

EightBitz's Dungeondraft Tools

Table of Contents

Overview	3
Credits	3
A Note about Security Software	4
Getting Started.....	4
Menus	6
Main Menu.....	6
Preferences	6
Common Controls	7
The Tools	8
Tag Assets.....	8
What Does it Do?	8
Important Details	8
Folder Organization.....	8
Colorable Assets.....	9
The Controls	10
Final Notes	13
Convert Assets	14
What Does it Do?	14
Important Details	14
Converting Organized Asset Folders	15
Converting Non-Asset Folders	16
The Controls	16
Convert Packs.....	16
What Does it Do?	17
Important Details	17
The Controls	17
Copy Assets	20
What Does it Do?	20
Important Details	20
Multiple File Versions.....	20

Portals	21
The Controls	22
Copy Tiles	24
What Does it Do?	24
Important Details	24
The Controls	24
Final Notes	27
Data Files	28
What Does it Do?	28
Important Details	28
The Controls	29
Final Notes	31
Map Details	31
What Does it Do?	32
The Controls	33
Final Notes	33
Pack Assets	34
What Does it Do?	34
Important Details	34
The Controls	34
Unpack Assets	36
What Does it Do?	36
Important Details	36
The Controls	36
Version History	37
Version 3.5	37
Version 3.0	37
Version 2.5	37
Version 2.2	37
Version 2.1	38
Version 2	38

Overview

This program offers 8 tools to facilitate creation and management of custom asset packs and to list information about custom assets used within specified maps. Each of these tools is documented in more detail later in this document, but here's a brief overview of each.

- **Tag Assets** will generate a default.dungeondraft_tags file for a given asset folder, based on that folder's subfolder structure.
- **Convert Assets** will convert files of various types to webp format.
- **Convert Packs** will unpack custom asset packs, convert the files within to web format, and repack the custom asset packs.
- **Copy Assets** will copy assets into a folder structure to prepare them for packaging by Dungeondraft. This tool focuses on placeable objects and portals.
- **Copy Tiles** is similar to Copy Assets, but it focuses on tiles used for terrain, floor patterns and floor tiles, and it will copy such assets into the appropriate folders to prepare them for packaging by Dungeondraft.
- **Data Files** will allow you to create/manage data files for tilesets and walls.
- **Map Details** will read a map file, and give you readable information about it, including when the map was created, which version of Dungeondraft it was created with, the grid size, which custom asset packs it uses, and which assets it uses from those packs.
- **Pack Assets** will allow you to create new packs from structured asset folders and to update existing packs.
- **Unpack Assets** will allow you to unpack custom asset packs to view and edit their contents.

Credits

Thanks to the reddit user GoodCat-BadCat for their Python script to unpack dungeondraft packs.

https://www.reddit.com/r/dungeondraft/comments/gjvlud/python_script_to_unpack_dungeondraft_pack_assets/

Thanks to the Discord user Ryex for the dungeondraft-pack.exe and dungeondraft-unpack.exe programs.

<https://github.com/Ryex/Dungeondraft-GoPackager/releases/tag/v1.1.0>

Thanks to the Discord user nooby for their Python script to display map details, and for their suggestion to add the same functionality to my tools.

<https://gist.github.com/noobanidus/7948054bffffbdaf3afd9ce6152112d09>

This program uses the Newtonsoft JSON framework for .NET

<https://www.newtonsoft.com/json>

This program uses the ImageProcessor .NET library.

<https://imageprocessor.org/>

Finally, thanks to everyone who has offered positive feedback and constructive criticism.

A Note about Security Software

I used Bitdefender for security against viruses, ransomware, and other types of malware. During the course of my testing, I've received several complaints from Bitdefender regarding "suspicious activity" from both my tool (DDTools.exe) and from Ryex's programs (dungeondraft-pack.exe and dungeondraft-unpack.exe). I have a report of the same type of issue from someone who uses Norton's security software.

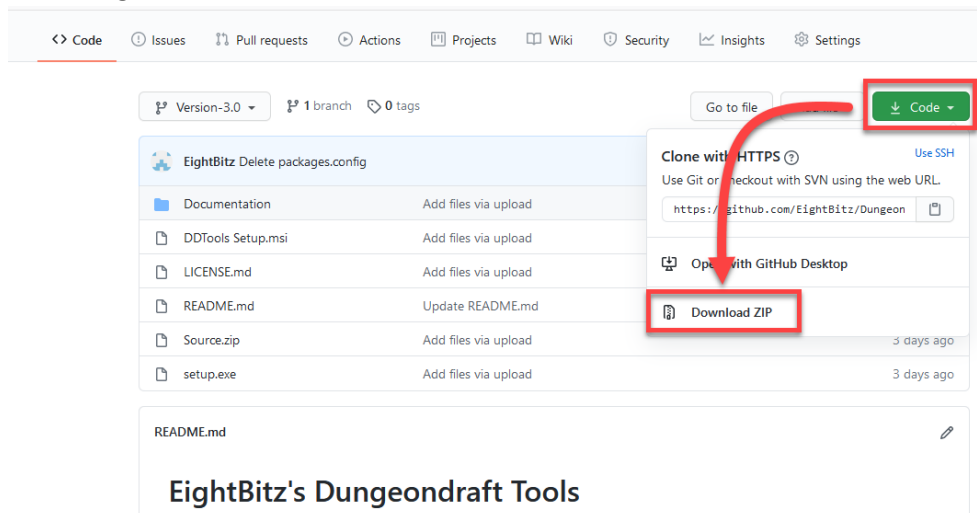
If you experience this issue, it is unfortunately beyond my control right now. My tool is designed to copy and modify files in large batches, and your security software is designed to protect files from being copied and modified in large batches.

The solution, for now at least, is to tell your security software to allow exceptions for these three programs. Usually, you can do that by clicking on the pop-up notification you'll receive, and clicking a button or a switch or something to allow the program or allow an exception for the program. It will look different and be worded differently, depending on which security software you're using.

Now, on with the show.

Getting Started

- Uninstall any previous version you might have installed. (Some earlier versions only required a download, not an installation, but if you have an installed version, uninstall it.)
- Browse to <https://github.com/EightBitz/Dungeondraft-Tools>
- Click the green "Code" button, then click "Download ZIP".



- Open the downloaded ZIP file, and extract the folder.
- Run Setup.exe
- Double-click the "DDTools" desktop icon.
- You will be greeted with the following window.



Menus

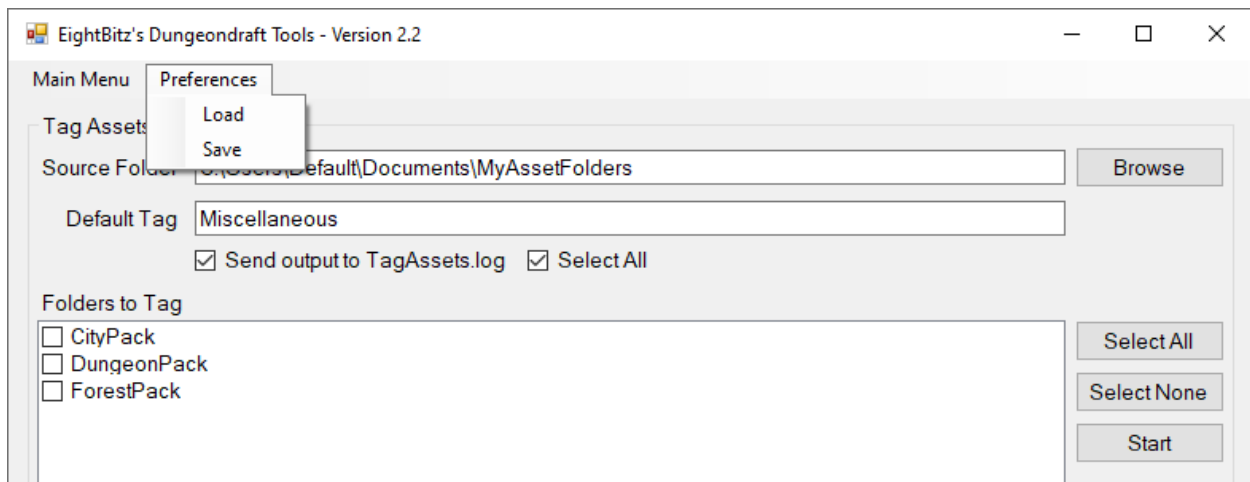
Main Menu

The Main Menu allows you to select which tool you want to use.



Preferences

Once you select a tool, there will be one or two text boxes you can fill out, and at least one or two check boxes you can select. The Preferences menu will allow you to save your existing values, or to load your previously saved values. Once you save your values for a given tool, they will be automatically loaded every time you switch to that tool from the Main Menu.



Note that the Preferences menu applies to the active tool. If the active tool is “Tag Assets”, then the Preferences menu will load and save preferences only for “Tag Assets”. If the active tool is “Copy Assets”, then the Preferences menu will load and save preferences only for “Copy Assets”.

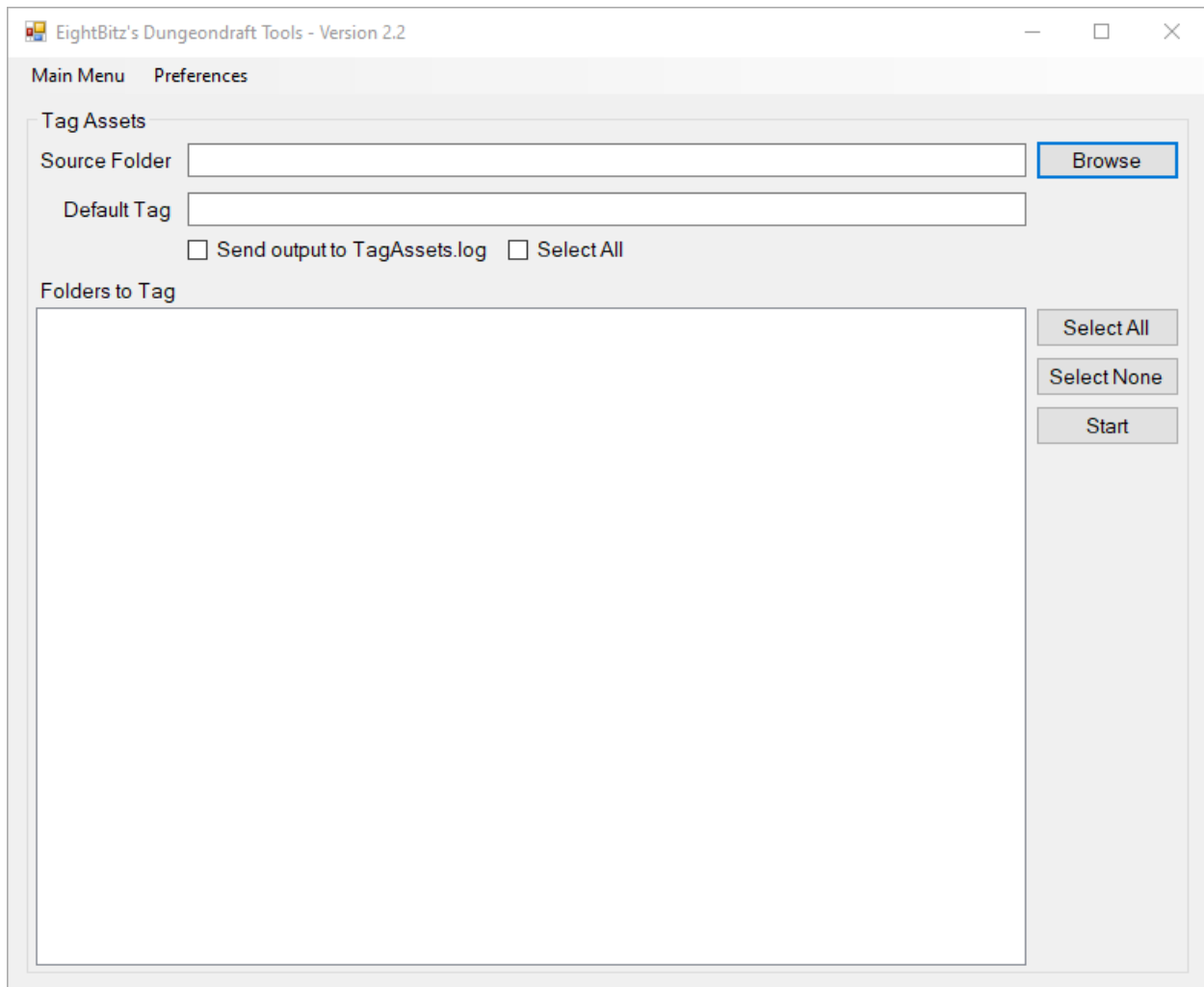
Common Controls

- **Source Folder:** This allows you to select your starting point. If you want to Tag Assets, the source folder lets you select where your asset folders are located. If you want to Copy Assets, the source folder lets you select where to copy them from. You can type in the name of the folder or click the corresponding Browse button to select it.
- **Destination Folder:** If you're going to create new files or folders of any sort, the destination folder allows you to select where they will be created. You can type in the name of the folder or click the corresponding Browse button to select it.
- **Send output to <log file name>:** Every tool generates output of some sort to tell you what it's doing or what it's done. Checking this box will save the output to a file as well as displaying on-screen. The log file will be saved to the folder where DDTools.exe is stored. The name of the file will be the name of the tool with no spaces and with a .log extension. For example: TagAssets.log or CopyAssets.log.
- **Checklist/Grid View box:** Once you select your source folder, selectable items should appear in the large box. Depending on the tool you're using, they will appear either as a simple list of items to check, or as a grid view (rows and columns, like a spreadsheet).
- **Select All button:** This button will select all the items in the checklist or the grid view. This button does not exist in the "Copy Tiles" tool, but the same function is provided with checkboxes to select or clear all items within a given row, all items within a given column, or all items within the grid view.
- **Select None button:** This button will clear all check selections in the checklist or the grid view. As with the Select All button, this button also does not exist in the "Copy Tiles" tool, but the same function is provided with checkboxes to select or clear all items within a given row, all items within a given column, or all items within the grid view.
- **Select All checkbox:** This checkbox offers the same functionality as the Select All and Select None buttons with one addition. The state of this checkbox is saved with your other preferences. You can use the buttons without affecting your saved preferences, but the state of the checkbox will be saved and loaded with your other preferences.
- **Start button:** Start processing the selected items.

The Tools

Tag Assets

This will present you with the interface for the Tag Assets tool.



