



Coffee Auto Patcher

User Guide

Release v1.0.0.29

August 3rd, 2019

CoffeeBns.com

The content of this manual if furnished for informational use only, is subject to change without notice and should not be construed as a commitment by its authors. The author assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

Firstly, thank you for purchasing the Coffee Auto Patcher system. If you have any requests, questions, comments or suggestions for this asset or any other assets, please feel free to contact me on

Twitter @CoffeeReclaimed

Through the CoffeeBnS Discord at <https://discord.gg/9WYC2sQ>

Email at support@coffeebns.com

Discord is the best location if you have any questions, comments or need any assistance setting up your package.

This guide includes full setup for configuring your Amazon AWS account and S3 bucket securely, as well as implementing the Coffee One Click Update system into your game.

For a full video walkthrough guide on setup check out

<https://youtu.be/iY6Ke3yzpHs>

! IMPORTANT !

Before you continue any further, create a new empty Project and Import the Coffee Auto Patcher into that Project. If you imported the Coffee Auto Patcher into your game, please remove all files from the Import other than this guide and the Coffee Game Files Folder.

When Unzipping the Source Files zip file, ensure you unzip it to a location outside of your Unity Assets Folder or it will cause issues with your Patcher upon building.

Table of Contents

If you are planning on using this with a CDN other than Amazon and do not plan on taking advantage of the Coffee One Click Update Program, skip to Page 9.

Amazon AWS and S3 Setup

AWS Account Creation	4
User Setup	4-5
S3 Setup and Config	6

Configuring Coffee One Click Update

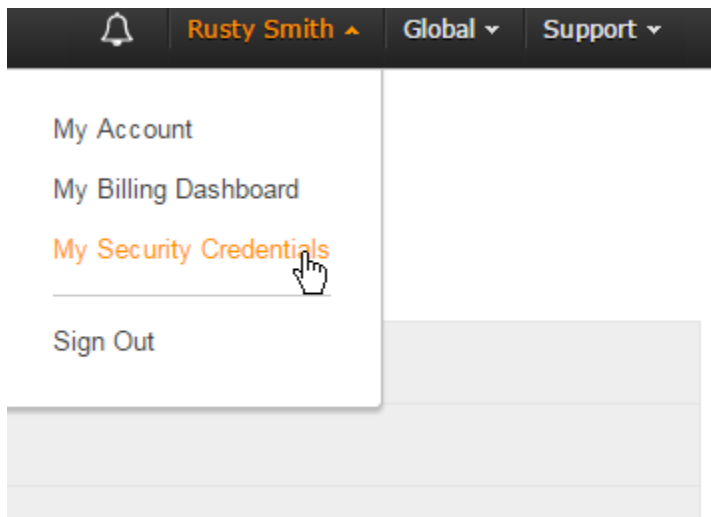
Coffee One Click Update Configuration	7
---------------------------------------	---

Coffee Auto Patcher Integration

Building the Unity Auto Patcher	9
Configuring Coffee Auto Patcher	10
Game Update Integration	11

Amazon AWS User Setup

1. Create your account at <https://aws.amazon.com>
2. Once signed in, click on your username and select My Security Credentials. If you have not already enabled two-factor authentication, it is highly recommended you set it up at this time.



3. From the Security Credentials Dashboard, click on Users then Add User. Ensure you check the box for Programmatic Access

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

[+ Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* ☒ **Programmatic access**

Enables an **access key ID** and **secret access key** for the AWS API.

☐ **AWS Management Console access**

Enables a **password** that allows users to sign-in to the AWS Managi

* Required

- You will now be prompted to select Permissions for this new User. Type S3 into the Policy Box and select AmazonS3FullAccess




Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.



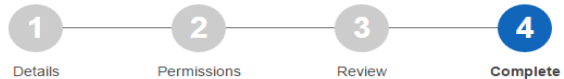
Attach one or more existing policies directly to the user or create a new policy. [Learn more](#)

[Create policy](#) [Refresh](#)

Filter: Policy type ▾		Q S3			
		Policy name ▾	Type	Attachments ▾	Description
<input type="checkbox"/>	▶	 AmazonDMSRedshiftS3Role	AWS managed	0	Provides access to manage S3 settings for Redshift endpoints for DMS.
<input type="checkbox"/>	▶	 AmazonS3FullAccess	AWS managed	1	Provides full access to all buckets via the AWS Management Console.
<input type="checkbox"/>	▶	 AmazonS3ReadOnlyAccess	AWS managed	0	Provides read only access to all buckets via the AWS Management Console.

