



HEXAGON

User Guide

GeoCompressor 2022

User Guide

Version 16.7.0
21 October 2021

Contents

What's New	5
2022	5
2020 Update 3	5
2020 Update 2	5
2020 Update 1	6
2020	6
2018 Update 3	7
2018 Update 2	7
2018 Update 1	7
2018	7
2016	8
2015	8
2014.1	8
Overview	9
Architecture	10
Supported Environments	11
Installation Guide	12
Windows	12
Installation	12
Uninstallation	16
Linux	17
Installation	17
Side-by-Side Installations	19
Uninstallation	20
Licensing	21
Overview	21
Functionality Matrix	22
Installation and Configuration	22
GeoCompressor Usage	23
Overview	23
Image Compressor Usage	26
Wizard Mode	26
Compression Mode: Batch compress one or more images.	27
Compression Mode: Mosaic a set of input images to a single output image.	29
Compression Mode: Mosaic a set of input images to multiple output images.	31
Compression Mode: Update a region within an existing ECW v3 file	32
Select Compression Options	33
Select Output File Attributes	36

Summary	38
Command-line Mode	40
Input	42
Output	42
Options	43
Reporting Example	53
Usage Examples	55
ECW v3 Update Example	56
XML Project File	59
Supported Input Formats	61
Null Block Analysis	63
Comparison	64
Point Compressor Usage	66
Wizard Mode	66
Command-line Mode	68
Reporting Example	69
Usage Examples	70
Upload Usage	72
GUI Mode	72
Command-line Mode	73
Input File	73
GeoCompressor Viewer Usage	74
Overview	74
Toolbar Functions	74
Standard Toolbar	74
View Toolbar	75
Navigation Toolbar	75
ECW or JP2 Toolbar	75
Viewing Images	75
Geolinking Images	76
Geolink None	76
Geolink Window	76
Geolink Screen	76
Apply to All	77
Viewing ECWP images	78
Map Properties	79
Layer Properties	79
ECW Layer Properties	83
Preferences	84
About	86
Appendix A: Mosaic to Multiple Output Workflow Example	88
Appendix B: Troubleshooting	95

Low Memory Issues During Compression	95
Unexpected Application Closure/Crash	95
Appendix C: FAQ	96
General	96
Image Compressor	97
HSCP Converter	100
Appendix D: Utilities	101
Benchmark Disk	101
ECW Header Editor CLI Parameters	103
Support	104
About Hexagon.....	104

What's New

2022

- Compression performance gains of up to 30% with the latest ECW JPEG 2000 SDK.
- Added multiple language support to the GeoCompressor and Viewer user interfaces. Current supported languages are English, German, Korean, Spanish, French, Italian, Portuguese, and Simplified Chinese.
- Added "Gigapixel" field to raster properties panel in Viewer for extended user information.
- A map view for visualizing the input and output file footprints has been added to the image compressor wizard. Footprints can be selected and edited.
- Consolidated opacity mode and shapefile region selection to a single page in the image compressor wizard. Region selection (from shapefile) is now defined only once for both output extents and opacity generation.
- Optimized the default output extents when compressing with a region. Output is now clipped to the intersection of the image and the bounding box of the input region, potentially producing smaller files.
- JP2 SIZ marker is now consistent with the JP2 profile (J2P1) declared in the file type box.
- Added capability to geolink multiple images in MDI mode using screen or image space in GeoCompressor Viewer.
- Added XML metadata view to ECW JP2 layer properties in GeoCompressor Viewer. This is primarily for viewing the contents of the "GML in JP2" box.
- Added conversion of point clouds to Hexagon Smart Point Cloud (HSPC) format and removed the obsolete Hexagon Point Cloud (HPC) format.
- Added Upload to HxDR utility, to upload HSPC files (and other supported formats) to the [Hexagon Digital Reality](#) (HxDR) platform. Removed upload to Smart M.App Chest account capability.
- Added a new Disk Benchmark utility for system verification of disk read I/O performance.
- Added a new Calculate Checksum utility to aid in verifying file integrity.

2020 Update 3

- Maintenance release.
- Added resampling options Nearest Neighbor, Bilinear, Cubic, Cubic Spline, Lanczos, Average, Mode, and Gauss when resizing data. #GC-1174
- ECW check is now more accurate when determining validity of files and prints better warnings. #EC-2444
- ECW check on Linux reports not enough disk space errors. #GC-1194
- GeoCompressor cannot remove bands from the input image and change the cell size in same task. #GC-1173
- If an existing JP2 file is opened for write, the file is not truncated and remains the same size as the original. #GC-1195
- Error messages from GDAL are now captured in the processing log. #GC-1218
- Added confirmation dialog when stopping/deleting tasks from the UI. #GC-1235

2020 Update 2

- Maintenance release.
- Optimized default profile for ECW version 3 files for faster decompression and lower storage requirements. #EC-2491
- Viewer crash on RGBA image when opacity channel selected in statistics. #GC-1131

- Made background color setting more consistent. Mosaic now honors background color setting and is applied to NULL block decode setting for ECW v3. #GC-426
- Added missing library dependencies for RHEL 8/CentOS 8 (new supported platform). #GC-1070
- Logfile now prints the tile size and compression metadata from GDAL. #GC-1133
- Block size for ECW files is now shown in Viewer properties page. #GC-1151
- Provided shapefile feature selector UI to region selection. #GC-1141

2020 Update 1

- Maintenance release.
- PNG images with alpha would crash the compressor. #GC-1101
- GeoCompressor not recognizing the SRS information of some JP2 images. #GC-1100
- Viewer: Double clicking a file in the ECWP Browser that is not in the default service generates an error. #GC-1096
- NODATA value doesn't work for thematic input. #GC-1091
- License refresh button crashes UI when no license sources configured. #GC-1087
- Viewer: Cannot change log level. #GC-1084
- Deleting a task from the task scheduler results in corrupted reporting. #GC-1044
- Can't JP2 compress a 27-bit IMG (stored as 32-bit). #GC-849
- TIFF with rotation compressed to JP2 differs on Linux and Windows. #GC-794
- Cannot compress to JP2 with bit-depth other than 8 or 16. # GC-69

2020

- First release of GeoCompressor Viewer, a free-to-use general purpose viewer for ECW and JPEG2000 files designed to support rapid user display but also QA from data providers.
- The previously known ERDAS APOLLO Utilities have now been migrated as a new GeoCompressor Utilities component for ECWCheck, HeaderEditor, and other supporting data preparation tools.
- GeoCompressor 2020 now enforces new core/thread restrictions based on detected license level. See licensing chapter for more details. #GC-969
- SOURCEDIR naming variable was erroneously applied in the wrong field. #GC-1059
- When compressing input data in EPSG:4326, the output projection was being lost due to mishandling of exponent GMLJP2 parsing. #GC-1074
- ECW v3 with stored histograms was incorrectly dropping the last bin value across all bands. #GC-1047
- Made numerous improvements to unit and cell size persistence across platforms and workflows. This resolves issues such as output unit being lost or the wrong cell size being applied. #GC-1014, GC-998
- Log button is now enabled during compression jobs to make it simpler to monitor for any issues. #GC-1012
- Reworked the general handling of bitdepths to return more meaningful error states and support int16 or greater in more situations. There were some cases where previous versions were casting in16 to uint16 unexpectedly. #GC-1071, GC-1024, GC-995
- Deprecated low memory handling and now returns a hard error when insufficient memory is detected. #GC-949
- Enhanced the reliability in resolving input georeferencing to expected EPSG codes. #GC-1075
- Installation technology has been changed from the cross-platform Qt installer to platform-specific, MSI (Windows) and RPM (Linux). #GC-872

- Resolved multithreading issue in reading GeoPackage Raster tables that caused random small white areas to appear in the output ECW. #GC-953
- General image compression throughput speeds can be expected to be 15% or higher on Intel® hardware.
- Variety of branding and other small user interface improvements.
- Miscellaneous bug fixes and platform updates.

2018 Update 3

- Added “-openoptions” parameter to command line compressor to pass parameters to GDAL input reader (refer to GDAL format driver documentation for available options for each format). #GC-800
- Calculating the default cell size in GUI results in incorrect values when map units are degrees. #GC-886
- Estimated time to completion always 0 for JPEG2000 output. #GC-836
- When output extents are specified in XML, the band list inputs are ignored. #GC-858
- Better reporting of license status in point compressor. #GC-851

2018 Update 2

- Fixed generation of GML in JP2 and GeoJP2 from GUI. #GC-840, GC-841
- Add additional supported input types to the GUI when selecting via the file chooser. #GC-837
- Add estimated progress completion time to GUI. #GC-836
- Fixed crash when mosaicking images with different input band lists. #GC-796
- Miscellaneous UI bugs, cleaned up and updated documentation.

2018 Update 1

- Fixed customer hang that could occur when mosaicking JPEG2000 input files. #GC-742/00025349
- Fixed inconsistent warning about thread count “exceeding CPU Core count.” #GC-740/00024548
- Clarified handling of input files where an unsupported GeoTransform is detected in the input. #GC-768/00026035
- Fixed UI errors when attempting to batch compress multiple images, or images without georeferencing. #GC-764, GC-765
- Null and data block counts are now correctly logged, where previously they were only printed to the console. #GC-762
- Numerous internal bugs were resolved to ensure binary consistency across Windows- and Linux-generated outputs. #GC-756, GC-422, GC-757
- Fixed crash caused by a custom bandlist order. It will now honor the band count and order specified when mosaicking. #GC-778

2018

- New workflow for creating multiple output files that are clipped to a polygon selected from a shapefile. Users can create an output file for each state or county, where a shapefile containing the polygons defining those states or counties is specified, from either a single file or a virtual mosaic of many input files. Opacity channels are also generated from the selected polygons.
- Output scaling/resampling to new cell size.

- Advanced customization options for JPEG2000 file creation, including progression order, quality layers, SOP/EPH markers, and tile and precinct dimensions.
- Control over output SRS format for JPEG2000 can now be set to be either GML in JPEG2000 (v1.0) or GeoJP2.
- Added ability to specify the input file band mappings in the GUI.

2016

- Support for specifying a transparent color and nodata values on input files.
- Speed up when mosaicking large amounts of input imagery.
- Added new feature to upload datasets to a Smart M.App Chest account for further processing on the cloud.
- Clip output image to polygon bounding box when compressing.
- AVX2 and FMA3 acceleration on supported CPUs.

2015

- “ERDAS Image Compressor” now known as “GeoCompressor.”
 - Product now available for purchase under a subscription, tiered licensing scheme
- Point cloud conversion to the streamable Hexagon Point Cloud (HPC) format has been added.
 - Windows only
- Partial region updates, or selective compression within an existing ECW v3 file is now supported.
- Image mosaicking now defaults to the tile compression method for vastly improved performance.
- User interface has been redesigned.
- Many bug fixes, performance and general workflow improvements.
 - Image Compressor report now breaks down user time into read, write, and reassembly time
 - Multi-band uint16 ECW v3 compression now uses per-band statistics to generate higher quality output
 - Region improvements to support multi-part features

2014.1

- Mosaicking now supports multiband input, multiband output. Previously, only RGB/RGBA output was supported.
- Mosaicking now supports greater than 8-bit output for ECW v3 and JPEG2000 files.
- Parsing of mosaic projects with thousands of datasets now 10 times faster.
- Overriding of SRS information now possible in Mosaic XML as a new output option.
- Added support for MBTile, NITF, and Terrashare RBD file input.
- 12-bit JPEG compressed file types now supported.
- Multi-part polygon regions now supported (e.g., polygons with holes).
- Further optimizations and bug fixes.

Overview

GeoCompressor is a high-performance geospatial image and point cloud compression application designed to simplify the creation of ECW, JPEG2000, and HSPC formats.

The application provides:

- Cross-platform, 64-bit support for both Windows and Linux with command-line and simple wizard user interfaces
- A stand-alone, decoupled plugin tool to use with existing data processing workflows, ideal for data providers
- A cost-effective solution that fills the void between developers acquiring an ECWJP2 SDK license and a full ERDAS IMAGINE license

The product's functional scope is intentionally narrow. GeoCompressor does not:

- Support 32-bit platforms
- Offer advanced mosaicking functions
 - You cannot feather, blend, color balance, dodge, adjust seamlines, preview output, or perform many other tasks typically performed in a mosaic process
- Reproject or warp imagery to different output coordinate systems

GeoCompressor can be seen therefore as a transcoder only of input data. For other image and point-cloud processing tasks, ERDAS IMAGINE or ImageStation remains the recommended product for end-to-end processing and compression.

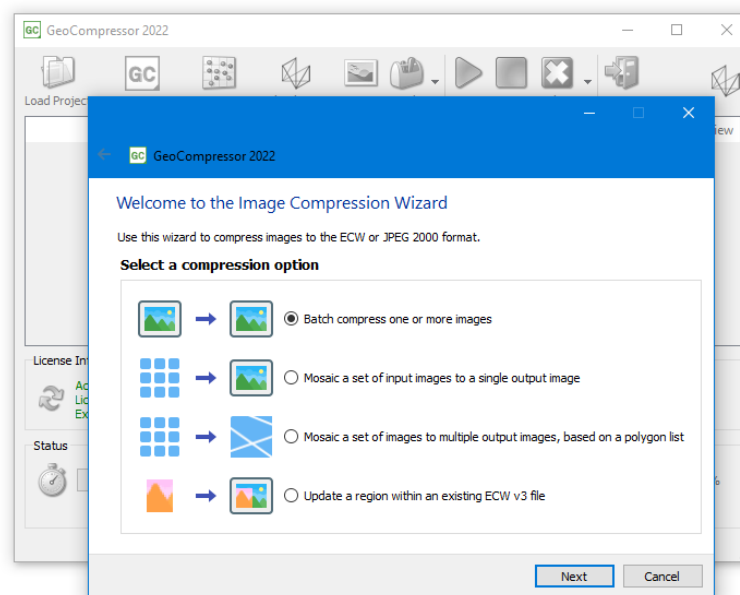


Figure 1 – Image Compression Wizard interface

Architecture

GeoCompressor can be broken into input readers, compressor logic, and output writers. The input readers ensure wide industry format support by leveraging the [GDAL](#), [libLAS](#), [LASZip](#), [ERDAS IMAGINE](#), and [ER Mapper](#) libraries.



The application is thread-safe¹ and has been deployed on processing workstations up to 64-core with excellent CPU scaling using the parallelized ECW compressor. Peak performance has been recorded in excess of 750MB/sec² encoding a 1000 gigapixel image and far exceeds the throughput of other third-party applications that implement the [ERDAS ECWJP2 SDK](#) due to optimizations implemented in the GeoCompressor architecture.

The provided user interface is a thin wrapper around several command-line executables, namely the ImageCompressor, HSPCConverter, and HxDRUploader executables. For intermediate to advanced users, it is expected that most will interface with the command line tools to give more control integrating with existing provider workflows.

¹ Not all GDAL supported formats are thread-safe. See FAQ.

² Compression throughput speeds vary and are heavily dependent on hardware and the input data format type and internal structure. This number is based on recording speed compressing an input test pattern.

Supported Environments

GeoCompressor supports the following operating systems. Please note that one is required.

- Microsoft Windows® 8.1, 10, 11 (64-bit)
- Microsoft Windows® Server 2016, 2019, 2022
- Red Hat® Enterprise Linux® / CentOS 8.x (64-bit)
- Red Hat® Enterprise Linux® / CentOS 7.x (64-bit)

Notes:

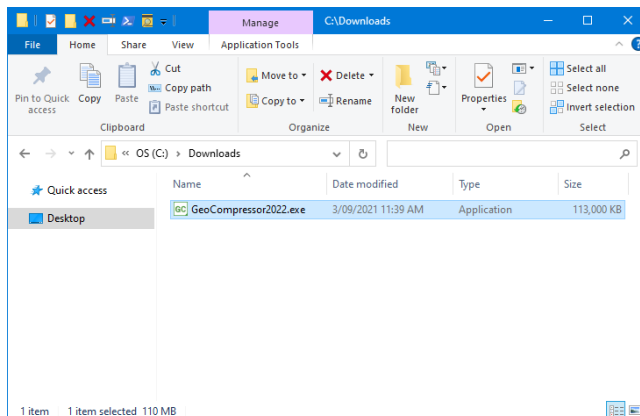
1. Other Linux distributions such as Debian, Ubuntu, Mint, Fedora, and OpenSuSE are considered *viable* platforms and may require additional installation requirements.
2. HSPC Viewer is not available on Red Hat® Enterprise Linux® 7 / CentOS 7.
3. Disk Benchmark tool is unavailable on Linux platforms

Installation Guide

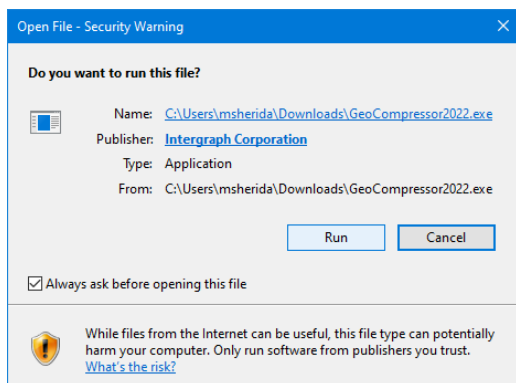
Windows

Installation

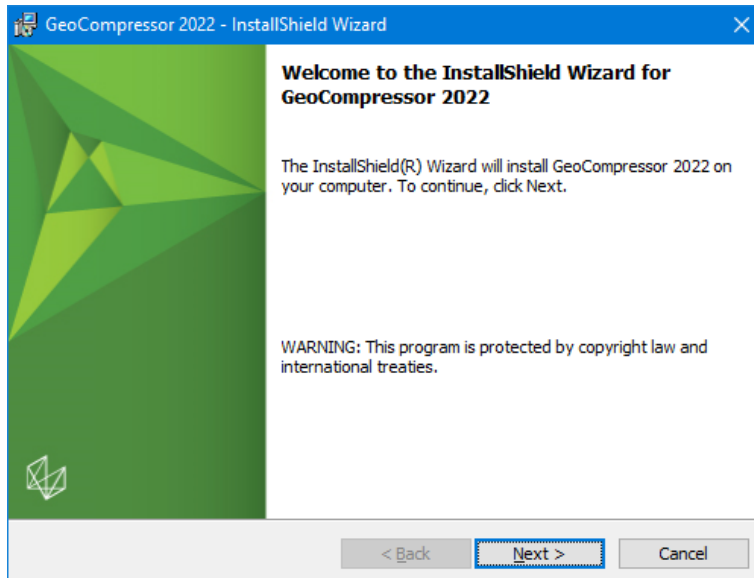
1. Download and execute the installation package from the product website.



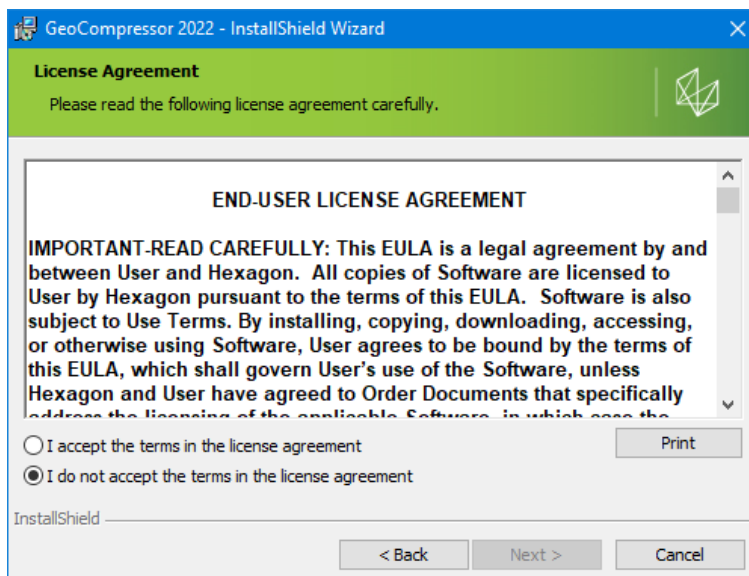
2. Windows will present a security warning. Verify the publisher is "Intergraph Corporation" and click "Run" to begin the installation.



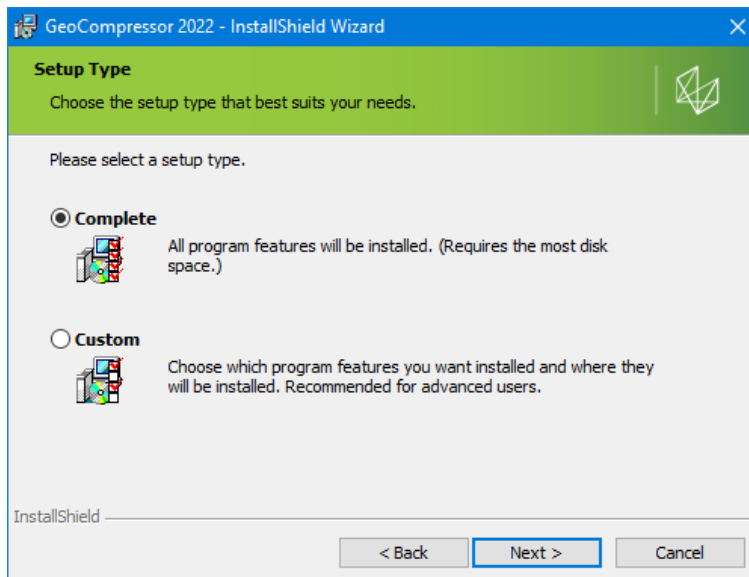
3. The installation wizard appears. Click “Next” to continue.



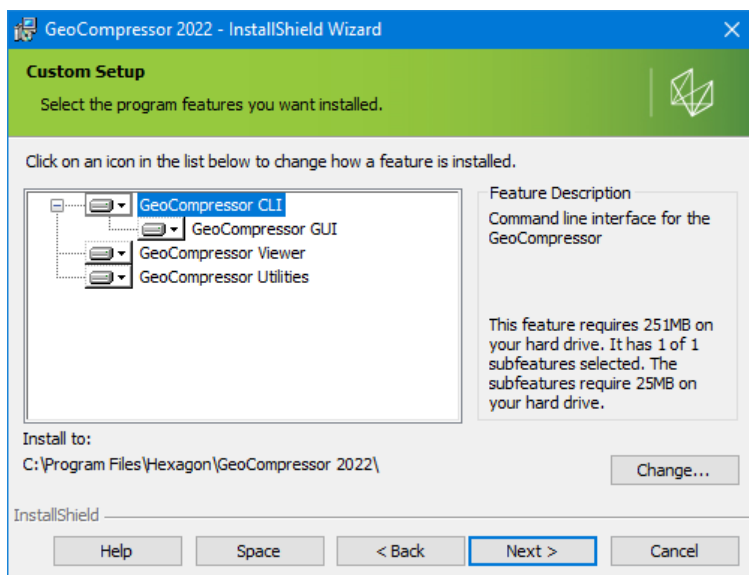
4. Read the end-user license agreement and select “I accept the terms of the license agreement” to continue. If you do not accept the license agreement, you cannot install GeoCompressor.



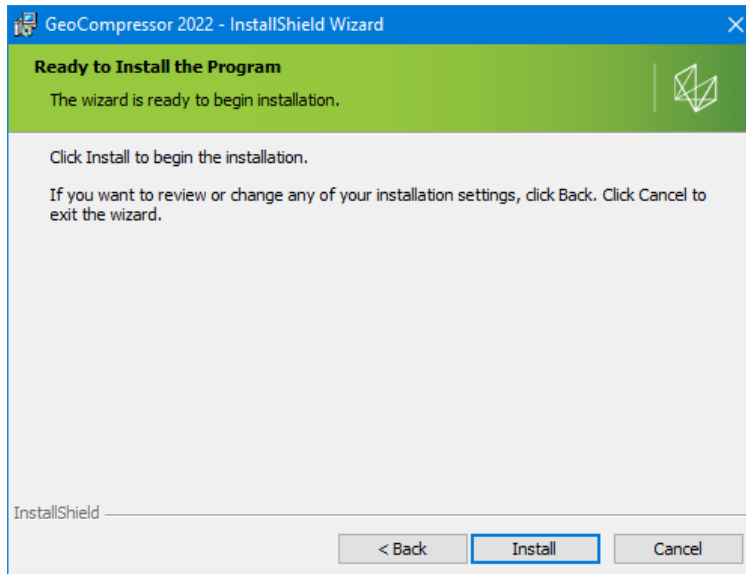
- You may select “Complete” to continue, or “Custom” to customize the installation. Click “Next” to proceed.



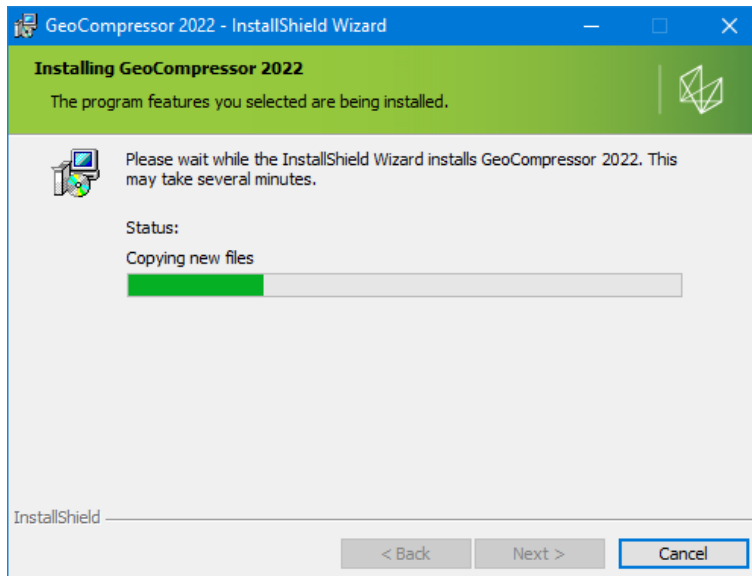
- If you have selected “Custom,” it will show a list of features that can be added/removed and has a “Change” button to specify a different installation directory.



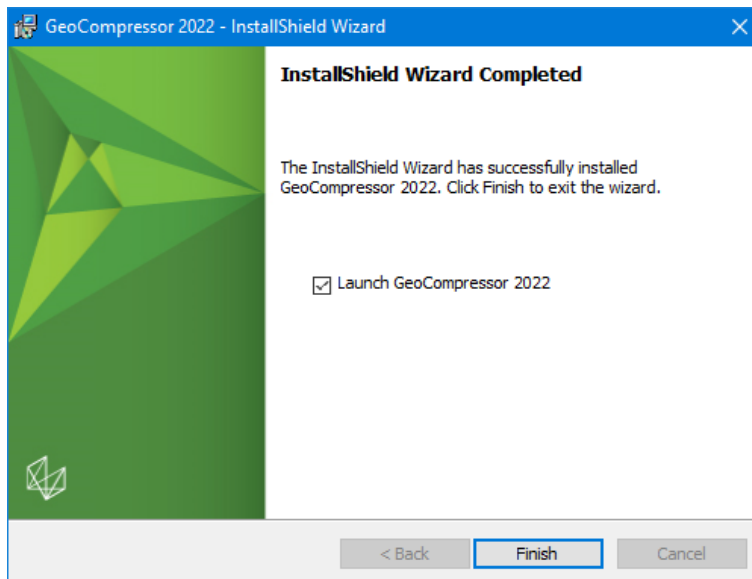
- Click “Next” to continue.



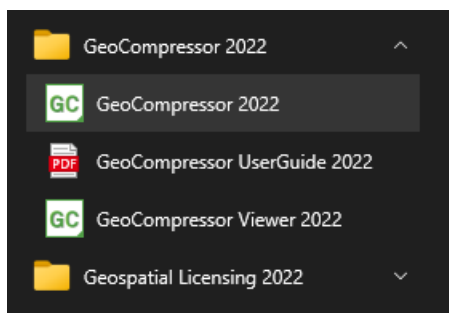
- Wait for the installation to complete.



9. Upon completion, select the “Launch GeoCompressor” checkbox to run the software. Click “Finish” to complete the installation.



GeoCompressor can be launched via the Windows Start Menu shortcut to launch the main user interface.



Or the executables can be launched directly from the command line, e.g.:

“C:\Program Files\Hexagon\GeoCompressor[year]\bin\ImageCompressor.exe”

“C:\Program Files\Hexagon\GeoCompressor[year]\bin\HSPCConverter.exe”

“C:\Program Files\Hexagon\GeoCompressor[year]\bin\HxDRUploader.exe”

See Licensing and Usage chapters for next steps.

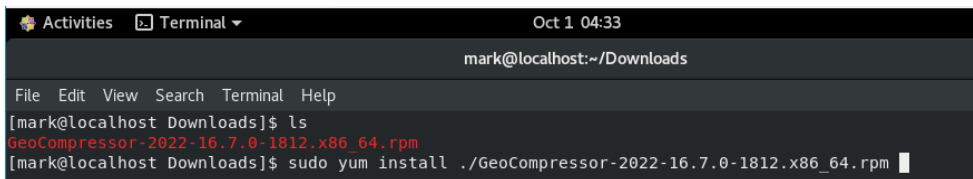
Uninstallation

To remove or uninstall, use the standard Windows “Add/Remove Programs” process.

Linux

Installation

1. Download the installer from the product website.
2. Ensure the installer has execute permission: `chmod u+x GeoCompressor.[version].x86_64.bin`.
3. Execute the .bin package `./GeoCompressor.[version].x86_64.bin`.
4. Read the EULA terms by pressing space to scroll through the text, and accept the terms to continue by typing "yes." An RPM package will be extracted.
5. Install the RPM:
 - A. On **CentOS / RedHat / Fedora**: using `yum install GeoCompressor.[version].x86_64.rpm`.