What Does it Do?

For people who want to create their own custom asset packs, the Tag Assets tool will create a properly formatted default.dungeondraft_tags file. This is the file that Dungeondraft uses to determine which assets in the object library are associated with which tags.

Important Details

This tool requires that Newtonsoft.Json.dll reside in the same folder as DDTools.exe.

Folder Organization

In order to benefit from this tool, your folders should be organized into subfolders. For instance, say you keep all your asset folders in a parent folder named "MyAssetFolders"

Within "MyAssetFolders", you might have folders named "CityPack", "DungeonPack" and "ForestPack".

Within each of those three folders, you will have a “textures\objects” path.

Within the “textures\objects” path, you can organize your assets into subfolders. For instance, you might have the following subfolders:

“ForestPack\textures\objects\Animals”

“ForestPack\textures\objects\Bushes”

“ForestPack\textures\objects\Trees”

The Tag Assets tool will use these subfolders as your tags. In this case, your tags will be “Animals”, “Bushes” and “Trees”. Anything files within those subfolders will be assigned to those tags, respectively. If you have subfolders for different types of trees, those files will still be tagged as “Trees”. For instance,

“ForestPack\textures\objects\Trees\dead”

“ForestPack\textures\objects\Trees\evergreen”

“ForestPack\textures\objects\Trees\fruit”

Anything within any of these deeper subfolders will still be tagged as “Trees”.

Colorable Assets

Anything within a “Colorable” folder will be tagged as colorable. For instance, anything within “ForestPack\textures\objects\Animals\Colorable” will be tagged as both “Animals” and “Colorable”.

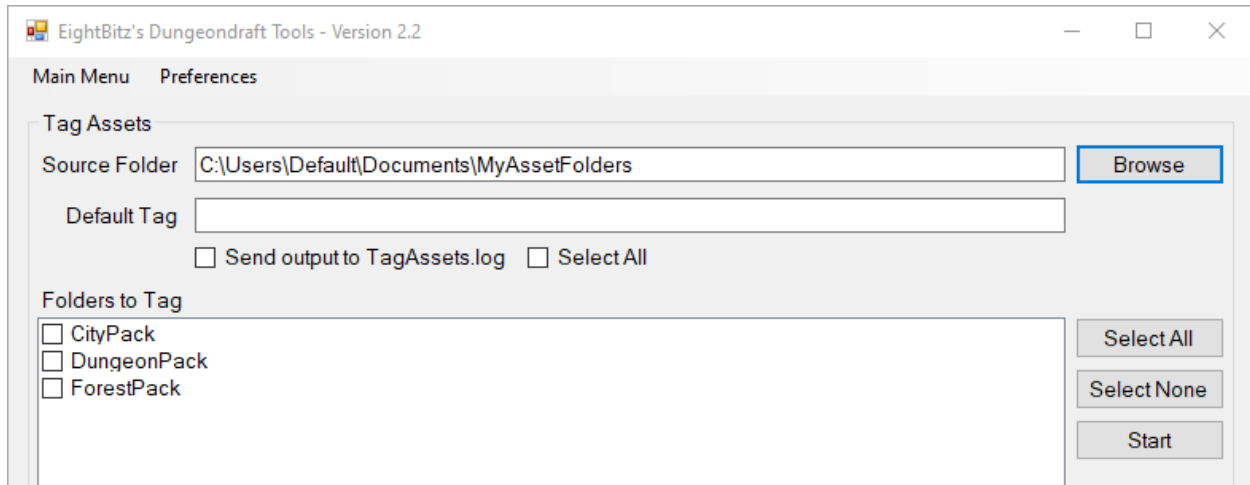
Anything within “ForestPack\textures\objects\Animals\Friendly\Colorable” will be tagged as both “Animals” and “Colorable”.

If you want something tagged as colorable, you can put it in its proper category, but make sure to place it in a “Colorable” folder within that category.

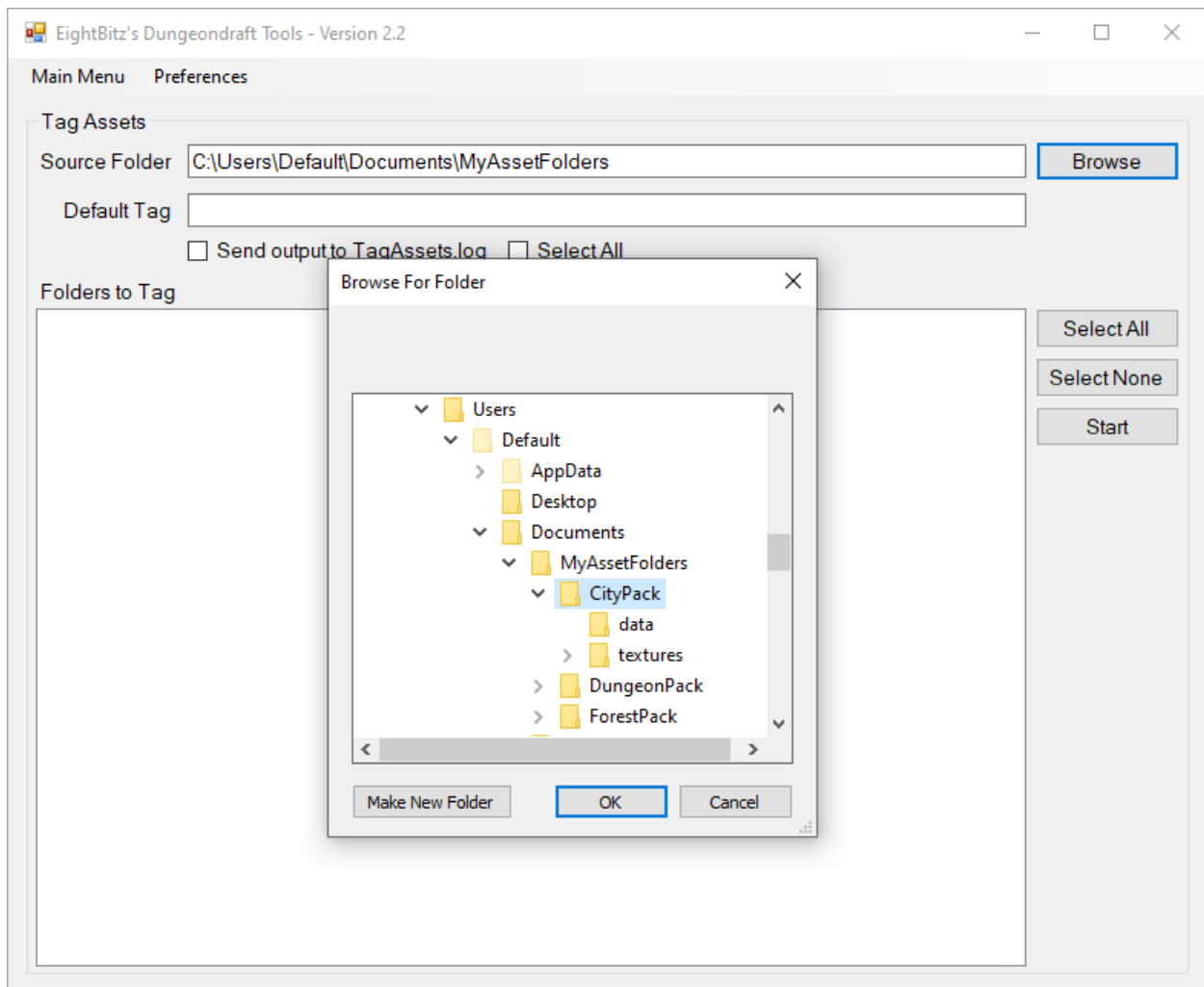
The Controls

Source Folder

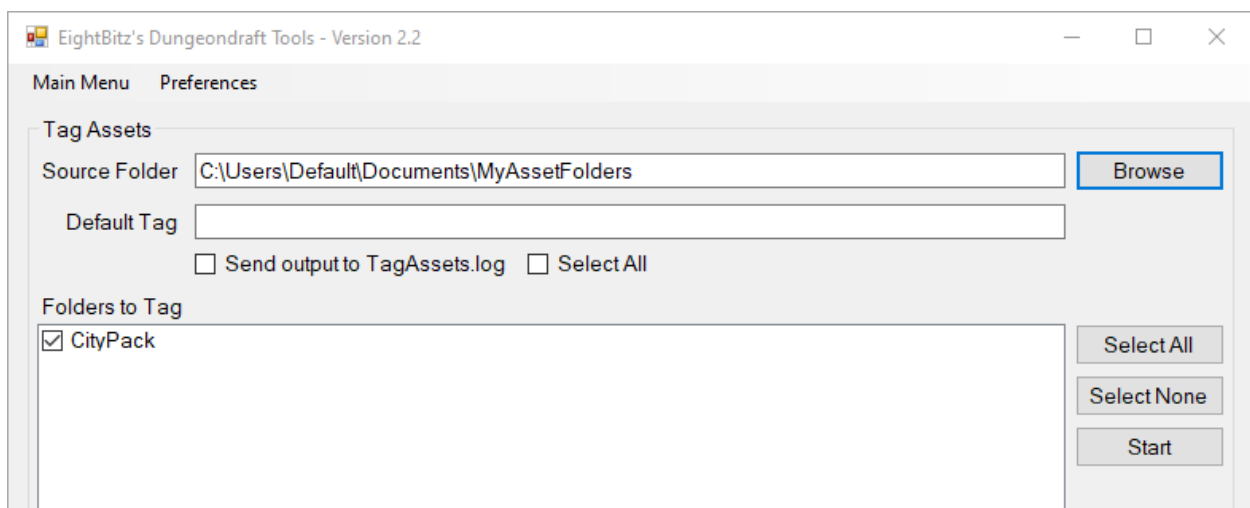
Using the example folders specified above, if I selected “MyAssetFolders” as my source folder, the tool will look for any folders within “MyAssetFolders” that have a “textures\objects” path. In this case, those folders will be “CityPack”, “DungeonPack” and “ForestPack”. Those three folders will appear in the checklist.



Alternatively, you can select one, specific folder as your source folder (as long as it contains a “textures\objects” path). Say, for instance, you select the “CityPack” folder.



The tool will add that folder, and only that folder (ignoring the others) to your checklist.



Default Tag

If you have any files that you just don't want to put in any subfolders, you can opt for a Default Tag for those files. Anything in "textures\objects", but not in a subfolder, will be tagged with the default tag.

If you leave the Default Tag blank, any files that are not in subfolders will not be tagged.

Send output to TagAssets.log

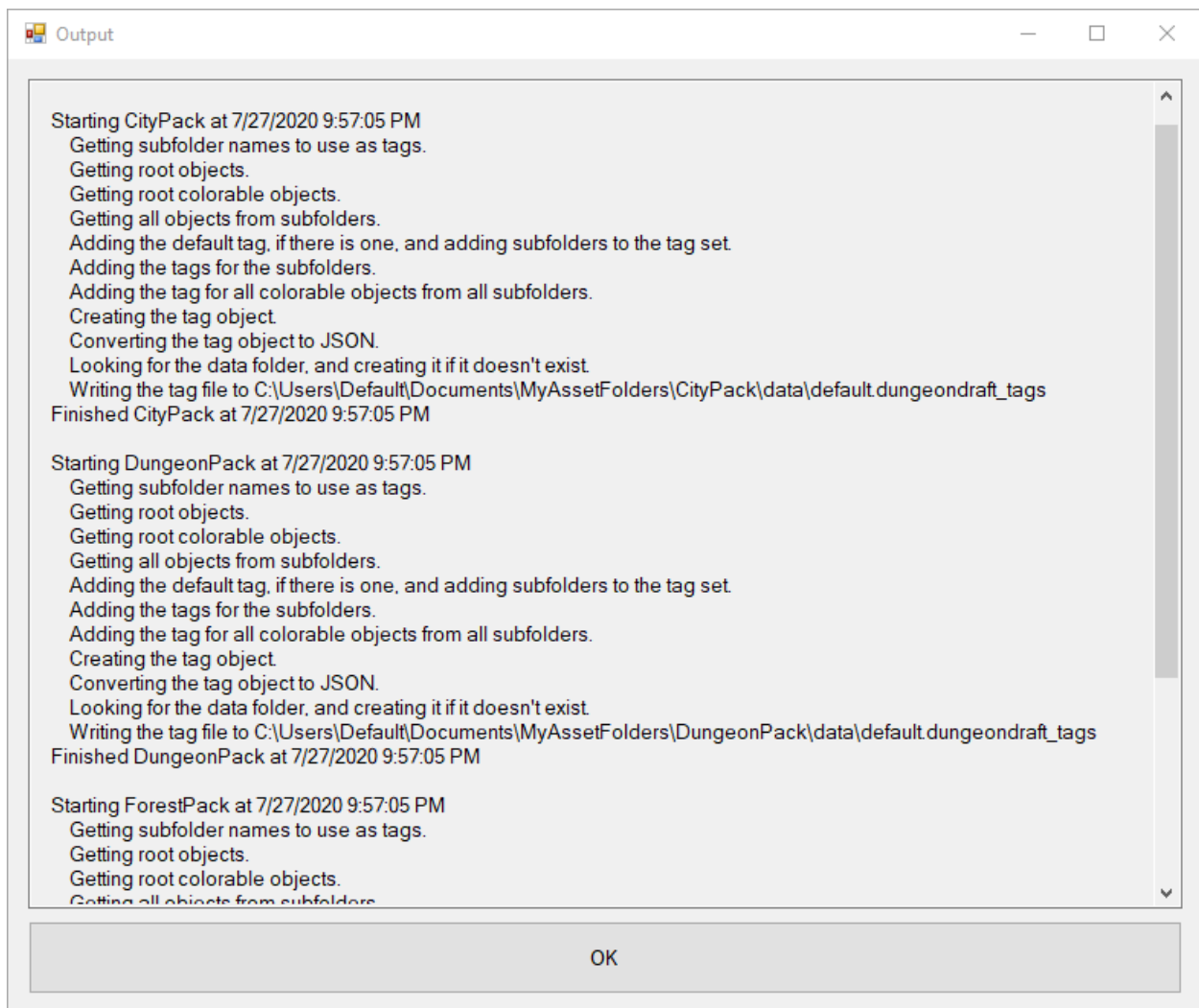
This will save output to a log file named "TagAssets.log", in the same folder as DDTools.exe.

Folders to Tag

When you select your source folder, any folders within it that have a "textures\objects" path will appear in this checklist.

