



git-scripts cheat sheet

adb-reset	Kill the adb.exe Process. Useful when developing mobile for Android and the debugger is not responding	origin <repo>	List remote tracking information for all branches
add <repo> <filemask>	Add all unmanaged files in the repo (normally you don't need this, as push does it automatically for you	prompt	Don't call this. It's a template script. Use install-scripts.cmd in the root of the repository to create a configured shortcut.
addsub <repo> <linkedrepo> [folder]	Add a sub module (linked repository)	pull <repo> [branch]	Pull the repo. Note: Use pull * to pull ALL repositories in the current folder!
branch <repo> [-l -d -D]	Create (and switch to) a new branch. Options: -l local branch -d delete branch (local) -D delete branch (local AND on remote)	purge-history <repo>	DANGER ZONE! Purge the entire history in the repo, local and on remote! Reduces the repository size to a minimum, but you only have the most current version available. No rollback!
clone	Don't call this, it's a template for the identity script. Use this to create your clone identities!	push <repo> "message"	Commit & Push to origin. Message is required. Note: Use push * to push ALL repositories in the current folder with the same message!
commit <repo> "message"	Perform local commit of changes. Message is required	resolve <repo>	Open the Tortoise-Git Resolve dialog if you face merge conflicts.
fetch <repo>	Perform fetch on the repo	revert <repo> [filemask]	Rollback your current changes
git-cache-reset	Resets the Tortoise-Git cache Process	sizeof <repo>	Determine/Estimate the real size of your repository (incl. history)
gms-delcache <project-name>	Delete all GameMaker caches from GMS , local and roaming AppData and from ProgramData folders for the specified project	status <repo>	Show status of your working copy (changed files, etc). NOTE: Perform status without any argument to show the status of ALL repositories in the current folder!
gms-rename <old-name> <new-name>	Renames a GameMaker project and also replaces the name in the .yyp and .yy files where needed	switch <repo> <branchname> [-m]	Switch to the specified branch. Use -m to merge your local changes into that branch.
identity	Create a new clone identity	tag <repo> <tag>	Set a tag on the repo, like for a new release.
list <repo>	List all known branches and tags of the repo	upsub <repo> [branch]	Update (pull) the linked submodules of the specified repo. Optionally update them from a specified branch
merge <repo> <from> <to> [-c -cp] [-2]	Merge <from> branch into <to> branch. Options: -c ...and commit or -cp ...and commit & push -2 ...after merge switch to second branch (<to>) By default, after merge, you are located in the first branch <from> after the merge	whereis <repo>	Shows you the current branch of the repo.