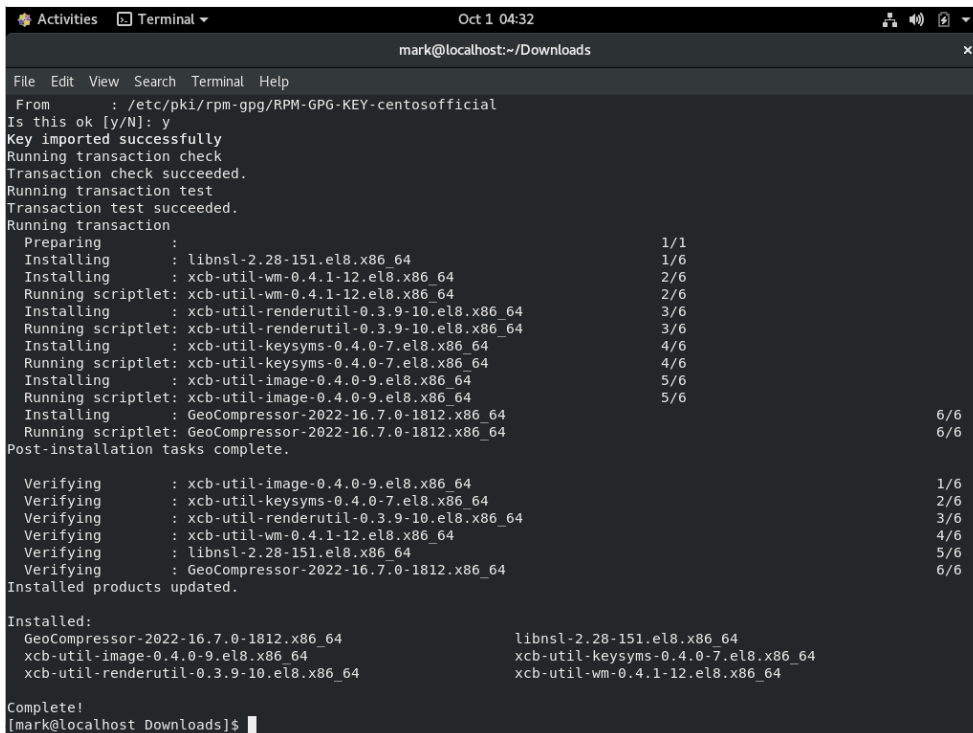


```
Oct 1 04:33
mark@localhost:~/Downloads

File Edit View Search Terminal Help

[mark@localhost Downloads]$ ls
GeoCompressor-2022-16.7.0-1812.x86_64.rpm
[mark@localhost Downloads]$ sudo yum install ./GeoCompressor-2022-16.7.0-1812.x86_64.rpm
```

`yum install` will automatically determine dependencies before installing GeoCompressor in the system. Type "yes" and press enter to download any identified dependencies.



```
Oct 1 04:32
mark@localhost:~/Downloads

File Edit View Search Terminal Help

From      : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/1
  Installing               : libnsl-2.28-151.el8.x86_64      1/6
  Installing               : xcb-util-wm-0.4.1-12.el8.x86_64  2/6
  Running scriptlet: xcb-util-wm-0.4.1-12.el8.x86_64        2/6
  Installing               : xcb-util-renderutil-0.3.9-10.el8.x86_64  3/6
  Running scriptlet: xcb-util-renderutil-0.3.9-10.el8.x86_64  3/6
  Installing               : xcb-util-keysyms-0.4.0-7.el8.x86_64  4/6
  Running scriptlet: xcb-util-keysyms-0.4.0-7.el8.x86_64      4/6
  Installing               : xcb-util-image-0.4.0-9.el8.x86_64  5/6
  Running scriptlet: xcb-util-image-0.4.0-9.el8.x86_64        5/6
  Installing               : GeoCompressor-2022-16.7.0-1812.x86_64  6/6
  Running scriptlet: GeoCompressor-2022-16.7.0-1812.x86_64    6/6
Post-installation tasks complete.

  Verifying                : xcb-util-image-0.4.0-9.el8.x86_64  1/6
  Verifying                : xcb-util-keysyms-0.4.0-7.el8.x86_64  2/6
  Verifying                : xcb-util-renderutil-0.3.9-10.el8.x86_64  3/6
  Verifying                : xcb-util-wm-0.4.1-12.el8.x86_64  4/6
  Verifying                : libnsl-2.28-151.el8.x86_64  5/6
  Verifying                : GeoCompressor-2022-16.7.0-1812.x86_64  6/6
Installed products updated.

Installed:
  GeoCompressor-2022-16.7.0-1812.x86_64      libnsl-2.28-151.el8.x86_64
  xcb-util-image-0.4.0-9.el8.x86_64          xcb-util-keysyms-0.4.0-7.el8.x86_64
  xcb-util-renderutil-0.3.9-10.el8.x86_64    xcb-util-wm-0.4.1-12.el8.x86_64

Complete!
[mark@localhost Downloads]$
```



On **Ubuntu 18/20** :

- a) `"sudo apt-get install alien"` (skip this if installed)
- b) `"sudo apt install libxcb-xinerama0"`
- c) `"sudo apt install libxcb-xinput0"`
- d) `"sudo alien -i --scripts GeoCompressor.[version].x86_64.rpm"`

Default installation location is `"/usr/local/hexagon/geocompressor[year]/."` Ensure the user installing the application has appropriate rights to write to this location (or use sudo).

GeoCompressor GUI can be executed via command line:

`"/usr/local/hexagon/geocompressor[year]/bin/GeoCompressor.sh"`

Or via the Application shortcut in the "Graphics" category on KDE and Gnome Window managers.



GeoCompressor CLI components can be executed via command line:

`"/usr/local/hexagon/geocompressor[year]/bin/ImageCompressor"`

`"/usr/local/hexagon/geocompressor[year]/bin/HSPCConverter"`

`"/usr/local/hexagon/geocompressor[year]/bin/HxDRUploader"`

To run these CLI components, it is recommended to use the paired shell scripts such as `./ImageCompressor.sh` to set the required environmental paths.



Side-by-Side Installations

Note that from 2022 onwards, GeoCompressor allows side-by-side installation of major versions (differentiated by year). For example, if you install GeoCompressor 2020 then install GeoCompressor 2022, you will have two installations, one in `/usr/local/hexagon/geocompressor2020` and another in `/usr/local/hexagon/geocompressor2022`. Previously, versions were installed with the package name "GeoCompressor," which meant that installation of a newer version would always try to upgrade the current version. From 2022 onwards, the installation uses the package name "GeoCompressor-[YEAR]." This means that installing an update of an installed version with the same year will automatically upgrade that version, but installations of a new version will be in a separate directory.

See Licensing and Usage chapters for next steps.

Uninstallation

A. CentOS/RedHat/Fedora use

a. “`sudo yum remove GeoCompressor-[YEAR]`”

```

Activities  Terminal  Oct 5 12:47
hexagon@localhost:~/Downloads

File Edit View Search Terminal Help
[hexagon@localhost Downloads]$ sudo yum remove GeoCompressor-2022.x86_64
[sudo] password for hexagon:
Dependencies resolved.

=====
Package                        Architecture  Version      Repository    Size
=====
Removing:
GeoCompressor-2022             x86_64        16.7.0-1799  @commandline  874 M
=====

Transaction Summary
=====
Remove 1 Package

Freed space: 874 M
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                               1/1
  Running scriptlet: GeoCompressor-2022-16.7.0-1799.x86_64 1/1
Pre-uninstallation tasks complete.

  Erasing                               1/1
  Running scriptlet: GeoCompressor-2022-16.7.0-1799.x86_64 1/1
Removing application shortcuts

  Verifying                               1/1
Installed products updated.

Removed:
GeoCompressor-2022-16.7.0-1799.x86_64

Complete!

```

B. Ubuntu use

a. “`dpkg -l | grep -i geocompressor`” (list the GeoCompressor packages installed)

b. “`dpkg -r GeoCompressor-[YEAR]`” (uninstall the given GeoCompressor package)

```

adminuser@adminuser-VirtualBox: /usr/local/hexagon/geocompressor2019/bin$ sudo dpkg -r GeoCompressor
(Reading database ... 167337 files and directories currently installed.)
Removing geocompressor (16.6.0-612) ...
Pre-uninstallation tasks complete.
Removing application shortcuts
Processing triggers for libc-bin (2.27-3ubuntu1) ...

```

Licensing

Overview

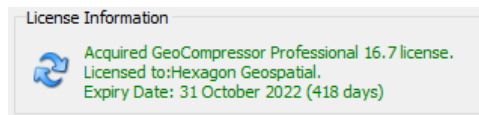
GeoCompressor is designed to complement existing ERDAS APOLLO installations and provide a decoupled, rapid compression tool. The product is dual-licensed with respect to the product codes to meet both use-cases.

Product Family	Required License Level	Version
GeoCompressor	Essentials	16.7
GeoCompressor	Advantage	16.7
GeoCompressor	Professional	16.7
ERDAS APOLLO	Essentials	16.7
ERDAS APOLLO	Advantage	16.7
ERDAS APOLLO	Professional	16.7

If node-locked or concurrent licenses are unavailable for any of the above products and versions, the compressor will fail. Where certain functionality is only available at a particular licensing tier and an operation is attempted, a license error will be returned. Refer to functionality matrix below for details.

In the event only an ERDAS APOLLO license is found, GeoCompressor will unlock the same functionality found at the GeoCompressor Professional level. For example, ERDAS APOLLO Essentials will unlock unlimited compression found at GeoCompressor Professional. This relationship, however, is not bi-directional; GeoCompressor licenses will not unlock ERDAS APOLLO.

Host-based licensing in the product ensures that if any of the above feature codes are already acquired on the same machine, GeoCompressor will validate but *not* check out an additional license. For example, GeoCompressor and ERDAS APOLLO Advantage can coexist under one ERDAS APOLLO Advantage license when run on the same machine at the same time. Equally, you can run parallel GeoCompressor processes on the same machine while using only one concurrent GeoCompressor license.




```
Administrator: C:\WINDOWS\SYSTEM32\cmd.exe
C:\Program Files\Hexagon\GeoCompressor 2022\lib\bin\x64URelease>ImageCompressor.exe
TBB Malloc loaded from C:\Program Files\Hexagon\GeoCompressor 2022\lib\bin\x64URelease\tbbmalloc.dll
Acquired GeoCompressor Professional 16.7 license.
Licensed to: Hexagon Geospatial.
Expiry Date: 31 October 2022 (418 days)
ImageCompressor <input> <output> [Options...]
Build: v16.7.0.1782
```

Figure 2 – License status in UI and command line

Functionality Matrix

The features available at each GeoCompressor license tier are as follows.

GeoCompressor				
	Free	Essentials	Advantage	Professional
ECW Compression Thread Limits	✗	4	8	Unlimited
Image Compression < 250 gigapixels	✗	✓	✓	✓
Image Compression < 500 gigapixels	✗	✗	✓	✓
Image Compression Unlimited	✗	✗	✗	✓
Image Mosaicking <small>Up to gigapixel limit</small>	✗	✓	✓	✓
Batch Image Compression <small>Up to gigapixel limit</small>	✗	✓	✓	✓
Point Compression Unlimited	✗	✗	✗	✓
ECW v3 Region Update	✗	✗	✗	✓
Concurrent License Only	✗	✓	✓	✓
Subscription Only	✗	✓	✓	✓
GeoCompressor Viewer (ECW * JP2)	✓	✓	✓	✓
Windows Explorer Shell Plugin	✓	✓	✓	✓
Utilities (File Check, Header Editor, Checksums, Disk Benchmark)	✓	✓	✓	✓
Upload to Hexagon Digital Reality (HxDR)	✓	✓	✓	✓

 The size of a file in gigapixels can be calculated by multiplying the number of rows (height) by the number of columns (width). For example, an image with 20,000 rows and 20,000 columns would equal 400,000,000 pixels or 0.4 gigapixels. The number of bands, cell type, and the number of “empty” areas are not considered.

Installation and Configuration

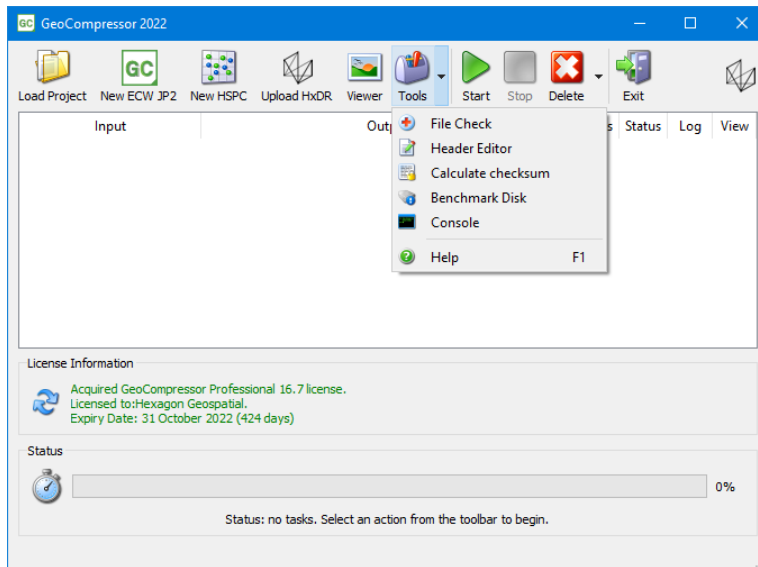
Refer to the [Hexagon Geospatial Licensing Portal](#) for installation instructions and troubleshooting information for the licensing server technology. Windows and Linux licensing information is available.

For most users with an existing product, no additional licensing steps should be required as the dependent product license codes should already be configured.

GeoCompressor Usage

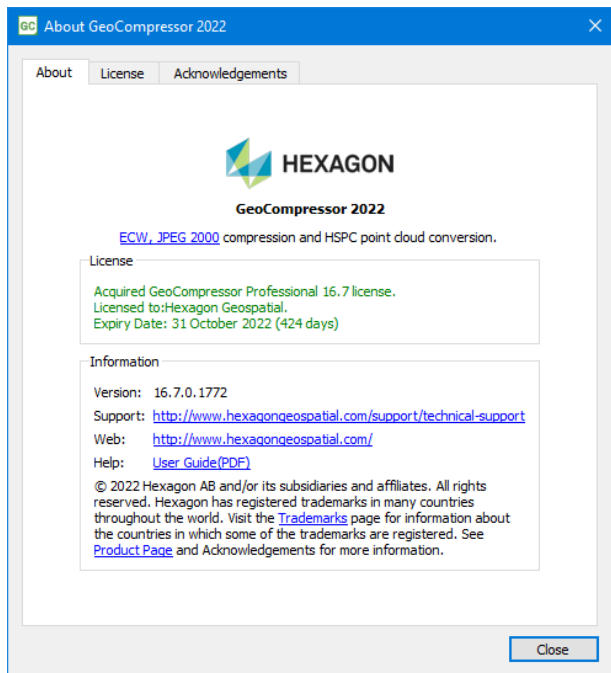
Overview

The GeoCompressor task window will list all previous and current compression tasks that have been defined and is the main user interface control outside of the direct command line interface.

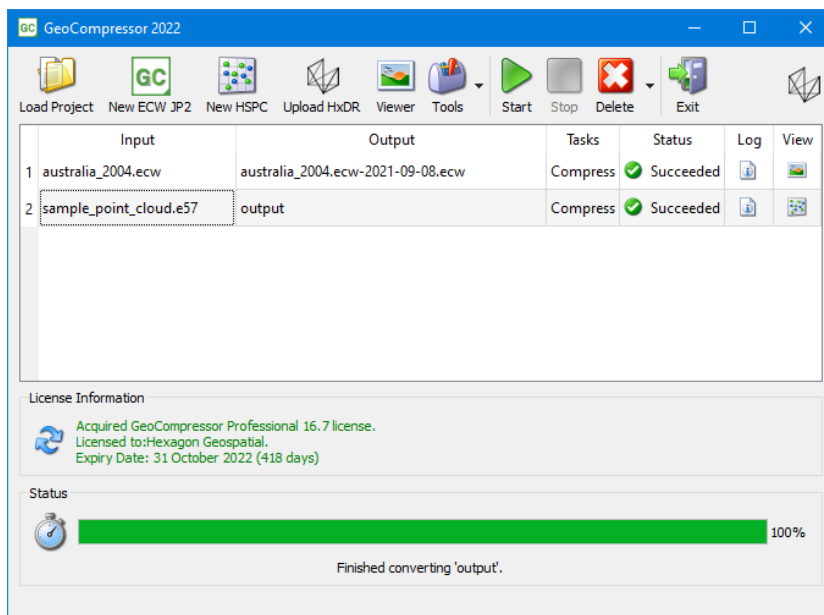


The toolbar icons on the main interface are as follows:

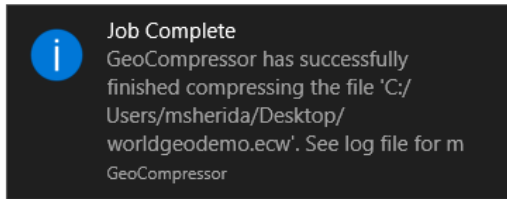
- “Load Project” – allows predefined GeoCompressor XML files that describe an image compression task to be loaded and run directly, bypassing the wizards.
- “New ECW JP2” – runs the Image Compressor Wizard to compress image files to ECW or JPEG2000 format.
- “New HSPC” – runs the Point Compressor wizard to convert a point cloud file to HSPC format.
- “Upload HxDR” – uploads a data file to the Hexagon Digital Reality platform (HxDR).
- “Viewer” – runs the GeoCompressor Viewer application to display ECW or JPEG2000 images, metadata, and other image properties.
- “Tools” – A tool menu of utilities consisting of the following. See Appendix D for more information.
 - “File Check” - checks the integrity of an ECW or JPEG2000 file to ensure it contains no errors.
 - “Header Editor” – edits the georeferencing or metadata information stored in an ECW or JP2 file.
 - “Calculate Checksum” – calculates the MD5 checksum of any file type.
 - “Benchmark Disk” – validates disk read I/O metrics to ensure peak GeoCompressor performance.
 - “Console” – opens a terminal/console to run command line utilities such as ECWCheck.exe, NCSFileInfo.exe, or the ImageCompressor.exe directly (Windows only).
 - “Help” – opens the PDF help manual.
- “Start” and “Stop” – allow you to begin or cancel currently running jobs, but note there is no resume function. Once a task is stopped, all output and temporary files are removed, and it must be restarted.
- “Delete Task” – removes the selected tasks from the main window (for example, to clear all previous jobs). Select the menu to show extended deletion options.
- “Exit” – closes GeoCompressor fully and does not minimize to the system tray (default behavior).
- “About” – confirms build, licensing, EULA, and acknowledgement information.



In the event of a failed task, the status will be reported with a link to the log file for easy review. On a successful compression task, the “View” column also becomes active and allows you to open the output file with the registered system application to open ECW, JPEG2000 or HSPC files, or open a web browser to the Hexagon Digital Reality login page for successful uploads.



GeoCompressor has a system tray icon and will report there via a pop-up balloon the status of the completed task (success or failure).



Additionally, if you right click the tray icon, there are further options to hide/show the compressor or launch a wizard for a new task.

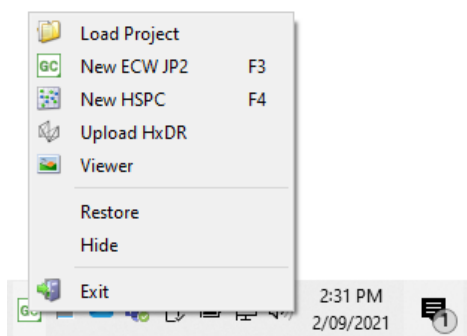
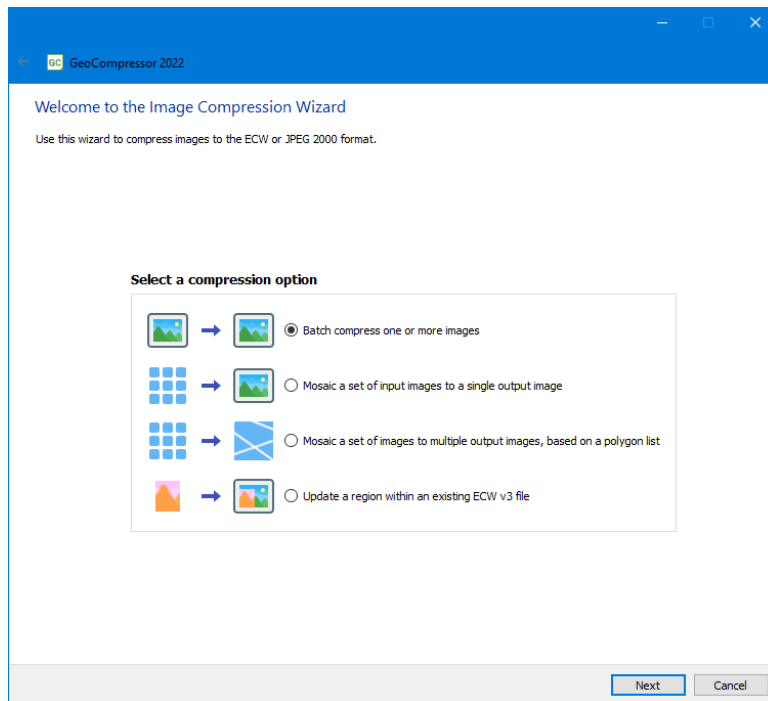


Image Compressor Usage

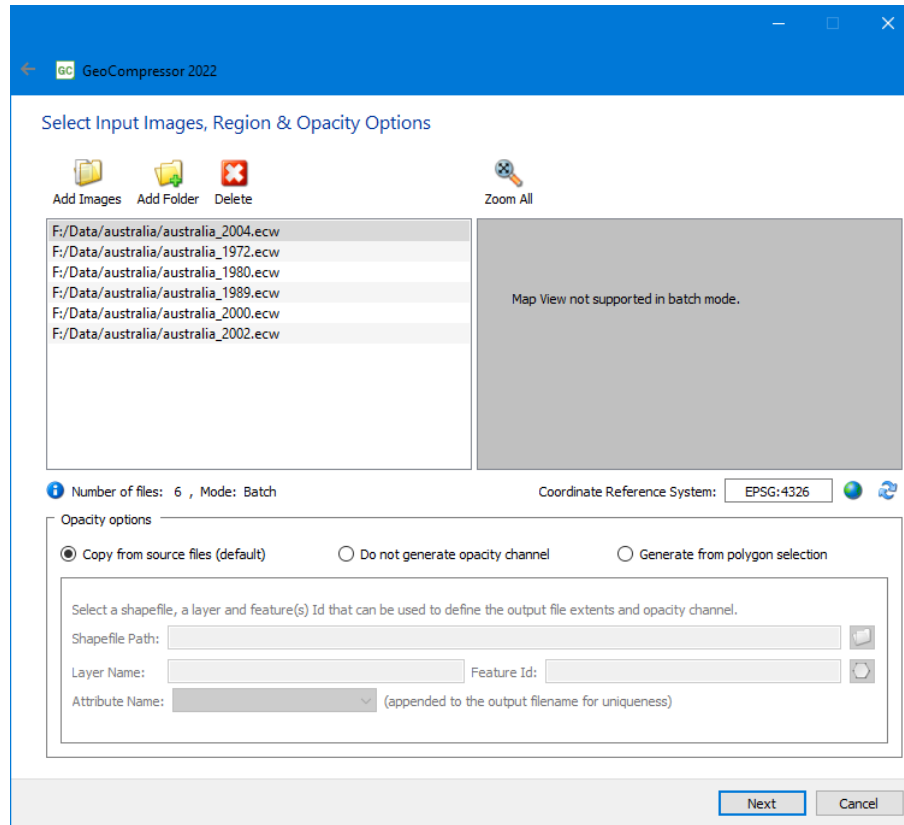
Wizard Mode

The ECWJP2 Task wizard comprises four separate task types:



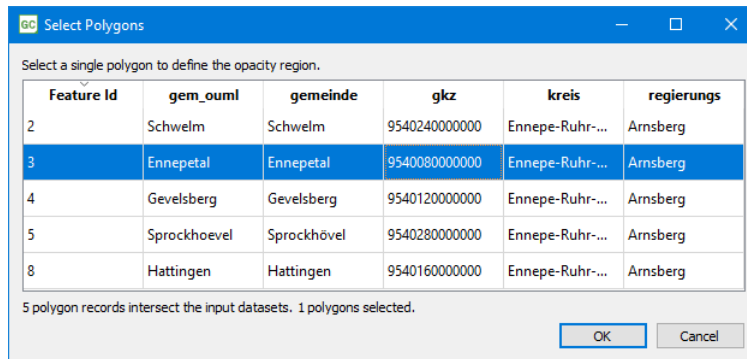
1. For “Batch compress one or more images,” GeoCompressor will create an output image for each input image. This provides a rapid way to compress thousands of images in batch.
2. For “Mosaic a set of images to a single output image,” GeoCompressor will combine all selected images into a single compressed output image. This mode was designed for tiled, preferably adjoining, images. The mosaic is created on the fly as part of the compression process; no intermediate files are created.
3. For “Mosaic a set of images to multiple output images, based on a polygon list,” the compressor will create a mosaic (as in option 2 if multiple files are input), but will output multiple images, based on a shapefile that contains multiple polygons that define the output extents of each image.
4. Use “Update a region within an existing ECW v3 file” to selectively update only a portion of an existing ECW file. This is particularly powerful for terapixel-sized ECWs to refresh only a portion of the image with new imagery, without recompressing the whole file.

Compression Mode: Batch compress one or more images.



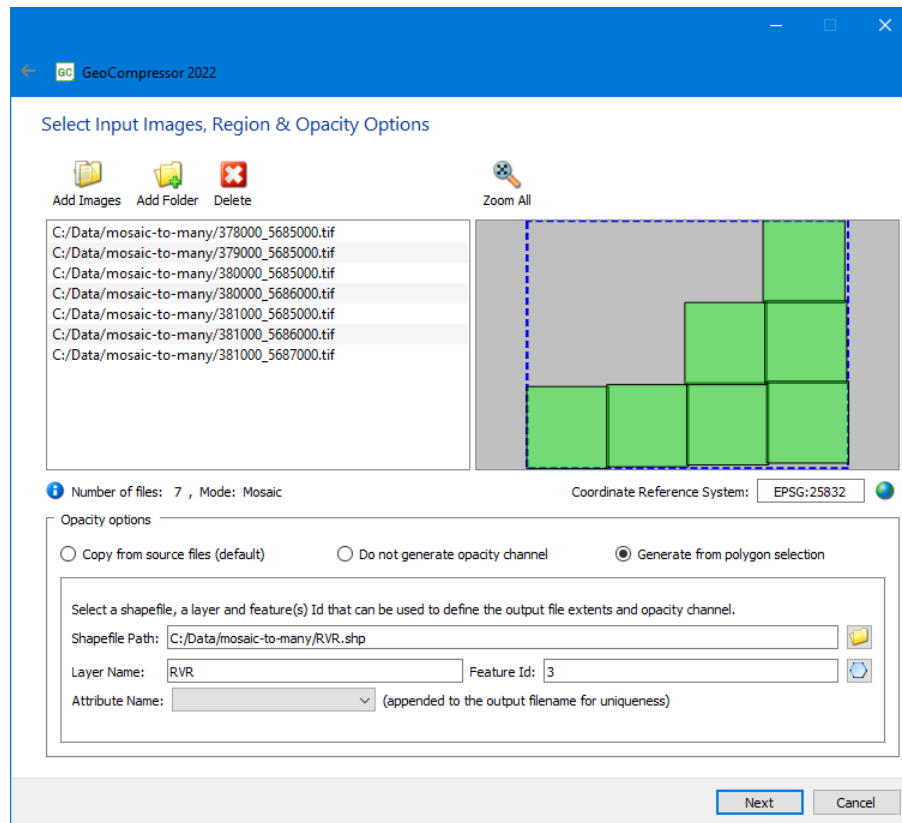
1. “Add images” allows individual file selection using the default “image filter” of common input file types (*.ecw, *.jp2, *.tif, *.alg, *.ers, *.img, *.vrt) or “All files” (refer to Supported Image Formats section for full list). The panel on the left displays the input images. If the wizard is in mosaic mode, the panel on the right displays a map with the footprints of the input images and the computed output file extents.
2. “Add folder” can be used to add thousands of files within a directory. If folders are found within the selected folder, you will be asked if you want to add images recursively.
3. For each input file in the list, an output file will be produced at the end of the images.
4. The number of input files and the compression mode (batch) is displayed, as well as the output Coordinate Reference System.
5. You can override the output Coordinate Reference System by entering an EPSG format code, such as “EPSG:4326.” Note that no actual re-projection of the data is performed; however, the image header metadata will contain this user-defined one. Click the icon next to the CRS definition to go to epsg.org to obtain more information about the projection. In batch mode, specifying a new CRS will apply to all images in the input list.
6. If the input data has no valid Coordinate Reference System, you will get the following warning message:
 - a. *Automatic lookup to EPSG failed. EPSG codes support georeferencing interoperability across software products. Press Cancel to define EPSG manually or OK to proceed.*
 - b. In this case, if you know the projection code, you should enter it before continuing.
7. Opacity options:
 - a. Copy from source – the output file will have an opacity channel if one is defined in the source image; otherwise, no opacity channel will be generated.
 - b. Do not generate opacity channel – no opacity channel will be generated, regardless of the input file structure.

- c. Generate from polygon selection – select a polygon from a shapefile. The shapefile must be in the same CRS as the input image(s), and you can only select a single polygon in batch mode.



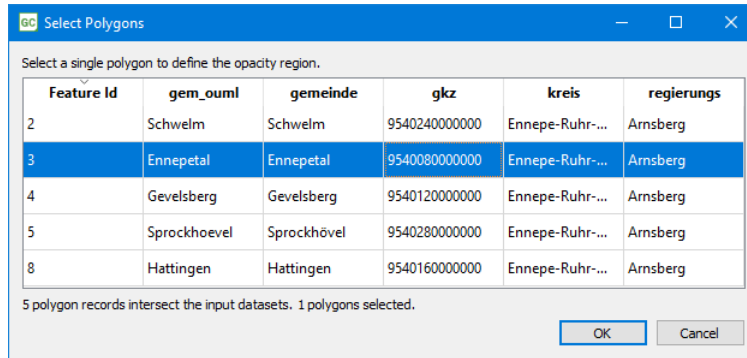
- d. If batch mode images are in different projections and different polygons are needed, then the compression wizard must be run again for each image/shapefile combination.

Compression Mode: Mosaic a set of input images to a single output image.