Start

Once you've selected all your other options, click "Start" will begin the tagging process. A new window will pop-up with a brief log of sorts.



Final Notes

The name of the asset folder will be used as the name of the tag set. For the “CityPack” folder, the tag set will be named “CityPack”. For the “DungeonPack” folder, the tag set will be named “DungeonPack”, etc.

The resulting tags file is written to where it should be. For “CityPack”, it will be written to “CityPack\data\default.dungeondraft_tags”. For “DungeonPack”, it will be written to “DungeonPack\data\default.dungeondraft_tags”.

If the data folder does not exist, it will be created.

If there is an existing default.dungeondraft_tags file, it will be overwritten.

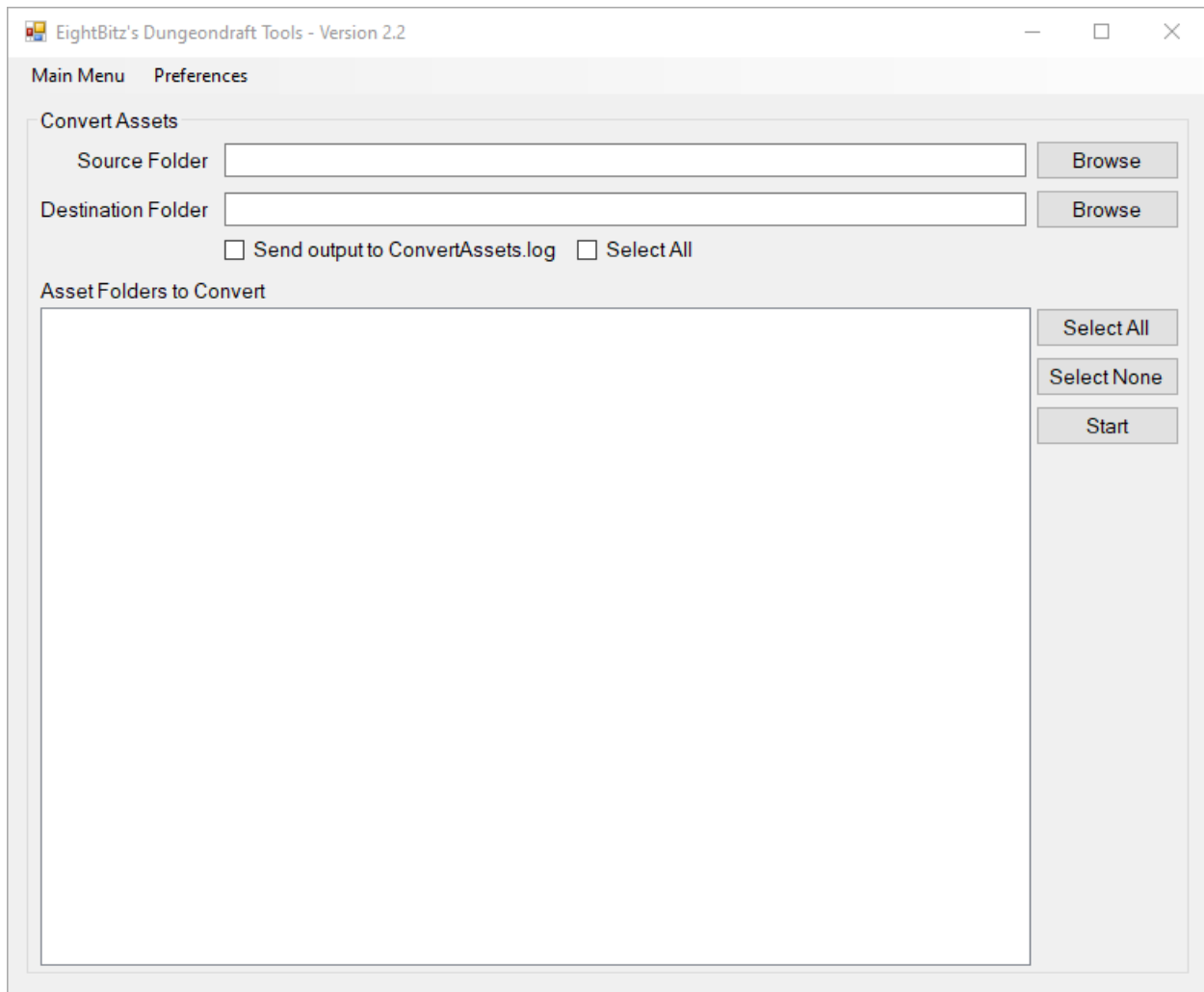
If there is an existing data folder, and there are other data files (for walls or tilesets), those data files will remain intact.

Consider this a beta. When it’s done doing its thing, take a look at the resulting files, and make sure things look OK.

Create the actual Package(s) with Dungeondraft, load them up, and make sure the tags are working.

Convert Assets

This will present you with the interface for the Convert Assets tool.



What Does it Do?

The Convert Assets tool will convert supported image formats in selected folders to .webp format. .webp offers lossless compression, so the files on disk end up being smaller. That being said, this will not save resources in Dungeondraft, as when .webp files are loaded into memory, they are uncompressed.

I added this function at the suggestion of someone on the Dungeondraft Discord server. Whether or not you find it ultimately beneficial, I will leave to you.

Important Details

If a pack.json file exists in the original asset folder, it will be rewritten in the converted folder to have a different pack ID. This means that if you package it with Dungeondraft, it will be considered a new pack. If you packaged a previous incarnation of this folder and used that pack for a map, that map will not recognize the new package created from the converted folder.

Converting Organized Asset Folders

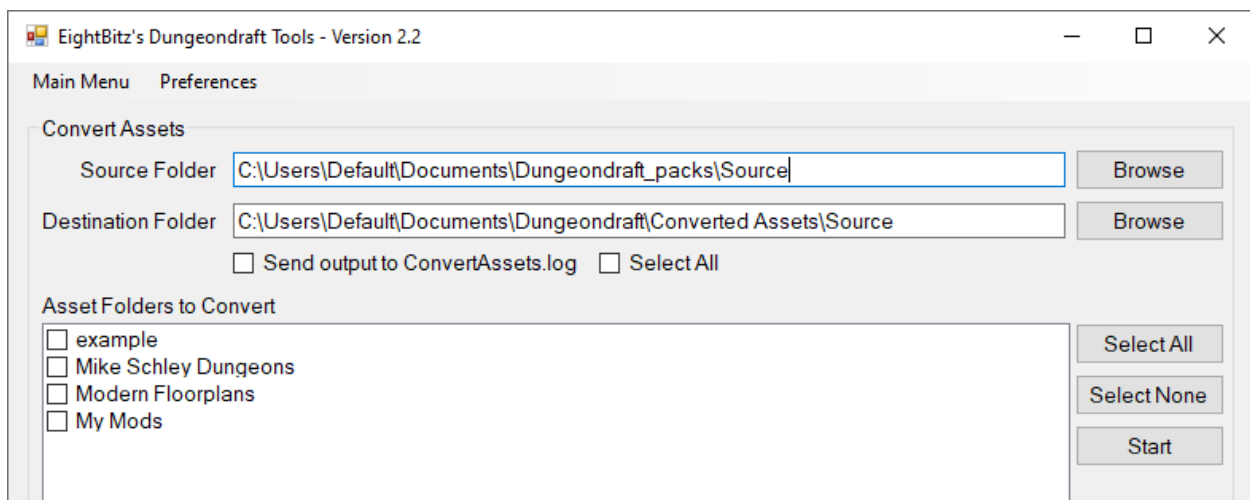
Dungeondraft supports .webp images in the objects library, but not for other types of assets (e.g. walls, tilesets, materials, et al).

If you are converting an asset folder that is already organized as a Dungeondraft asset folder, this tool will recognize that. It will only convert images in the textures\objects folder. All other images will be copied instead of converted.

All data and metadata files will be copied as well. All default.dungeondraft_tags files will be updated to reflect the extension for converted files.

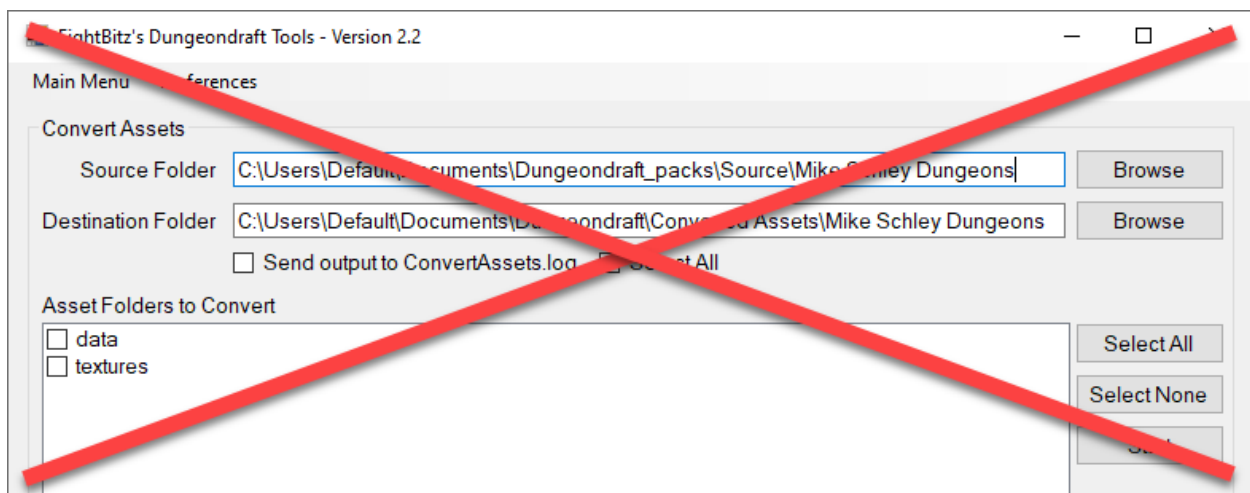
Also, this tool does a recursive conversion, meaning it will find all subfolders within the folder you select. This is particularly important to note if you want to convert a folder that's organized as an asset folder. In that case, you want don't want to drill down into the actual asset folder.

You want to stop at the point where you see your asset folder names in the list, like this:



You can go higher than that if you want, but not lower.

You don't want to go into your asset folders where you see the textures and/or data folders.



If you do drill down into an individual asset folder, the textures folder will be converted, but the data folder will be ignored. I may workout how to fix this later, but for now, be aware.

Converting Non-Asset Folders

If you convert a folder that is not organized as an asset folder, all supported image types will be converted to webp, and all other files will be ignored.

The Controls

Source Folder

This specifies the folder that contains the folders and files you want to convert.

Destination Folder

This specifies the folder that will contain the converted files.

Send output to ConvertAssets.log

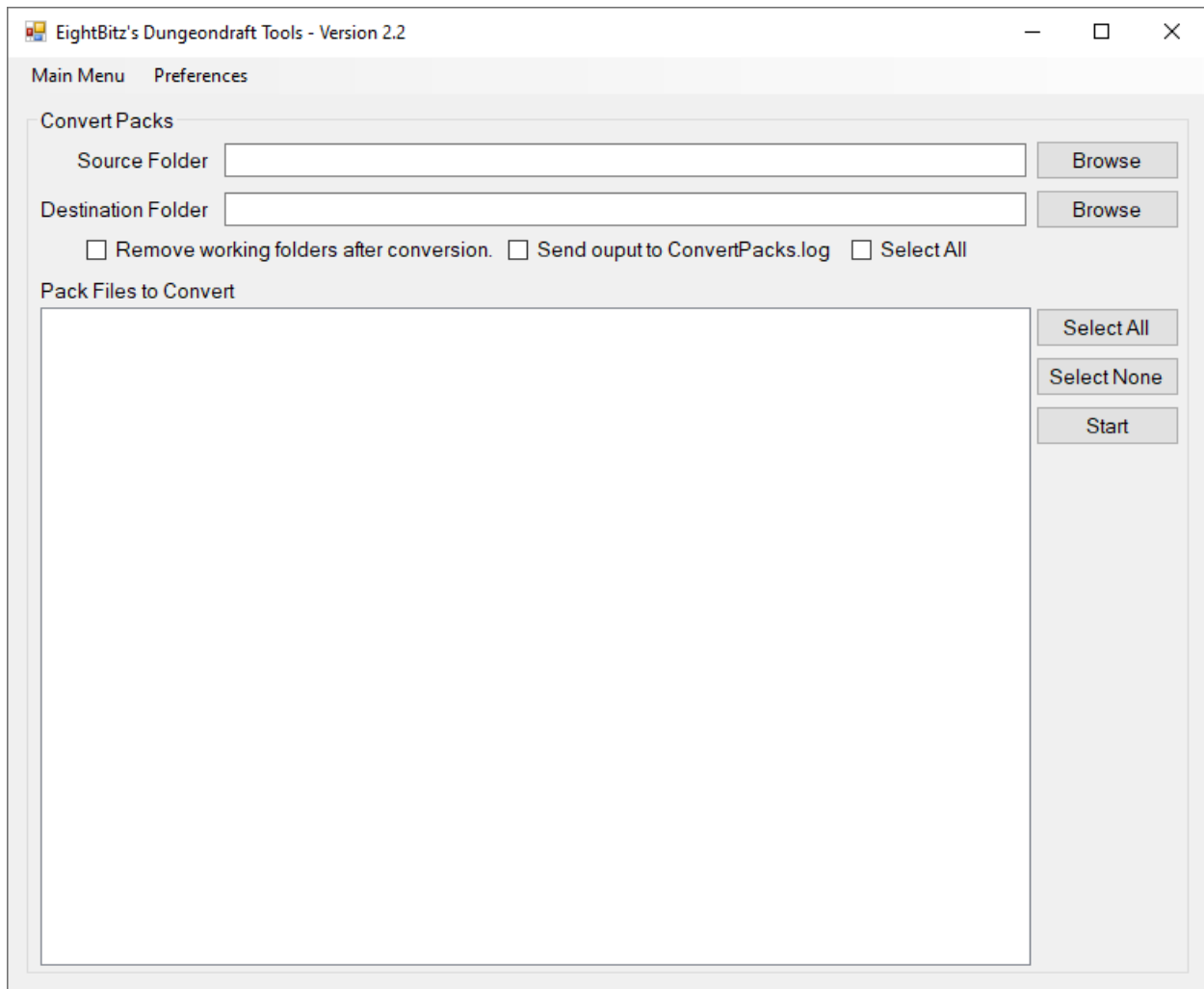
This will save output to a log file named “ConvertAssets.log”, in the same folder as DDTools.exe.

