
GSP
Git Simple Packager
- 2020 -

Git Simple Packager

Take some stress off

George Murphy
Software Cloud Engineer

- Thanks to God
- Beautiful Family
- ColdFusion Engineer
- Federal Agency Contractor
- Fishing Specialist
- Infrastructure as Code

Git Simple Packager 2020

Definition of GSP

GSP is a simple system designed for clients who are not allowed to connect out to the internet or have any dot git folders on their production servers. No CI/CD allowed. How many of you fit that mold? How do you manage getting files to production without missing any and backing up what is being overwritten?

GSP

Git Simple Packager

Git simple packager is a system designed to get added and modified files into a package that can be easily deployed to your production servers. Once the files arrive there you can be assured that you will be able to create a package backup of all files that will be overwritten. It is written in GOLang and it is available for all OS's. It is an executable file that needs to be on your path..

How do you deploy?

There are multiple ways to deploy GSP.

Download the repository and create your own binary. https://github.com/murpg/gsp.git

I) Dependencies:

a) Git Bash choco install git.install -y

b) Make

c) GoLang choco install golang -y

Download the executables from Github.

II) Releases:

d) gsp_1.0.0_darwin_amd64.zip

f) gsp_1.0.0_freebsd_amd64.zip

g) gsp_1.0.0_linux_amd64.zip

h) gsp_1.0.0_SHA256SUMS

i) gsp_1.0.0_windows_amd64.zip

j) Source Code (zip)

K) Source Code (tar.gz)

Install Configuration Files

File number one:

The first file (.gsp-config.json) should be in the same directory as the gsp binary and the root of your home directory. You can remove the .sample from the filename (.gsp-config.sample.json) to rename it to .gsp-config.json. The executable file (gsp.exe) needs to be in a directory that is on the path of that operating system. My preferred choice for installing almost all programs is installing this via Chocolatey. If you installed via Chocolatey the location for that gsp.exe file would be c:/ProgramData/chocolatey/bin/gsp.exe.

Install Configuration Files

File number two:

If you are on Windows the location for the second file .gsp-config.json would be at C:/Users/$env:username variable. If you have more than one user needing to use GSP this gives each user his own configuration file if you place a file in the users home directory.

Git Simple Packager

How do I create release packages?

In order to create a release package, you should have some files you are ready to move to production in one of your repositories. Okay let's go to one of your repositories. For me, I will use the repository I set up for CFSummitDev to walk you through the process.

Git Simple Packager

Package Creation

Let's call the first site CFSummit Development which is running and available on my local machine. Okay now create a package that has the first 3 git commits. The first thing we want to do is setup our user Configuration file. Let's grab a base file from here.

GIT Simple Packager

Configuration Options

```
{
   "commitsCount": 0,
   "diffFilter": "AM",
   "directoryNames": [
   "handlers/",
   "views/"
   ],
   "gitHashNewest":
"77c471f31c4b717f6906e49599ea12ec7b2da875",
   "gitHashOldest":
"bb1948f2776cdbcc8544aede37ea0a726e5710a9",
   "outputPath": "path-to-store-releases",
   "repositoryPath": "path-to-repository"
}
```

```
{
   "commitsCount": 0,
   "diffFilter": "AM",
   "directoryNames": [
   ""
   ],
   "gitHashNewest":
"77c471f31c4b717f6906e49599ea12ec7b2da875",
   "gitHashOldest":
"bb1948f2776cdbcc8544aede37ea0a726e5710a9",
   "outputPath": "path-to-store-releases",
   "repositoryPath": "path-to-repository"
}
```

```
{
   "commitsCount": 4,
   "diffFilter": "AM",
   "directoryNames": [
   "handlers/",
   "views/"
   ],
   "gitHashNewest": "",
   "gitHashOldest": "",
   "outputPath": "path-to-store-releases",
   "repositoryPath": "path-to-repository"
}
```

Git Simple Packager

Releases

Preview Release

Let's create our first package. Just run this command inside of your repository.

>gsp -n

this is the command for a dry run. You will get an out that looks like this.

Git Simple Packager

Releases

Actual Release

Now to create your release run this.

>gsp

The output should look like the next slide

Git Simple Packager

Preview Package Release

>gsp -n

```
Release - November 16, 2020 12:19:11 EST
========================================
Changed files:
- views/stepThree/list.cfm
- views/stepTwo/index.cfm
- views/stepTwo/list.cfm
- handlers/stepThree.cfc
- views/stepThree/index.cfm
- views/stepOne/index.cfm
- views/stepOne/list.cfm
- handlers/stepOne.cfc
- handlers/stepTwo.cfc
=> Not creating a release archive as --dry-run.
```

Git Simple Packager

Actual Package Release

```
>gsp
```

```
Release - November 16, 2020 12:40:47 EST
========================================
Changed files:
- handlers/stepThree.cfc
- handlers/stepTwo.cfc
- views/stepOne/index.cfm
- views/stepTwo/list.cfm
- handlers/stepOne.cfc
- views/stepThree/index.cfm
- views/stepThree/list.cfm
- views/stepTwo/index.cfm
- views/stepOne/list.cfm
```

Move the release to production

Cloud Release

Now let’s move the release files and directory to the CFSummit Production web site. I will copy my release folder to this location (C:\Users\$env:username\Documents\releases).