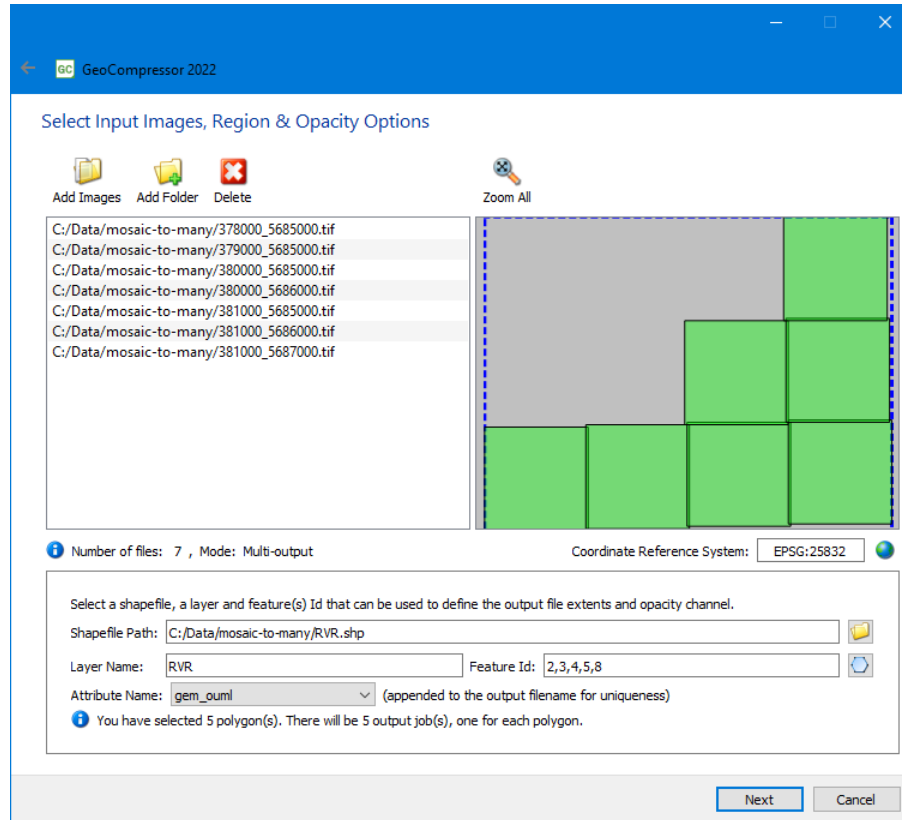


1. Selection of input files is the same as in batch mode.
2. In this mode, a map view with footprints of the input datasets (green) and the output file extents (dashed blue) representing the minimum bounding rectangle of all the input is shown on the right side.
3. The order in which the datasets are displayed in this list box represents the display order of the output mosaic, where datasets displayed at the "top" will be rendered on "top" of the output image.
4. You can use the contextual right click menu or drag and drop to reorder the stacking order. You can delete images by selecting items in the list and typing "Del" on the keyboard, right clicking and selecting "Delete," or clicking the "Delete" button on the toolbar.
5. You can select a rectangle on the map, and it will highlight the corresponding dataset in the list and vice versa. You can pan and zoom the map (left button drag or mouse wheel) to zoom in on features, or select the "Zoom All" tool button to reset the map to full output extents.
6. Areas where opacity will be generated (if selected) will be in grey areas within the blue rectangle.
7. The number of input files and the compression mode (Mosaic) is displayed, as well as the output Coordinate Reference System.
8. You can override the output Coordinate Reference System by entering an EPSG format code, such as "EPSG:4326." Note that no actual re-projection of the data is performed; however, the image header metadata will contain the user-defined one. Click the icon next to the CRS definition to go to epsg.org to obtain more information about the projection.
9. If the input data has no valid Coordinate Reference System, you will get the following warning message:
 - *Automatic lookup to EPSG failed. EPSG codes support georeferencing interoperability across software products. Press Cancel to define EPSG manually or OK to proceed.*
 - In this case, if you know the projection code, you should enter it before continuing.
10. Opacity options:

- Copy from source – the output file will always have an opacity channel, defined from the blank areas that are not covered by input imagery.
- Do not generate opacity channel – no opacity channel will be generated, regardless of the input file structure.
- Generate from polygon selection – select a polygon from a shapefile. The shapefile must be in the same CRS as the input image(s), and you can select one single polygon in mosaic mode (one output file).



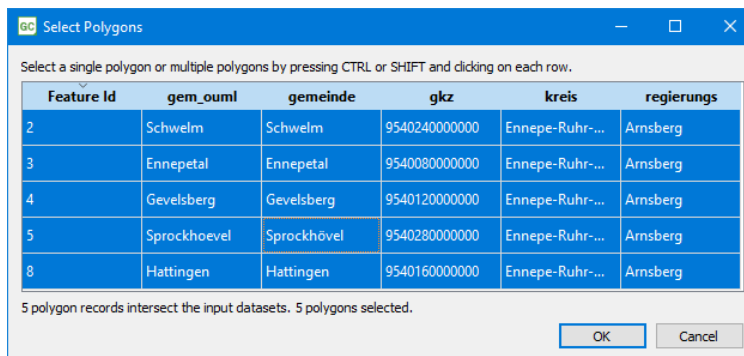
Compression Mode: Mosaic a set of input images to multiple output images.



1. In this mode, an output image will be generated for each region (polygon) that the user selects from a shapefile.
2. Selection of input files is the same as in previous modes. The user can enter in any number of images, which will be mosaicked into a single image for input to the process. However, there will be multiple output files created representing the polygon features selected.
3. The footprints of the input datasets are shown on the map view, the same as in mosaic mode.
4. The number of input files and the compression mode (Multi-output) is displayed, as well as the output Coordinate Reference System.
5. You can override the output Coordinate Reference System by entering an EPSG format code, such as "EPSG:4326." Note that no actual re-projection of the data is performed; however, the image header metadata will contain the user-defined one. Click the icon next to the CRS definition to go to epsg.org to obtain more information about the projection.
6. If the input data has no valid Coordinate Reference System, you will get the following warning message:
 - a. *Automatic lookup to EPSG failed. EPSG codes support georeferencing interoperability across software products. Press Cancel to define EPSG manually or OK to proceed.*
 - b. In this case, if you know the projection code, you should enter it before continuing.

7. Opacity options:

- a. In multi-output mode, an opacity channel will always be generated.
- b. Select a set of polygons (one or more) from a shapefile. The shapefile must be in the same CRS as the input image(s). You can select multiple polygons, which implies the number of output images that will be generated, with the opacity based on each individual polygon. Each output image will be clipped to the extents (footprint) of the selected polygon.
- c. Take note of the table column header names. One of the attributes must be used when naming the output file; be sure to choose something unique. In this example below, use the “gem_ouml” attribute in the output filename for easy identification of the images.

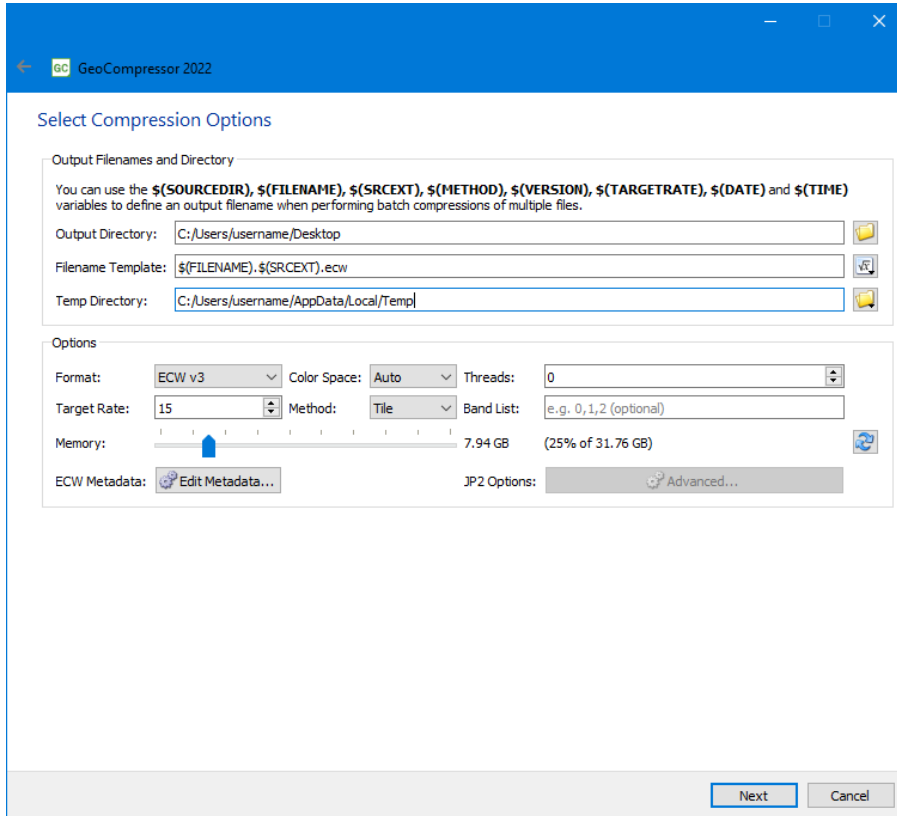


Compression Mode: Update a region within an existing ECW v3 file

1. Selection of input files is the same as in previous modes. Input files represent the new images that will be overwritten onto the destination file. Multiple input images will be mosaicked into the output file.
2. The number of input files and the compression mode (ECW Update) is displayed, as well as the output Coordinate Reference System.
3. You cannot update the output coordinate system in this mode.
4. Opacity options: you cannot override the opacity channel options in this mode:
 - a. If an opacity channel exists in the output file, it will be updated with new information from the input opacity channels.
 - b. If no opacity exists in the output file, the opacity channel information in the inputs is ignored.

Select Compression Options

Compression Mode: Batch, Mosaic or Mosaic-to-many



The screenshot shows the 'Select Compression Options' dialog box in GeoCompressor 2022. The dialog is titled 'Select Compression Options' and has a blue header bar with the GeoCompressor logo and title. Below the header, there are three main sections: 'Output Filenames and Directory', 'Options', and 'Buttons'.

Output Filenames and Directory: This section contains three text input fields with browse buttons on the right. The first field is 'Output Directory' with the value 'C:/Users/username/Desktop'. The second field is 'Filename Template' with the value '\$(FILENAME).\$(SRCEXT).ecw'. The third field is 'Temp Directory' with the value 'C:/Users/username/AppData/Local/Temp'. Above these fields is a note: 'You can use the \$(SOURCEDIR), \$(FILENAME), \$(SRCEXT), \$(METHOD), \$(VERSION), \$(TARGETRATE), \$(DATE) and \$(TIME) variables to define an output filename when performing batch compressions of multiple files.'

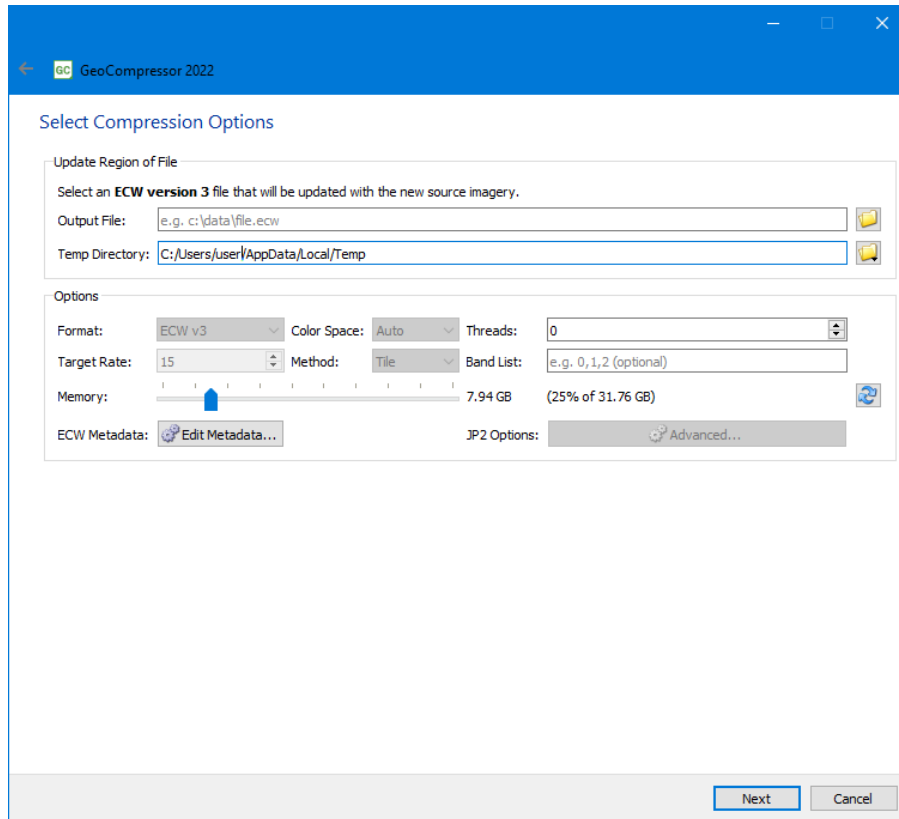
Options: This section contains several settings:

- Format:** A dropdown menu set to 'ECW v3'.
- Color Space:** A dropdown menu set to 'Auto'.
- Threads:** A numeric input field set to '0'.
- Target Rate:** A numeric input field set to '15'.
- Method:** A dropdown menu set to 'Tile'.
- Band List:** A text input field with the value 'e.g. 0,1,2 (optional)'.
- Memory:** A progress bar showing '7.94 GB' (25% of 31.76 GB).
- ECW Metadata:** A button labeled 'Edit Metadata...'.
- JP2 Options:** A button labeled 'Advanced...'.

Buttons: At the bottom right of the dialog are two buttons: 'Next' and 'Cancel'.

1. The output filename and directory can be defined using standard file locations or with templates, which are powerful ways to ensure the output names are unique.
2. Select a pre-defined template by clicking the "formula" button on the right side of the input template string, or create your own. E.g., "\$(FILENAME).\$(SRCEXT).\$(DATE).ecw" for an input file called "australia_1980.tif" will produce an output file named "australia_1980.tif-2021-09-07.ecw."

Compression Mode: ECW v3 Update



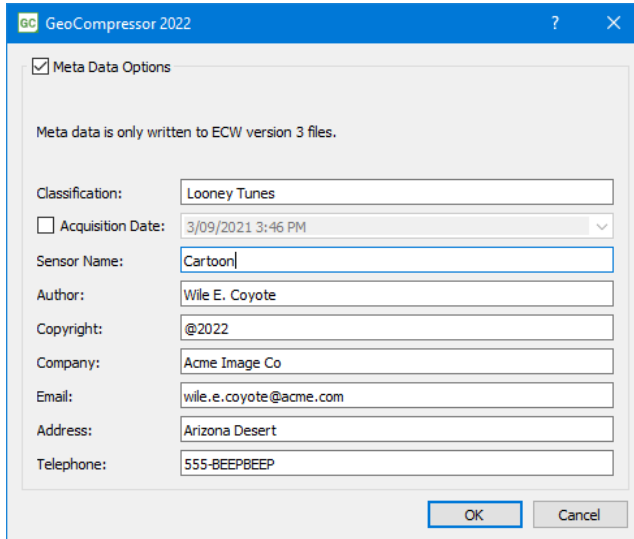
The screenshot shows the 'Select Compression Options' dialog box in GeoCompressor 2022. The title bar is blue with the 'GC' logo and 'GeoCompressor 2022'. The main area is white with a blue header 'Select Compression Options'. Below this, there's a section 'Update Region of File' with a sub-header 'Select an ECW version 3 file that will be updated with the new source imagery.' It contains two text boxes: 'Output File:' with the value 'e.g. c:\data\file.ecw' and 'Temp Directory:' with the value 'C:/Users/user/AppData/Local/Temp'. Below this is an 'Options' section with several controls: 'Format:' set to 'ECW v3', 'Color Space:' set to 'Auto', 'Threads:' set to '0', 'Target Rate:' set to '15', 'Method:' set to 'Tile', and 'Band List:' with the value 'e.g. 0, 1, 2 (optional)'. There is a 'Memory:' slider set to 7.94 GB (25% of 31.76 GB). At the bottom of the options section are 'ECW Metadata:' with an 'Edit Metadata...' button and 'JP2 Options:' with an 'Advanced...' button. At the very bottom of the dialog are 'Next' and 'Cancel' buttons.

In “Update Region” mode, the output file is the file that is being updated with new imagery. It must exist, and be in the same CRS as the input image(s). Ensure the image is not in use by any other applications, or it will fail to update.

For all modes, the compression options are displayed next. These options control the compression process and the output file metadata.

1. The options listed are identical to the command line equivalent. Refer to the Command Line Options section below for more information.
 - a. Format: ECW or JPEG2000
 - b. Color Space: RGB, Greyscale, Multiband, or Auto (automatically selected from the source)
 - c. Threads: the number of CPU threads to use during compression (tile mode only). ‘0’ will choose an appropriate number of threads for the machine (2 x physical CPUs).
 - d. Target Rate: the target compression ratio for the output file. This higher the ratio, the more compression, but the more visually lossy the image becomes.
 - e. Method: Tile or Line. Tile-based compression is faster and uses multiple threads (this is the default). The line-based compressor is single-threaded but may be required for particular data types.
 - f. Band List: Hard code the number of bands in cases where the compressor cannot work out the correct RGB bands from the input data. The index is zero-based.
2. You can select the maximum amount of memory the compression process will attempt to use here. The default is 25% of physical RAM. For large compressors, larger values will be required.

3. Select “Edit Metadata Defaults” when encoding ECW version 3 files if you would like to add extra metadata regarding the source imagery to the files. The metadata window is shown below.



GeoCompressor 2022

☒ Meta Data Options

Meta data is only written to ECW version 3 files.

Classification:

☐ Acquisition Date:

Sensor Name:

Author:

Copyright:

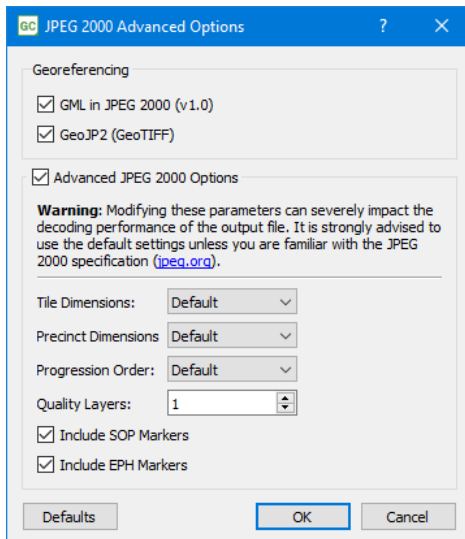
Company:

Email:

Address:

Telephone:

4. Select the “JP2 Options: Advanced...” button to customize the JPEG2000 format-specific options. You should only modify these parameters if you understand the impact they will have. Changing these parameters can impact the decoding performance of the output file and in some cases may produce a file that is not a valid JPEG2000 image.



GeoCompressor 2022

JPEG 2000 Advanced Options

Georeferencing

☒ GML in JPEG 2000 (v1.0)

☒ GeoJP2 (GeoTIFF)

☒ Advanced JPEG 2000 Options

Warning: Modifying these parameters can severely impact the decoding performance of the output file. It is strongly advised to use the default settings unless you are familiar with the JPEG 2000 specification ([jpeg.org](https://jpeg.org/jpeg2000/)).

Tile Dimensions:

Precinct Dimensions:

Progression Order:

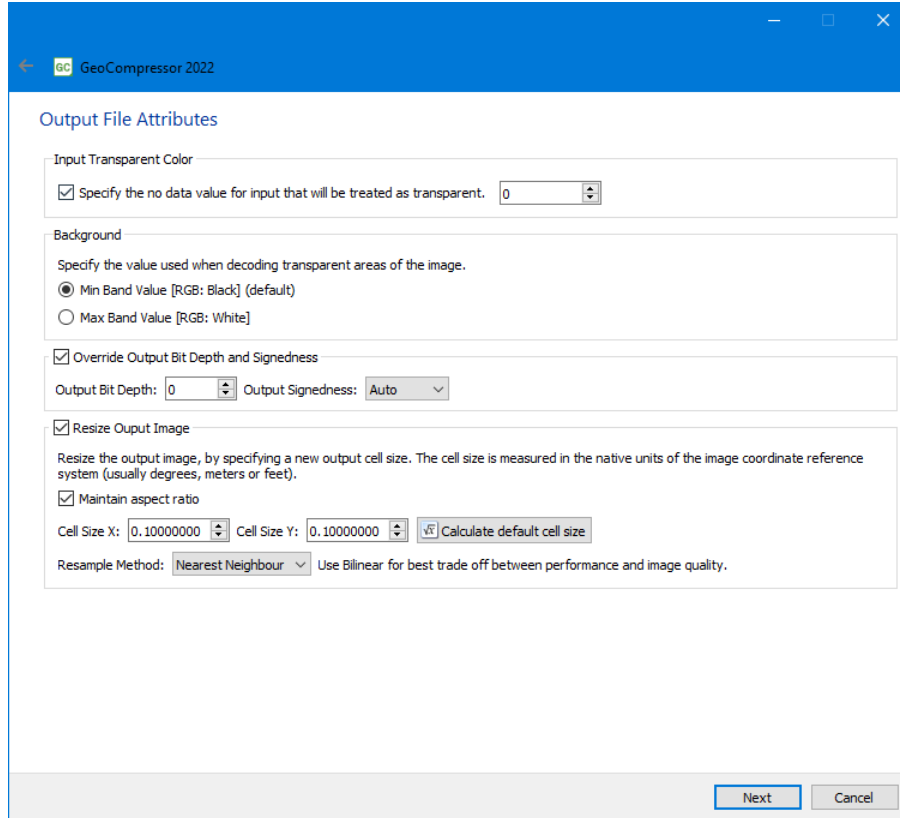
Quality Layers:

☒ Include SOP Markers

☒ Include EPH Markers

5. GML in JP2: Write an output box to the JP2 file containing the ISO standard GML in JP2 coordinate reference system reference box.
6. GeoJP2: Write an output box containing the common GeoJP2 standard (using TIFF tags) coordinate system reference box.
7. Advanced JPEG 2000 Options: adjust the layout parameters for the output JP2 file. Consult the JP2 documentation at <https://jpeg.org/jpeg2000/> for information on the specific meanings of these parameters.
8. If using the ECW Update Wizard, the output options will be disabled, as you cannot modify the compression parameters in this mode. The “Output File” represents the existing ECW v3 file as it will be updated in-place. No copies will be created, so it is recommended to back up images when testing.

Select Output File Attributes

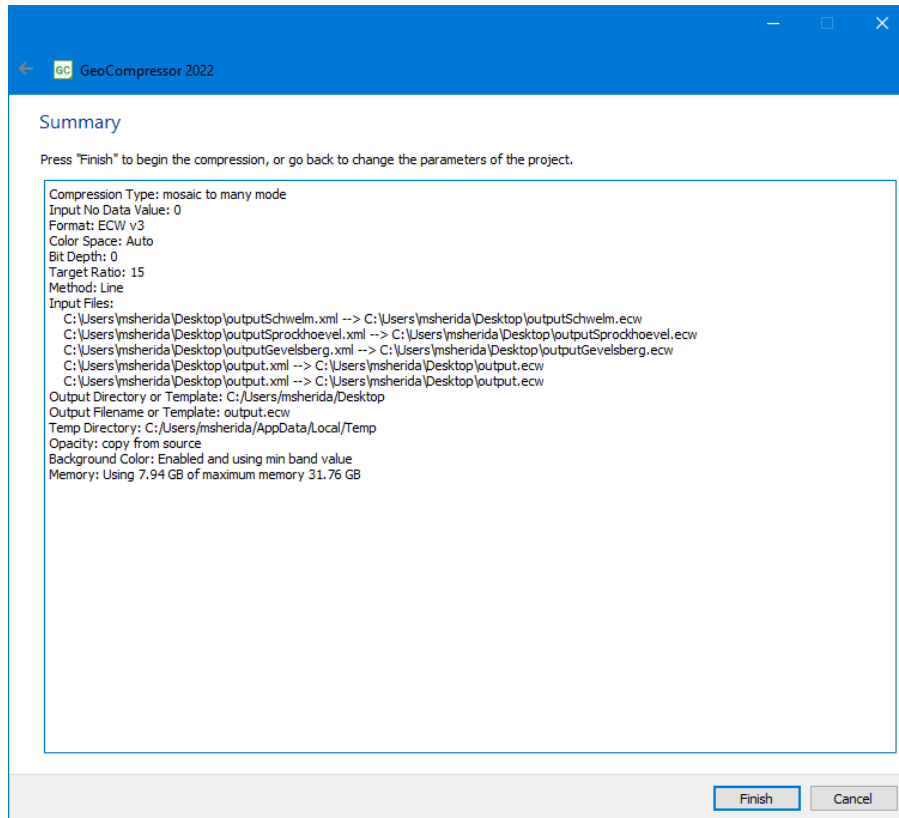


1. **Input Transparent Color:** If the input file has a transparent color (e.g., black [0,0,0] is often considered transparent), then you can specify it here. The output file will have the corresponding pixels set as transparent in the opacity channel. This is a per-band value (so individual colors cannot be specified).
2. **Background:** For ECW v3 images, when an image has large areas with no data, the ECW format does not store any data ("null blocks"), and the opacity channel always returns transparent. Applications that ignore or do not understand opacity channels will see these as blocks of black or white, depending on the input specified here. The default is '0' (black).
3. **Override Output Bit Depth and Signedness:** Specify a different output bit depth and sign from the input dataset(s). This may be required where the compressor is unable to determine the correct output range, or the input data range is smaller than the full output type (e.g., 12-bit data stored in UInt16 data type), in which case overriding this may help achieve more optimal compression rates. The compressor will not scale the data up or down (e.g., from UInt16 to UInt8) but will truncate. This is an advanced option and should be used with caution. ECW supports only unsigned 8- and 16-bit output. JPEG2000 supports from 1- to 28-bit output.
4. **Resize Output Image:** resample the output image to the new pixel/cell dimensions.
 - a. **Maintain Aspect Ratio:** For certain data types, the pixel/cell size may not be equal in both dimensions. You should usually leave this selected for most data types.
 - b. Select "Maintain aspect ratio" to ensure the cell size remains uniform. Some data types may have non-square pixels; in this case, unselect the checkbox and enter the desired output values.
 - c. Note: not all viewing applications support images with non-square cell sizes.
 - d. **Cell Size:** these values are in the default map units of the output image projection (e.g., meters, feet, etc). 1.0 would be equivalent to no resampling for a single image.
 - e. **Resize (resample) the output image** by specifying a new cell size (in the natural units of the projection). The default input cell size is pre-calculated. Select the resample method to use when reading the data. The value can be one of Nearest Neighbor, Bilinear, Cubic, Cubic Spline, Lanczos, Average, Mode, or Gauss. Nearest

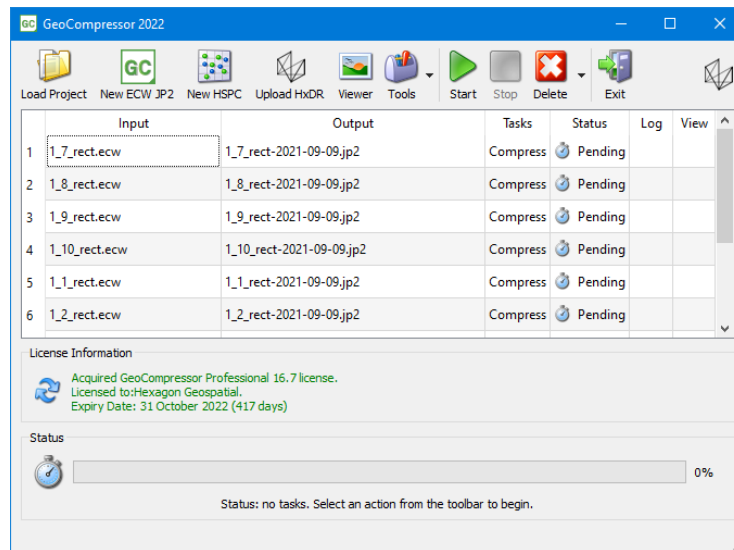


Neighbor is the lowest quality, but the fastest speed. Bilinear is recommended for most application types with medium speed and high quality. You should experiment with the method based on the data being compressed.

Summary



A summary page is shown to confirm the selected compression task settings. If this is correct, select 'Finish' to add the compression task to the job list. You can go back through the wizard pages to make any adjustments if there are corrections to be made.



Press the “Start” button to begin the compression jobs, which will then execute the jobs in sequential order. Executing jobs in parallel is not supported by the UI but can be achieved through direct command line access.

Command-line Mode

Command-line mode offers fine-grain control over the compression process. Where possible, defaults are set to ensure that in many cases only the *<input>* and *<output>* files are required; however, careful consideration of all options is recommended. When the user interface is used, the output compression log includes the command line options provided. To see the command line options, open a shell or command prompt in the “bin” directory of the software installation and type “imagecompressor.exe” without arguments. The command line options are shown below.

Note: The available options are identical for both the Windows and Linux versions of GeoCompressor unless otherwise specified.