- Once you have selected Permissions the new User will be created. You will be given an Access Key and Secret Key. Click Download .csv and save this file to a secure location on your computer, you will use this file to upload files to Amazon S3.

Add user



✓ **Success**
You successfully created the users shown below. You can view the console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: [I](#)

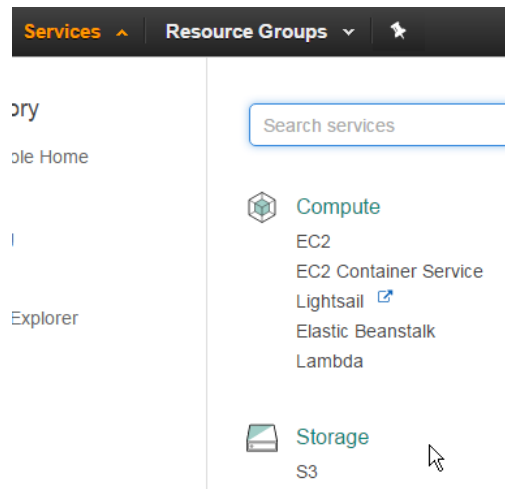
[Download .csv](#)

	User	Access key ID	Secret access key
▶	✓ s3admin	AKIAI44QH8DHBVS7G13GR	***** Show

[Close](#)

Amazon S3 Setup

1. Now that your S3 user has been created, you can create the S3 Bucket that you will be using for your Patching. Select Services then click on S3. I recommend using a name like patch.yourgame.com for your bucket name.



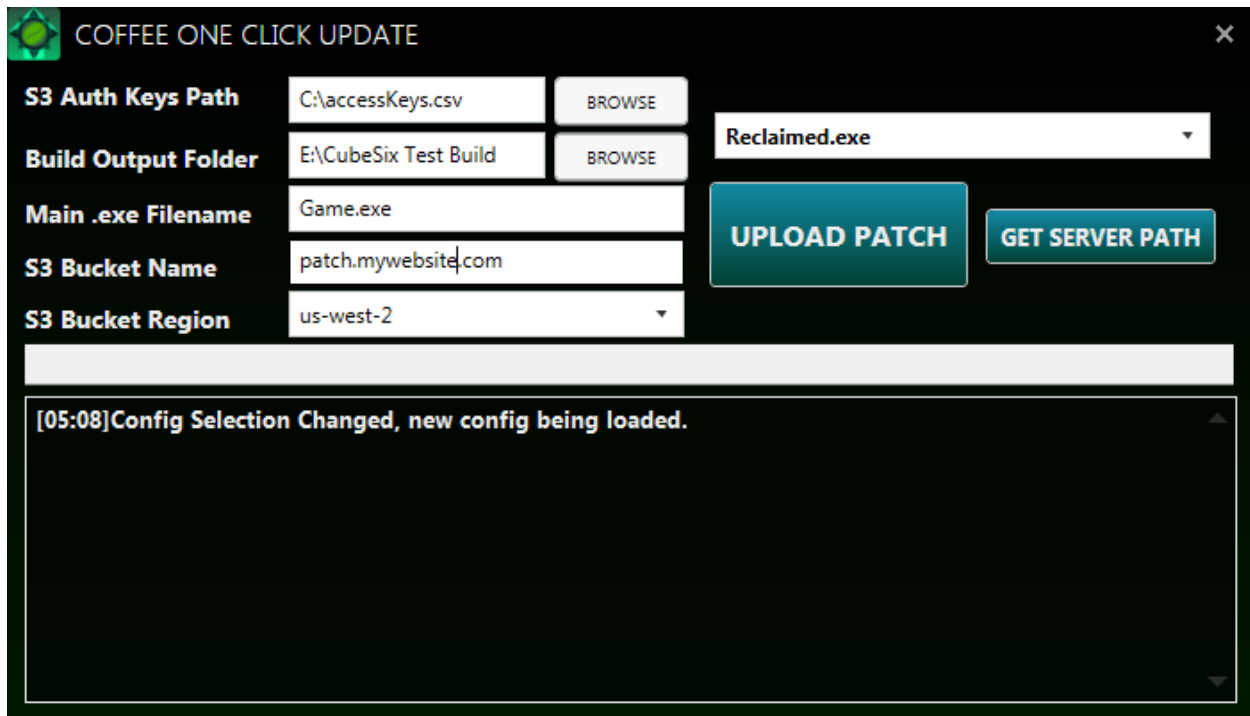
2. Once the bucket has been created, select the bucket, click Properties on the Upper Right, then click Permissions > Add New Permissions. From here, your new permission is going to be for **Everyone** and you are going to give them the permission to **List only**. Ensure that they do not have any other permissions.
3. S3 is now setup and you can move on to configuring the Coffee Deploy Application.