Start

This begins the conversion process.

Convert Packs

This will present you with the interface for the Convert Packs tool.



What Does it Do?

This works similar to the Convert Assets tool, except it works on files that are already packed. This tool calls dungeondraft-unpack.exe to unpack selected files, then it calls ImageMagick to convert assets to webp, then it repacks the converted asset folders.

Important Details

This will create a new .dungeondraft_pack file with a unique pack ID, different from the original pack. When you select this package in Dungeondraft, it will be considered a new pack. If you packaged a previous incarnation of this folder and used that pack for a map, that map will not recognize the converted pack.

The Controls

Source Folder

This specifies the source folder that contains the asset packs you want to convert.

Destination Folder

This specifies the root folder that will contain the converted packs.

Remove working folders after conversion

While this program runs, it creates three folders in the destination path. It creates an “Unpacked Assets” folder, a “Converted Folders” folder, and a “Converted Packs folder.

When it finishes, by default, it removes the “Unpacked Assets” folder and the “Converted Folders” folder. It basically cleans up after itself.

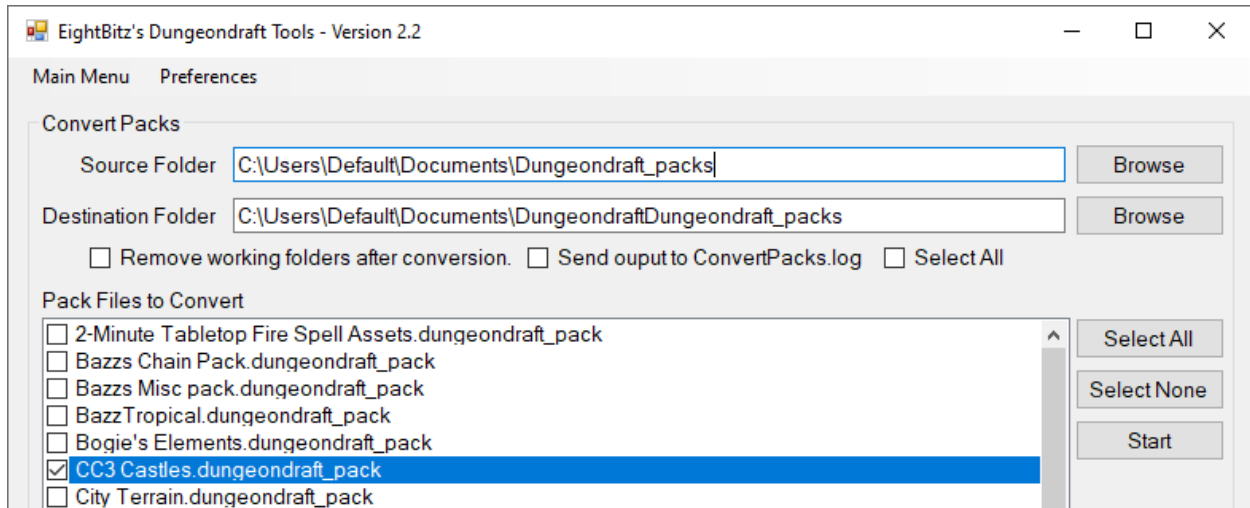
If you want to leave those folders in place, uncheck this box.

Send output to ConvertPacks.log

This will save output to a log file named “ConvertPacks.log”, in the same folder as DDTools.exe.

Pack Files to Convert

When you select the folder that contains the packs you want to convert, the packs will show up in the listbox as checkable items.

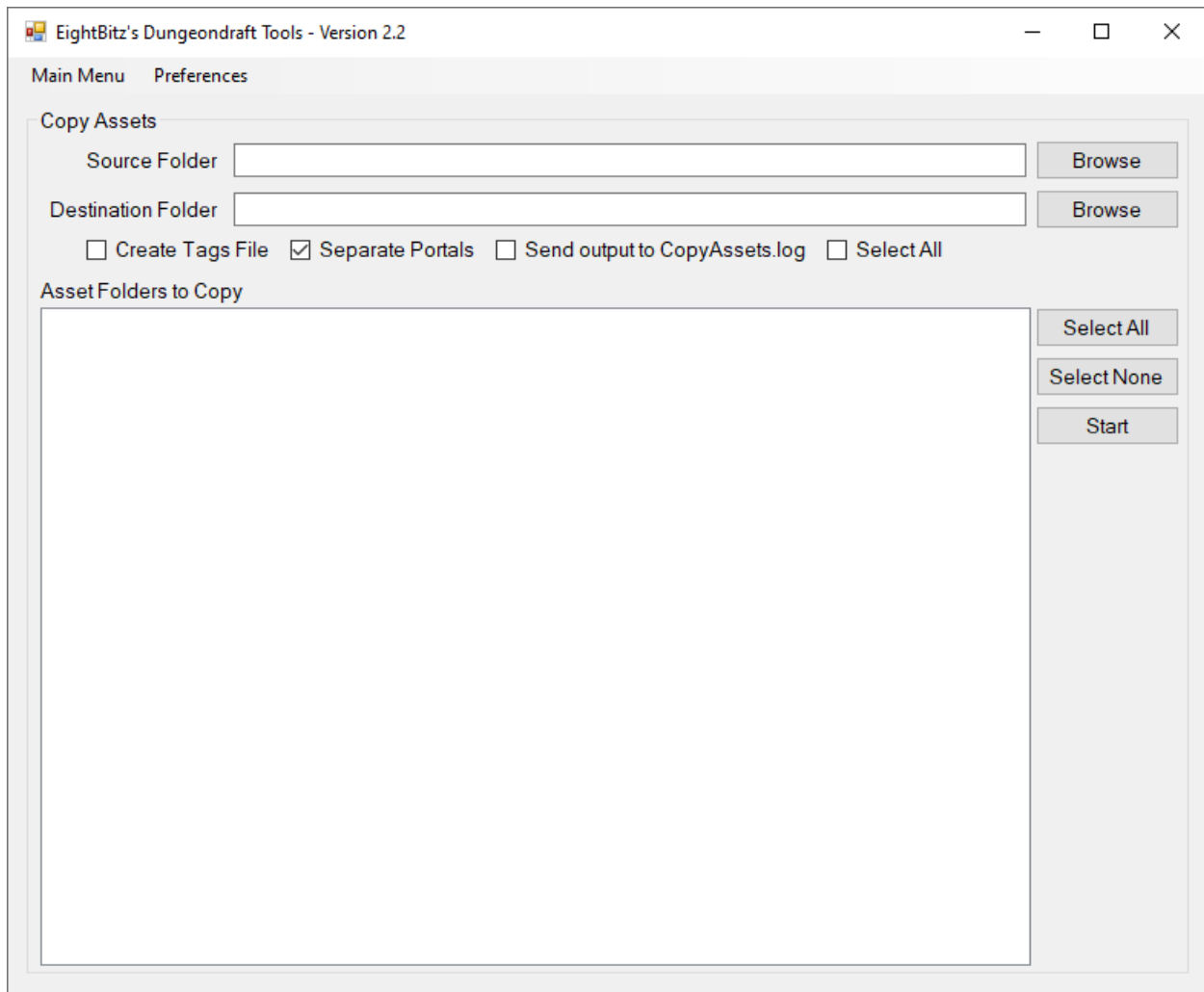


Start

Start the conversion process.

Copy Assets

This will present you with the interface for the Copy Assets tool.



What Does it Do?

This tool is primarily designed to copy Campaign Cartographer 3 Plus (CC3+) assets into a textures\objects and/or textures\portals folder structure for Dungeondraft. This tool will work for other assets as well, but Campaign Cartographer has two important naming conventions that this tool uses to manage how it copies assets.

Important Details

This tool will still work for copying assets from a generic source, but if you are copying files from a CC3+ source, you might want to be aware of the following.

Multiple File Versions

CC3+ has duplicate files of different quality. For instance, if there's a "House1" image, there will be up to five versions of House1.

- House1_VH.PNG

- House1_HI.PNG
- House.PNG
- House_LO.PNG
- House_VL.PNG

When this tool copies CC3+ assets, it will only copy one version of the file, and that version will be the best quality it can find while completely ignoring _LO and _VL versions. Those are the two lowest-quality versions, and they're useless in Dungeondraft, so this tool is really only concerned about _VH.PNG files, _HI.PNG files, and .PNG files that are neither _VH nor _HI.

- If all three of those versions exist, only the _VH.PNG version will be copied.
- If only two of those versions exist, the highest-quality version will be copied, whichever one it may be.
- If only one of those versions exists, then of course, that will be the only version that gets copied.

Portals

CC3+ names doors so their names start with "Door " or "Doors ", and it names windows so their names start with "Window " or "Windows ".

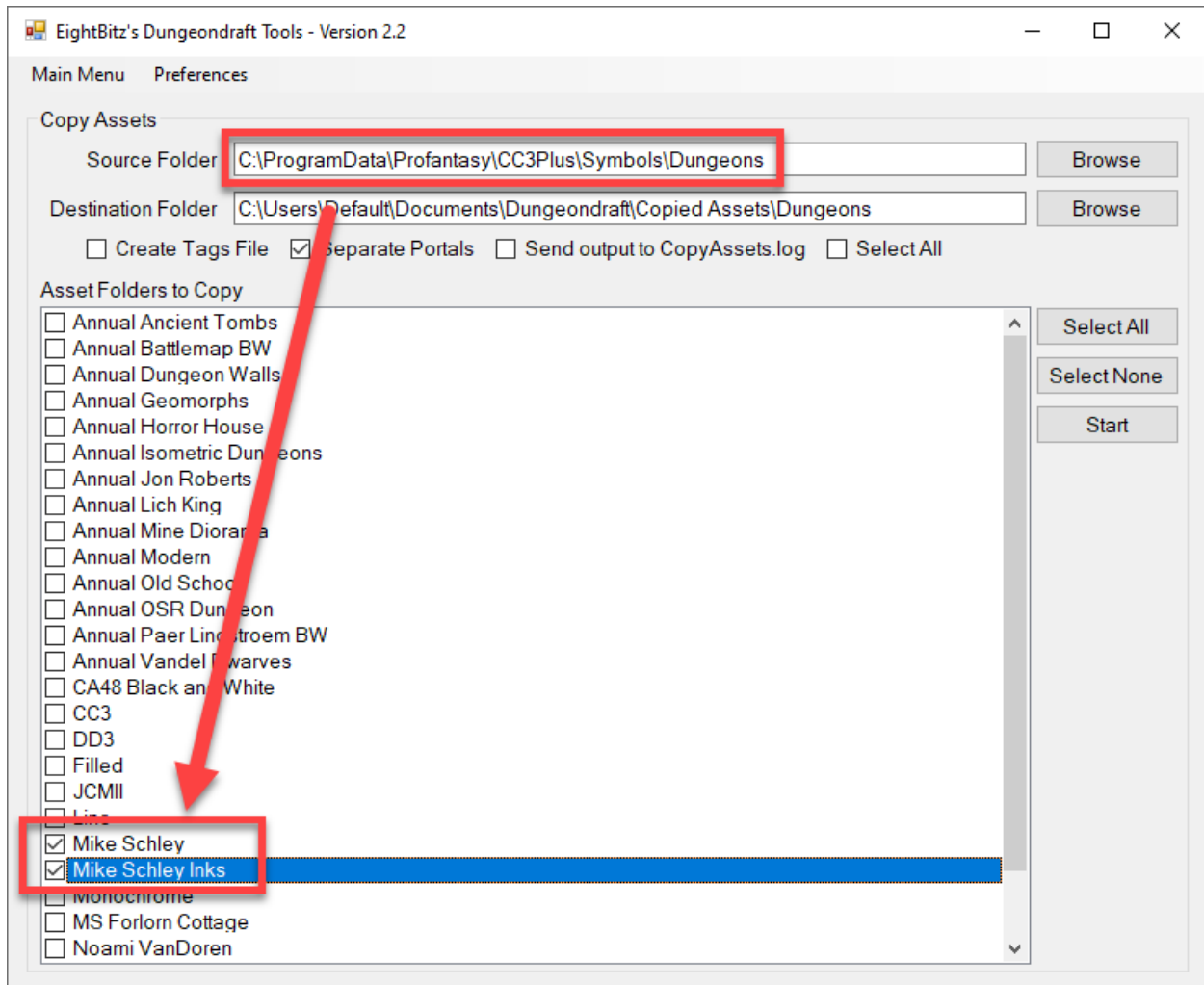
This tool has an option to "Separate Portals" which, if checked, will copy doors and windows to "textures\portals" instead of "textures\objects".

The Controls

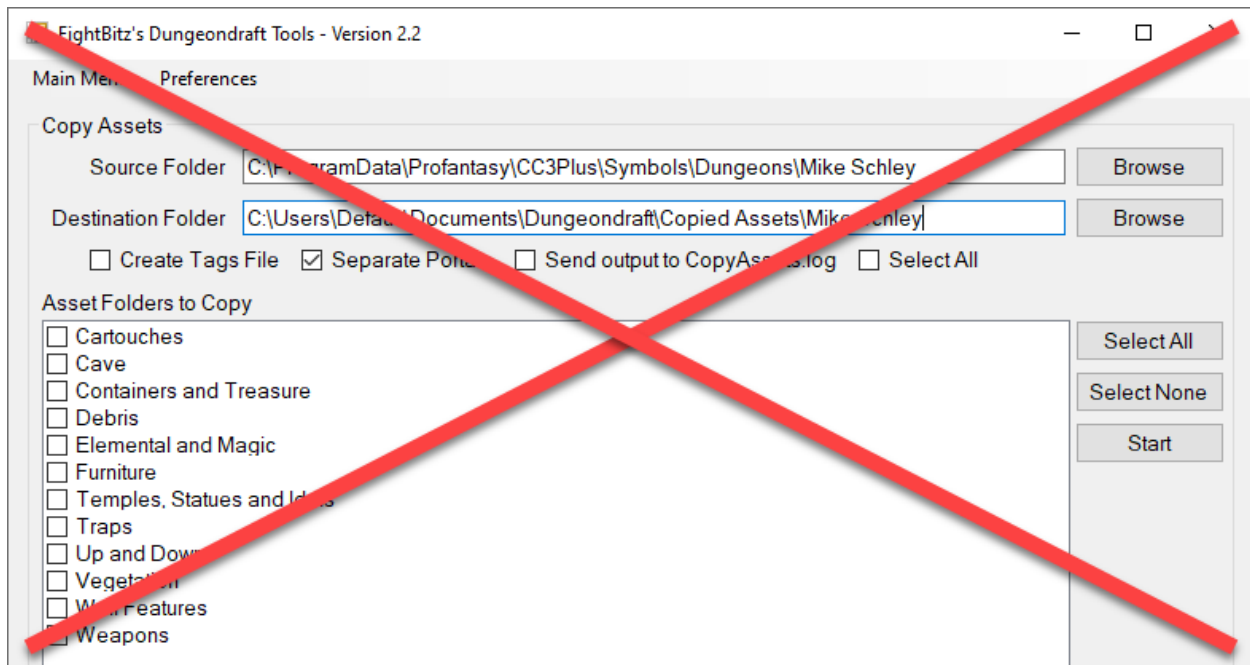
Source Folder

This specifies the folder that contains the folders you want to copy.