Get Connected

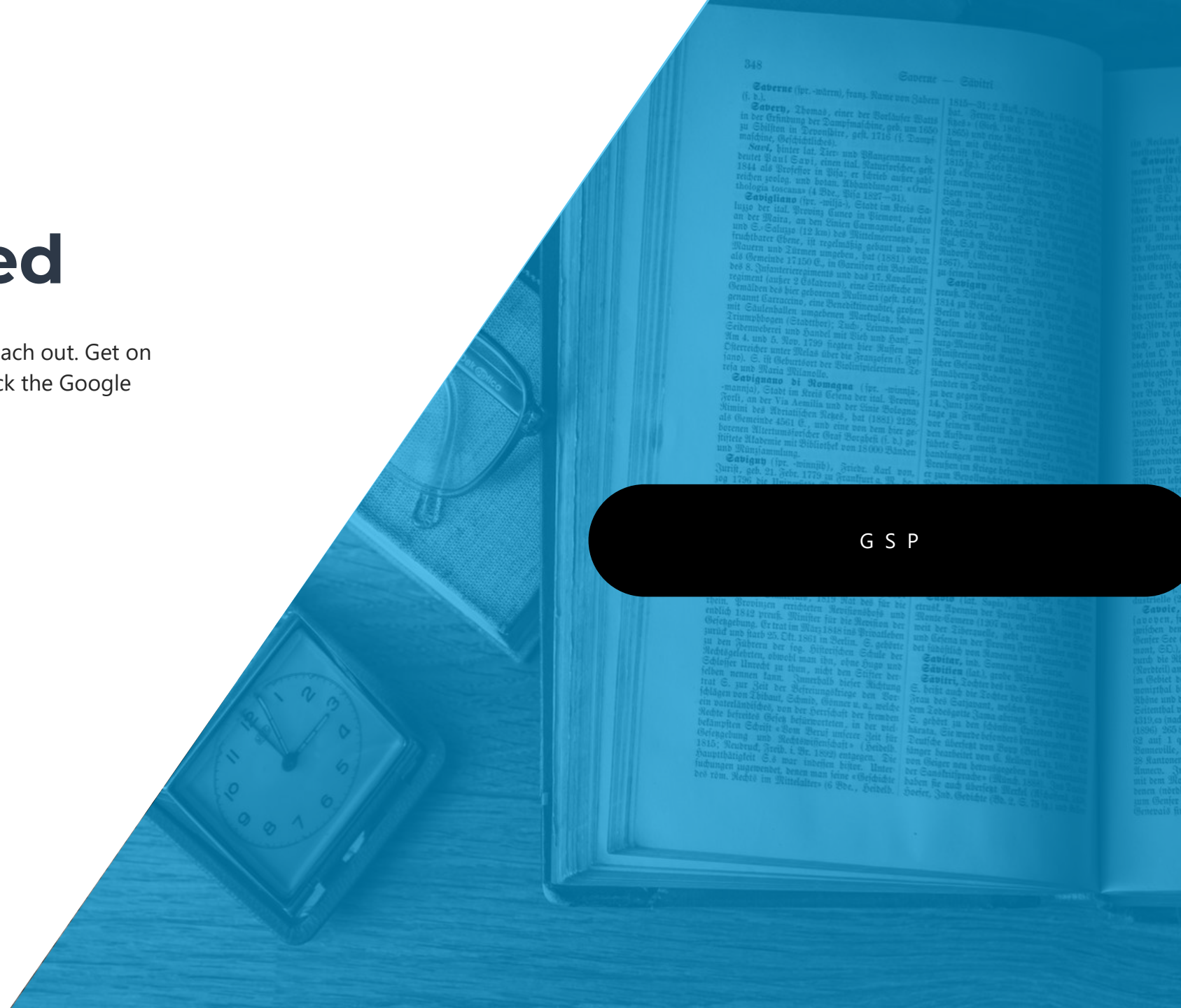
If you are interested in our project, please reach out. Get on the mailing list to receive **gsp-backup.exe**. Click the Google Forms Link below to receive notification.
Google Forms

703-478-6867

dbgdevops@websbygeorge.com

Northern Virginia

https://github.com/murpg/gsp



Thank You

For your attention & watching

- 2020 -