Public access

Group	List objects	Write objects	Read bucket permissions
<input type="radio"/> Everyone	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No	<input type="checkbox"/> No

Coffee One Click Update Configuration

If you intend on using the Coffee One Click Update Application to deploy your new game builds via Amazon S3, these are the fields required to prepare for Deployment. Unzip the Coffee Patch Deploy WPF zip file found in Source Files > Coffee Patch Deploy WPF into a folder outside of your Unity Project (it will cause errors if unzipped within your project) and run this from the Coffee One Click Update App folder.



COFFEE ONE CLICK UPDATE

S3 Auth Keys Path C:\accessKeys.csv BROWSE

Build Output Folder E:\CubeSix Test Build BROWSE

Main .exe Filename Game.exe

S3 Bucket Name patch.mywebsite.com

S3 Bucket Region us-west-2

Reclaimed.exe

UPLOAD PATCH GET SERVER PATH

[05:08]Config Selection Changed, new config being loaded.

S3 Auth Keys Path – This is the path to the .csv file containing the ACCESSKEY and SECRETKEY that you saved from the new Amazon S3 user creation.

Build Output Path – This is the directory that contains all files included in the build of your game.

Main .exe Filename – This is the full filename of the main exe file for your game that you want to launch after patching is complete.

S3 Bucket Name – This is the name that you chose for the Amazon S3 bucket that your game files will be stored in.

S3 Bucket Region – This is the region that your Amazon S3 Bucket resides in. To find this go to your S3 management Panel click Properties > Static Website Hosting. The Endpoint URL that you see will have a region near the end of the URL

Endpoint: patch.playreclaimed.com us-west-2.amazonaws.com



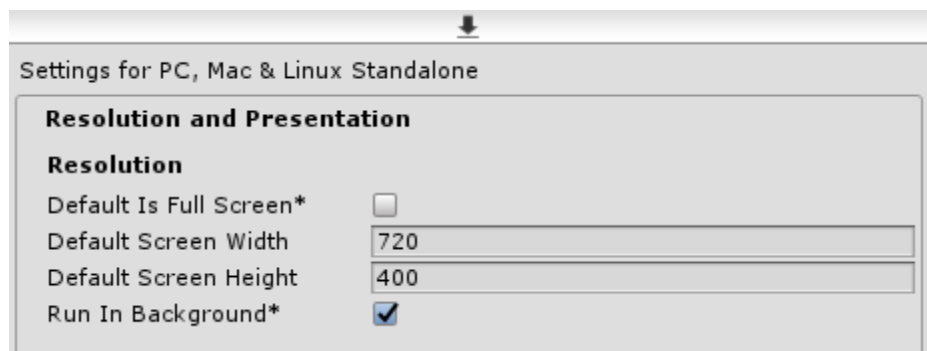
1. Once you have

successfully entered all of the information for your S3 server, click the GET SERVER PATH button to get your full fileList patch copied to your clipboard. One Click Deploy configuration is now complete. Now you will make a new build of the Patcher for your game.