Note: Not all options are available in command-line mode. More flexibility is achieved by using a compressor XML input file to describe the compression process.



```
C:\Program Files\Hexagon\GeoCompressor 2022\lib\bin\x64URelease>ImageCompressor.exe
```

```
Acquired GeoCompressor Professional 16.7 license.  
Licensed to:licenseserver.  
Expiry Date: 32 October 2022 (225 days)
```

```
ImageCompressor <input> <output> [Options...]  
Build: v16.7.0.100
```

Options:

```
-method (tile | line)  
    The method used for compression. Default "tile".  
-listinputformats (true | false)  
    List the available input formats. Default "false".  
-targetrate (15)  
    Target compression rate where 30 represents 30:1. Default "15".  
-opacityband (4)  
    Force opacity band definition from the input source.  
-version (2 | 3)  
    Specify ECW format version for writing. Default "2".  
-tempdir (C:\temp\)  
    Define location to write intermediate files before assembling  
    final output. Default C:\Users\ctweedie\AppData\Local\Temp.  
-threads (4)  
    Number of CPU threads used for compression. Tile based algorithm only.  
    Line will always use 1 CPU thread. Default value is CPU core count.  
-memcache (0.25)  
    Percentage of total RAM allocated for compression where 0.25 represents  
    25% of the system memory.
```

Region Options:

```
-opacity (0 | 1 | 2 | 3)  
    Manage the opacity band and/or fill transparent areas according to input  
    region. 0 and 2 will force the generation of an opacity band in the output.  
    - 0 generate an opacity band based on the input and a region definition.  
    - 1 set transparent areas to -nullvalue based on a region definition.  
    - 2 (default) does both 0 and 1.  
    - 3 Strip the opacity band from the output if there is one in the input.  
-datasetregionfile (region.txt)  
    ASCII file containing dataset coordinates defining a closed non-null  
    polygon; see the user guide for more info.  
-shapefile (vector.shp,layername,fid)  
    Esri Shapefile that defines a non-null polygon FID representing the  
    dataset active area.  
-wktfile (region.txt)  
    OGC well known text that defines a closed non-null polygon.
```

Advanced Options:

```
-bitdepth (8 | 16)  
    Force output bitdepth to 8, 16. Default to input bitdepth.  
-signed (true | false)  
    Force output to be signed. Default "false".  
-colorspace (greyscale | multiband | rgb)  
    Force color space for the output. No value will retain input colorspace.  
-units (degrees | meters | degrees)  
    Force units in ECW v2. This value is only informational.  
-bandlist (0,1,2)  
    Comma separated band indices to write to the output file. Default is null  
    which retains all bands from input.  
-logfile (C:\temp\compression.log)  
    Writes compression log file to the location.  
-loglevel (debug | info | warn | error | fatal)  
    Log level setting. Default "info".  
-genstats (true | false)  
    Generate statistics with full histogram as part of compression. Valid for  
    ECW v3 output only. Default value "true".  
-inputnodata (0)  
    Set up NoData value that will be applied to all bands.  
-nullvalue (min | max)  
    min is 0, max is the largest value available for each band. Default "min".  
-srs (EPSG:2180)  
    sets destination spatial reference system to the one specified. Note: this  
    will not reproject the image.  
-xml (c:\data\mosaic.xml)  
    Use the XML file as input for the compression process. The contents of the  
    XML overrides all other command line options.  
-json (c:\data\mosaic.json)  
    Use the JSON file as input for the compression process. The contents of the  
    JSON overrides all other command line options.  
-generatexmltemplate  
    Generate a generic compression configuration template in xml format.  
-generatejsontemplate  
    Generate a generic compression configuration template in json format.  
-validatexml (c:\data\mosaic.xml)  
    Validate xml file as input.  
-validatejson (c:\data\mosaic.json)  
    Validate json file as input.  
-compressionheaps (number greater than zero)  
    The number of heaps the compressor should use internally.  
-decompressionheaps (number greater than zero)  
    The number of heaps the ecw/jp2 decoder should use internally.  
-openoptions (option1=value1,option2=value2)  
    Passes options to the library opening the input file.
```

Input

For supported input format types, see the Supported Input Formats section. To compress any of these formats, pass in the file location, and the Image Compressor will attempt to load it via one of the supported Input Readers. If reading fails, the compressor will exit. The File location should be enclosed in quotes if the path contains spaces.

Output

Starting at version 2020 Update 2, the default format for ECW output is version 3, and the internal layout of ECW version 3 files now uses a larger internal block size (128x128). This has some side effects, such as on average, the overall compression time is faster (up to twice as fast); however, the amount of memory required to complete the compression will be approximately 80% more (as the block size is doubled). The major advantage to this change, however, is a reduction of around 10-15% in storage space. The quality and decoded image are unchanged.

Based on the output file type (ECW v2, ECW v3, or JPEG2000), different compression options will become available according to the following format capabilities.

Capability	ECW v2	ECW v3	JPEG2000
Line compression	✓	✓	✓
Tile compression	✓	✓	✗
8-bit unsigned	✓	✓	✓
16-bit unsigned	✗	✓	✓
16-bit signed	✗	✗	✓
Visually lossless	✓	✓	✓
Numerically lossless	✗	✗	✓
Null block support	✗	✓	✗
Opacity band support	✓	✓	✓
Data statistics, histogram	✗	✓	✗
RPC storage	✗	✓	✗
Custom metadata	✗	✓	✓
Region update	✗	✓	✗
Geo-referencing	GDT	GeoTIFF Tags	GML in JP2, GeoJP2
Color space support	Greyscale, RGB, Multiband	Greyscale, RGB, Multiband	Greyscale, RGB, Multiband
Largest known image	32 terapixels	48 terapixels	756 gigapixels

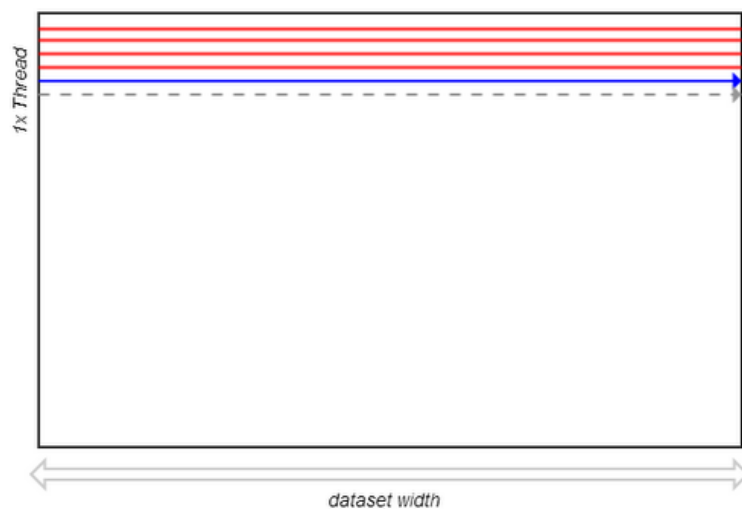
Options

-method line|tile

Image Compressor implements two types of image compression algorithms.

“Line” represents a scan line-based reader that reads across each scan line, compresses, and continues to the next scan line until the end of file. This approach has been used since ECW’s inception.

Line encoding

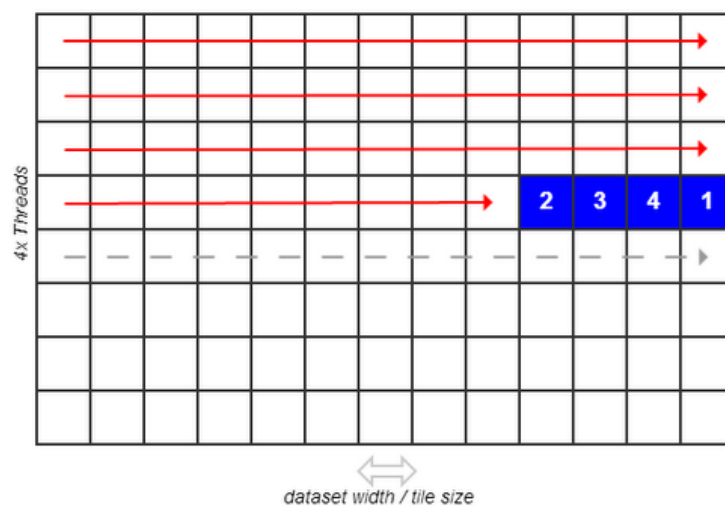


Line is inherently single-threaded and cannot scale across multiple CPU cores; however, it benefits from lower memory requirements to compress. Line is particularly suited for scan line structured input data such as Strip TIFF or file formats that do not perform well with multi-threaded readers.

Note: Line must be used for the compression of JPEG2000 files.

“Tile” is a new parallel algorithm introduced in ERDAS ECWJP2 SDK v5 that reads input data in discrete tiles across multiple reader threads specified by the -threads option. Each thread processes independently in a thread-pool across the width of the dataset and is then repeated down the image.

Tile encoding





The new algorithm results in scaling across CPU cores; however, it requires more memory to compress the same input as "line." There is a further trade-off with disk I/O, as the concurrent threads increase load and require more data to be processed than line, increasing the likelihood of reaching a disk bottleneck. It's also possible that the input data readers have not been optimized for multi-threaded reading, creating additional bottlenecks. See FAQ for formats that are not thread safe.

Fast I/O is a requirement in order to feed data to the multiple worker threads; otherwise, CPU utilization will be low, and performance may be slower than the line algorithm. Where data I/O is sufficient, the tile algorithm can be more than 400% faster than line, depending on hardware and input format used.

Both algorithms are suited for different situations. In order to determine the ideal method for your situation, benchmarks should be performed, particularly where compression speed is an important measure.

Note: Output ECW files are binary identical irrespective of the compression method used.

-listinputformats true

Retrieves a list of supported input formats across all data readers. See Supported Formats section for a formatted version. This option is mutually exclusive from all others.

-targetrate 15

The target compression rate is expressed as a ratio, where 15 represents 15:1, or in other words, a 94% size reduction. The compressor uses this target to maintain image compression quality, which can result in the actual compression ratio being higher or lower than the provided value. This is expected and is commonly seen on datasets with no-data areas or other areas with spectral characteristics that are highly compressible without sacrificing quality. A 15:1 target compressed image always produces comparable image quality to any other 15:1 target image even though actual rate may vary.

For RGB or Multiband input, 15:1 and 20:1 target compression rates, respectively, produce visually lossless output results. For greyscale, 10:1 is recommended.

Note: To compress numerically lossless JPEG2000, specify a targetrate value of 1. Any other value will revert to the lossy compression type. ECW does not support numerically lossless compression, so a targetrate of less than 5 is not recommended.

-opacityband 4

The compressor will by default attempt to retain any opacity or alpha band present in the input file. This option can be used to force the detection of the opacity channel if it cannot be determined automatically from the input reader. For example, where an input 4-band image is tagged as Bands #0 to 3, where Band #3 is actually a mislabeled opacity band.

-version 2 | 3

The ECW file format has two versions available. ECW v2 is the legacy format with the widest industry support and ensures interoperability with all existing ECW software. ECW v3 was introduced in 2012 and adds new capabilities such as null blocks for improved performance and space savings, improved metadata storage, and support for uint16 cell types.

Defaults to ECW v2. This option is ignored for JPEG2000 output.

-tempdir C:\temp

As part of every compression process, GeoCompressor must maintain a certain amount of intermediate data on disk before reassembling to form the final output format. These intermediate files are all stored in the tempdir location that defaults to the system %TEMP% environment location.

For performance reasons, it's strongly recommended to write temporary data to a different physical disk than the input or output drives to maintain throughput. Users should allocate the same amount of storage at tempdir

as at the final target destination. For example, if a 10 GB ECW output is estimated, then at least 10 GB is required at the tempdir location.

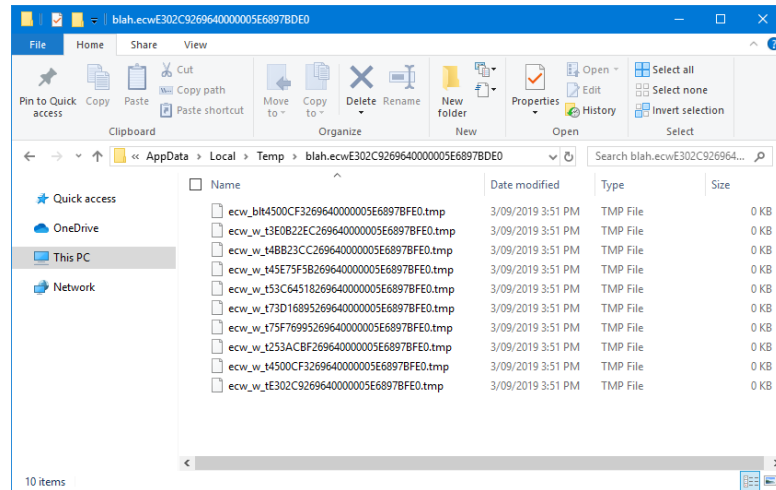


Figure 3 – Example ECW temp files written to tempdir

GeoCompressor manages the files written to this location. On completion of all compression jobs, the temp files will be deleted; however, if the compression process is interrupted, temporary files may be left on disk. If no compression is currently running, it is recommended to delete any orphaned temp files.

-threads 4

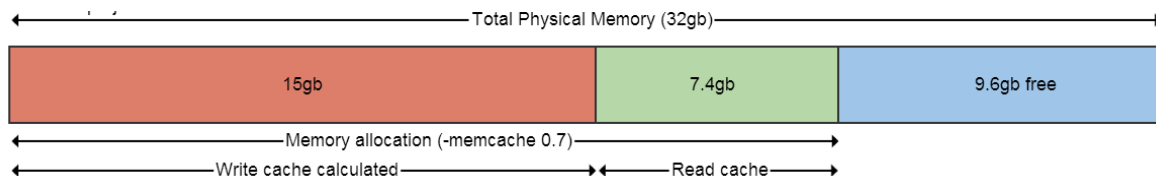
The number of threads used when compressing via the tile compression method. Defaults to the detected CPU core count on one processor but can be adjusted to determine ideal concurrency level. Like memcache, using all threads may not yield best performance, particularly if disk I/O is inadequate. On large deployments with 32 cores or more, optimal throughput may be found setting threads to 8 or 16. Testing is always recommended.

Value is ignored when -method is set to "line" as it will always be one. See -method above for more information.

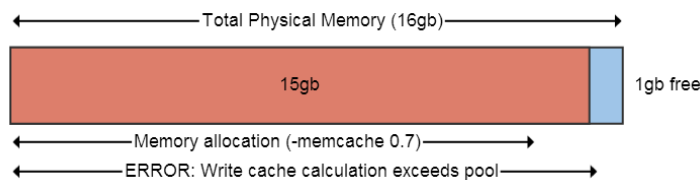
-memcache 0.25

Allocate a percentage of total system memory to a memory pool for the compression process. GeoCompressor will calculate the required memory for the Write Memory Cache based on the input image dimensions and compression method specified (see table). Any remaining memory in the pool will then be assigned to the Read Memory Cache.

On a machine with 32 GB system memory, for example, a memcache of 0.7 can be visualized as follows:

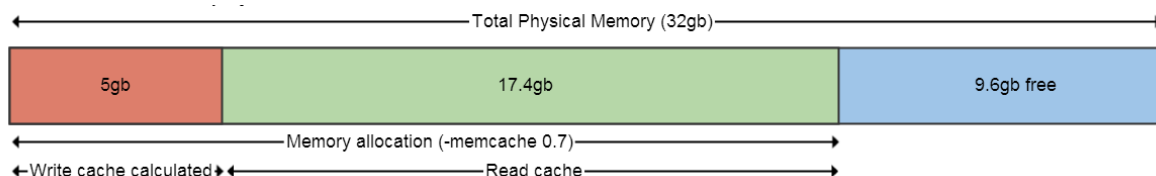


If the same input image was then compressed on a machine with only 16 GB system memory, the same memcache value of 0.7 would yield:

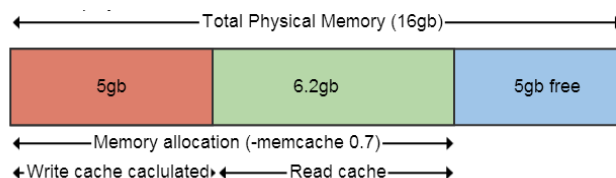


Which would result in an error since the fixed Write Memory Cache requirement of 15 GB is larger than the whole memory pool budget (16 GB * 0.7 = 11.2 GB). To remedy, a memcache value of 0.95 (15.2 GB) is needed, which would require careful management to ensure the system does not begin to swap to pagefile. Adding additional memory would be recommended in this situation, or switching to the line compression method to reduce the memory requirement but potentially sacrificing throughput performance.

To compress a smaller hypothetical image on the 32 GB machine with the same memcache value of 0.7, the write cache is reduced and the read cache size increases dramatically to take the remainder of the memory pool.



The smaller image will now also be able to be run on the 16 GB machine as the Write requirements no longer exceed the memcache budget:



The size of the Read Memory Cache is secondary, and increasing this value does not always yield an increase in performance. In other words, if a -memcache of 0.25 is sufficient to compress your input image, it is not always advised to specify -memcache of 0.9. It is dependent on the type of input format, how well that dataset driver responds to a higher read memory cache, and disk I/O performance. In some cases, performance could drop given additional overheads managing a large cache versus simply reading data from disk.

It is always recommended to leave a memory buffer to ensure long-running compression jobs do not reach 100% memory usage. GeoCompressor can fluctuate within a degree of tolerance of the memory budget before purging, so assigning 99% memory is never advised. This buffer size also depends on other applications that may be running on the machine, which is not considered by default.

The following table highlights the fixed write memory cache requirements for different conversion types.

Input image size	Bands	Gigapixels	Output Format	Compression Method	Write Memory Cache (MB)
50,000 x 50,000 px	3	2.5	ECW v2	Tile	247
	3	2.5	ECW v2	Line	136
100,000 x 100,000 px	3	10	ECW v2	Tile	443
	3	10	ECW v2	Line	223
500,000 x 100,000 px	3	50	ECW v2	Tile	2,007
	3	50	ECW v3	Tile	3,706
	1	50	ECW v2	Tile	704
	1	50	ECW v3	Tile	1,269
	3	50	ECW v2	Line	918
	8	50	ECW v2	Tile	5,150
	3	50	ECW v2	Tile	443
	3	50	ECW v3	Tile	801
100,000 x 500,000 px	1	50	ECW v2	Tile	182
	1	50	ECW v3	Tile	300
	3	50	ECW v2	Line	223
	3	50	JPEG2000	Line	192
	8	50	ECW v2	Tile	1,070
	3	250	ECW v2	Tile	2,017
	3	250	ECW v3	Tile	3,717
	3	250	ECW v2	Line	918
500,000 x 500,000 px	3	250	JPEG2000	Line	763
	8	250	ECW v2	Tile	5,324
	8	250	JPEG2000	Line	1,955
	3	1000	ECW v2	Tile	4,136
	3	1000	ECW v3	Tile	7,372
	3	1000	ECW v2	Line	1,792
	8	1000	ECW v2	Tile	10,547

Note:

- The drop in memory when using line over tile compression method.
- The drop in memory when the width of the input changes, despite identical gigapixel value.
- The large increase in memory requirements moving from 3 to 8 bands.
- The small memory drop using our default JPEG2000 profile over ECW in line mode.
- Although not shown, write memory requirements do not change depending on the -targetrate. A target of 10:1 will require the same memory as 50:1.
- ECW v3 defaults to a larger internal block size, requiring more encoding memory than ECW v2 by default; however, encoding performance will increase, and the output file size will be approximately 15% smaller.

Region Options

-opacity 0 | 1 | 2 | 3

The opacity value defines the behavior of the input region defined by -datasetregionfile, -shapefile and -wktfile.

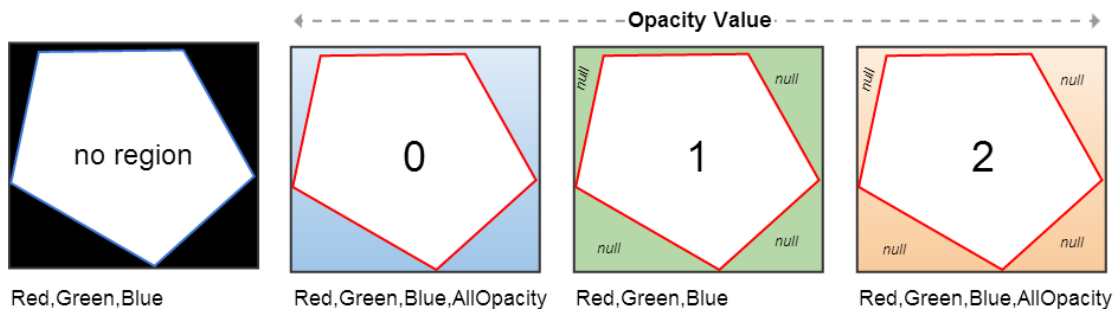
- -opacity 0: create an opacity band from the input
- -opacity 1: null blocks only (*ECW v3 only*)
- -opacity 2: opacity and null blocks (*if writing ECW v3*) [DEFAULT]
- -opacity 3: do not create an opacity band

When writing ECW v3, a value of 2 is recommended to benefit from an opacity band as well as reduced storage requirements and possible performance improvements using the null block capability. Be aware that null blocks introduce additional calculations in compression, so where the null area represents a small percentage of the total dataset area (ratio to data), or the region complexity is large (a high number of vertices), a value of 0 may compress faster. Testing is recommended to quantify this difference for your situation. The output quality will be identical between 0 and 2. ECW users will be unaware whether null blocks are present, and it will behave like any other ECW file.

When writing ECW v2 and JPEG2000 output, a value of 0 will ensure the output has an opacity band to make the background transparent.

If no region and no opacity value are specified, Image Compressor will recreate the input opacity information when detected; however, a region will override this automatic translation.

The following diagram visualizes each available option where the red polygon is the input region:



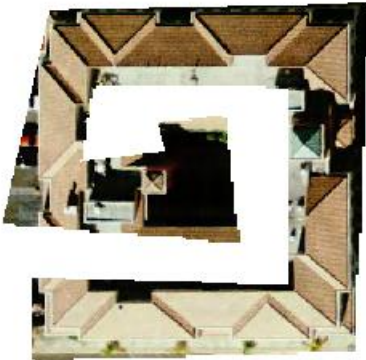
“No Region” or “Opacity 3” and “Opacity 1” have opaque backgrounds with no opacity band in the output. Opacity 1 will have reduced storage requirements due to null blocks but will still have a solid background color (defined by -nullcolor). Only opacity values of 0 or 2 will create an opacity band and allow underlying data to be shown in this area.

Default value is “2” for ECW v3 output and “0” for ECW v2 and JPEG2000

-datasetregionfile region.txt

A file on disk that represents a closed polygon in image coordinates (pixel space) defining the active or data area of the image. All area outside of this polygon will be tagged as null and/or transparent as per the -opacity option.

GeoCompressor will not clip the output image dimensions to the extents of the input region, so it should not be used to subset larger images.

Dataset coordinates	Output result
<pre> 563,1642; 1071,1681; 1040,1408; 899,1423; 915,1486; 727,1509; 758,1337; 1181,1345; 1181,1775; 625,1760; 625,1939; 1321,1916; 1353,1165; 664,1181 </pre>	

Note: Each coordinate pair must be specified on a new line with a semicolon separator.

-shapefile vector.shp, layername, fid

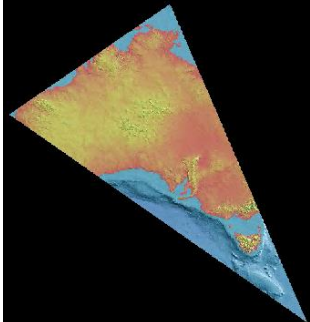
Provide an Esri® Shapefile to define the non-null or active area region defined by a closed polygon feature.

Requirements:

- Shapefile projection must match the coordinate system of the input raster
- FID identifier must map to a polygon feature

-wktfile region.txt

Identical to “-worldregionfile,” but in OGC [WKT format](#). Coordinates must match the dataset source projection.

Dataset coordinates	Output result
<pre>POLYGON((112.65148514850258721 - 23.95247524751998469,135.48118811879865575 - 8.2336633663325216,154.94257425741173506 - 53.51881188118210986,112.65148514850258721 - 23.95247524751998469))</pre>	

Advanced Options

-bitdepth 8 | 16

Force the output cell type to the provided value. If undefined, GeoCompressor will maintain the input celltype, so this option is only required in instances where the input type is incorrectly detected. Unsigned Int16 is supported in ECW v3 and JPEG2000 output.

Note: Setting -bitdepth to 8 (uint8) when the input range is uint16 will not rescale the data. GeoCompressor will merely truncate the data values and a warning will be logged.

-signed true | false

Used in conjunction with “-bitdepth 16” to set signed or unsigned int16 output. Signed INT16 is only supported in JPEG2000 output.

-colorspace greyscale | rgb | multiband

Both ECW and JPEG2000 formats support three colorspace, which are dependent on the band count. When undefined, GeoCompressor will maintain the colorspace of the input data. This option can be used in conjunction with -bandlist to subset multiband input to create RGB or greyscale output.

-bandlist 0,1,2

Explicitly select which input bands are to be compressed, indexed from 0. Default option is to compress all bands in original order.

This option allows bands to be reordered, such as BGR to RGB (e.g., 2,1,0), or to be targeted selectively when used with the -colorspace option. For example, a 7-band Multiband input can be mapped to a RGB 7,4,2 output by specifying -colorspace RGB -bandlist 7,4,2.

-logfile C:\log.txt

All information printed to the console will also be logged to file, useful for auditing or performance purposes.

-genstats true | false

When “true,” GeoCompressor will keep count of histogram bins and data statistics such as mean, mode, and standard deviation for writing to the output file. This is calculated during the compression process and is only available when writing ECW v3 files (with embedded metadata support). For performance reasons, the

calculated statistics are based on the input pixels rather than the output compressed pixels. This trade-off will be improved in a future release.

Default value “true” for ECW v3.

-inputnodata (nodata value)

Set the “nodata” value that will be applied to all bands for each input file. This represents a value in the input file that will be treated as null value when compressing. (E.g., “255” would be equivalent to white, “0” to black.) This is useful when mosaicking images to remove black or white areas around the dataset (often caused by reprojection).

-nullvalue min | max

The color definition to assign to null blocks when -opacity is set to 1 or 2. The min/max represents the smallest or largest pixel value for the specified bit depth range. For uint8, min will be 0 (black) and max 255 (white).

Default value “min.”

-interactive true | false

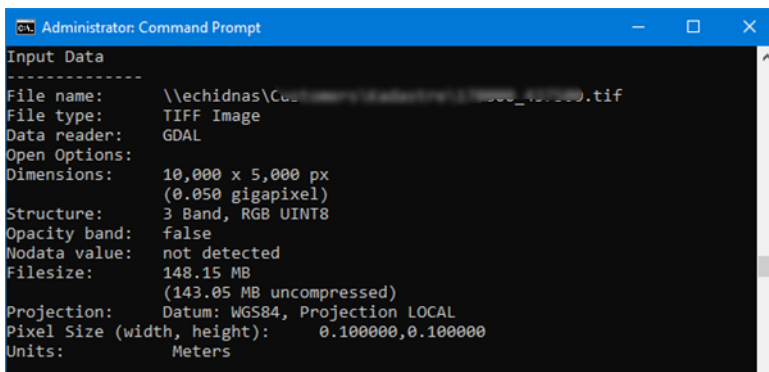
Compressor will prompt for user input when the following conditions are met:

- Insufficient memory cache
- Output file already exists

When using batch compression mode via the GUI, interactive defaults to “false.” Command line defaults to “true.”

-srs EPSG:2180

Override the input projection definition and use the specified EPSG code when writing the georeferencing information in the output file. This parameter should only be used where Image Compressor is unable to detect the source projection correctly. For example, the following IMG input lists a projection code of WGS84 / LOCAL, indicating an offset was detected but EPSG lookup failed:



```

Administrator: Command Prompt
Input Data
-----
File name:      \\echidnas\Cus..._10000.tif
File type:      TIFF Image
Data reader:    GDAL
Open Options:
Dimensions:     10,000 x 5,000 px
                (0.050 gigapixel)
Structure:      3 Band, RGB UINT8
Opacity band:   false
Nodata value:   not detected
Filesize:       148.15 MB
                (143.05 MB uncompressed)
Projection:     Datum: WGS84, Projection LOCAL
Pixel Size (width, height):  0.100000,0.100000
Units:          Meters
  
```

Where input files are known to have an EPSG code but the code was not detected correctly, the srs value (-srs EPSG:28992) can be passed into the compressor to set this value explicitly, and a relevant warning will be shown in the log.

```
Administrator: Command Prompt
Input Data
-----
File name:      \\echidnas\c001\img\2019\20190801\20190801.tif
File type:      TIFF Image
Data reader:    GDAL
Open Options:
Dimensions:     10,000 x 5,000 px
                (0.050 gigapixel)
Structure:      3 Band, RGB UINT8
Opacity band:   false
Nodata value:   not detected
Filesize:       148.15 MB
                (143.05 MB uncompressed)
Projection:     Datum: WGS84, Projection LOCAL
Pixel Size (width, height):  0.100000,0.100000
Units:          Meters

WARNING: Overriding spatial reference. Setting it to EPSG: 28992
```

Note: GeoCompressor will not perform any reprojection of the source to target projection systems. This option should not be used when the software is correctly listing the input projection code. For reprojection needs, ERDAS IMAGINE is recommended. For modifying existing ECW files, refer to the ECW Header Editor Utility.

-xml (C:\data\mosaic.xml)

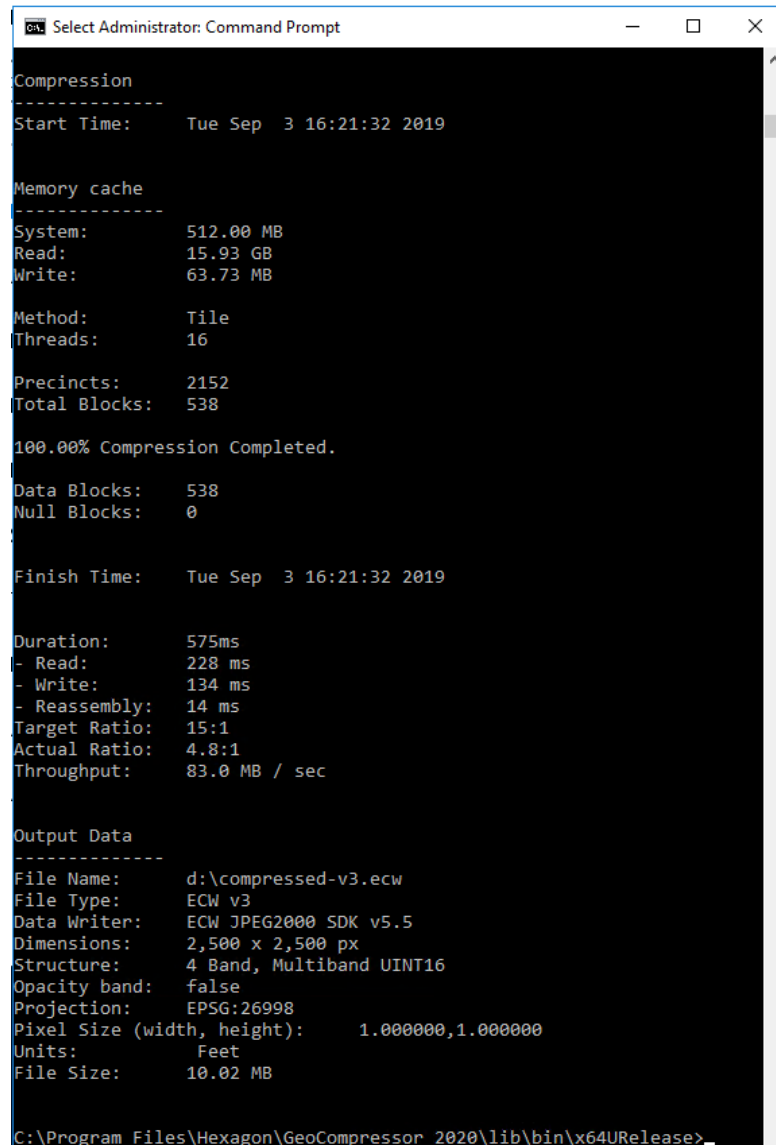
Define all compression and mosaic parameters within the XML project file and pass into the compressor. This option is exclusive of all others, including <input>, <output>, and other options.