As with the other tools, when you select your source folder, the items that appear as checkboxes should be the folders that will become your asset packs. For instance, if I want to copy my Mike Schley assets from my CC3+ folder, I would specify my source folder as “C:\ProgramData\Profantasy\CC3Plus\Symbols\Dungeons”, then I would click the checkboxes for the Mike Schley folders.



What I don't want to do is click into the Mike Schley folder and select the subfolders.



The first method will give me a destination folder structure with “Mike Schley\textures\objects”, then under the objects folder, I’ll have all the subfolders. Cartouches, Cave, Containers and Treasure, etc. That’s what I want my destination folder to look like.

The second method will give me several destination folders that I don’t want.

“Mike Schley\Cartouhes\textures\objects”

“Mike Schley\Cave\textures\objects” (etc)

That second method is what you want to avoid.

Destination Folder

This specifies the folder that will contain the copied assets.

Create Tags File

If you check this, then for each folder that is copied, a default.dungeondraft_tags file will be generated and placed in the appropriate data folder for each copied folder.

Separate Portals

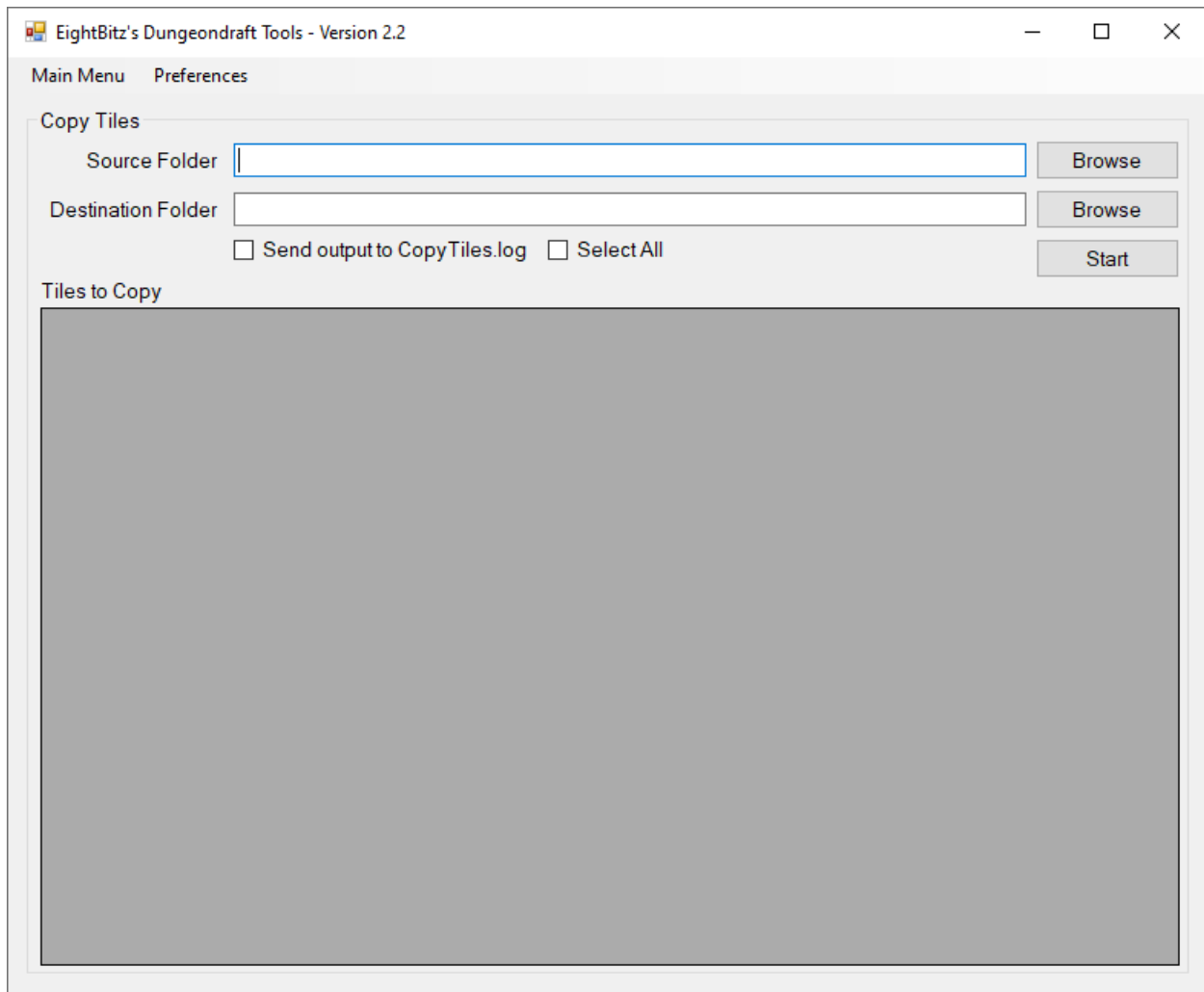
If you check this, doors and windows will be copied to “textures\portals” instead of “textures\objects”.

Send output to CopyAssets.log

This will save output to a log file named “CopyAssets.log”, in the same folder as DDTools.exe.

Copy Tiles

This will present you with the interface for the Copy Tiles tool.



What Does it Do?

Important Details

Like the Copy Assets tool, this tool is geared toward CC3+ and its naming conventions. It will filter through multiple copies of the same image, copying only the highest-quality version.

Also like the Copy Assets tool, this tool will copy assets into a folder structure ready to be packaged for Dungeondraft.

The Controls

Source Folder

This specifies the source folder that contains the tiles you want to copy.

Destination Folder

This specifies the root folder you want to copy the tiles to.

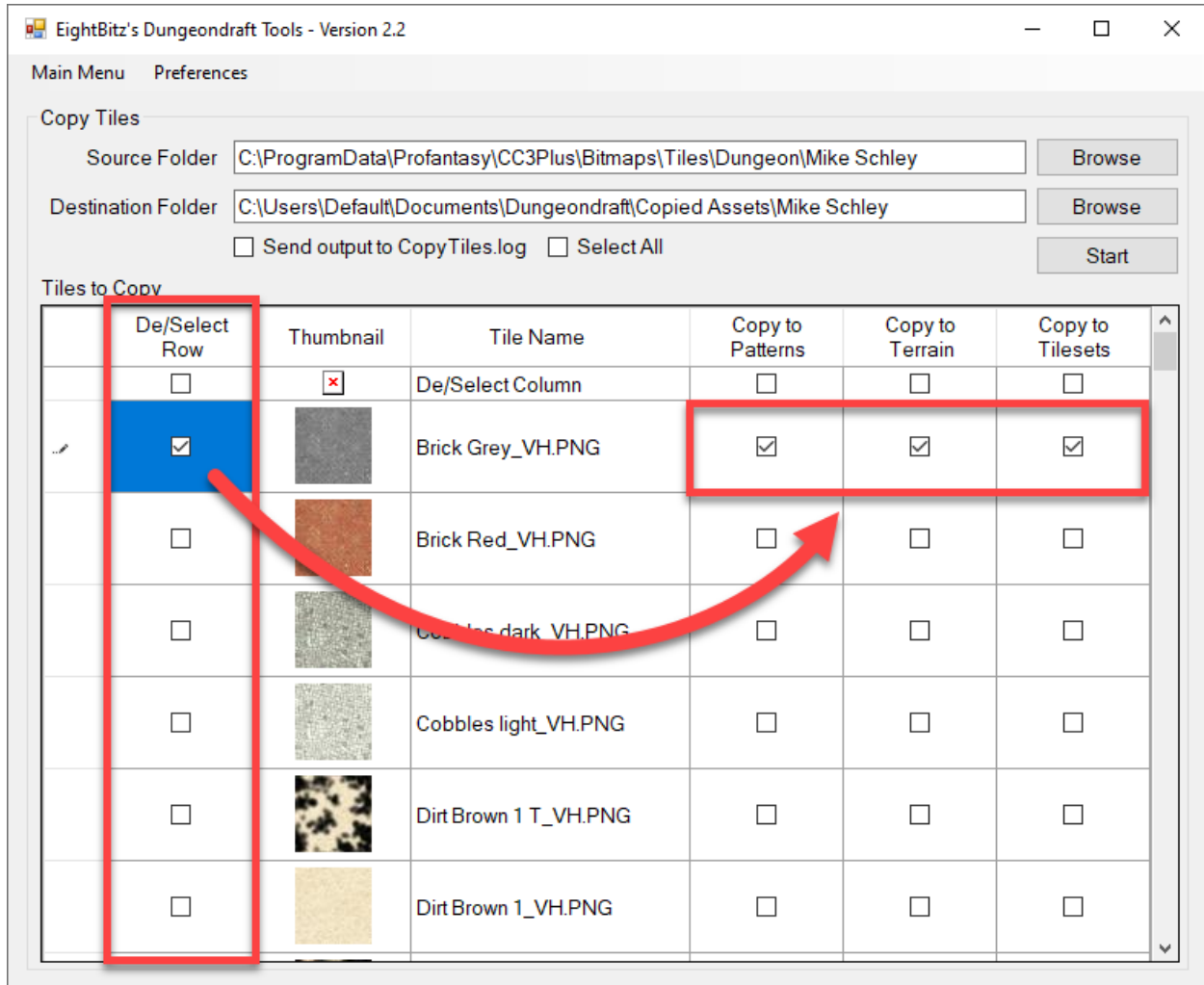
[Send output to CopyTiles.log](#)

This will save output to a log file named “CopyAssets.log”, in the same folder as DDTools.exe.

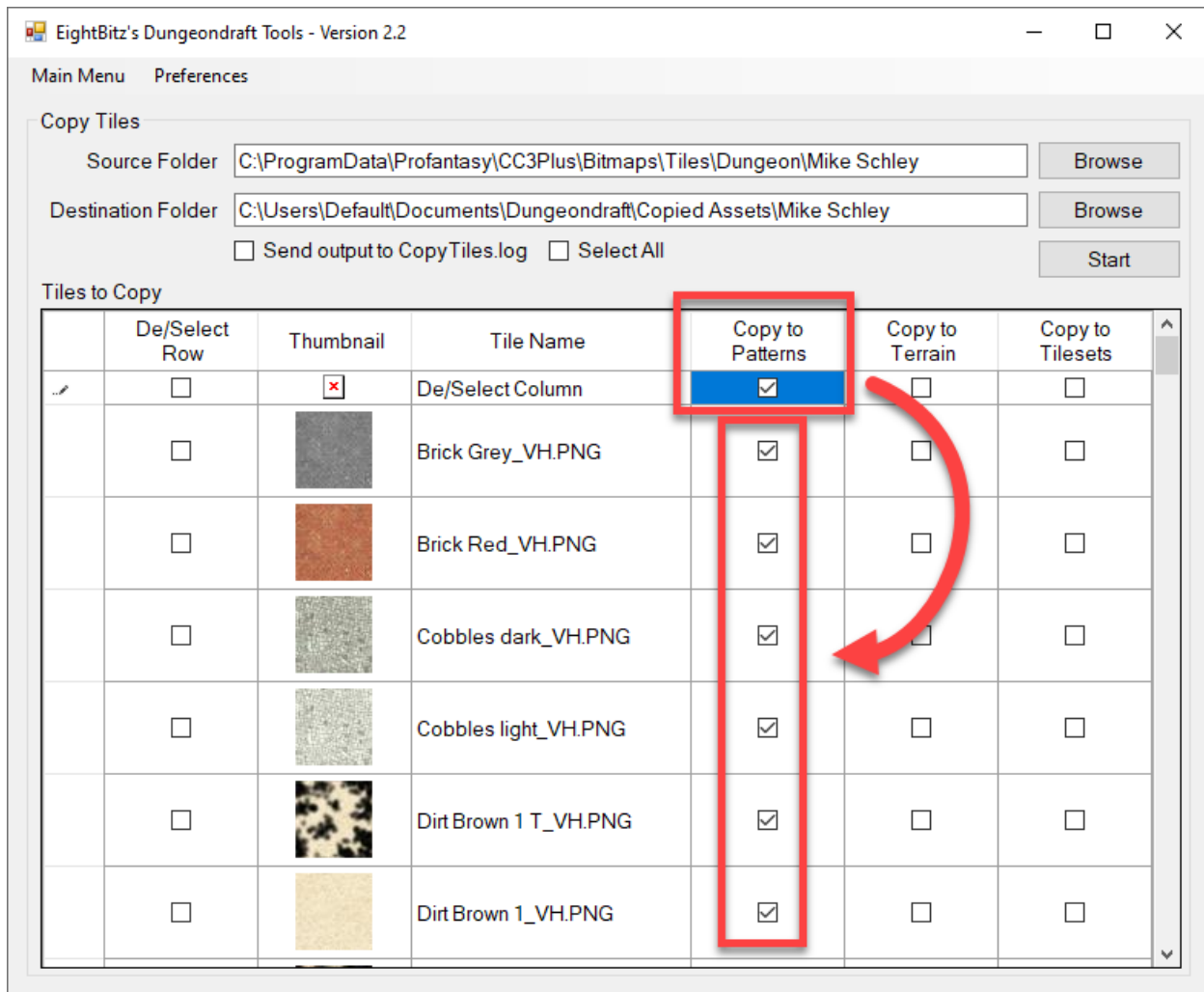
[The Grid View](#)

For each tile, you can check the column for “Copy to Patterns”, “Copy to Terrain” or “Copy to Tilesets”.