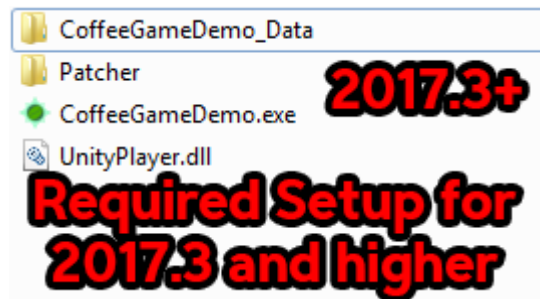
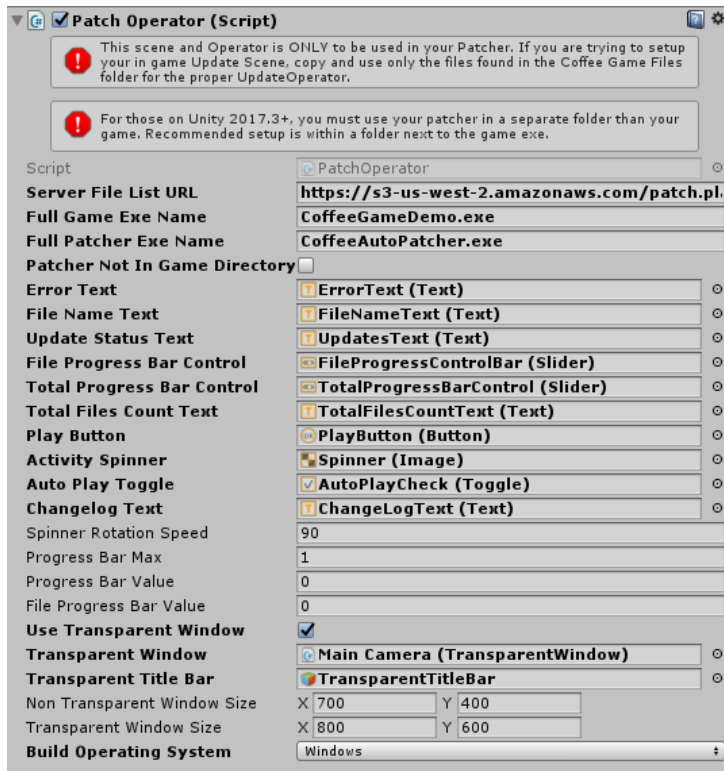
Building the Unity Auto Patcher

In order to begin setting up the new Coffee Auto Patcher for your game, first you are going to open up another copy of Unity and Create a new Project. Name this project whatever you would like to use as the name for your Patcher/Launcher.

1. Once you have created the new project, import in the full Coffee Auto Patcher Package.
2. Click on File > Build > Player Settings and click the Resolution and Presentation Tab. It is recommended you uncheck Full Screen and set Width to 800 and Height to 600. Check the box for Run in Background. **For those on Unity 2019.1.13+ ensure you disable the setting for Use DXGI Flip Model Swapchain for D3D11 as it will cause issues with transparency.**



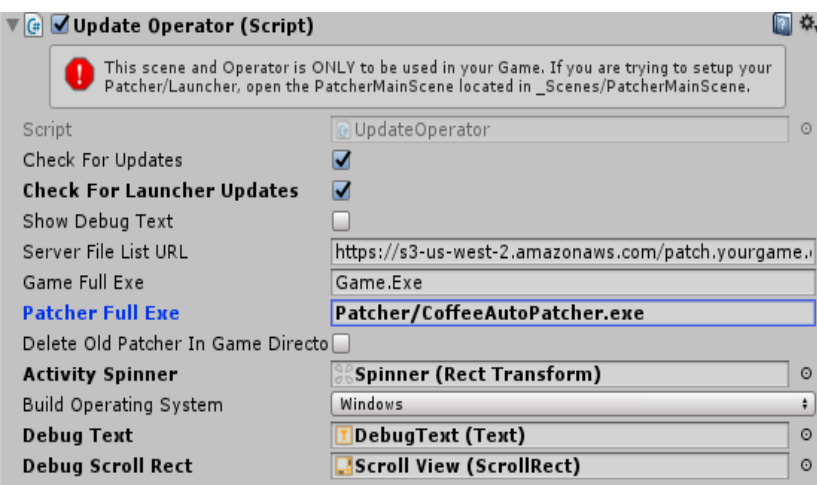
3. Open up the Coffee Auto Patcher Scene and select the PatchOperator Object. Paste in the Server File List URL that was generated by the Coffee Patch Deploy if you are using it or place in the full root path of your CDN/FTP if you are using one followed by fileList.txt. Example: <https://myFTPorCDN.com/gamepatch/fileList.txt>
4. Under the full Game Exe Name, input the full executable name of your Game. For windows builds this will end in .exe, for Mac it will have no extension and be located in the game.app/Contents/MacOS/ folder and for Linux it will end in .x86