Example usage: ImageCompressor.exe -xml c:\data\mosaic.xml

Reporting Example

GeoCompressor records a variety of metrics of the input, output, hardware, and throughput for each compression job. When reporting issues or concerns regarding processing speed, always include the full report.

The information is also written to a log file when used with the -logfile option, which includes additional information, such as the timing for each percentage increment.



```

Select Administrator: Command Prompt

Compression
-----
Start Time:      Tue Sep  3 16:21:32 2019

Memory cache
-----
System:          512.00 MB
Read:            15.93 GB
Write:           63.73 MB

Method:          Tile
Threads:         16

Precincts:       2152
Total Blocks:    538

100.00% Compression Completed.

Data Blocks:     538
Null Blocks:     0

Finish Time:     Tue Sep  3 16:21:32 2019

Duration:        575ms
- Read:          228 ms
- Write:         134 ms
- Reassembly:    14 ms
Target Ratio:    15:1
Actual Ratio:    4.8:1
Throughput:      83.0 MB / sec

Output Data
-----
File Name:       d:\compressed-v3.ecw
File Type:       ECW v3
Data Writer:     ECW JPEG2000 SDK v5.5
Dimensions:      2,500 x 2,500 px
Structure:       4 Band, Multiband UINT16
Opacity band:    false
Projection:      EPSG:26998
Pixel Size (width, height): 1.000000,1.000000
Units:           Feet
File Size:       10.02 MB

C:\Program Files\Hexagon\GeoCompressor 2020\lib\bin\x64URelase>

```

Figure 4 – Command line processing report

Key reporting elements include:

- The data reader used (GDAL, ERMLib, ECWJP2 SDK)
- The size and structure of the input data
 - The internal block size (512x512) for tile, (10000x1) for strip

- Compression used, if any (RLE, LZW, JPEG, Packbits)
- Band count, cell type, opacity, and projection information
- Uncompressed filesize and gigapixels
- Calculated memory cache settings
- Output compression results in terms of throughput, time, and actual compression rate achieved
- Duration is now broken down into sub-components to give further insights into the processing bottlenecks. The sample shows that the compression task took 9 minutes, 2 seconds to complete. Of interest are three components:
 - “Read” is the CPU time the ECWJP2 SDK compression threads are waiting for input.
 - “Write” is the CPU time the ECWJP2 SDK compression threads are performing the DWT, writing to disk, or, in general terms, performing the compression.
 - “Reassembly” is the user time needed to reassemble the file at the end of a compression. This phase will always take the last 5% of each compression job and be associated with low CPU and high levels of disk I/O reading from the tempdir location and writing to the output location.

```
Duration:      575ms
- Read:       228 ms
- Write:      134 ms
- Reassembly: 14 ms
Target Ratio: 15:1
Actual Ratio: 4.8:1
Throughput:   83.0 MB / sec
```

Figure 5 – Output timing breakdown

The above example indicates the majority of the time was waiting on data input. The actual compression (Write) was a small relative percentage of the CPU time, which is very important to differentiate. This could indicate slow or remote storage, poorly formatted input structure, bottlenecks in the mosaicking pipeline, threading issues in the data reader (GDAL), or all of the above.

An ideal target is for the Read/Write to be almost the same; however, depending on the situation, this might be unattainable. The critical piece of information to highlight is that the **ECW Compression pipeline is almost always bottlenecked by reading data from disk**. Contact Support for advice or recommendations to improve throughput if speed is an important criteria for your business.

Note: Read + Write + Re-assembly Time will not equal the Duration. See http://en.wikipedia.org/wiki/CPU_time



Usage Examples

GeoTIFF to ECW v2 using the line encoder

```
ImageCompressor c:\data\test.tif e:\data\test.ecw -method line
```

ArcInfo Binary Grid (AIG) to lossless JPEG2000 INT16

```
ImageCompressor c:\dem\w001000.adf e:\data\dem.jp2 -targetrate 1 -signed true
```

4 band BGRN UINT16 IMG HFA to RGB ECW v3 with target rate of 20:1 and band stats created

```
ImageCompressor c:\multi\bgrn.img e:\data\rgb.ecw -version 3 -bandlist 2,1,0 -targetrate 20 -genstats true
```

3 band ERMapper ALG to ECW v3 with null blocks and opacity

```
ImageCompressor c:\alg\mosaic.alg e:\data\mosaic.ecw -version 3 -opacity 2 -shapefile  
c:\alg\region.shp,region,0
```

Mosaic XML project to ECW v2 at 20:1 target ratio (defined within the XML file)

```
ImageCompressor -xml project.xml
```

Panchromatic IMG to lossy JPEG2000 with opacity defined by WKT region

```
ImageCompressor c:\jp2\data.img e:\data\output.jp2 -wktfile c:\jp2\region.txt -opacity 0
```

Multiband 7 band TIF to RGB ECW with band selection

```
ImageCompressor multiband.tif rgb.ecw -bandlist 7,4,2 -colorspace rgb
```



ECW v3 Update Example

The following example shows an existing ECW v3 file being updated with 0.15% new data. Previously, this use case required the whole project to be recompressed; however, now, with the selective algorithm, it resolves a major limitation of the format in how customers could use ECW as a “living” format as new data was received. The larger the ECW, the greater the problem this limitation becomes. The speed performing the update is comparable to the speed compressing only the updated region and, in this example, only took 13 seconds.

When an ECW v3 file is updated, the previous data is inaccessible. End users are equally unaware that the file has been updated as it behaves just like any other ECW file. All applications that can read ECW v3 can also read updated files.

Note: There is no restriction to the amount of new data that can be updated, but if more than 30% of the original is to be updated, it's recommended to use the standard compression task rather than update due to processing overheads. Update is more suited for extremely large ECW files with update regions in the few percentage (in terms of coverage).

Source Data

File name: roads.ecw
 File type: ECW Image
 Dimensions: 243,536 x 247,155 px
 (60.191 gigapixel)
 Structure: 4 Band, RGB UINT8
 Opacity band: true
 Statistics: true
 Filesize: 713.07 MB
 (168.17 GB uncompressed)
 Projection: EPSG:2240



Update data

File name: ECWUpdate.xml
 Dimensions: 25,000 x 5,000 px
 (0.125 gigapixel)
 Structure: 4 Band, RGB UINT8
 (2 image files)
 Opacity band: true
 Filesize: 0.79 KB
 (357.63 MB uncompressed)
 Projection: EPSG:2240
 Update data area: 0.149%



Result

Duration: 0 hours 0 mins 13 seconds
 - Read: 8930 ms
 - Write: 3035 ms
 - Reassembly: 400 ms
 Throughput: 36.6 MB / sec

Output Data

File Name: roads.ecw
 File Type: ECW v3
 Data Writer: ECW JPEG2000 SDK v5.2
 Dimensions: 243,536 x 247,155 px
 Structure: 4 Band, RGB UINT8
 Opacity band: true
 Projection: EPSG:2240
 File Size: 717.57 MB

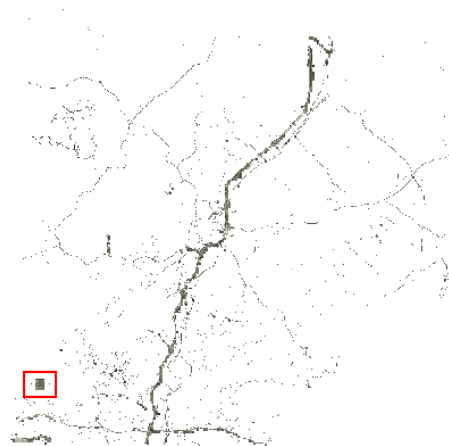


Figure 6 – Updated ECW V3 with highlighted addition



Figure 7 – Zoomed in view of updated area



Figure 8 – 1:1 Native resolution shows seamless addition on left with the original data on the right. Quality is unchanged.

XML Project File

GeoCompressor uses its own XML Project file to define image compression parameters, which is the only way to define a mosaic or update input task. If using the GUI, the wizard interfaces will create the XML for you as part of the wizard. The XML offers an easy integration point for third-party systems to define compression tasks in a programmatic way and pass in the XML to the command line tool using the -XML switch.

Note:

- Metadata and RPC elements are optional and are only relevant for ECW v3 output.
- A mosaic operation is implied when there is more than one input file listed, while a batch operation is implied when there is only one input (operation="create").
- To update a region within an existing ECW V3 file, operation must be set to update (operation="update"). The output file tells the existing ECW v3 that you wish to update with the data listed as inputs. Multiple inputs are supported, as GeoCompressor will mosaic and update on the fly.
- In most situations, GeoCompressor will default to an appropriate value; otherwise, it will exit when parsing the XML.

```
<?xml version="1.0" ?>
<imagecompressor version="1.1">

  <!--compression task: attributes describe the options for the process-->
  <compresstask operation="create|update"
    interactive="true" logfile="C:/temp/compressor.log"
    loglevel="Info" method="tile" tempdir="C:/temp"
    threads="4" memcache="0.25">

    <!--inputs: multiple files determine mosaic mode-->
    <inputs>
      <file bandlist="0,1,2" path="C:/dir/file_rgb1.ecw" zindex="0" nodata="0"/>
      <file bandlist="0,1,2" path="C:/dir/file_rgb2.ecw" zindex="1" nodata="0"/>
      <file bandlist="7,4,2" path="C:/dir/file_landsat5.ecw" zindex="2" nodata="255"/>
    </inputs>

    <!--output options-->
    <output path="C:/data/output/compressed.ecw" version="2">
      <!--output bandlist-->
      <bandlist>
        <band description="red" id="0"/>
        <band description="green" id="1"/>
        <band description="blue" id="2"/>
      </bandlist>

      <!--region definition from shapefile-->
      <region layerid="1" layername="file" name="region1"
        path="file.shp" type="shape"/>

      <!--region definition from WKT-->
      <region name="region2" path="file.wkt" type="wkt"/>
    </output>

    <!--metadata: ecwv3 only-->
    <metadata>
      <item name="classification" value="raster image"/>
      <item name="acquisitiondate" value="2013-09-12"/>
      <item name="acquisitionSENSORNAME" value="Landsat 7"/>
      <item name="author" value="John Smith"/>
      <item name="copyright" value="Intergraph"/>
      <item name="company" value="Intergraph"/>
      <item name="email" value="compressor@imagery.net"/>
      <item name="address" value="2 Abbotsford St, West Leederville WA 6007"/>
      <item name="telephone" value="(08) 9388 2900"/>
    </metadata>

    <!--Embedded RPC: ECW v3 only-->
    <rpccdata>
      <errorbias>12.23</errorbias>
      <errorrandom>0.48</errorrandom>
      <lineoffset>3522</lineoffset>
      <sampleoffset>4406</sampleoffset>
      <latitudeoffset>35.2298</latitudeoffset>
      <longitudeoffset>-80.8601</longitudeoffset>
      <heightoffset>186</heightoffset>
    </rpccdata>
  </compresstask>
</imagecompressor>
```



HEXAGON

```
<linescale>3639</linescale>
<samplescale>4421</samplescale>
<latitudescale>0.0780</latitudescale>
<longitudescale>0.1020</longitudescale>
<heightscale>501</heightscale>

<!--the below four coefficient need to have 20 comma separated parameters -->
<linenumeratorcoefficients>
  -2.082755E-03,-2.790828E-02,...-5.147002E-07
</linenumeratorcoefficients>
<linedenominatorcoefficients>
  1.000000E+00,-3.544814E-05,...+2.449699E-07
</linedenominatorcoefficients>
<samplenumeratorcoefficients>
  -7.918007E-05,+1.000871E+00,...+6.943991E-08
</samplenumeratorcoefficients>
<sampledenominatorcoefficients>
  +1.000000E+00,-2.142645E-04,...+0.000000E+00
</sampledenominatorcoefficients>
</rpcdata>

<!--options-->
<options>
  <option name="bitdepth" value="8"/>
  <option name="targetrate" value="30"/>
  <option name="colorspace" value="rgb|greyscale|multiband"/>
  <option name="genstats" value="true"/>
  <option name="blocksize" value="64"/>
  <option name="blocksizey" value="64"/>
  <option name="opacity" value="0|1|2|3"/>
  <option name="nullblocks" value="region2"/>
  <option name="nullvalue" value="min"/>
  <option name="qualitylayers" value="50"/>
  <option name="srs" value="EPSG:4326"/>
</options>
</output>
</compresstask>
</imagecompressor>
```

Supported Input Formats

Hexagon Formats

- Enhanced Compressed Wavelet ECW (.ecw) version 2 and 3
- JPEG2000 (.jp2, .jpc, .j2k, .jpf, .j2c, or .jpx) files
- ERMapper Algorithm: .alg
- ERMapper Raster: .ers
- Terrashare Raster Backdrop file .rbd

Other Industry Formats

- MrSID : Multi-resolution Seamless Image Database (MrSID)
- VRT : Virtual Raster
- GTiff : GeoTIFF
- NITF : National Imagery Transmission Format
- RPFTOC : Raster Product Format TOC format
- ECRGTOC : ECRG TOC format
- HFA : ERDAS IMAGINE Images (.img)
- SAR_CEOS : CEOS SAR Image
- CEOS : CEOS Image
- JAXAPALSAR : JAXA PALSAR Product Reader (Level 1.1/1.5)
- GFF : Ground-based SAR Applications Testbed File Format (.gff)
- ELAS : ELAS
- AIG : Arc/Info Binary Grid
- AAIGrid : Arc/Info ASCII Grid
- GRASSASCIIGrid : GRASS ASCII Grid
- SDTS : SDTS Raster
- DTED : DTED Elevation Raster
- PNG : Portable Network Graphics
- JPEG : JPEG JFIF
- MEM : In Memory Raster
- JDEM : Japanese DEM (.mem)
- GIF : Graphics Interchange Format (.gif)
- BIGGIF : Graphics Interchange Format (.gif)
- ESAT : Envisat Image Format
- BSB : Maptech BSB Nautical Charts
- XPM : X11 PixMap Format
- BMP : MS Windows Device Independent Bitmap
- DIMAP : SPOT DIMAP
- AirSAR : AirSAR Polarimetric Image
- RS2 : RadarSat 2 XML Product
- SAFE : Sentinel-1 SAR SAFE Product
- PCIDSK : PCIDSK Database File
- PCRaster : PCRaster Raster File
- ILWIS : ILWIS Raster Map
- SGI : SGI Image File Format 1.0
- EIR : ERDAS IMAGINE Raw
- DIPEX : DIPEX
- LCP : FARSITE v.4 Landscape File (.lcp)
- GTX : NOAA Vertical Datum .GTX
- LOSLAS : NADCON .los/.las Datum Grid Shift
- SRTMHGT : SRTMHGT File Format
- Leveller : Leveller heightfield
- Terragen : Terragen heightfield
- ISIS3 : USGS Astrogeology ISIS cube (Version 3)
- ISIS2 : USGS Astrogeology ISIS cube (Version 2)
- PDS : NASA Planetary Data System
- VICAR : MIPL VICAR file
- TIL : EarthWatch .TIL
- ERS : ERMapper .ers Labelled
- L1B : NOAA Polar Orbiter Level 1b Data Set
- FIT : FIT Image
- GRIB : GRIdded Binary (.grb)
- RMF : Raster Matrix Format
- WCS : OGC Web Coverage Service
- WMS : OGC Web Map Service
- MSGN : EUMETSAT Archive native (.nat)
- RST : Idrisi Raster A.1
- INGR : Intergraph Raster
- GSAG : Golden Software ASCII Grid (.grd)
- GSBG : Golden Software Binary Grid (.grd)
- GS7BG : Golden Software 7 Binary Grid (.grd)
- COSAR : COSAR Annotated Binary Matrix (TerraSAR-X)
- TSX : TerraSAR-X Product
- COASP : DRDC COASP SAR Processor Raster
- R : R Object Data Store
- MAP : OziExplorer .MAP
- KMLSUPEROVERLAY : Kml Super Overlay
- Rasterlite : Rasterlite
- MBTiles : MBTiles
- PLMOSAIC : Planet Labs Mosaics API
- CALS : CALS (Type 1)
- WMTS : OGC Web Map Tile Service
- SENTINEL2 : Sentinel 2
- MRF : Meta Raster Format
- PNM : Portable Pixmap Format (netpbm)
- DOQ1 : USGS DOQ (Old Style)
- DOQ2 : USGS DOQ (New Style)
- GenBin : Generic Binary (.hdr Labelled)
- PAux : PCI .aux Labelled
- MFF : Vexcel MFF Raster
- MFF2 : Vexcel MFF2 (HKV) Raster
- FujiBAS : Fuji BAS Scanner Image
- GSC : GSC Geogrid
- FAST : EOSAT FAST Format
- BT : VTP .bt (Binary Terrain) 1.3 Format
- LAN : ERDAS .LAN/.GIS
- CPG : Convair PolGASP
- IDA : Image Data and Analysis
- NDF : NLAPS Data Format

- NTV2 : NTV2 Datum Grid Shift
- CTable2 : CTable2 Datum Grid Shift
- ACE2 : ACE2
- SNODAS : Snow Data Assimilation System
- KRO : KOLOR Raw
- ROI_PAC : ROI_PAC raster
- ENVI : ENVI .hdr Labelled
- EHdr : ESRI .hdr Labelled
- ISCE : ISCE raster
- ARG : Azavea Raster Grid format
- RIK : Swedish Grid RIK (.rik)
- USGSDem : USGS Optional ASCII DEM (and CDED)
- GXF : GeoSoft Grid Exchange Format
- PLSCENES : Planet Labs Scenes API
- HTTP : HTTP Fetching Wrapper
- NWT_GRD : Northwood Numeric Grid Format .grd/.tab
- NWT_GRC : Northwood Classified Grid Format .grc/.tab
- ADRG : ARC Digitized Raster Graphics
- SRP : Standard Raster Product (ASRP/USRP)
- BLX : Magellan topo (.blx)
- SAGA : SAGA GIS Binary Grid (.sdat)
- XYZ : ASCII Gridded XYZ
- HF2 : HF2/HFZ heightfield raster
- OZI : OziExplorer Image File
- CTG : USGS LULC Composite Theme Grid
- E00GRID : Arc/Info Export E00 GRID
- ZMap: ZMap Plus Grid
- NGS GEOID : NOAA NGS Geoid Height Grids
- IRIS : IRIS data (.PPI, .CAPPi etc)
- DB2ODBC : IBM DB2 Spatial Database
- GPKG : GeoPackage

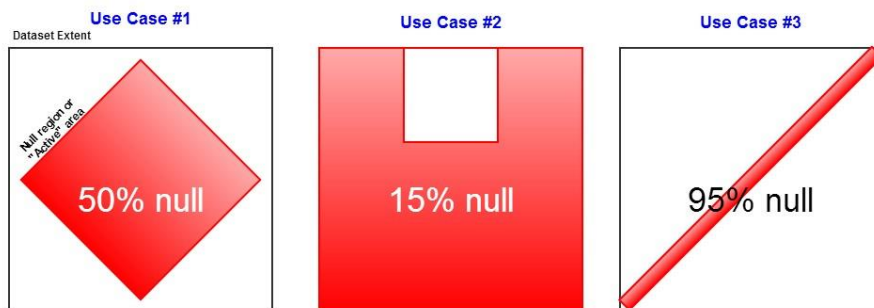
Note: When mosaicking input data, not all GDAL supported formats listed are available due to non-thread safe operations. The list above is supported for batch processing or single input only. If random errors occur when processing these input format types in tile mode, setting -method to line is recommended.

Null Block Analysis

A major improvement with ECW v3 is the introduction of null blocks that can offer further file storage savings and compression performance compared with ECW v2 or JPEG2000 without sacrificing image quality. The key criteria as to whether null blocks should be enabled is the relationship of the input data extent to the amount of null or no-data areas and the size of the input image. Generally, the higher amount of null area defined by the input region with increasing image input size, the greater the gains that enabling null blocks will provide. This feature is particularly suited to imagery corridor projects such as rivers, roads, transmission lines, pipelines, railways, or large national mapping projects.

The compressor output lists two important region metrics to help identify suitability:

1. Number of vertices, which indicates the complexity of the input region. The greater the vertices, the more expensive spatial intersection tests will be. The compressor has implemented a variety of optimizations, but where possible, a simplified region will ensure the fastest throughput. The compressor has been tested with polygons with tens of thousands of vertices; however, it's expected most use cases will on average only require vertices in the hundreds if not less.
2. Ratio of null area to data, which can be visualized in the following diagrams, where the red area is the region passed into the compressor and the white is the remaining area to be tagged as "null blocks."



Important in these examples are the first and third use cases. Both have four vertices, but clearly the percentage ratio to data is a lot higher in the third example at 95%. Therefore, null blocks will provide the greatest benefit to the third image both in terms of additional file storage savings and compression speed. The first example will still benefit and is still a good candidate for null blocks, but it will not see gains as significant. Null blocks in this way will always provide *varying* levels of optimizations, depending on input, as highlighted in the following examples.

Comparison

Corridor mapping example

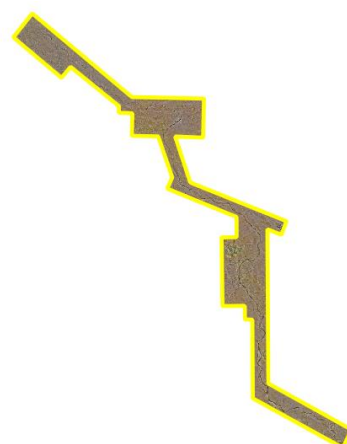
Input image

Dimensions: 372,535 x 477,806 px
(177.999 gigapixel)
Structure: 4 Band, RGB UINT8
Opacity band: true
Projection: EPSG:28350



Shapefile region (yellow)

Null Vertices: 33
Ratio to data: 89.731%



Null blocks enabled				Null blocks disabled			
Hardware				Hardware			
Platform:	Windows 7 / Server 2008 R2			Platform:	Windows 7 / Server 2008 R2		
CPU Model:	Intel(R) Xeon(R) CPU	E5410	@	CPU Model:	Intel(R) Xeon(R) CPU	E5410	@
	2.33GHz				2.33GHz		
CPU Cores:	8			CPU Cores:	8		
Memory:	16,380.00 MB			Memory:	16,380.00 MB		
Memory cache				Memory cache			
System:	512.00 MB			System:	512.00 MB		
Read:	2,319.47 MB			Read:	2,319.47 MB		
Write:	1,775.53 MB			Write:	1,775.53 MB		
Threads: 8				Threads: 8			
Precincts: 57967752				Precincts: 57967752			
Total Blocks: 14491938				Total Blocks: 14491938			
Data Blocks: 1506776				Data Blocks: 14491938			



Null Blocks: 12985162

Duration: 0 hours 53 mins 45 seconds

Target Ratio: 15:1

Actual Ratio: 209.4:1

Throughput: 210.6 MB / sec

Output Data

File Name: g:\corridor-null.ecw

File Type: ECW v3

Data Writer: ECW JPEG2000 SDK v5.2

Dimensions: 372,535 x 477,806 px

Structure: 4 Band, RGB UINT8

File Size: 3,242.55 MB

Null Blocks: 0

Duration: 3 hours 41 mins 19 seconds

Target Ratio: 15:1

Actual Ratio: 169.3:1

Throughput: 51.2 MB / sec

Output Data

File Name: g:\corridor-no-null.ecw

File Type: ECW v3

Data Writer: ECW JPEG2000 SDK v5.2

Dimensions: 372,535 x 477,806 px

Structure: 4 Band, RGB UINT8

File Size: 4,011.66 MB

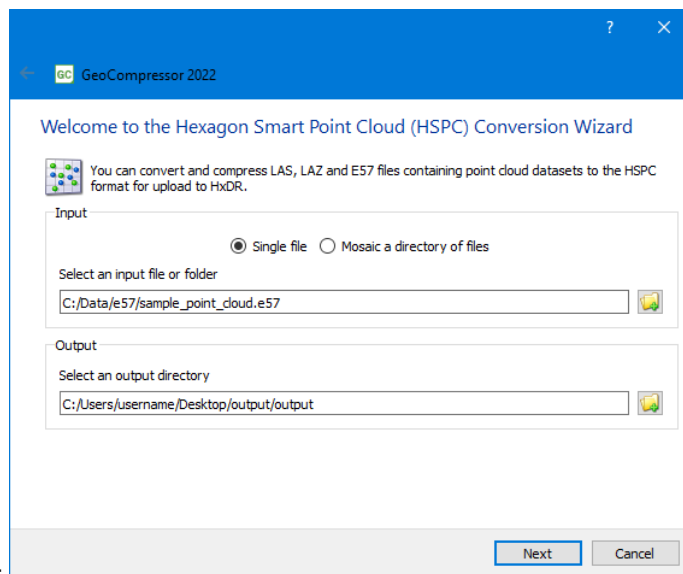
1. File savings: 769 MB (~ 19% smaller)
2. Time savings: 2 hours 48 minutes (~ 410% faster)

This example shows the strengths of enabling null blocks. It has a relatively simple input region and a high level of null data of 89%. Unlike the previous example, we can now observe significant gains to both compression speed and file savings with no degradation to image quality. Although not tested here, decompression speed will also be improved.

Point Compressor Usage

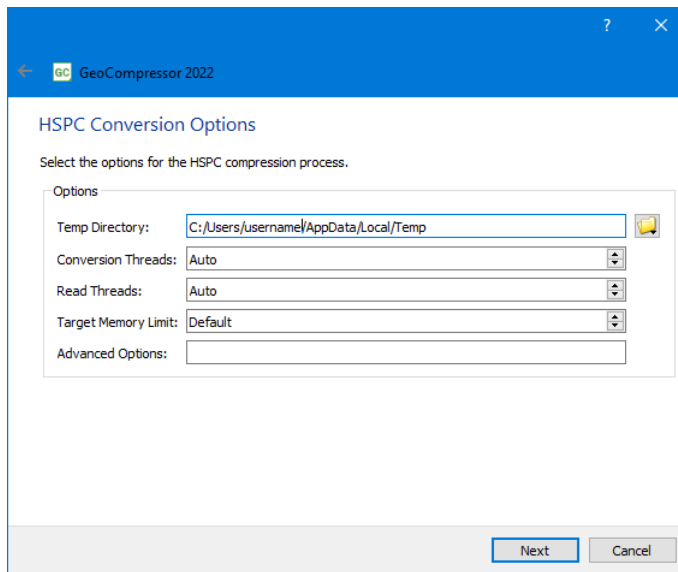
Note: Conversion to HPC format is no longer supported in the 2022 release and has been replaced by Hexagon's new HSPC format.

Wizard Mode



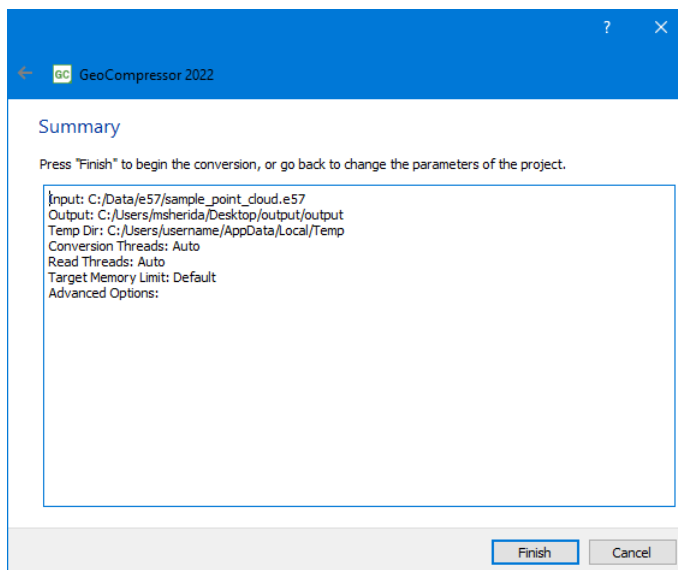
The HSPC Task wizard supports conversion of LAS, LAZ and E57 file inputs.

1. Enter in either a single input file or a folder containing files that will be merged into a single HSPC output.
2. Select Next to continue to the conversion options. The next page confirms the available conversion settings.



- a. "Temp Directory" is the location that intermediate files are written to during conversion. They are cleaned up on completion of each task. This folder selected must be empty. All items contained in the folder at the end of the compression will be deleted.
- b. "Conversion Threads" specifies the number of threads to use when performing the conversion. Auto will choose an appropriate number based on the resources of the target machine.
- c. "Read Threads" specifies the number of threads for data input. Auto will choose an appropriate number based on the resources of the target machine.
- d. "Target Memory Limit" specifies the target memory usage in Gigabytes; the default is 10 GB.
- e. Advanced Options specifies advanced parameters that can be passed to the conversion process. See the output of the conversion executable help for more information (e.g., *HSPCConverter.exe -help*).

3. Select "Next" to continue to the summary page.



4. A summary of the conversion process is shown. Select "Finish" to add the task to the main windows list.

Command-line Mode

The HSPC converter, like its Image Compressor sibling, is a transcoder of point cloud information. Its main use case is for the conversion to the Hexagon Smart Point Cloud (HSPC) format used across other Hexagon software and on the Hexagon Digital Reality platform.