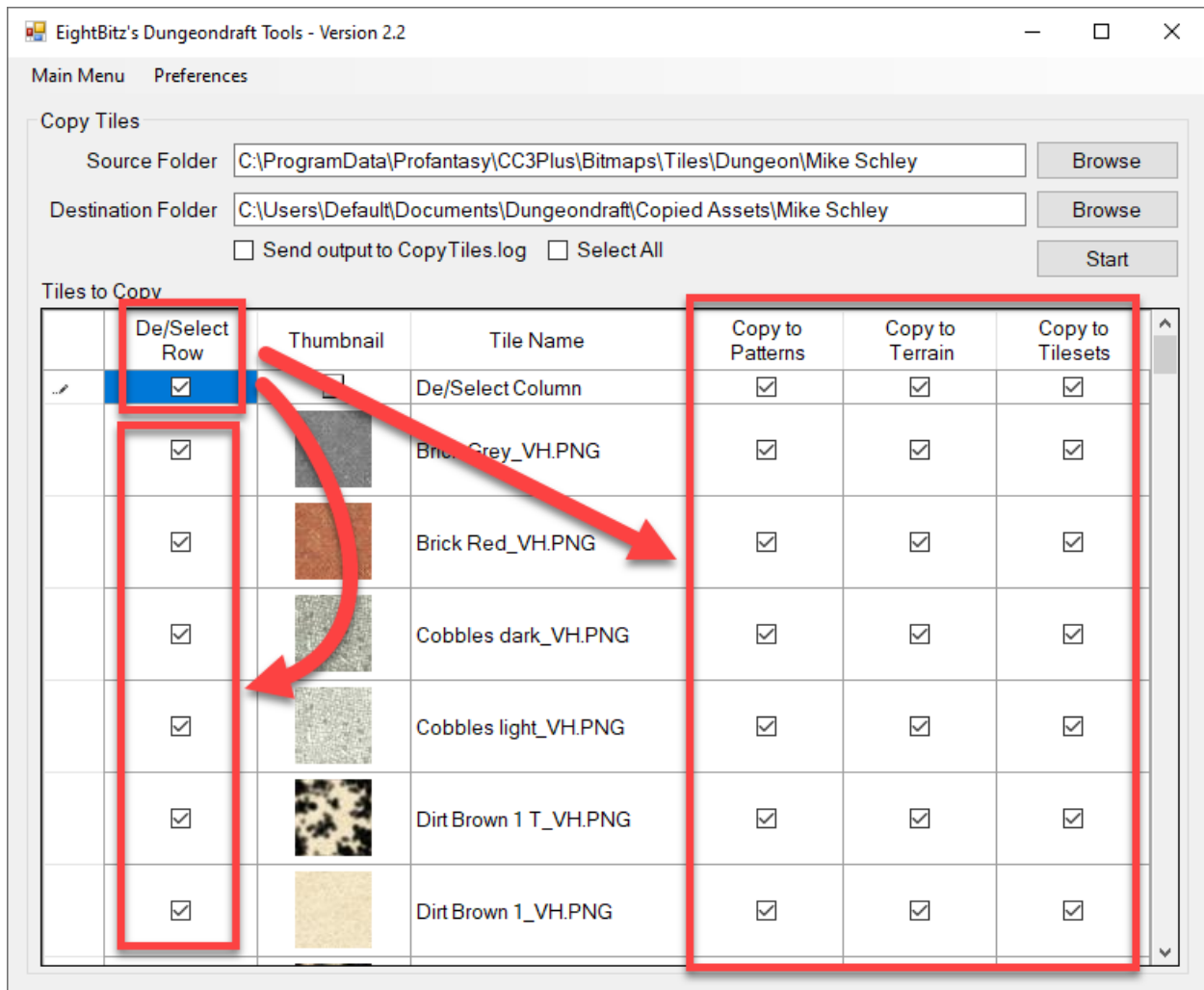
The left-most checkbox of any given row will select or clear all the other checkboxes in that row.



The top-most checkbox of any given column will select or clear all the other checkboxes in that column.



The top-left checkbox will select or clear all other checkboxes.



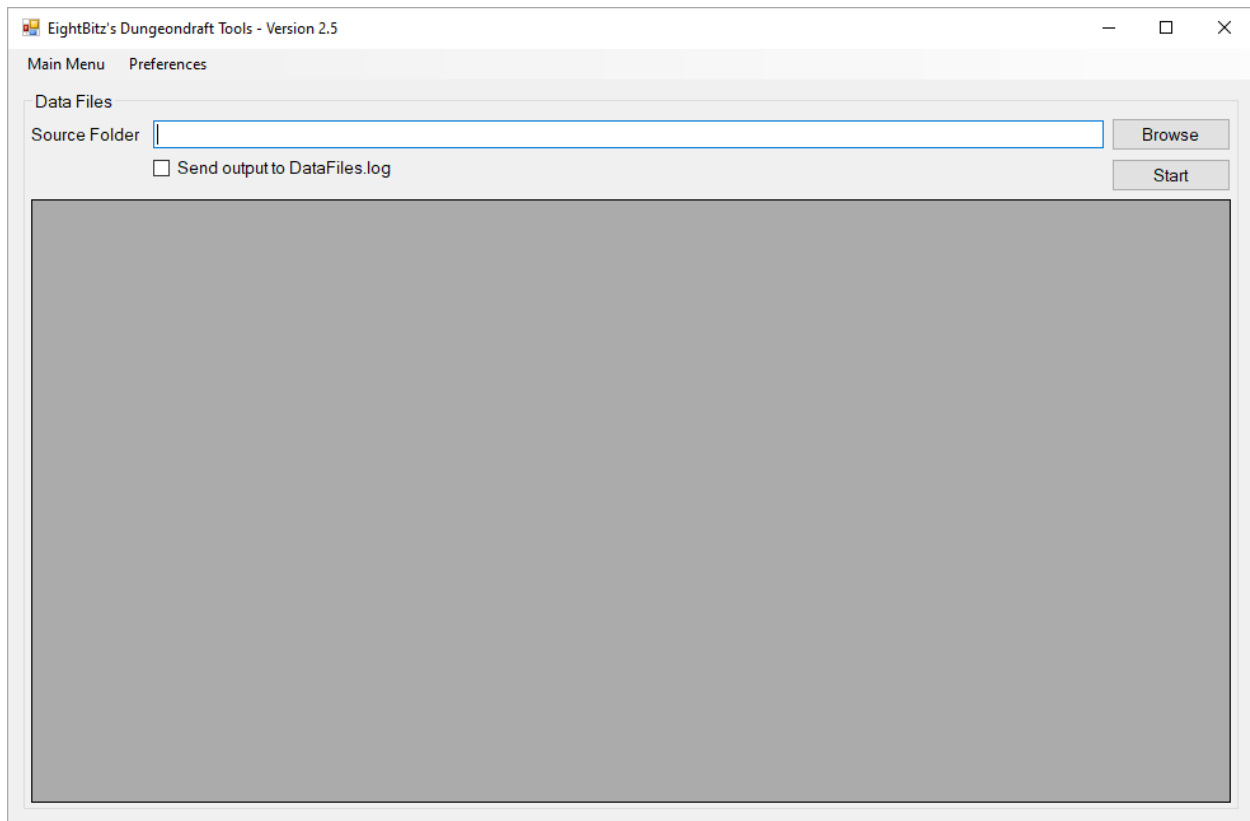
Final Notes

Any tiles for which you select "Copy to Tilesets" will be copied as simple tilesets. When Dungeondraft loads simple tilesets, it also shows them in the "Patterns" tool. Any tile for which you select both "Copy to Patterns" and "Copy to Tilesets" will show up twice in the Patterns tool. Once for the simple tileset, and once for the pattern.

A data file will be generated for each simple tileset. The data file will specify the tileset as normal (rather than custom_color) and will specify the color as white (ffffff). This will present the tileset in Dungeondraft as it would normally appear, with no custom coloring.

Data Files

This will present you with the interface for the Data Files tool.



What Does it Do?

When you're working with custom asset packs, each tileset and each wall requires its own data file.

Important Details

The source folder you specify should be the root folder for the asset pack. If you planning to package a "Mike Schley" folder, then the source folder should be "Mike Schley", not "Mike Schley\textures" or "Mike Schley\data", just "Mike Schley".

When loading information, this tool does the following in the order listed:

- This tool will first get a list of all the files from "textures\tilesets" and all its subfolders and from the "textures\walls" subfolder.
- For "textures\walls", it will skip any files whose names end in "_end". So if you have "sample_wall.png" and "sample_wall_end.png", then the only file it will read will be "sample_wall.png". It will skip "sample_wall_end.png".
- The tool will then assign default values to all relevant fields.
- The tool will then look for relevant data files in "data\tilesets" and "data\walls", and load all the information from any it finds.
- Then it looks for matches with the files it loaded, and for any matches it finds, it replaces the default values with the values found in the data files.
- Then it presents you with the final results, allowing you to make any edits or changes you desire.

- Finally, before it writes the new data files for tilesets and walls, it will delete the existing ones. This will ensure that if you changed any filenames, you don't end up with any stray data files.

The Controls

Source Folder

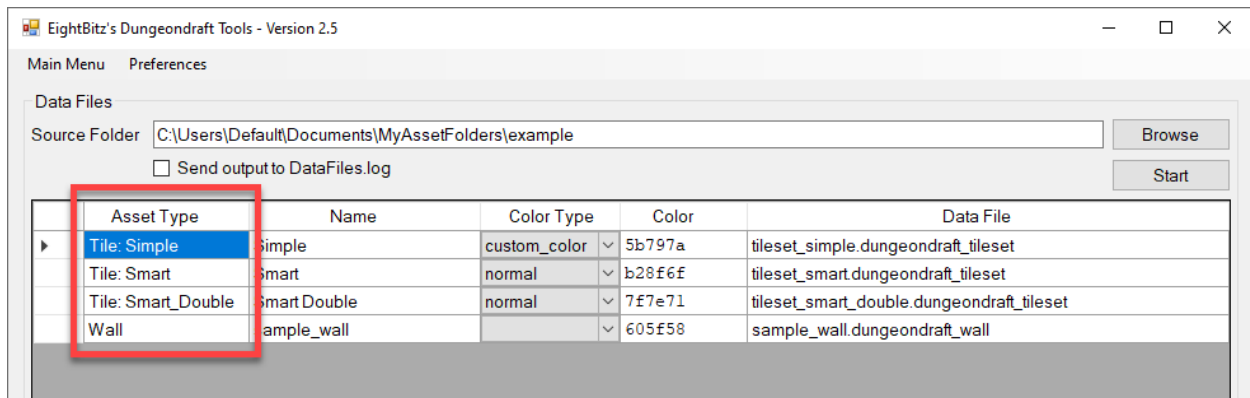
Select as your source folder the folder that you will package as your custom .dungeondraft_pack file. This will be the folder that contains your textures folder and your data folder.

Send output to DataFiles.log

This will save output to a log file named "DataFiles.log", in the same folder as DDTools.exe.

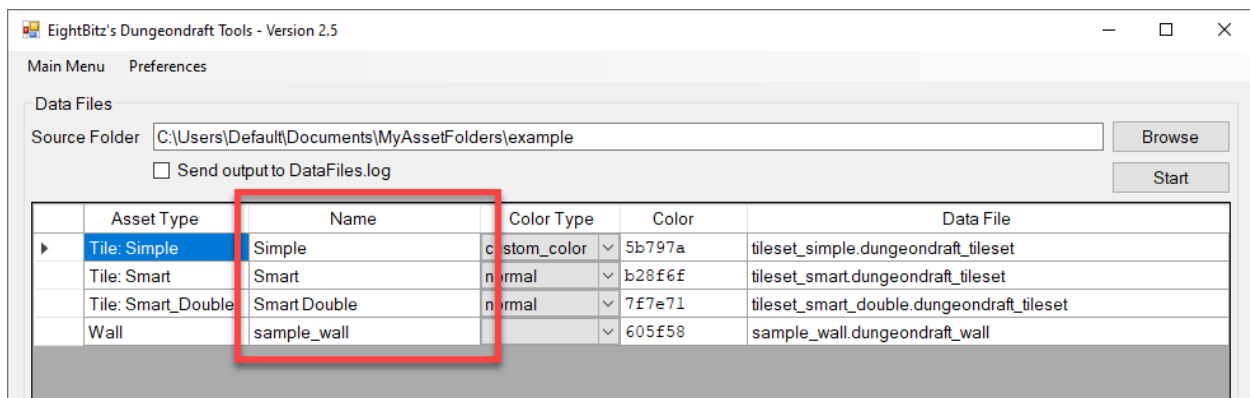
The Grid View

The "Asset Type" column will tell you what kind of asset is in each row. This column is read-only. You cannot edit the value.

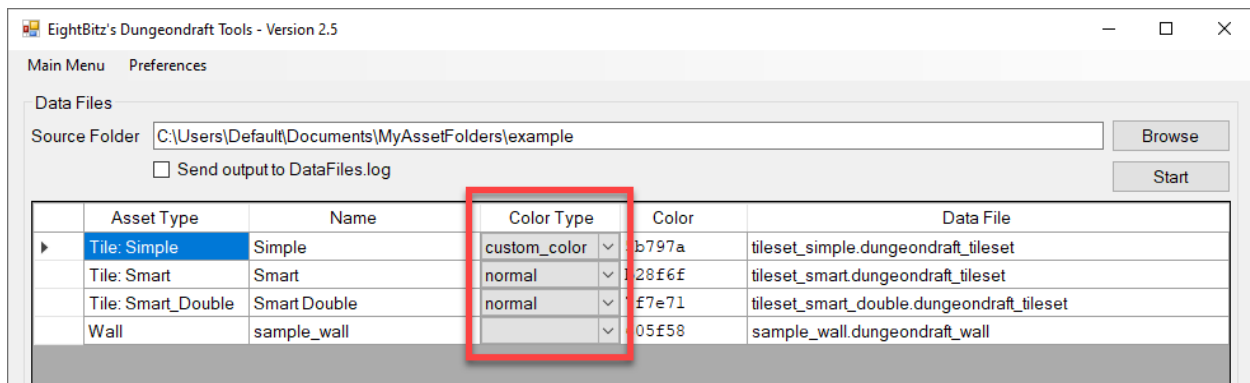


The "Name" column lists the name of the tileset or the wall. For tilesets, this defaults to the file name, without the extension, unless there's a data file with an overriding value. For walls, this defaults to the file name, without the extension. The data file for walls does not contain a name value.