5. Once you have customized and configured the Patcher to your liking, add the Coffee Auto Patcher scene as the main scene and Build the Patcher under whichever name you like and build it to a folder (Create a new folder called Patcher for simplicity) within the directory your Game is built to and/or will be built to. Take note of the exact executable name and path used for your Patcher as you will be putting it into values in your game shortly. **Image above for reference and recommended setup for all users.**
6. *Transparent Windows are only available on Windows Builds and do not function on Linux or Mac Builds. If you do not plan on using a Transparent window background, uncheck the box for Use Transparent Windows*
7. **!IMPORTANT!** *If you are using Unity 2017.3+, you MUST have your Patcher in a folder within your game build directory as seen above for Windows OS Builds and you must check the box for Patcher Not In Game Directory.*
8. Now that your patcher has been built, you can open up another copy of Unity and pull up your Game Project.

Game Update Integration

1. Open your Game in the Unity Editor and Import in just the Coffee Game Files folder from the Coffee Auto Patcher Package. If you imported the full asset into your game prior to reading this, feel free to delete anything other than the Coffee Games Files and User Guide Folders to avoid confusion.
2. Create a new scene or open the UpdateExample in the Coffee Game Files > Scenes folder and add it to your Build in Unity. Set it as Scene 0 to ensure it always starts up first. If you have a startup Scene already, add the UpdateOperator Prefab to the scene or add the UpdateOperator.cs script to any object.
3. The UpdateOperator requires you put in the full Game exe, Patcher Full Exe Path (whatever Patcher folder and name you choose as seen below.) and paste in the Server File List URL that was copied from the Deployment program or the URL to your CDN hosting the fileList.txt.



- The Check for updates checkbox will disable checking for updates entirely if unchecked.
- The check for Launcher Updates being unchecked will make the Game Client not check for Launcher Updates.
- The Build Operating System is to be set for the Operating System you are building for.
- If Show Debug Text is unchecked, the debug text window will auto hide on scene start.
- **IMPORTANT!** If moving from a pre-2017.3 build to post 2017.3 build where the Patcher's new location is within a folder in the game directory, you can check the box for Delete Old Patcher In Game Directory to

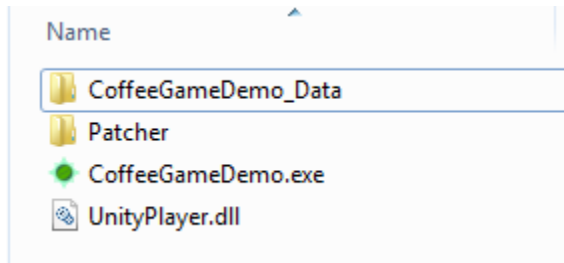
have it delete the older patcher exe. This is highly recommended to avoid issues with users getting confused and running the wrong patcher.

For Mac Builds – For Mac builds under the Game and Patcher Full Exe values you are going to input the file names located in <game>.app/Contents/MacOS/<game>