For information on the command line options, refer to the help:

HSPCConverter.exe --help

```
Administrator: C:\WINDOWS\SYSTEM32\cmd.exe
C:\Program Files\Hexagon\GeoCompressor 2022\lib\bin\x64URelase>HSPCConverter.exe -help
usage:
  HSPCConverter.exe options

where options are:
  -i, --input_path <input path>          The directory containing one or more input files (default: input,)
  -f, --input_filter <input filter>      A regular expression that can be used to filter the input files
                                         (default: .*)
  -e, --extension <file extension>      File extension. (default: las)
  -r, --recursive                        Recursively scan input paths (default: false)
  -l, --file_type <file type>           Input file type (default: <auto>)
  -x, --input_extra_options <extra options> Extra options for input parsing (default: )
  -t, --temp_path <temp path>           Path for temporary data (default: temp)
  -a, --attributes <attributes>         List of attributes to include (default: <RGB or Intensity determined
                                         from the input data>)
  -n, --name <name>                     Name include in the output file (default: <basename of first input
                                         file>)
  -c, --comment <comment>               Comment to include in the output file (default: )
  -s, --node_size <node size>            Target size of output nodes (default: 262144)
  -m, --max_node_size <max node size>    Maximum size of output nodes (default: <node size> * 1.2)
  -k, --keep_duplicates                  Keep duplicate points (default: 0)
  -j, --thread_count <threads>           Number of threads for conversion, 0 means <auto> (default: 0)
  --reading_thread_count <reading threads> Number of threads for data input, 0 means <auto> (default: 0)
  --target_memory <target memory>        Target memory usage in GigaBytes (default: 10 GB)
  -o, --output_path <output path>        The destination path (default: output)
  -q, --purge_output_path                Purge destination path if exists (default: 0)
  -w, --clean_temp_path                  Clean temp folders after conversion (default: true)
  -b, --source_projection <source EPSG> Projection of the source data (default: 0)
  -p, --output_projection <output EPSG> Projection of the output data (default: 0)
  -d, --custom_projection <epsg:proj4> Custom Proj4 projection in the form <epsg>:<proj string> (default: '')
  -v, --verbose <verbose level>         Verbose level, 0=quiet, ..., 5=Debug (default: 1)
  --version                             Version/build
  -?, -h, --help                        display usage information

C:\Program Files\Hexagon\GeoCompressor 2022\lib\bin\x64URelase>
```

Reporting Example

HPSC Converter outputs details about the conversion process for the input point cloud dataset. The following example shows the output report for an E57 to HSPC conversion with default options.

```

Input files(1):
  C:/Data/e57/sample_point_cloud.e57
Build information
  Compile time      = "Aug 24 2021 00:22:10"
Configuration settings
  Environment:
    Working dir     = "C:\Program Files\Hexagon\GeoCompressor 2022\lib\bin\x64URelease"
  Input:
    Path[0]         = "C:\Data\e57\sample_point_cloud.e57"
    Input files     = 1
    Filter           = ".*"
    Extension       = ".las"
    Recursive       = false
    File type       = <auto>
    Extra options   = ""
  Conversion
    Temp path       = "C:\Users\username\AppData\Local\Temp\hspctemp"
    Attributes      = "INTENSITY:F32,R:U8,G:U8,B:U8"
    Node size       = 262144
  Output:
    Path            = "C:\Users\username\Desktop\output\output"
    Purge           = false
    Projection EPSG = 0
  Verbose level    = 5
  Target memory limit= 10 [GB]
  Input thread count = 0
  Encoding thread count = 0
  Log file         = -
  Input files list:
    [0] "C:/Data/e57/sample_point_cloud.e57"
HSPC:   Info: origin=(-165.113110,-269.866600,-2.580233); scale=(0.000010,0.000010,0.000010)
HSPC:   Info: target memory=10737418240
HSPC:   Info: batch size=1073741824
HSPC:   Info: worker count=16
HSPC:   Info: distribution worker count=16
HSPC:   Info: lodMin=13; topLodMin=18
HSPC:   Info: min lod=13
HSPC:   0/100: Run started
HSPC:   Info: Actual raw sort: elapsed: 5 [s]
HSPC:   Info: max point count per voxel=127886
HSPC:   Info: Voxel merging: elapsed 0 [s]
HSPC:   Info: totalVoxelCount=3935
HSPC:   Info: minTopTreeLod=18
HSPC:   Info: leaf array construction: elapsed 0 [s]
HSPC:   Info: max lod =26
HSPC:   Info: top tree lod=18
HSPC:   Info: read array count=2
HSPC:   Info: raw buffer count=19
HSPC:   Info: point count=7915943 filtered point count=0
HSPC:   50/100: Finished raw sort after 5[s].
HSPC:   Info: maxBufferCount: 19 bufferCountLimit: 2560 mergeThreadCount: 1
HSPC:   Info: Max raw buffer count per read: 19 memory limit for merge: 5368709120 buffer size: 94187879
HSPC:   100/100: Finished converting point nr. 7915943
HSPC:   Info: HSPC Info octree structure:
  lod[32] : size : 1
  lod[31] : size : 2
  .....
  lod[1] : size : 0
  lod[0] : size : 0
  nbOfNodes : 91
  merge phase:          elapsed: 2 [s]
  tree construction: elapsed: 0 [s]

HSPC:   Info: Rawsort:          elapsed: 5 [s].
HSPC:   Info: Distribution: elapsed: 2 [s].
HSPC:   100/100: Done
HSPC conversion finished, time elapsed: 7 [s].
Cleaning temp folder.
HSPC: info: program exited normally.

```

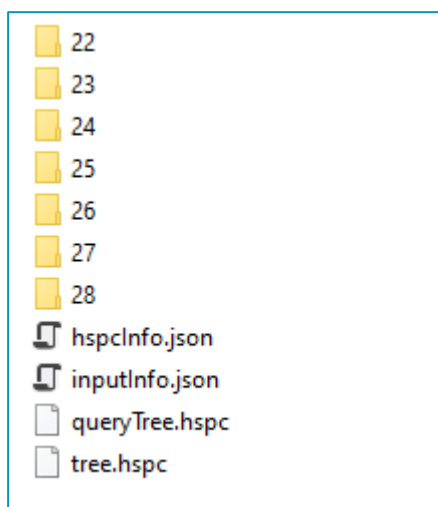
Usage Examples

Create a HSPC File from a Single LAS File

Convert a LAS file to a HSPC file with the following command line:

```
HSPCConverter.exe -i C:\Data\PointCloud\mosaic\input.las -o C:\Data\PointCloud\hspc1\
```

The output folder will contain a hierarchical set of files describing the point cloud, optimized for fast visual display in 3D applications and web-based applications (e.g., HxDR, LuciadRIA). The folder contains .json files containing the point cloud metadata and two .hspc files describing the HSPC output. The point cloud data is contained in the set of subfolders.



Create a HSPC Mosaic from a Folder of LAS files

Assuming C:\Data\PointCloud\mosaic contains a folder of LAS files, all in the same coordinate space and with the same attributes, you can create a single output HSPC mosaic of all the files with the following command line:

```
HSPCConverter.exe -i C:\Data\PointCloud\mosaic\ -o C:\Data\PointCloud\hspc_mosaic
```

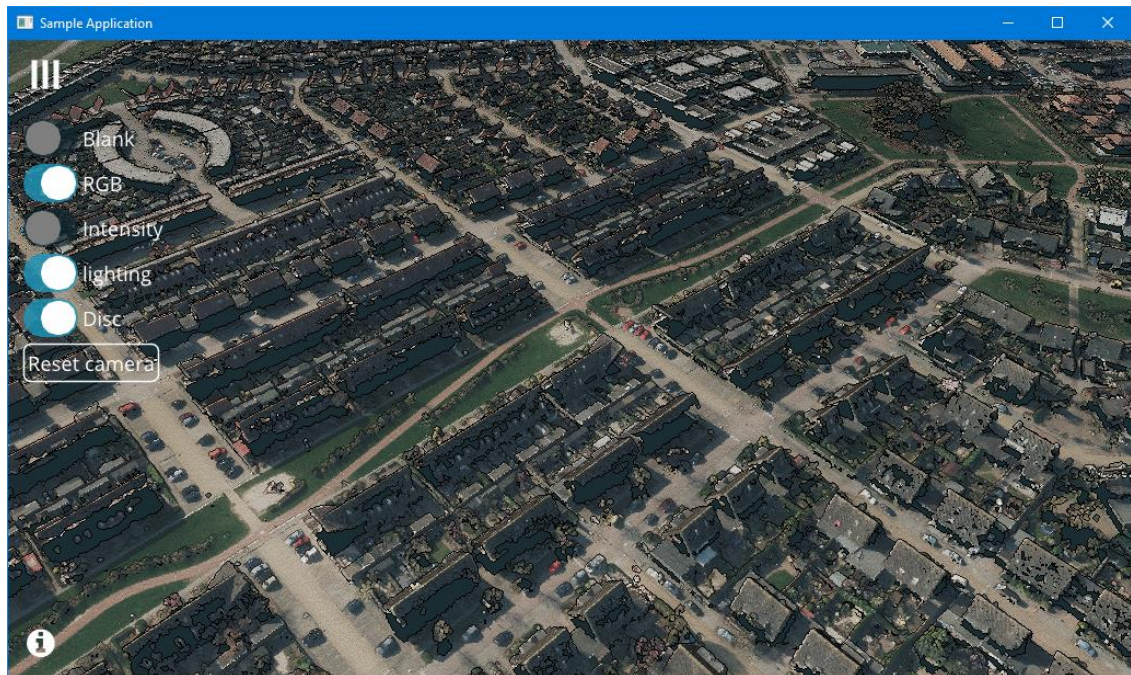
As above, the output is created in the folder C:\Data\PointCloud\hspc_mosaic.

Visualizing a HSPC Dataset

To view the output of the HSPC conversion process, use the command line:

```
HSPCViewer.exe -datadir C:\data\PointCloud\hspc_dataset\
```

Where the `datadir` is the folder name of the output HSPC dataset. The viewer will search for the `.hspc` files to determine if the dataset is supported.



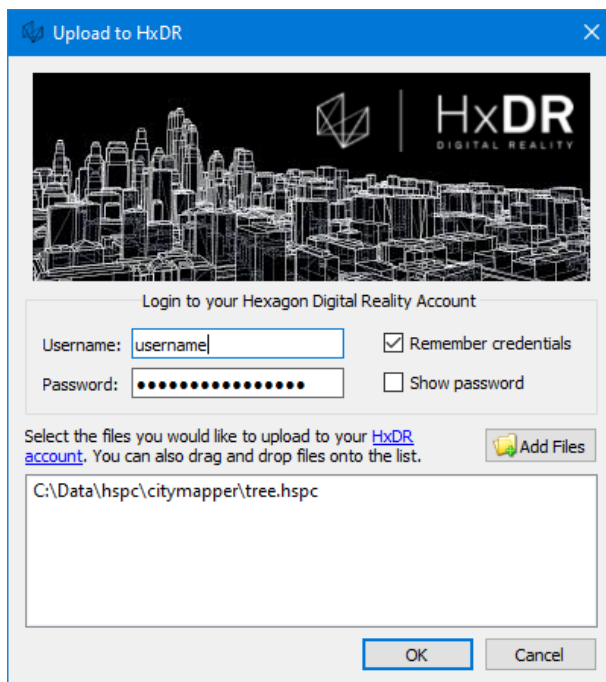
Upload Usage

Hexagon Digital Reality (HxDR) is Hexagon's cloud-based visualization and collaboration platform for spatial data and services. It is a space where the concepts that transform the way we live are placed within accurate digital representations of the real world.

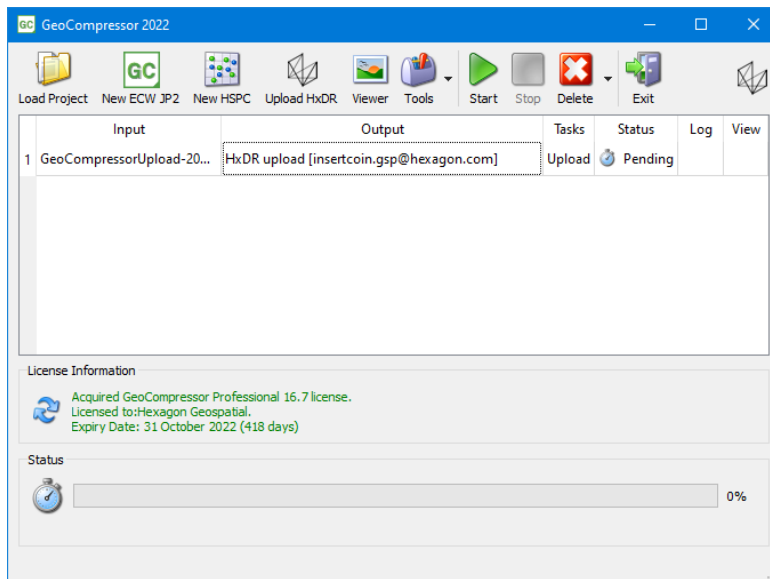
GeoCompressor now includes the ability to upload content to HxDR. A user must have a valid account in order to use the upload capability. Although a user can upload data via a web browser from their account, the GeoCompressor upload feature allows uploads larger than 5 GB and features multiple concurrent HTTP connections for faster uploads.

GUI Mode

1. On the main window for the GeoCompressor click the “Upload HxDR” button to open the upload window.



2. Enter your credentials and click “Authenticate.”
3. Enter your credentials into the “Username” and “Password” fields.
4. Optionally, select “Remember Credentials” to allow GeoCompressor to remember your account credentials.
5. Optionally, select “Show Password.”
6. Select “Add Files” and browse for files to add to the list. You can also drag and drop files from the file explorer to the download window.
7. Select “OK” to begin authorization to the HxDR platform. If the credentials are accepted, the upload window will close and the task is added to the main window.
8. The files are queued for processing, along with any other conversion tasks that were already queued. Click the Start button to begin uploading files.



9. When the upload is complete, you can click the Log File button to view the log file. To view the file online in your HxDR account, click the View button.

Command-line Mode

The functionality to upload data files to a HxDR account has been supplied as a separate command-line application so that users can script the upload of files in their own workflows. The application HxDRUploader.exe can be found in the bin directory of the GeoCompressor installation.

The format of the command is:

```
HxDRUploader.exe -u <username> -p <password> -xml <config.xml> [options]
```

Input File

The input file is an XML file describing a set of files and the upload parameters.