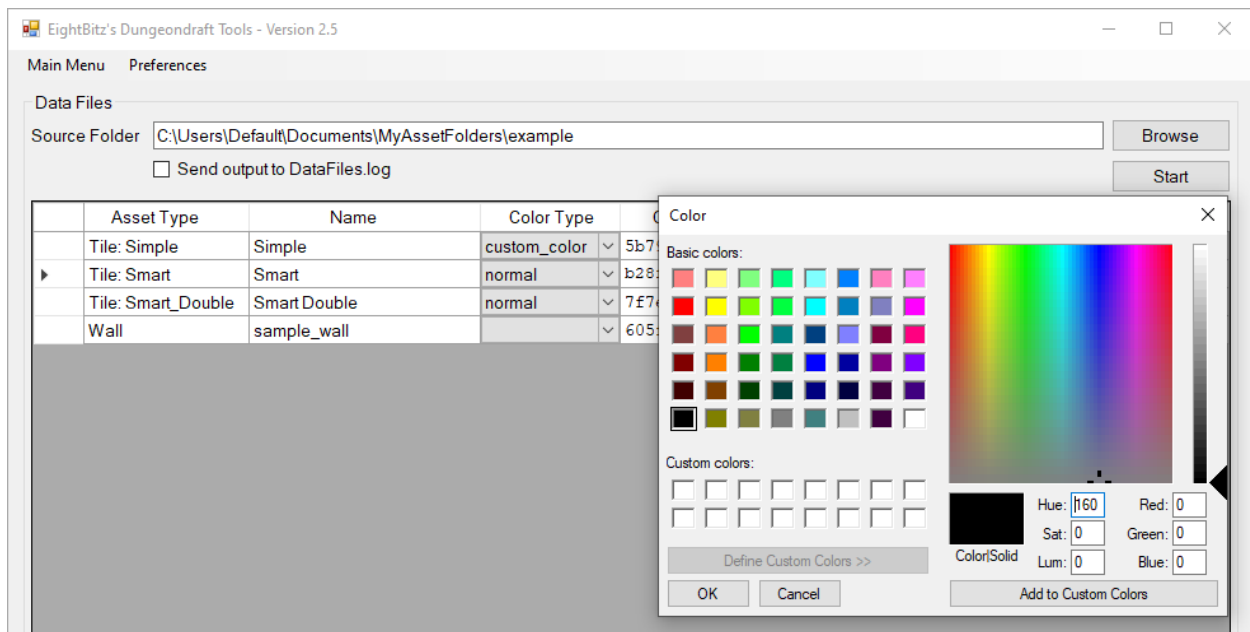
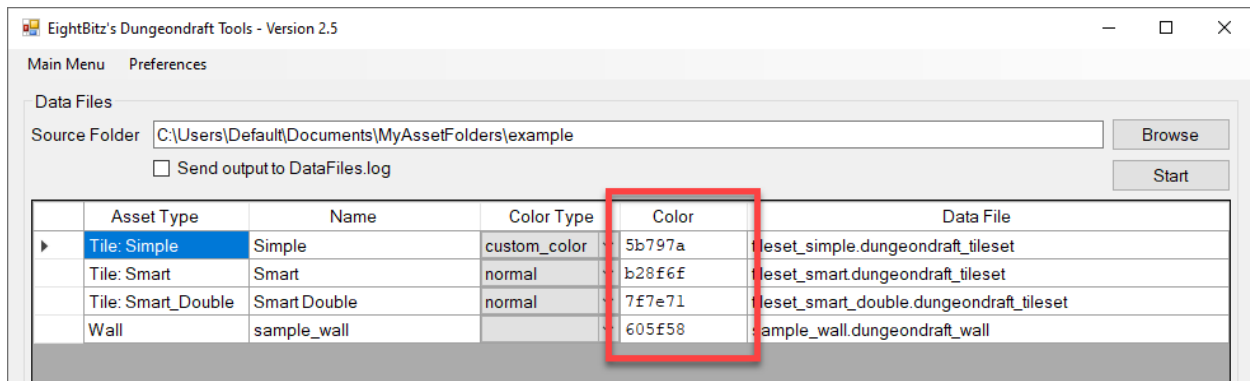
This value can be changed for tilesets, but for walls, it's read-only. If you want to change the name of a wall, you'll have to change the file name.



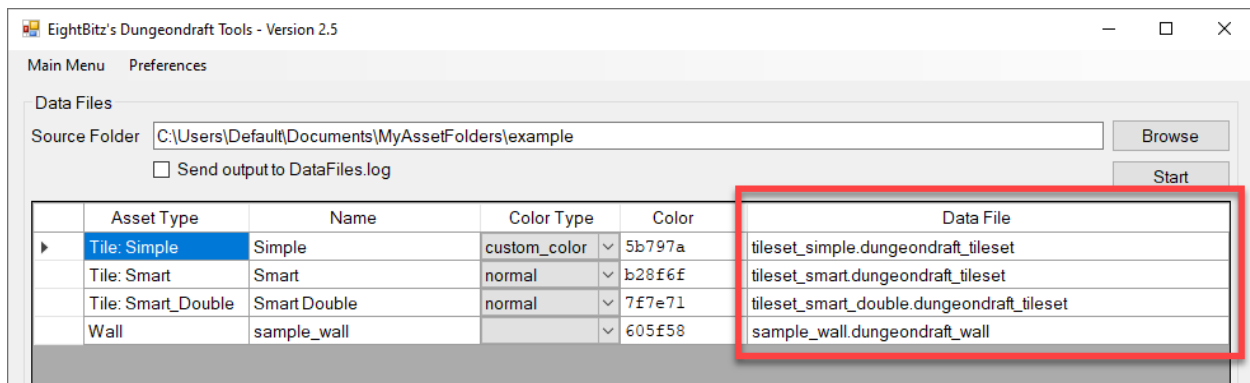
The "Color Type" column determines whether a tileset is colorable within Dungeondraft, or will have a fixed color within Dungeondraft. For walls, this value is blank, and it's read-only.



The “Color” column defines the default color for the tileset or wall. If you want to edit the color for a tileset or wall, clicking the value once will let you edit the hex code. Double-clicking will open a color-picking dialog.



The “Data File” column will allow you to edit the name of the data file for the tileset or wall.



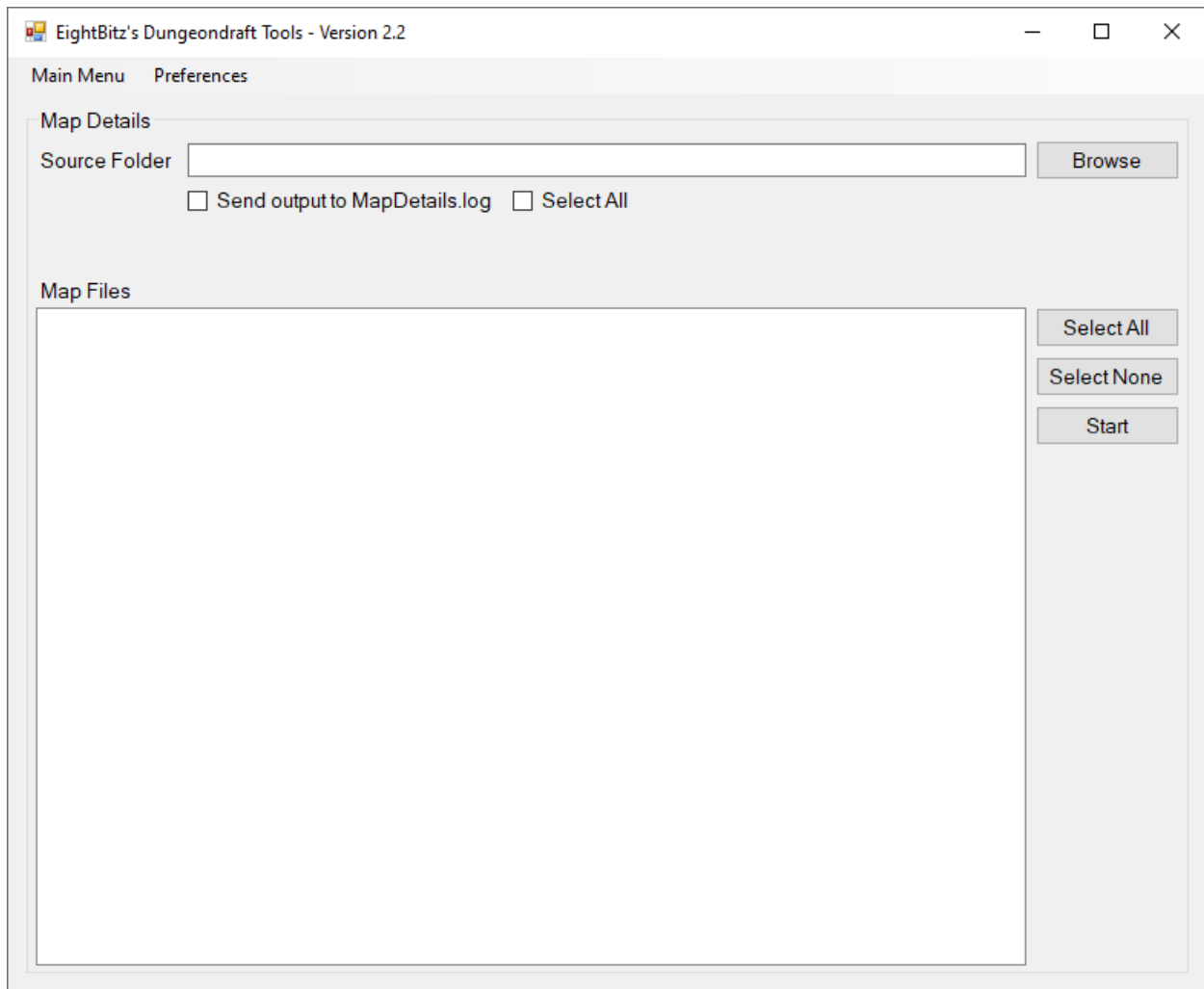
Start

Write all the specified information to all the listed data files. This will first purge all the existing data files in case you edited any filenames. This may not be the most elegant way to make sure you don't end up with stray data files, but it works.

Final Notes

Map Details

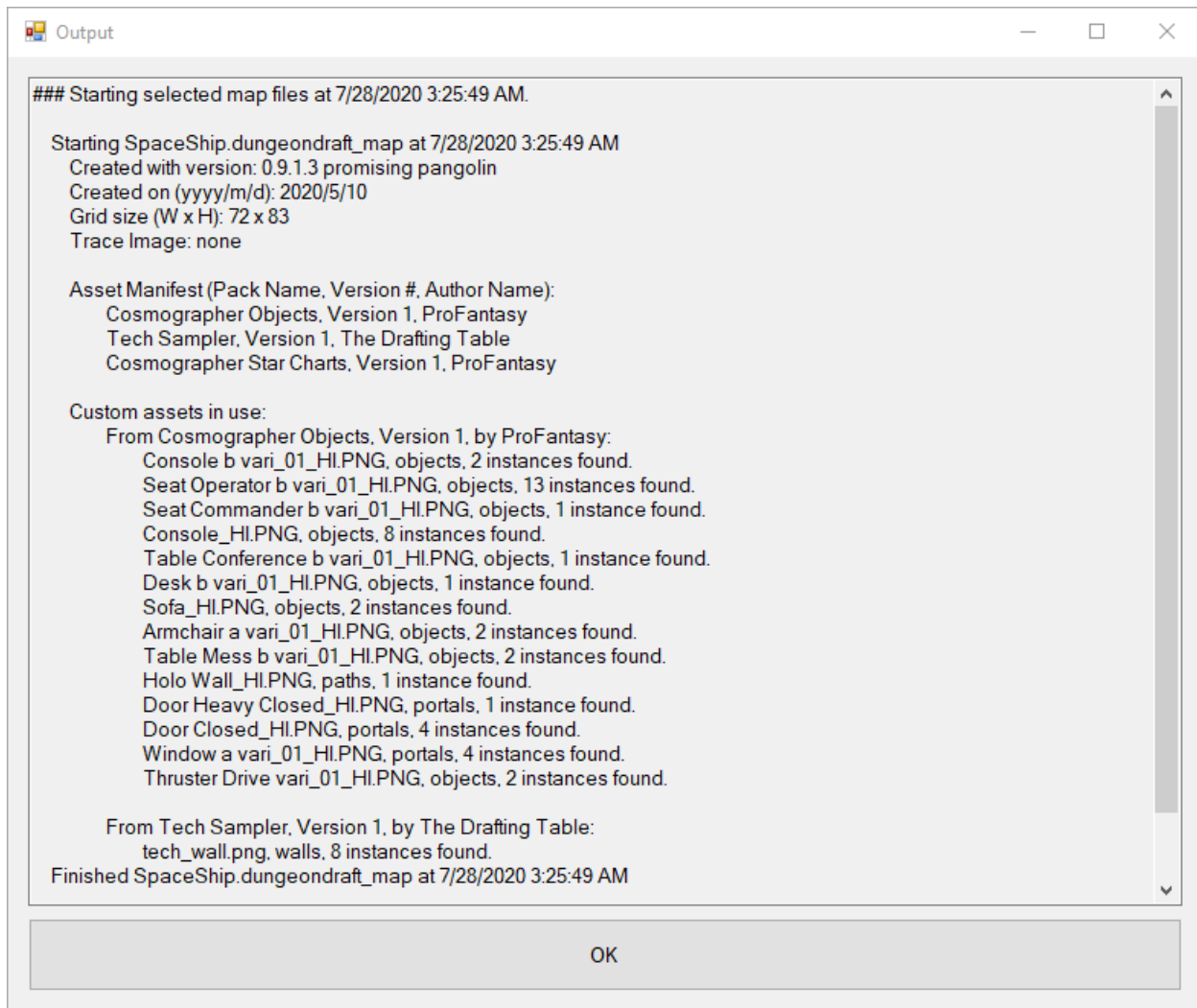
This will present you with the interface for the Map Details tool.



What Does it Do?

This tool presents you with information about a selected map file, including:

- What version of the Dungeondraft created the map.
- The map's creation date.
- The map's grid size.
- The map's asset manifest.
- Which custom assets are actually in use on the map.



The Controls

[Source Folder](#)

This specifies the source folder that contains the tiles you want to copy.