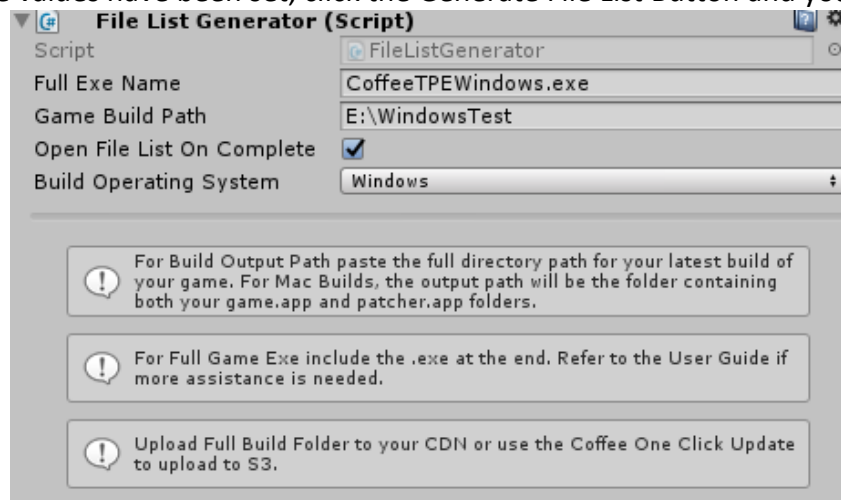
<patch>.app/Contents/MacOS/<patch>

The full name of your game should not have any extension on the end for Mac Builds. For Linux Users, input the name of your executable with the extension .x86 or .x86_64 depending if you are using a 32bit or 64bit build of your game and patcher.

4. Once these settings have been set build a copy of your game to the directory containing your Patcher folder as seen below.



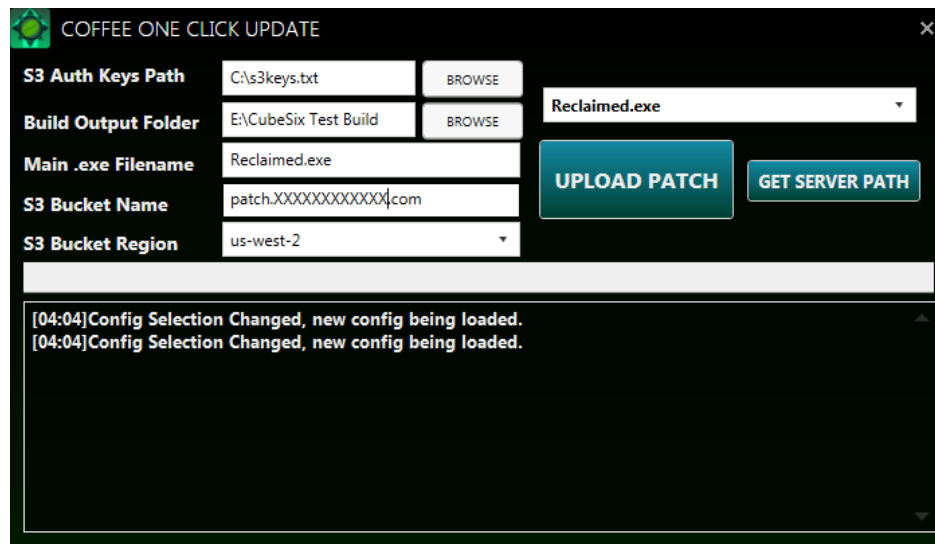
5. If you are using the Coffee One Click Update Program, Skip to Step 8. If you are not using Amazon S3 or choose not to use the Coffee One Click Deploy application, navigate to the Coffee Game Files > File List Generator Folder and select the FileListGenerator Prefab object.
6. Following the same instructions as above, input the Full Game Exe Name (this will match exactly the value that is inside of the UpdateOperator object.) Copy paste in the full path of the folder that contains all data from the builds of your Game and Patcher and ensure you select the Operating System for your Game Build.
7. Once the values have been set, click the Generate File List Button and you should see a



fileList.txt generated in the same folder as your game and patcher builds.

8. You should now have a folder containing both your Patcher and Game Files. If you are using the changelog feature, update your changelog.txt (or create a changelog.txt in the root folder containing your game exe file.) with your new version and changes in the build. I've included an example changelog in the Source Files folder.