Sample input XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<hxduploader version="1.0">
  <uploadtask logfile="c:/temp/logfile.log">
    <inputs>
      <file path="C:\Data\hpsc\" />
      <file path="C:\Data\hpsc\queryTree.hpsc" />
      <file path="C:\Data\pointcloud\city.e57"/>
    </inputs>
  </uploadtask>
</hxduploader>
```

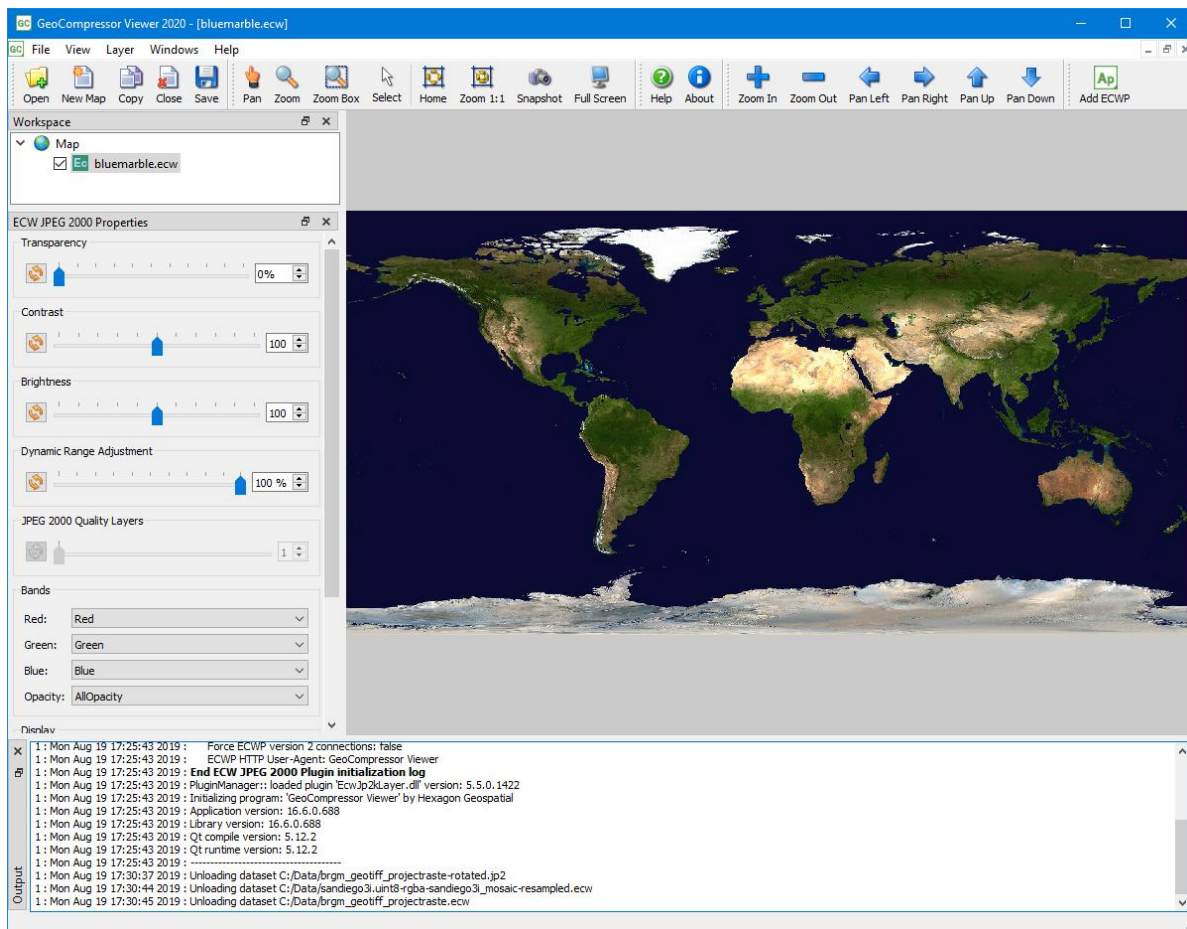
Multiple input files can be specified within the `<inputs>` element. For HSPC files, either the folder name or the queryTree.hpsc file can be specified.

GeoCompressor Viewer Usage

Overview

The GeoCompressor Viewer application is a simple ECW and JPEG2000 image viewer. It can also view imagery over the internet via the ECWP protocol from an ERDAS APOLLO server. This component does not require a GeoCompressor license to use.

The application is an MDI (multiple document interface) application that allows you to pan and zoom around very large geospatial imagery. You can add multiple layers (for images in the same projection).



Toolbar Functions

Standard Toolbar

- Open: open an image file from local the machine.
- New Map: create an empty map window.
- Copy: copy the current map window to a new map window.
- Close: close the current map window.

- Save: save the current map window, including extents and properties, to an XML file, which can be loaded later back into a new map window.

View Toolbar

- Pan: pan the image up, down, left, or right by clicking the left mouse button and dragging across the map. Right click will instigate interactive zoom node (see below).
- Zoom: interactively zoom in or out of the image by clicking the left mouse button and dragging down or up respectively.
- Zoom Box: zoom to a region of interest by dragging a rectangle on the map window.
- Select: select a feature or profile the image by clicking the left mouse button and dragging across the map. The current map SRS and cursor coordinates will display in the lower left corner of the application status bar.
- Home: zoom to the full extents of all layers.
- Zoom 1:1: zoom to the 1:1 pixel resolution (one pixel on screen equals one pixel in the source image) of the currently selected raster layer. If no layer is selected, the first raster layer found will be used.
- Snapshot : save a screen snapshot of the current map window to your desktop in PNG format.
- Full Screen: put the viewer into full screen mode. This maximizes the application so that no borders are present and takes over the entire screen. This allows the user to maximize the amount of imagery that can be displayed in the map window.

Navigation Toolbar

- Zoom In, Zoom Out: zoom in or out of the map incrementally with each click of the button.
- Pan Left, Pan Right, Pan Up, Pan Down: pan incrementally with each click of the button.

ECW or JP2 Toolbar

- Add ECWP: browse an APOLLO server and open an image via the ECWP protocol.

Viewing Images

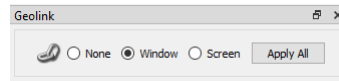
If you have registered the application to handle ECW and JPEG2000 images on install, double clicking an image in the file browser will automatically launch the application. You can open an image from the local machine from within the application by clicking the “Open” button on the main toolbar or selecting File → Open from the main menu. The selected image will appear in a new map window.

You can also drag and drop an image from the operating system file browser onto the main application, and it will open in a new window.

To add an image as a new layer in the current map view, drag and drop the image from the file browser onto the workspace window (tree view with map and layers). If no map window is currently open, a new window with the image is created. If a map window is active, the image will be added as a new layer to that map view.

Geolinking Images

The Geolink dock panel gives the user the ability to have multiple windows open and link panning and zooming operations on those windows for analysis purposes. This is only possible when the application is in windowed MDI mode (not tabbed MDI mode; see Preferences). To show the Geolink panel, right click on the toolbar or menu area and select “Geolink”:

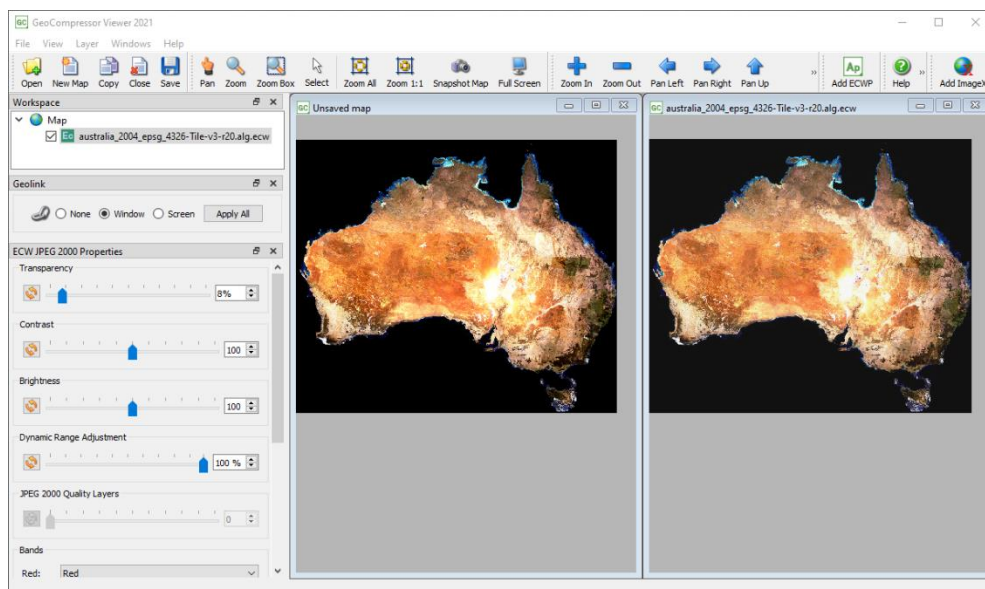


Geolink None

Windows are not linked. Panning and zooming has no effect on other windows.

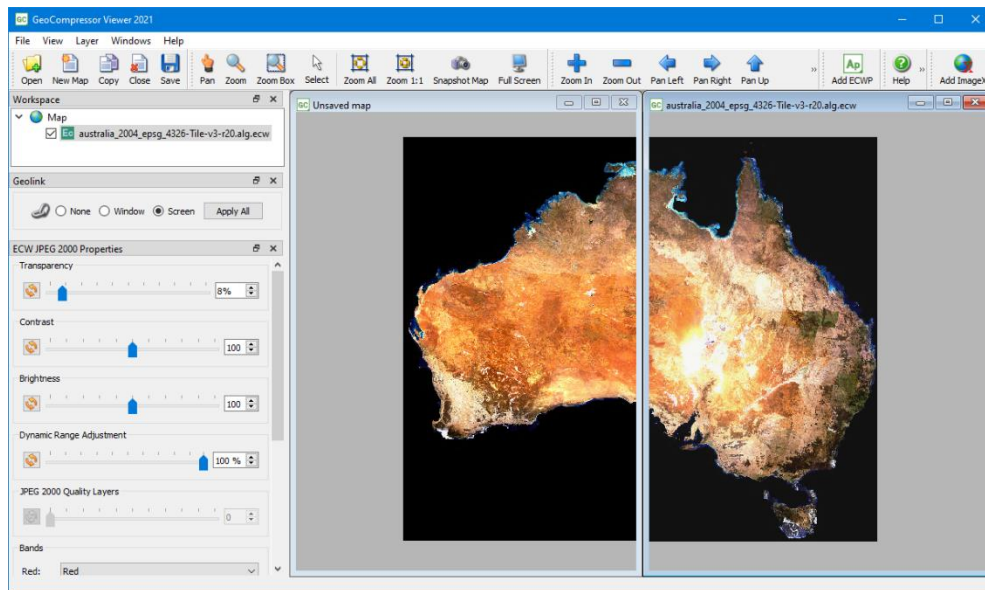
Geolink Window

Windows are linked with the same geographical extents. Panning or zooming on a window will reflect the same coordinate extents of other windows in the same coordinate system. Linked windows use the center point and scale of the source window.



Geolink Screen

Windows are linked in geographical screen space. Panning or zooming on a window will update other windows using the screen as the common coordinate system. Linked windows use the same scale as the source window, but the extents reflect the source window if it were moved in logical coordinates to the destination.



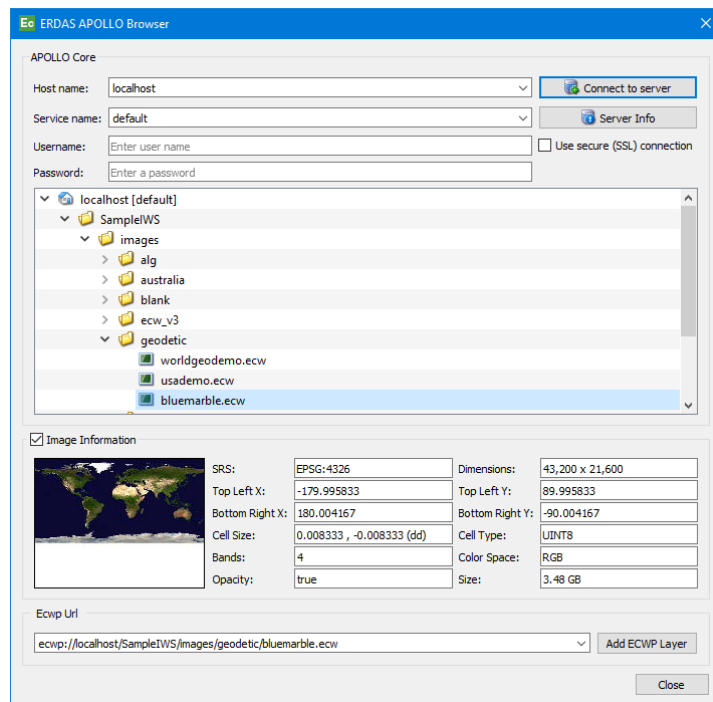
Apply to All

When the user selects None, Window, or Screen, the geolink mode is immediately applied to the current window. Selecting a mode and clicking “Apply All” will set that mode in all windows in the application area. Only windows in compatible coordinate systems will be linked to each other (when the map EPSG codes match).

Viewing ECWP images

Select “Add ECWP” from the main toolbar, or right click the “Map” icon in the Workspace view and select “Add Layer → Add ECWP Image” from the menu. You can also select “Add Layer” from the Layer menu on the main toolbar. The ERDAS APOLLO Browser window will open.

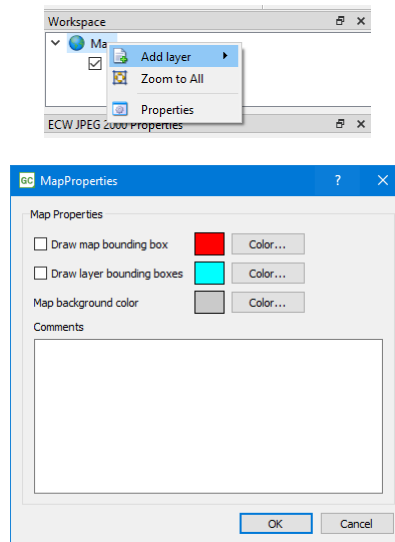
The viewer supports opening ECWP streamed images from ERDAS APOLLO.



1. Enter a server and (optional) port name (e.g., demo-apollo.hexagongeospatial.com).
2. If using SSL (ECWPS), select “Use secure (SSL) connection.”
3. Optionally, select “Server Info” to display the APOLLO server version and metadata.
4. Select the service name from the drop-down combo box (if not using the default service) from which you would like to browse images.
5. Open the root node and browse through the tree hierarchy to find the image you want to display.
6. Select an image to show its metadata and thumbnail. Double click an image to add it to the current map display and close the browser. Alternatively, select “Add ECWP Layer,” then continue browsing and add further images as new layers to the map. When done, select Close.
7. You can also paste an ECWP(S) URL directly into the URL box at the bottom of the dialog and select “Add ECWP Layer” if you know the URL of the image resource.

Map Properties

The map properties dialog allows editing of some basic map properties. To open, right click the “Map” item in the Workspace window and select “Properties.”



Current properties settings:

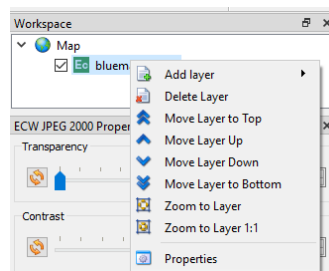
- Draw map bounding box and color
- Draw layer bounding boxes and color
- Comments about the map (saved into the map XML)

Layer Properties

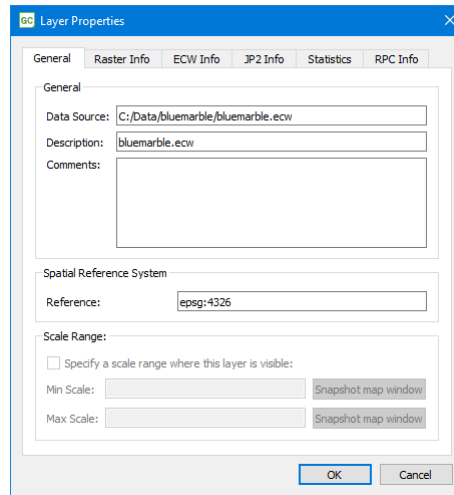
The layer properties dialog allows editing of some basic layer properties. To open, right click the layer in the Workspace window and select “Properties.”

To view the layer properties dialog:

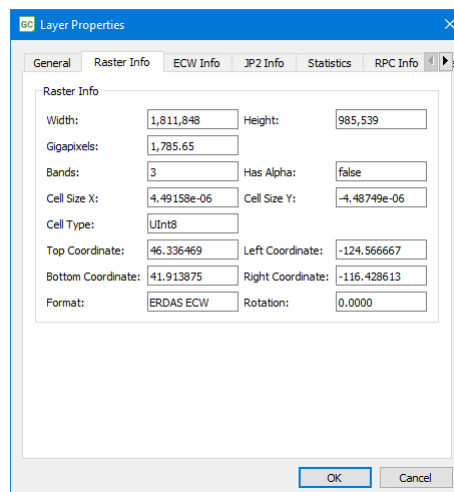
- Select the layer in the Map Workspace
- Right click on the Layer and select Properties, or
- Go to the Layer menu on the main toolbar and select Properties



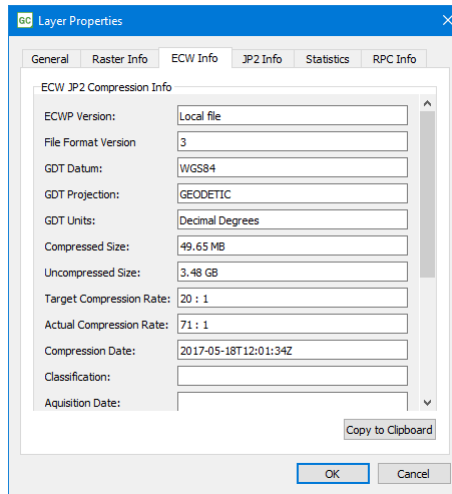
The General tab of the properties dialog shows the data source, the description, and comments. It also shows the native spatial reference system for the layer.



The Raster Info tab displays the generic raster properties for the layer, such as width, height, cell size, coordinate bounding box, and format.



The ECW Info tab displays the metadata specific to ECW files, including the ER Mapper GDT datum and projection (if defined), as well as compression information and ECW version 3 metadata such as acquisition date, classification, etc. Select "Copy to Clipboard" to copy the information in a text format to the system clipboard.



Layer Properties

General Raster Info ECW Info JP2 Info Statistics RPC Info

ECW JP2 Compression Info

ECWP Version: Local file

File Format Version: 3

GDT Datum: WGS84

GDT Projection: GEODETTIC

GDT Units: Decimal Degrees

Compressed Size: 49.65 MB

Uncompressed Size: 3.48 GB

Target Compression Rate: 20 : 1

Actual Compression Rate: 71 : 1

Compression Date: 2017-05-18T12:01:34Z

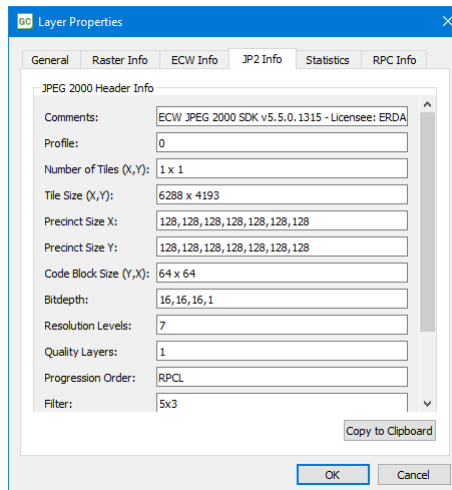
Classification:

Acquisition Date:

Copy to Clipboard

OK Cancel

The JP2 info tab displays the metadata specific to JPEG2000 files, such as the profile, tile and precinct sizes, resolutions, and quality layers, etc. Select “Copy to Clipboard” to copy the information in a text format to the system clipboard.



Layer Properties

General Raster Info ECW Info JP2 Info Statistics RPC Info

JPEG 2000 Header Info

Comments: ECW JPEG 2000 SDK v5.5.0.1315 - Licensee: ERDA

Profile: 0

Number of Tiles (X,Y): 1 x 1

Tile Size (X,Y): 6288 x 4193

Precinct Size X: 128, 128, 128, 128, 128, 128

Precinct Size Y: 128, 128, 128, 128, 128, 128

Code Block Size (Y,X): 64 x 64

Bitdepth: 16, 16, 16, 1

Resolution Levels: 7

Quality Layers: 1

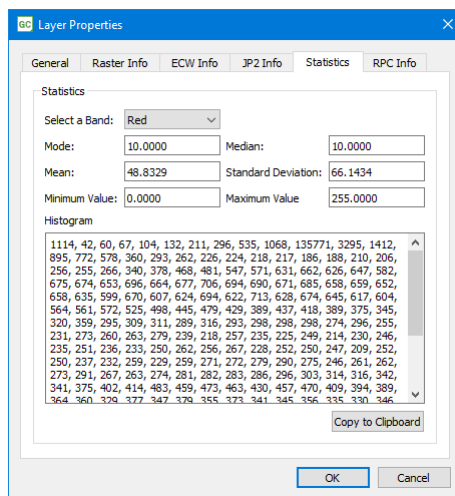
Progression Order: RPCL

Filter: 5x3

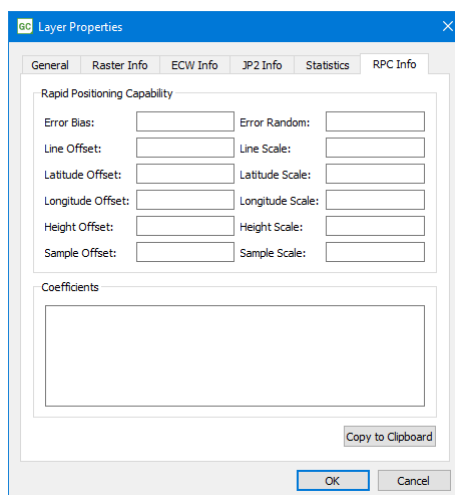
Copy to Clipboard

OK Cancel

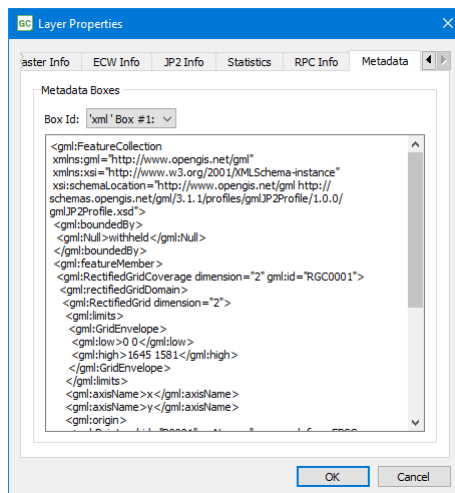
The Statistics tab displays the embedded statistics information contained in ECW v3 files (if present). You can select which band to show from the drop-down combo box containing the band descriptions. Select “Copy to Clipboard” to copy the information in a text format to the system clipboard.



The RPC Info tab displays the Rational Polynomial Coefficients and associated metadata that can be stored in ECW v3 files. Select “Copy to Clipboard” to copy the information in a text format to the system clipboard.

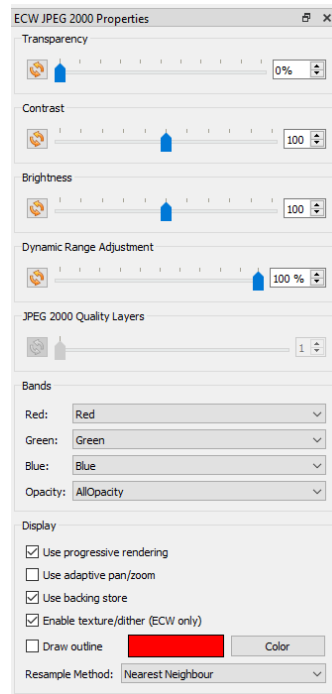


The Metadata tab displays any XML boxes that are in the file. For JPEG2000, this is typically the “GML in JP2” georeferencing metadata. You can select and copy the text to the system clipboard.



ECW Layer Properties

The ECW JPEG2000 Properties panel displays extra properties specific to the ECW or JP2 layer type display.

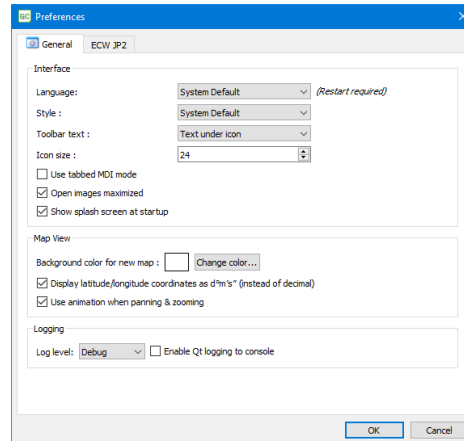


The first group of properties allow some post processing on the layer. These are transparency, contrast, brightness, dynamic range adjustment, and JPEG2000 quality layers. The second set of properties allow the user to specify band combinations and opacity (alpha) channel display. The third set of properties are custom layer properties for ECW and JP2 files. These are:

- **Use Progressive Rendering:** display imagery as it comes in from the internet. Unselecting this will cause each pan and zoom to block until all imagery has been downloaded and decoded.
- **Use Adaptive Pan/Zoom:** when in non-progressive mode, use lower-resolution decoding to speed up display.
- **Use Backing Store:** draw a low-resolution bitmap in the background to fill the display area.
- **Enable Texture/Dither (ECW only):** apply texturing to the decoded image. For some images (especially over-compressed images), this may provide a better-quality image display.
- **Draw outline:** draw a colored box around the extents of the layer.

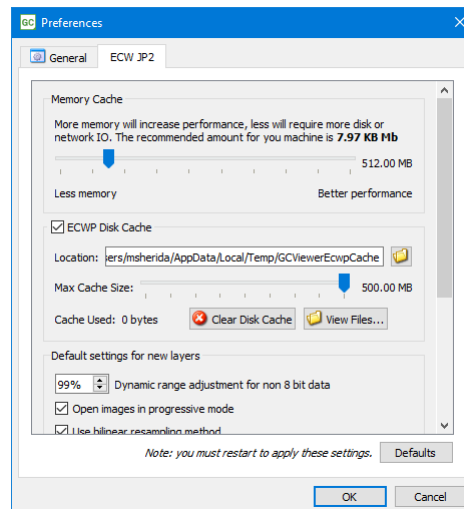
Preferences

Open the preferences dialog by selecting “File” then “Preferences” from the main menu bar.



The General tab allows you to set some preferences for the whole program.

- **Language:** specify the language that will be used in both the GeoCompressor and Viewer user interfaces. Only the GUI is localized; log files and errors will still appear in English.
- **Style:** select the icon style in the toolbars.
- **Toolbar text:** show text under or next to icons.
- **Icon size:** change the size of the main toolbar icons.
- **Use tabbed MDI mode:** image windows are aligned along the top of the main window with tabs. If unselected, traditional MDI (multiple document interface) will be enabled.
- **Open images maximized:** when in MDI mode, maximize the image window when creating a new map.
- **Show splash screen at start-up.**
- **Background color for new map:** select a map background color default.
- **Display latitude/longitude coordinate as DMS (instead of decimal degrees).**
- **Use animation when panning and zooming:** when you double click the map to zoom in, use the keyboard shortcuts (up, down, left, right), or use the mouse scroll wheel, the map will use animated panning. Unselect this to jump directly to the new map location/view.
- **Log Level:** set the log level used to display information in the output console. “Information” is the default. This can aid in determining if there are errors or issues with JPEG2000 and ECW files.
- **Enable Qt logging to console:** allow the Qt user interface toolkit to log information to the output window. This is for debugging purposes only and should be disabled for general usage.



The ECW JP2 tab is specific to the ECW imagery layer plugin. The preferences are:

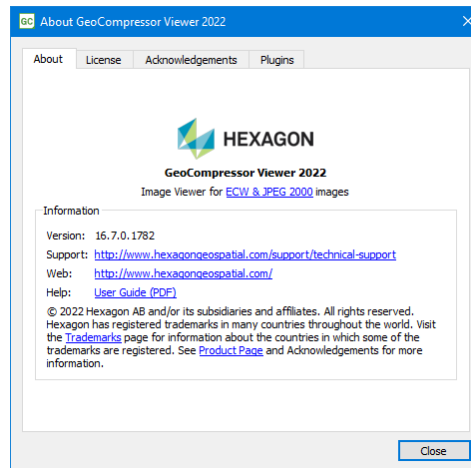
- **Memory Cache:** choose the maximum amount of system memory that will be allocated to the ECW cache.
- **ECWP Disk Cache**
 - **Location:** the location on the local file system to store the cache files. Default is the user temp dir.
 - **Max Cache Size:** maximum amount of disk space to use for the persistent cache.
 - **Clear Disk Cache:** clear the contents of the cache to reclaim disk space.
 - **View Files:** view the files in the disk cache.
- **Dynamic range adjustment for non-8-bit data:** when viewing imagery with higher bit depth per channel, select a transform percentage when scaling down to 8-bit for display. This is effectively a simple DRA (dynamic range adjustment) using a percentage left-right clip.
- **Open images in progressive mode:** display imagery as it comes in from the internet. Unselecting this will cause each pan and zoom to block until all imagery has been downloaded and decoded. The layer preference overrides this default value.
- **Use bilinear resampling:** smooth imagery when resampling using a bilinear filter. The layer preference overrides this default value.
- **Use adaptive pan/zoom:** decode at lower view resolutions for faster display. The layer preference overrides this default value.
- **Enable texturing when decompressing ECW images:** apply texturing to the decoded image. For some images (especially over-compressed images), this may provide a better-quality image display. The layer preference overrides this default value.
- **Force ECWP version 2 connections:** fallback to the old protocol (current is v3). This is not recommended and is for testing purposes only.
- **Use WinHTTP in the ECWP Client:** Use WinHTTP over WinInet. This is not recommended and is for testing purposes only.
- **ECWP HTTP User-Agent:** set the user agent for ECWP streams. Some firewalls may block certain user agents. You can work around this by setting a custom user agent here.

- Log Information Level: set the level for logging. Change this only when diagnosing problems, as it impacts performance of the ECW and JPEG2000 decoders. Default level is “Info.”

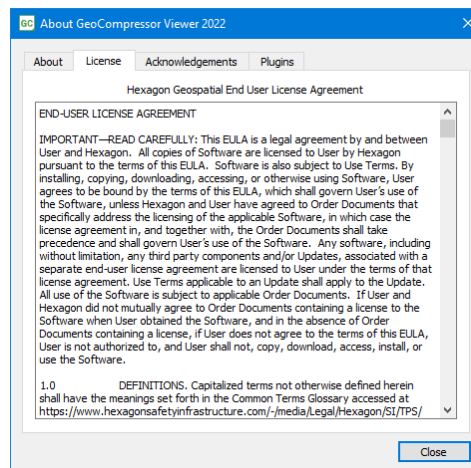
About

Select “About” from the Help menu on the main window to display the about dialog.

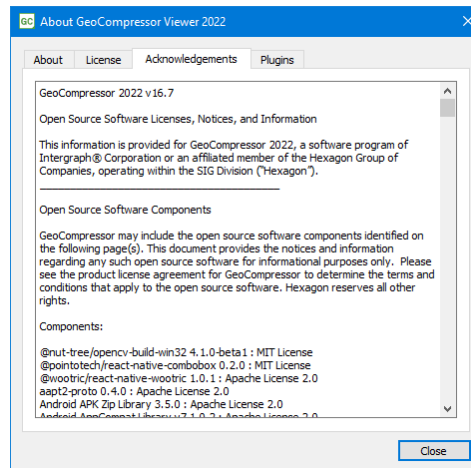
About: Version information is displayed, as well as links to the technical support web site, the Hexagon web page, the User Guide, and the Acknowledgements.



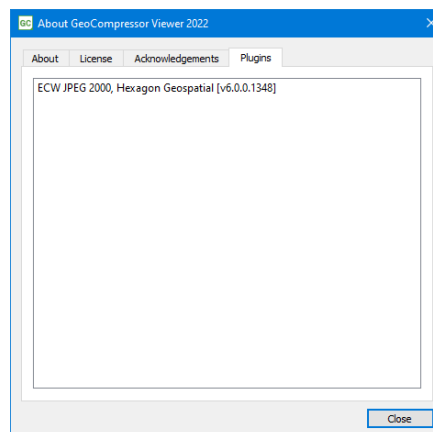
License: displays the full text of the End User License Agreement.



Acknowledgements: displays the open-source software licenses, notices, and information.



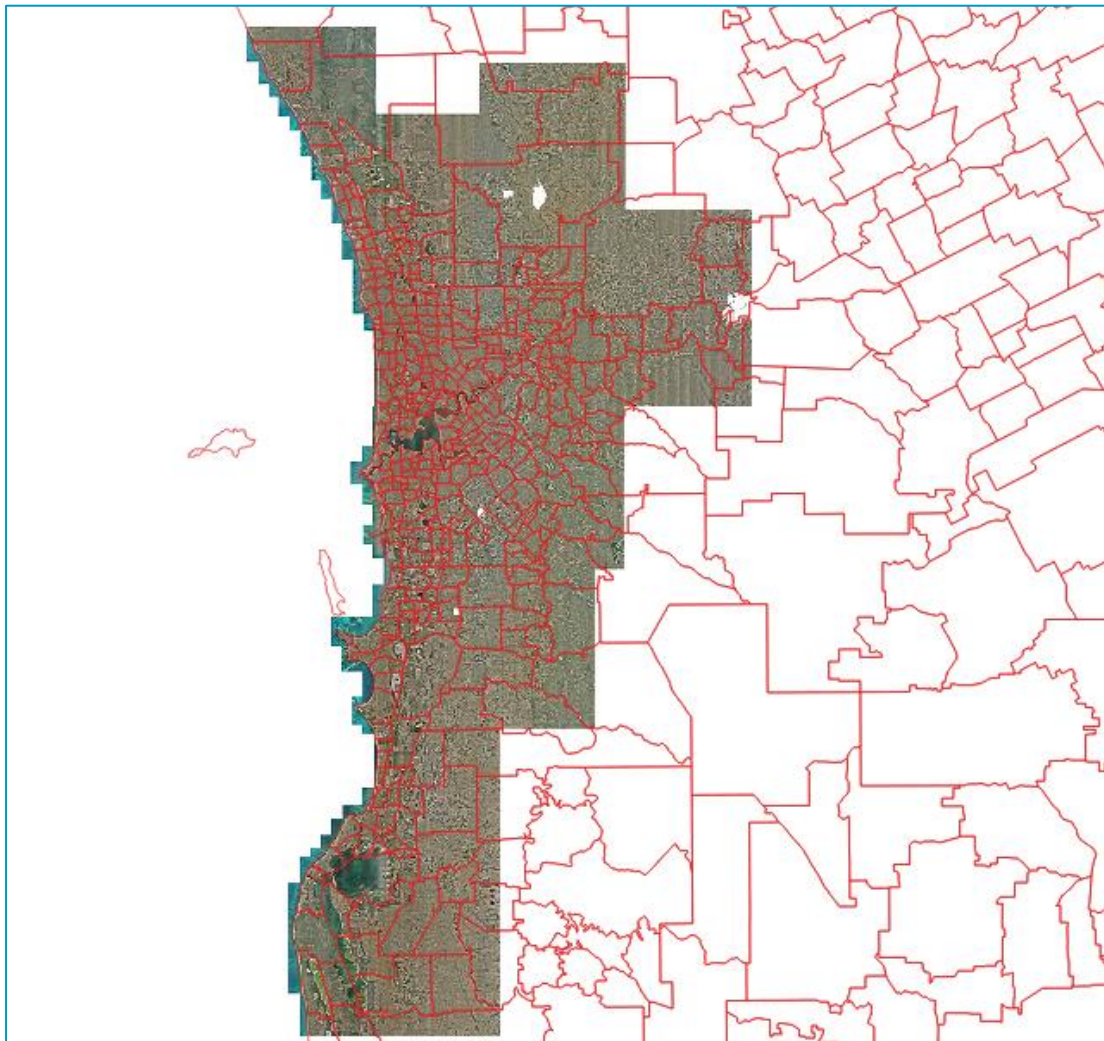
Plugins: displays version information about the installed layer plugins.



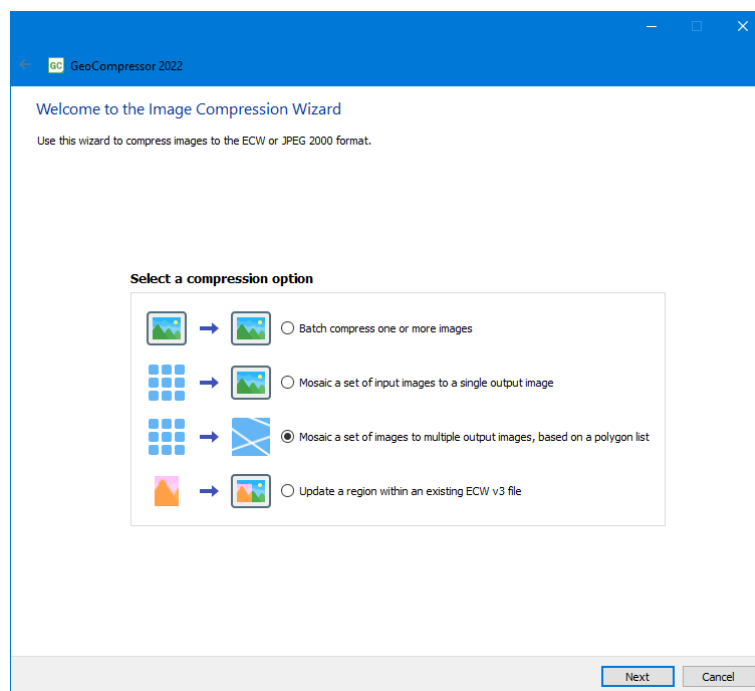
Appendix A: Mosaic to Multiple Output Workflow Example

The production of many output files from an input mosaic (or single image) is where each output file is clipped (and opacity channel generated) to a polygon defined in a shapefile. For example, you may have 1000 TIFF files as inputs to a mosaic, and you have a shapefile containing 100 polygons, where each polygon represents a ward or county boundary. The output will be 100 ECW or JP2 files, each one clipped to the shape of the polygon.

Given the following example, the input mosaic of files will be compressed to a set of output files, where each output file corresponds to a polygon in the shapefile (in this case, a ward boundary). Outputs will only be generated where there is valid data in the intersection between the polygon and input raster dataset.

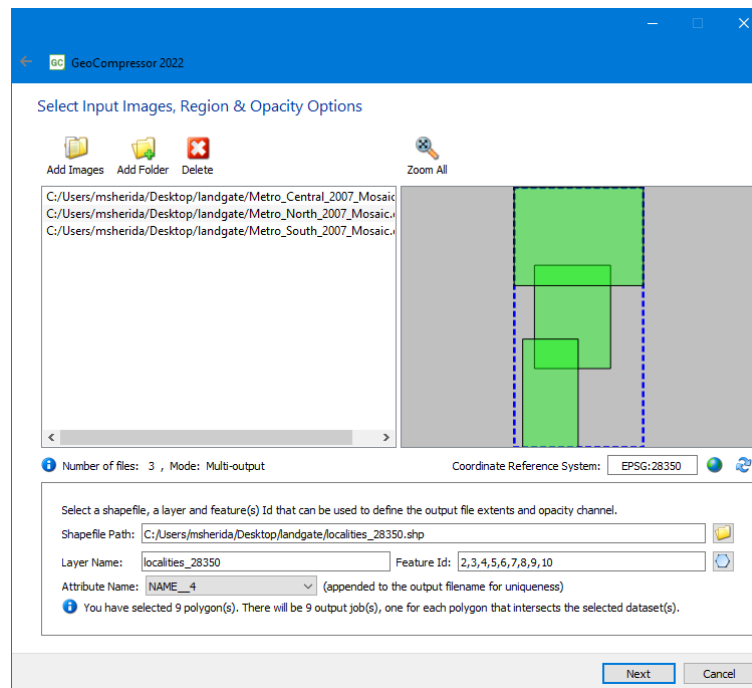


1. Open the compression wizard, by clicking “New ECW JP2” from the main toolbar.



2. Select “Mosaic a set of images to a single or multiple output images”
3. Click Next.

4. Select your input datasets on the “Select Input Images, Region & Opacity Options” page. You can drag-and-drop them from file explorer onto the input list or use the “Add Images” button to browse for files.

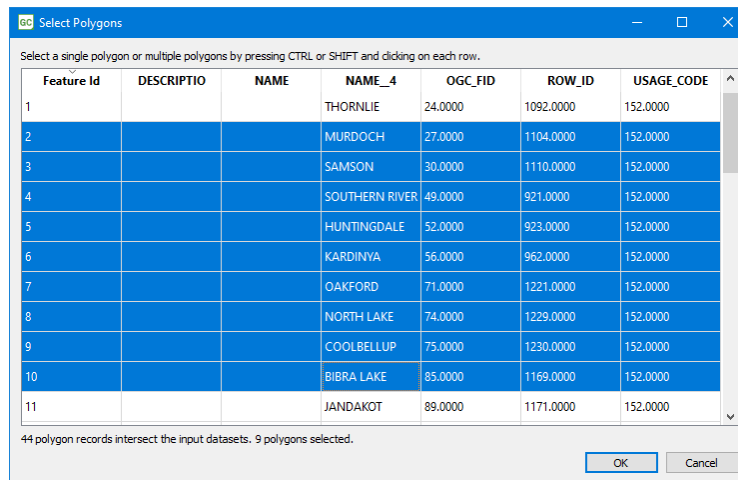


Note the footprints of the input images (green rectangles) and the output extents of the full mosaic (dashed blue rectangle).

5. The output CRS of the mosaic (and hence all the output files) is shown below the map. If the output CRS is incorrect, you can override it by typing in a new EPSG value. Click the link button to browse the definition of the CRS at <https://epsg.org/>, or click the reset button to return to the default value.

Note: overriding the EPSG value does not reproject the image; it only saves the new value in the output image header metadata.

6. Select a shapefile containing regions in the same CRS and bounds as the input images.
7. Click the “Select Polygon” button to select the polygons that will determine the number of output files and their extents.



- The information at the bottom of the list displays the total number of records found that intersect the input image list and the number of selected items. You can select multiple polygons using SHIFT-click or CTRL-click.

Note: Look for a unique attribute (column header) to use to uniquely identify the output images. It must be of string type and must have valid values. In this case, we will use the “NAME__4” attribute, which specifies the locality name.

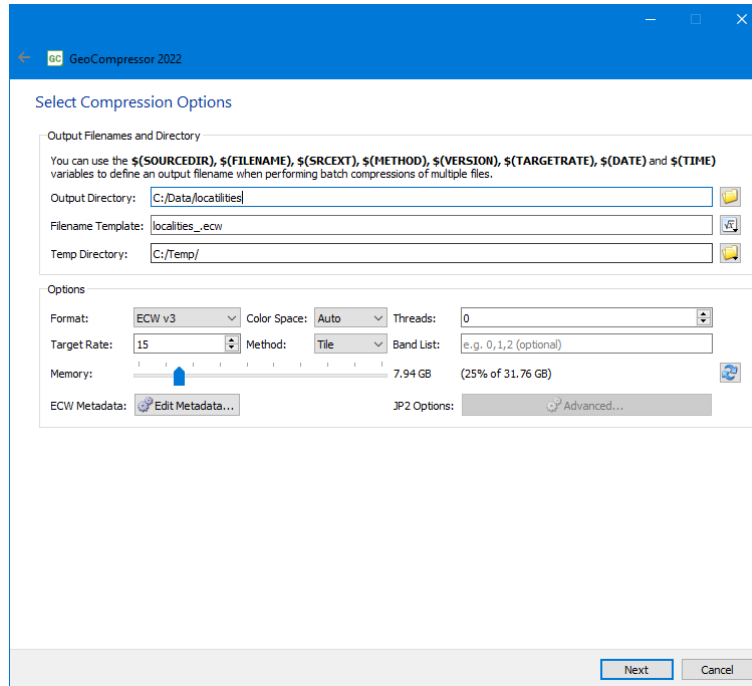
- Select OK to accept the polygon list selection. The output polygon IDs are shown in the “Feature Id” list.
- Select an attribute name from the drop-down list, in this case “NAME__4.” These attribute names (e.g., MURDOCH) will be appended to the output filename to ensure uniqueness.

Note: The information item at the bottom of the page displays the final number of jobs that will be created.

- Click Next.

12. Select the compression options, specifying the format and compression parameters. Optionally, edit the metadata and JP2 options as described in the “[Select Compression Options](#)” section.

Note: The attribute name is automatically appended to the filename template.



The screenshot shows the 'Select Compression Options' dialog box in GeoCompressor 2022. The dialog is titled 'Select Compression Options' and has a blue header bar with the GeoCompressor 2022 logo. It is divided into two main sections: 'Output Filenames and Directory' and 'Options'.

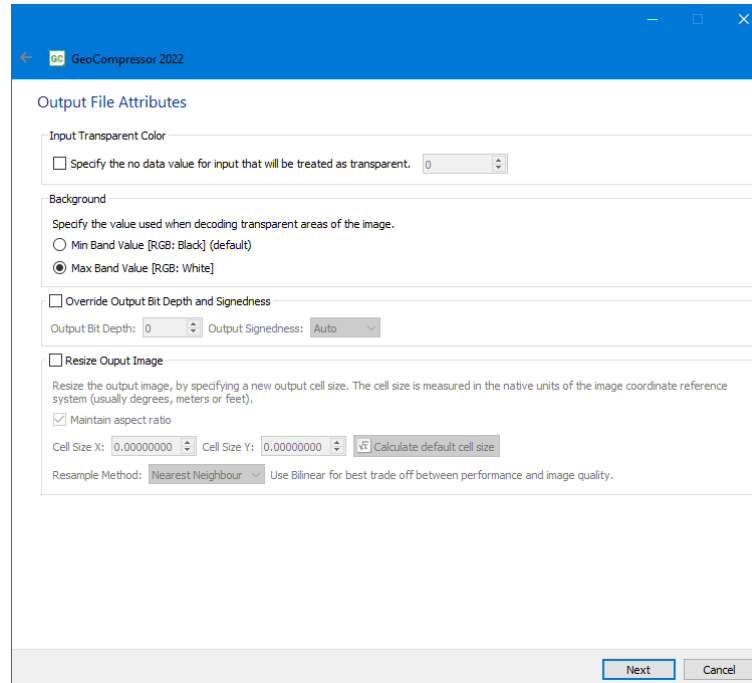
Output Filenames and Directory: This section contains three text input fields with browse buttons (folder icons) to the right. The 'Output Directory' field is set to 'C:/Data/localities'. The 'Filename Template' field is set to 'localities_ecw'. The 'Temp Directory' field is set to 'C:/Temp/'. A note above the fields states: 'You can use the \$(SOURCEDIR), \$(FILENAME), \$(SRCEXT), \$(METHOD), \$(VERSION), \$(TARGETRATE), \$(DATE) and \$(TIME) variables to define an output filename when performing batch compressions of multiple files.'

Options: This section contains several controls for compression settings. It includes a 'Format' dropdown menu set to 'ECW v3', a 'Color Space' dropdown menu set to 'Auto', and a 'Threads' spinner box set to '0'. Below these are a 'Target Rate' spinner box set to '15', a 'Method' dropdown menu set to 'Tile', and a 'Band List' text input field containing 'e.g. 0, 1, 2 (optional)'. A 'Memory' section shows a progress bar and text indicating '7.94 GB (25% of 31.76 GB)'. At the bottom of this section, there are two buttons: 'Edit Metadata...' (with a gear icon) and 'Advanced...' (with a gear icon).

At the bottom right of the dialog, there are two buttons: 'Next' and 'Cancel'.

13. Select Next to continue.

14. Select the output file attributes. In this example, we only select the “Background Color” to be white. We will not resample or override any other settings.



Output File Attributes

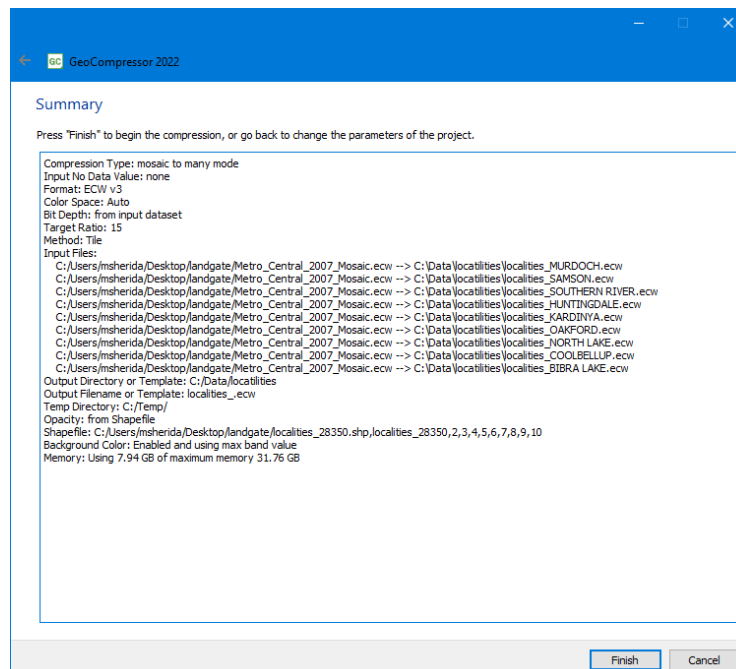
☐ Input Transparent Color
Specify the no data value for input that will be treated as transparent. 0

Background
Specify the value used when decoding transparent areas of the image.
☐ Min Band Value [RGB: Black] (default)
☒ Max Band Value [RGB: White]

☐ Override Output Bit Depth and Signedness
Output Bit Depth: 0 Output Signedness: Auto

☐ Resize Output Image
Resize the output image, by specifying a new output cell size. The cell size is measured in the native units of the image coordinate reference system (usually degrees, meters or feet).
☒ Maintain aspect ratio
Cell Size X: 0.00000000 Cell Size Y: 0.00000000
Resample Method: Nearest Neighbour Use Bilinear for best trade off between performance and image quality.

15. Select Next to continue. The summary page is presented.



Summary

Press "Finish" to begin the compression, or go back to change the parameters of the project.

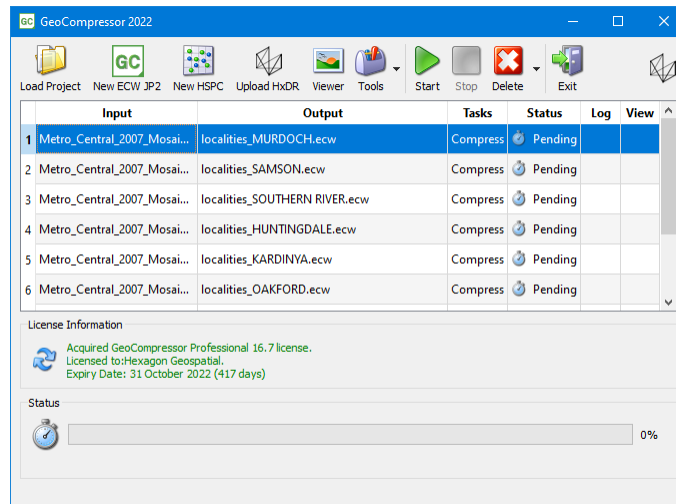
Compression Type: mosaic to many mode
Input No Data Value: none
Format: ECW v3
Color Space: Auto
Bit Depth: from input dataset
Target Ratio: 15
Method: Tile

Input Files:
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_MURDOCH.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_SAMSON.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_SOUTHERN RIVER.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_HUNTINGDALE.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_KARDINYA.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_OAKFORD.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_NORTH LAKE.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_COOLBELLUP.ecw
C:\Users\msherida\Desktop\landgate\Metro_Central_2007_Mosaic.ecw --> C:\Data\localities\localities_BIBRA LAKE.ecw

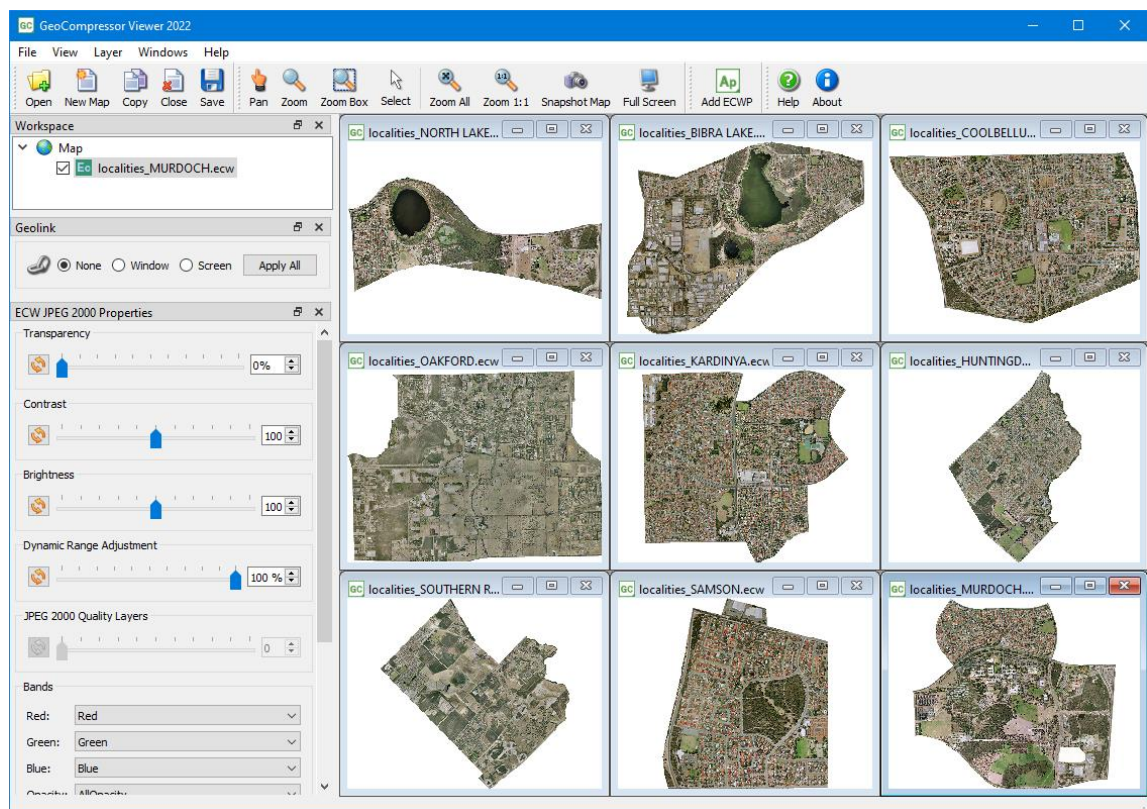
Output Directory or Template: C:\Data\localities
Output Filename or Template: localities_*.ecw
Temp Directory: C:\Temp/
Opacity: from Shapefile
Shapefile: C:\Users\msherida\Desktop\landgate\localities_28350.shp,localities_28350,2,3,4,5,6,7,8,9,10
Background Color: Enabled and using max band value
Memory: Using 7.94 GB of maximum memory 31.76 GB

16. Note the output files listed correspond to each polygon selected and have the correct attributes appended to the filename.
17. Click Finish to queue up the compression tasks in the main window.

18. The task list shows the compression task queue, one task for each polygon, with the output name appended with the attribute value of each polygon. Select “Start” to begin the compression process.



19. When the compression tasks have completed, click the “View” button to open each image in the Geocompressor Viewer. You should have an ECW or JPEG2000 file corresponding to each polygon, clipped to the polygon extents, each with a matching opacity channel, similar to the following:



Appendix B: Troubleshooting

Low Memory Issues During Compression

GeoCompressor uses Intel Thread Building Blocks for memory allocation and management, which is faster than the normal system routines under high multi-threaded load. However, it may use more memory than the normal system routines. If you are running low on memory or compressions fail with a message about not enough system memory, you can remove the Thread Building Blocks library and the compressor will run as usual. It may be slower, but it will use less memory. The library can be found at:

<INSTALLDIR>/lib/bin/x64URelease/tbbmalloc.dll (Windows)

<INSTALLDIR>/lib/tbbmalloc.so (Linux)

You should ensure you make a backup of the library before deleting it so you can restore it later.

Unexpected Application Closure/Crash

In the unlikely event of a processing job suddenly exiting for no apparent reason, refer to your GeoCompressor installation crashdump folder (usually found in C:\Program Files\Hexagon\GeoCompressor [year]\log\crashdump). In the event of a crash, .dmp files (Windows) will be written that allow the product team to triage and understand what caused the issue. Please submit these files to Hexagon support team if found.

Appendix C: FAQ

General

Should I use GeoCompressor or ERDAS IMAGINE Professional?

The answer is really dependent on the required functionality. If basic mosaicking or batch compression is required, GeoCompressor is perfectly suitable. If you require more control over cut-lines and color balancing and need to perform other image/point processing tasks prior to compression, ERDAS IMAGINE is recommended. GeoCompressor is designed to complement existing workflows rather than address end-to-end image or point cloud needs.

Are files created by GeoCompressor any different from those created by ERDAS IMAGINE?

No. There may be slight variations to the metadata stored in the file headers, but otherwise an ECW, JPEG2000 compressed image will be identical regardless of what application is used. Both applications use the same underlying toolkits for conversion/compression.

What relationship does GeoCompressor have with the legacy ER Mapper Image Compressor application?

None. GeoCompressor is a completely redesigned and rebuilt compression application. Despite the similar product naming, the capabilities are not equivalent, and this is not a direct upgrade from the legacy product.

Can GeoCompressor be used on a headless Linux machine?

Yes. Starting with 2020, we now deploy RPM installers that fully support headless, or command-line-only, execution. In this deployment mode, we still deploy the user interface, but it does not need to be used.

What window managers are supported on Linux?

The GeoCompressor GUI is built using the [Qt](#) cross-platform library and has been tested across Gnome, KDE, and Unity platforms.

Why are some Linux platforms listed as viable?

Due to time constraints, platforms that are viable are deemed to be usable and, in some cases, tested, but not to the same degree as fully supported platforms. They may require additional libraries to be installed before they function correctly.



Image Compressor

What advantages does GeoCompressor provide over using the ECWJP2 SDK within GDAL?

Although GDAL has excellent support for the ECWJP2 SDK, a key feature it lacks is the multi-threaded tile encoder and region management for null block and opacity generation. There are also other performance implications that limit throughput if GDAL is used for both reading and writing. GeoCompressor will remain faster at compressing equivalent input even if the GDAL reader is used. Mosaicking is significantly faster.

What mosaic file format gives the best performance?

GeoCompressor supports both ER Mapper ALG and GDAL VRT file formats. Both support a wide range of dynamic capabilities, including the ability to mosaic separate files together to present them as a single virtual file or mosaic. The Image Compressor mosaicking process uses the ER Mapper ALG engine internally for this capability and was chosen for performance reasons and because it's thread-safe. VRT is supported but is known to perform slower than an equivalent ALG, especially for large input sizes. VRT files can also only be compressed using the line encoder.

Why can't I define a multi-polygon for opacity/null definition?

You can! This previous limitation has been resolved in the 2015 release. You can now provide vector regions with multi-part shapes (e.g., holes or multiple geometries).

How does GeoCompressor calculate pixel/polygon intersections?

GeoCompressor calculates the intersection based on the top left coordinate of each pixel. Therefore, you may need to adjust your polygons by half a pixel or more to ensure that the bottom right corner encapsulates the entire raster data you wish to include.

I use complex ER Mapper Algorithms that link to Virtual ERS rasters, kernels, and so on. Will GeoCompressor compress these files?

Yes. GeoCompressor uses the full ER Mapper Library, but in a 64-bit environment. GeoCompressor can be used to compress large datasets where memory requirements exceed 32-bit limits but still benefit from the powerful visualization tools that ERDAS IMAGINE provides.

Is GeoCompressor faster than [product X]?

GeoCompressor compresses faster than the legacy ER Mapper 7.2 (64-bit), ER Mapper Mosaic Balance Compressor (MBC), Safe FME, and Global Mapper products, all of which are used heavily throughout the industry for compressing to ECW and JPEG2000. These products use the legacy ECWJP2 SDK v3.x, which was released over 14 years ago (2006). In comparison, ERDAS IMAGINE and GeoCompressor are always built on the latest ECWJP2 SDK and benefit from years of further research, optimizations for latest hardware, bug fixes, and other improvements. GeoCompressor is maintained by the same development team as the ECWJP2 SDK, ensuring the product is the reference, best-of-breed implementation of the underlying compression engine.

Why does GeoCompressor prevent some input formats from being compressed in tile mode?

Using the tile compression method forces multi-threaded accesses on the input data. For a variety of reasons, many input formats are not thread-safe. Where we are aware of formats that are not thread-safe, we default to the line compression method. GeoCompressor will return an error if tile is used.

The GDAL VRT format is one such example of an input format that compresses without issue with a single thread but causes significant problems when called from multiple threads. This problem is a general GDAL Data Reader problem and is not specific to GeoCompressor (see [GDAL documentation](#)). As improvements are made to data readers like GDAL, GeoCompressor will re-enable these formats.

Why is georeferencing sometimes lost in the output files?

GeoCompressor relies on the data readers to provide georeferencing information in order to resolve back to an EPSG code. For a variety of reasons, this process is not always successful. In such cases, GeoCompressor will create output files in a "WGS84/LOCAL." This definition can be altered to the correct EPSG code by updating the file header after the fact; a recompression is not required. Alternatively, the -srs command line parameter can be defined prior to compression to force the output definition to the given EPSG code.

If no georeferencing is detected at all and the input data is raw, "RAW/RAW" projection/datum pairing will be written to the output.

To confirm, check the Compressor log and look for the "Projection" value set within the "Input Data" metadata area.

What settings should I use to get best throughput?

This question is dependent on many variables and can only be answered after benchmarks and comparisons conducted on customer hardware with customer input data. Contact Hexagon support for assistance, and include an example log file of a compression task you would like to optimize.

Are pyramids or overviews required for compression?

No. GeoCompressor only reads the input data at native resolution. If files such as RRD, OVR, or internally defined overviews are present, they will be ignored. Generating these files is a redundant processing step for compression workflows; they do not speed up the compression of ECW or JPEG2000 files and are not recommended unless required to QA prior to compression. It is a waste of time and storage space.

Why is the output actual compression ratio significantly different from the target when using ECW v3 null blocks?

Depending on the region supplied and the amount of area defined as null, it is not uncommon to specify 20:1 target and obtain an actual compression rate of 50:1 or more. When null blocks are enabled, the amount of data written to the output file can be substantially reduced, which is reflected in the actual compression ratio since this divides the uncompressed input size by the output size. It makes no allowances between null or data areas.

This behavior can be quantified by customers using the ERDAS APOLLO Image Quality utility or by your own visual quality assessment by overlaying a sample image with and without null blocks enabled. The image quality of the "data" areas will remain identical to the same image compressed to 20:1 without null blocks even though its actual target rate may be closer to the target rate and the file size is substantially different.

When should null blocks be enabled?

Null blocks were designed to shrink the output file size and speed up compression speed, but these benefits are directly related to the characteristics of the input. To help identify when null blocks should be enabled, the compression output will report the ratio to data as well as the number of vertices that make up the selected region. Where the ratio to data is low, enabling null blocks will not yield much change to the output file size,

and in many cases will increase compression time due to the additional spatial intersection checking. Similarly, if the number of vertices is in the thousands, performing spatial intersection tests against very complex polygons could adversely impact the compression time, negating the compression speed improvements.

The degree to which these trade-offs affect throughput is heavily dependent on the input data but also the hardware that the compression is performed on. The only reliable way to answer this question is to perform two identical compression tests, one with -opacity 0 (or with no region specified) and one with -opacity 2. This will allow direct comparison of the benefits that null blocks provide. See the Null Block Analysis section for more information and comparisons.

Why does Image Compressor need to calculate min/max values?

To compress signed or unsigned 16-bit output, the compressor needs to know the data range to ensure high-quality output when the actual data range is a smaller sample (for example, 11-bit stored in a 16-bit range). This step is not required for 8-bit output.

Depending on the size and format type, this range calculation may take a few minutes before the actual compression begins. This calculation uses an approximate algorithm by taking an overview image of approximately 2,500 pixel samples to determine the range. If the range reported appears wildly inaccurate, it is recommended to calculate full statistics prior to compression. GeoCompressor will read existing statistics when detected.

GeoCompressor supports reading statistics stored in *.aux.xml, *.aux, ECW v3, ERS, and ALG files.

When compressing large inputs, why is there a delay when the process reaches 95%? Or the opposite problem where the progress completes 95-100% extremely quickly?

For large inputs, 100 gigapixels or more, there can be a noticeable delay when reassembling the output file. We now report the “Re-assembly time” in the output report and count this phase as part of the compression time. Previously, re-assembly would be performed once progress was at 100% which created unnecessary confusion with customers who believed the process had hung. If the re-assembly time is greater than a few % of the total duration, consider improving the disk storage sub-system, as the bottleneck clearly is reading from the temporary location and writing to the output file location.

Why do I see a warning regarding NUMA?

GeoCompressor detects when multiple NUMA nodes are enabled and defaults to the thread count of one NUMA node only. This is done for scalability reasons as the current tile encoder cannot scale across processor groups. Reducing the -threads to the single processor core count will remove this warning.

Why do I sometimes see background pixels along edges of the opacity region with files that contain an opacity channel?

In the image below, the background of the map is red (for demonstration purposes), but the background color of the ECW image is defined as white at compression time. The polygon defining the region to use as an opacity channel lines up exactly with the edge of the raster data, causing white to bleed into the image.



This is due to the nature of the wavelet transform. The color (white) on the background edge of the image will affect the pixels along the “image” edge when the discrete wavelet transform is done on compression. This often happens when the image data bounds and the opacity region are exactly aligned to the same pixel.

To alleviate this affect, make the opacity channel defining region slightly smaller than the raster region. Leaving a few pixels as a boundary is usually enough to eliminate the effect and get a seamless DWT (discreet wavelet transform). If the polygon defining the region is always smaller than the input raster area, you will not see this problem.

HSCP Converter

What is the Hexagon Smart Point Cloud (HSPC) format?

HSPC is a point cloud format processing and visualization framework from Hexagon.

It enables flexible workflows while reducing processing time. HSPC is based on a specialized coordinate encoding particularly suited to point collection by scanners. The resulting storage format has been optimized for data transfer, cloud-based storage, rapid spatial queries, and fast visualization.

A highly efficient spatial tree structure allows HSPC to exhibit several beneficial non-functional properties:

- Encoding performance ranging from twice to ten times as fast as the legacy HPC format.
- Built-in compression reduces data size by a factor of two or more compared to other spatially structured formats, close to that of LAZ (which is not spatially structured).
- Spatial queries that massively outperform those of HPC, LAS, etc.
- Excellent rendering performance by leveraging available GPU resources when visualized by Hexagon software.
- “Unlimited” scalability through streaming of the spatially divided data.
- Target most platforms such as browsers, mobile devices, AR, and VR when visualized by Hexagon software.

All attributes, except for position, are untouched, and compressed losslessly during encoding. Attributes can also be modified post-encoding.

HSPC data sets are currently stored as exploded directly structures, meaning many files. While impractical to move around, this allows for:

- Simplicity and efficiency when exposing data for streaming
- Maximum benefit from the efficiency of the file system
- Low impact when doing attribute editing

HSPC as a framework is continuously evolving through new features and further improvements to non-functional characteristics. Its adoption will continue to increase across Hexagon developer software products.

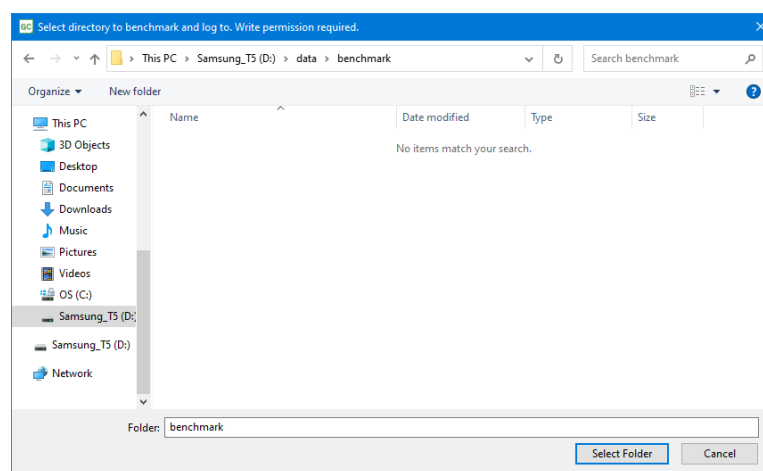
Appendix D: Utilities

GeoCompressor includes a variety of embedded, unlicensed support utilities to help users complete distinct tasks relevant to large format compression tasks.

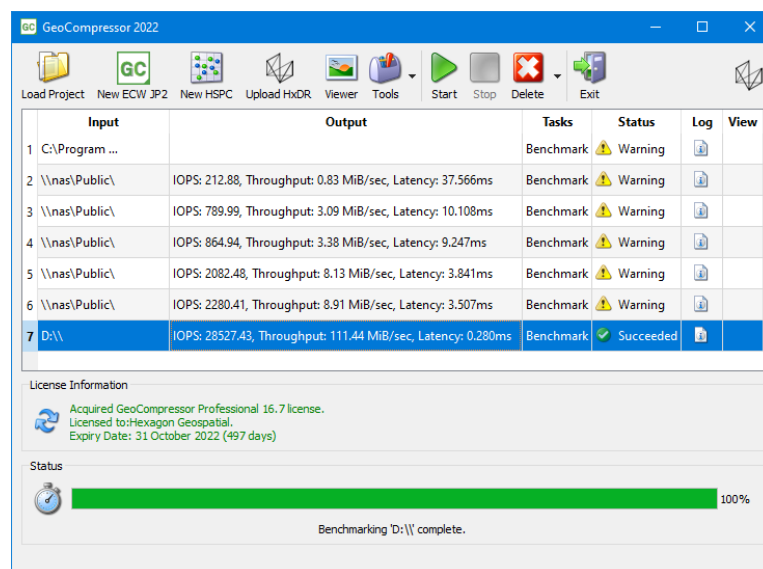
Benchmark Disk

The Benchmark Disk utility introduced in v2022 simulates a performance test with an access pattern matching long-running GeoCompressor jobs. This tool reuses Microsoft's [diskspd](#) but simplifies its usage so customers or prospective users of GeoCompressor can ensure their disk infrastructure meets two distinct thresholds.

This utility, given an input folder location, will create a test 2 GB sample file (geocompressor_diskspd.dat).



Once the GeoCompressor task is started, it will run four concurrent threads randomly reading 4 KB blocks of data from the test file location. The test will take approximately 120 seconds to complete and then return a summary result such as below.



The log attached is the raw output that contains additional information; however, we focus on three inter-related metrics only: IOPS (Input/Output Operations Per Second), Throughput, and Latency (at 95% percentile).



GeoCompressor implements the following thresholds to help guide users when the input storage medium is deemed slow, moderate, or fast. These boundaries align with real-world performance results collected by the development team.

I/O Threshold	Status	IOPS	Throughput	Latency
Slow	Warning	< 500	< 2 MiB/sec	> 60ms
Medium	Warning	500 -> 5000	2 <-> 10 MiB/sec	20 <-> 60ms
Fast	Succeeded	> 5000	> 10 MiB/sec	< 20ms

GeoCompressor will work with any reported threshold, but observed performance will closely align with disk read I/O. In almost all cases, for large image mosaic creation, reading input data takes ~85% of the total time. When a project could take 20 days to compress, it is critically important to benchmark read I/O prior to a large compression project.

While sourcing tens of terabytes of fast storage can remain costly, these figures can be used as a general system validation tool together with customer IT departments. Due to the variety of storage mediums available, we do not recommend a specific system or hardware configuration. If it meets these thresholds, it does not matter if the storage is direct connect, networked, or virtualized. We simply want to educate users by illuminating more easily the performance characteristics so customers can make a more informed decision.

If storage performance is constrained, it is generally recommended to revise the CPU threads used for image compression. If the workstation has 32 CPU threads available, but the input I/O is benchmarked as slow, it is highly likely that 32 compression threads will perform slower than 8 CPUs due to the concurrent reads becoming bottlenecked.

Further notes:

- Real-world compression results will also depend on the input format structure, its dimensions, compression type, and other parameters. GeoCompressor will not read only in 4 KB chunks; the test is an approximation only.
- The temporary `geocompressor_diskspeed.dat` test file contains random binary data and will be automatically deleted once you close the GeoCompressor application.
- Advertised disk speed metrics are often quoted via artificial tests of a particular type. If the GeoCompressor benchmark results are wildly slower than expected, it's likely our test parameters are wildly different. This is most commonly seen when customers compare sequential read speeds against random reads, for example. It is important to compare carefully and use a reproducible benchmark.
- If your GeoCompressor compression speeds are slower than expected or CPU utilization remains low, it is highly likely that disk read performance is the cause. Run this utility and open a support ticket for more information.

ECW Header Editor CLI Parameters

The ECW header editor allows you to edit the georeferencing information contained in an ECW or JP2 file header without recompressing the file.

To display the help for the generic parameters for ECW or JP2 files, use the following command line:

```
> ECWHeaderEditorCLI.exe --header -help

Header editor operation:
--help                Display header editor help message
--input arg           Path of the dataset to update
--projection arg       The new projection string
--datum arg           The new datum string
--origin-x arg        The new X coordinate of the top-left
                      corner of the top-left pixel
--origin-y arg        The new Y coordinate of the top-left
                      corner of the top-left pixel
--cell-increment-x arg The new world units per pixel in the X
                      axis
--cell-increment-y arg The new world units per pixel in the Y
                      axis
--update-ers arg (=0) Should a .ers side-car file also be
                      updated
--allow-invalid-datum-projection-pairs arg (=1) Allow any datum and projection
                      combinations in the dataset
```

To display the help for the specific parameters for JP2 files, use the following command line:

```
> ECWHeaderEditorCLI.exe --jp2 --help

JP2 advanced operations:
--help                Display JP2 advanced operations help
                      message
--input arg           Path of the JP2 dataset to update
--show-cdef-box       Show the cdef box contents
--validate-cdef-box    Validate that the cdef box contents are
                      correct
--correct-cdef-box-rgb-in-multiband Correct the cdef box where RGB channels
                      are associated to bands for non RGB
                      datasets
--backup-dataset arg (=1) For operations that modify the input
                      backup the file first
```



Support

GeoCompressor product support is available to all customers with an active GeoCompressor subscription or Software Maintenance on ERDAS APOLLO. See the [Hexagon support](#) page for more information on how to raise support requests.

When reporting problems, please include the GeoCompressor output log, which will include all relevant information to help diagnose possible causes. Input data may be required to reproduce.

About Hexagon

Hexagon is a global leader in digital reality solutions, combining sensor, software and autonomous technologies. We are putting data to work to boost efficiency, productivity, quality and safety across industrial, manufacturing, infrastructure, public sector, and mobility applications.

Our technologies are shaping production and people-related ecosystems to become increasingly connected and autonomous — ensuring a scalable, sustainable future.

Hexagon's Safety, Infrastructure & Geospatial division improves the performance, efficiency and resilience of vital services. Its Safety & Infrastructure solutions enable smart and safe cities. Its Geospatial software leverages the power of location intelligence.

Hexagon (Nasdaq Stockholm: HEXA B) has approximately 21,000 employees in 50 countries and net sales of approximately 3.8bn EUR. Learn more at [hexagon.com](https://www.hexagon.com) and follow us [@HexagonAB](#).

© 2021 Hexagon AB and/or its subsidiaries and affiliates. All rights reserved. Hexagon and the Hexagon logo are registered trademarks of Hexagon AB or its subsidiaries. All other trademarks or service marks used herein are property of their respective owners.