[Send output to MapDetails.log](#)

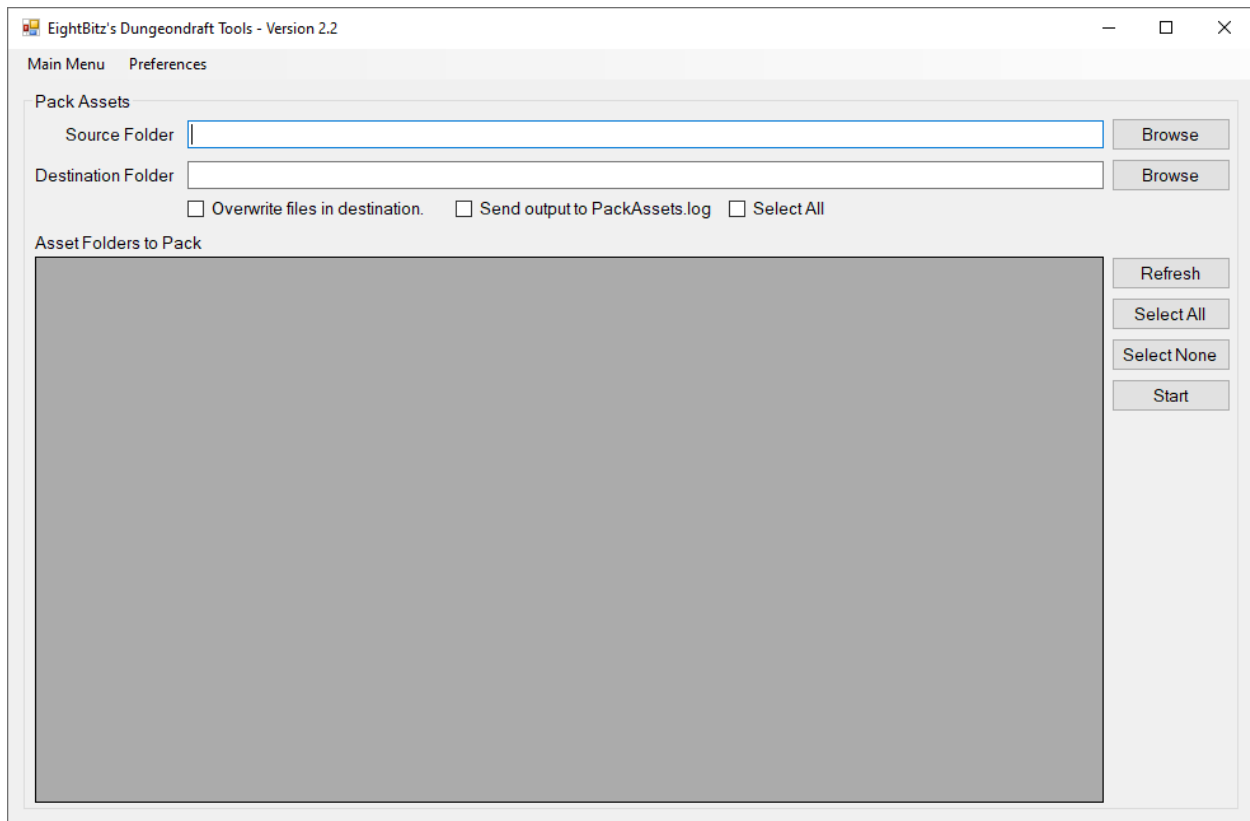
This will save output to a log file named "MapDetails.log", in the same folder as DDTools.exe.

Final Notes

As far as which custom assets are in use, this tool will display everything except tilesets. I cannot find a reliable way to determine which custom tilesets are in use and which custom asset packs they're from. This information is not stored in the map in the same manner as it is with other custom assets.

Pack Assets

This will present you with the interface for the Pack Assets tool.



What Does it Do?

This tool will create .dungeondraft_pack files from selected asset folders.

Important Details

dungeondraft-pack.exe must reside in the same directory as DDTools.exe

The Controls

Source Folder

This specifies the folder that contains the asset folders to be packed.

Destination Folder

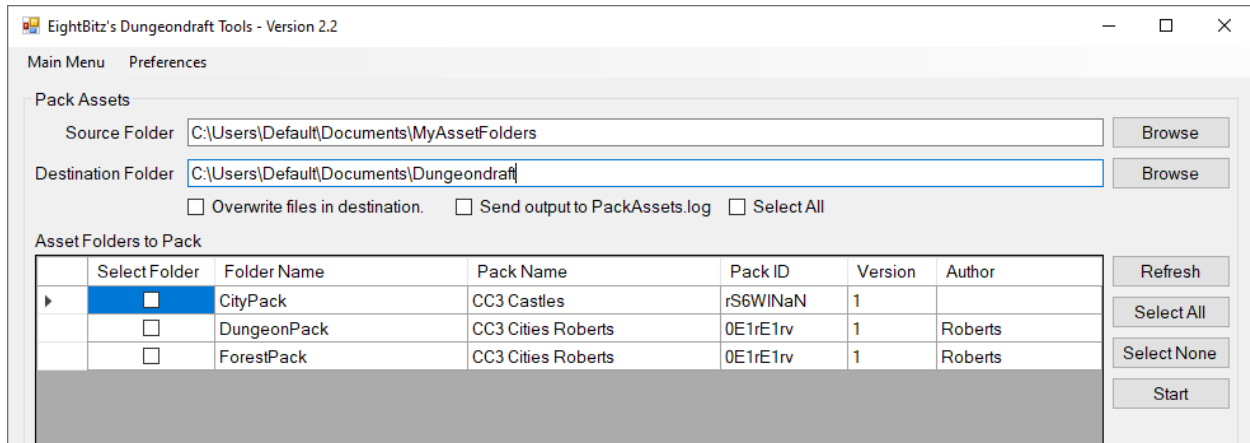
This specifies the folder that will contain the packed asset folders.

Overwrite files in destination.

If this checkbox is checked, then any existing packages in the destination folder will be overwritten by the creation of new packages.

The Grid View

Once you select your source folder, the Pack Assets tool will search within to find any subfolders that themselves contain a “textures” subfolder.



The columns you can edit are Pack Name, Version and Author. You cannot edit Folder Name or Pack ID.

For any folders that have been previously packaged, their information will be read from their pack.json files. For any folders that have not been previously packed, the Pack Name will be copied from the folder name, the Pack ID will be blank, the Version will be 1, and the Author will be blank.

If you edit the Pack Name of any item with a Pack ID, the pack.json file will be updated to give the folder a new Pack ID, then a package will be created with the new Pack Name and the new Pack ID.

If you leave the name unchanged but edit the Version or the Author, then the folder's pack.json file will be updated with the new Version and Author, but the Pack ID will remain the same. Then a package will be created with the updated information.

For any folders that have not been previously packaged and do not contain a pack.json file, a Pack ID will be generated, and a pack.json file will be created with the specified Pack Name, Version and Author. Then a package will be created with the specified information.

Refresh

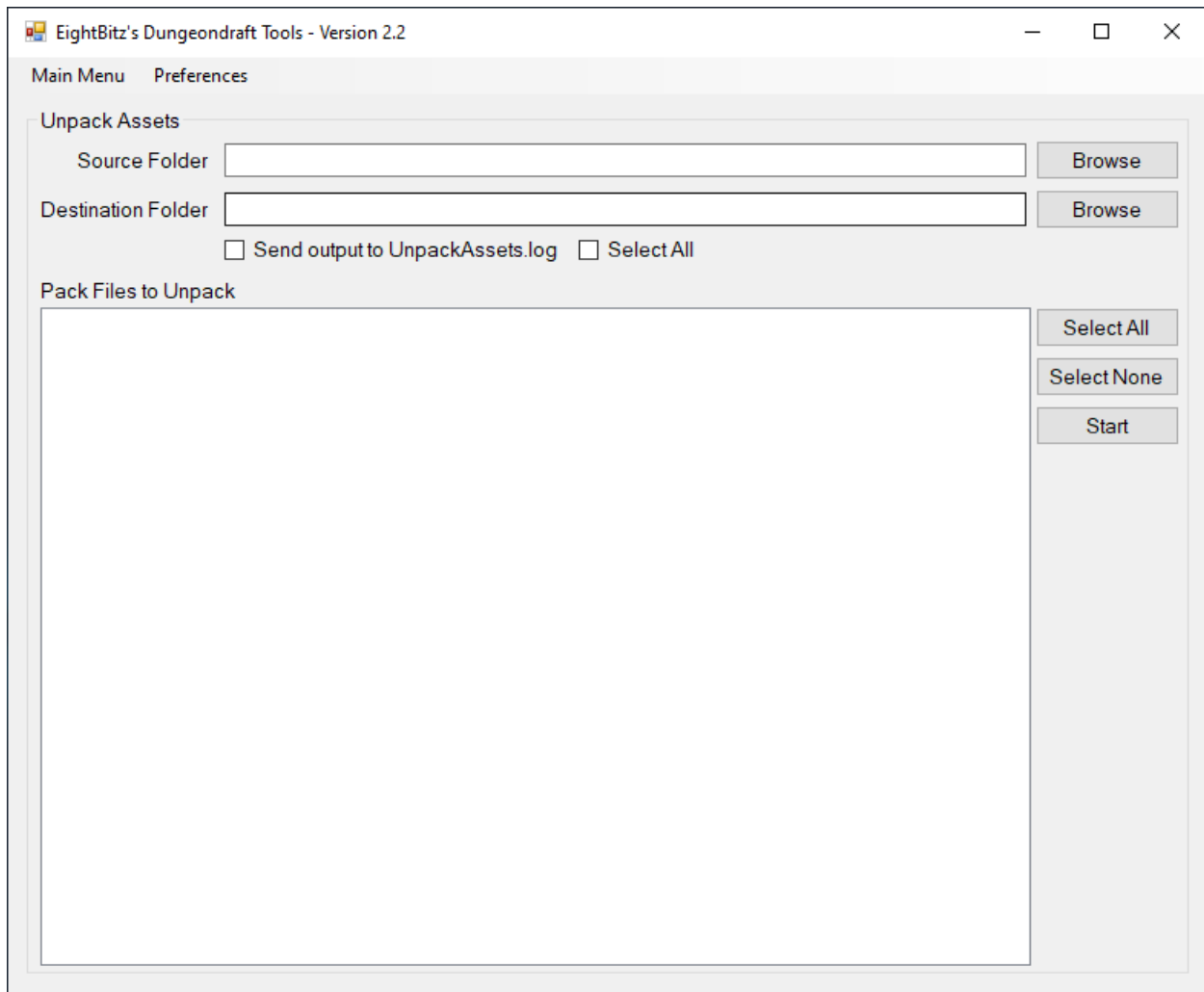
If you have edited any information in the grid view, but have not yet clicked Start, to update or create any files, you can click Refresh to clear your edits and restore the all the values as they exist in the current files.

Start

Start packing the selected folders.

Unpack Assets

This will present you with the interface for the Unpack Assets tool.



What Does it Do?

This tool will unpack .dungeondraft_pack files.

Important Details

dungeondraft-unpack.exe must reside in the same directory as DDTools.exe

The Controls

Source Folder

This specifies the folder that contains the .dungeondraft_pack files to unpack.

Destination Folder

This specifies the folder that will contain the unpacked asset folders.

Pack Files to Unpack

Once you select your source folder, this will contain a list of .dungeondraft_pack files from which you can select which one(s) you wish to unpack.

Start

This starts the unpacking process.

Version History

Version 3.5

- Conversion from PNG to webp is now completely internal and self-contained. ImageMagick is no longer needed.
- Converting an Asset Folder or an Asset Pack will rewrite the pack.json file with a new ID while preserving the name, author and version of the original folder or pack.
- File names, folder names and pack names are now appended with “ (webp)”.
- Fixed a bug in “Convert Packs” where after it unpacked and converted, it repacked the original content instead of the converted content. (Yes, I know. I did a severe facepalm. I’m embarrassed. I’m sorry.)
- Added more asset types for conversion. Previously, only objects were being converted, because I knew webp did not work for some asset types. I finally took a deeper dive into which ones work and which ones do not.
 - Lights, objects, paths, patterns, portals and terrain are included in conversion.
 - Materials, tilesets and walls are excluded from conversion. Materials and tilesets would cause crashes, and walls just did not work properly. These files are simply copied instead of converted.

Version 3.0

- Map Details now displays the hex codes for any custom colors you’ve used.
- Copy Tiles now supports thumbnails for webp files.
- The program must now be installed with setup.exe.
- The config file is saved to %appdata%\EightBitz\DDTools.
- A Help menu has been added.

Version 2.5

- Changed the behavior of check boxes so they only require a single click to check or uncheck.
- The Map Details tool will now list embedded assets (the assets that the 0.9.3.3 update allows you to drag-and-drop onto the map from the file system).
- Fixed some issues with how the program saves to and loads from the config file.
- Added a tool to create/manage data files for tilesets and walls.

Version 2.2

- Fixed an issue where when the Preferences menu was used to load preferences, the preferences would load but wouldn’t take effect.
- When “Create Tags File” is checked while copying assets, the “DefaultTag” value will be read from the preferences for “Tag Assets”. If there is no saved setting, the “DefaultTag” will not be applied.
- When asset packs are listed from the asset manifest, the version number and author of each pack are now listed with the pack name.

- When custom assets are listed by asset pack, the version number and author of each pack are now listed with the pack name.
- Fixed an issue where some blocks of code were referencing the wrong controls.
- “Select All” is no longer assumed for any of the tools, but it can now be saved as a preference.
- The value for “Destination Folder” will no longer be automatically changed unless it’s blank.
- Fixed an issue where the “Finished” message for Copy Tiles did not display.
- Fixed an issue where Pack Assets did not list the items it was packaging.
- Fixed an issue where copied tiles were not properly named.
- Various other logging issues, details and oversights.

Version 2.1

- Fix an issue where, when a tool was selected, the preferences would load but wouldn’t take effect.

Version 2

- The "Select Tool" menu has been renamed to "Main Menu".
- Copy Tiles: This is a new tool that will copy tile assets in the same manner that Copy Assets copies objects and portals. You have the option of copying the tiles to Patterns, Terrain or Simple Tilesets.
- Map Details: This is a new tool that will show you some basic information about your map. The date it was created, which version of DD it was created with, the grid size, which custom asset packs were selected when the map was saved, and which custom assets are actually in use on the map.
- Pack Assets: Since Ryex updated the dungeondraft-pack.exe utility to use the same algorithm for generating unique IDs, I reworked the Pack Assets tool to take full advantage of this. You can now edit and create packs outside of Dungeondraft.
- Preferences menu: For all of you who are sick of browsing to the same folders every time you load the tool, you can now save your preferences, not just for the folder locations, but for all the available options.