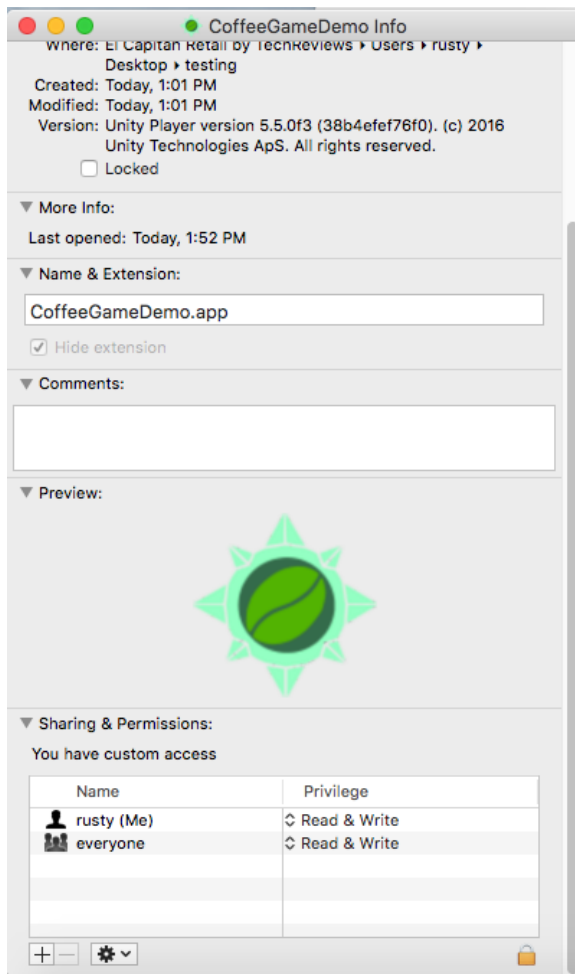
9. From here if you are using the Coffee One Click Deploy program, verify that all values are correct and click Upload Patch. If your settings are correct you should see it upload all of your game and patcher files to your Amazon S3 bucket. It will have to upload all of the files the first time, but after that when you make adjustments to your game it will just upload the files that have changed since the prior build.
10. For those that are not using Amazon S3 and/or not using the Coffee One Click Update.



The screenshot shows the 'COFFEE ONE CLICK UPDATE' application window. It features a dark-themed interface with a sidebar on the left containing a green cube icon and the title. The main area contains several input fields and buttons. The fields are: 'S3 Auth Keys Path' with 'C:\s3keys.txt' and a 'BROWSE' button; 'Build Output Folder' with 'E:\CubeSix Test Build' and a 'BROWSE' button; 'Main .exe Filename' with 'Reclaimed.exe'; 'S3 Bucket Name' with 'patch.XXXXXXXXXXXXX.com'; and 'S3 Bucket Region' with 'us-west-2'. To the right of these fields is a dropdown menu showing 'Reclaimed.exe'. Below the fields are two buttons: 'UPLOAD PATCH' and 'GET SERVER PATH'. At the bottom, there is a log window with a black background and white text showing two entries: '[04:04]Config Selection Changed, new config being loaded.'

Once you have generated a fileList.txt from within either the Patcher or your Game using the FileListGenerator, and updated your changelog if you are using it, copy all files from your Patcher and Game Build folders over to your FTP/CDN. The FileListGenerator will also generate a list of all files that need to be reuploaded to your FTP for you if you give it the URL of your fileList.

11. Your Game is now fully integrated with the Patcher and Update Scene. Get the new version of your game with the Patcher out to your users and from this point forward they will automatically be patched when you push out new builds for your game.
12. If you have additional Operating Systems to do builds for in your game, create new folders for each build Operating System and ensure you update the UpdateOperator and PatchOperator in both your Patcher and Game to show the correct file extension (none for Mac, .exe for Windows and .x86 for Linux) as well as selecting the proper operating system for the Build Operating System value. All other steps will be the same.
13. If you are building for Mac, when your users install the game and the patcher they must enable read/write privileges for everyone as the patcher and game will not be ran using their credentials when being auto started.



Thank you again. If you have any requests, questions, comments or suggestions for this asset or any other assets, please feel free to contact me on

Twitter @CoffeeReclaimed

Through the CoffeeBnS Discord at <https://discord.gg/9WYC2sQ> (Preferred and recommended method of support)

Email at support@coffeebns.com

Discord is the best location if you have any questions, comments or need any assistance setting up your package.