git_manager: deploy and back up from the admin

Source file: doc/eng-us/features/6.0/extensions/git_manager.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/features/6.0/extensions/git_manager.md

This page is for administrators of servers where nobody has a shell, or where a shell is not the way you want to work. `git_manager` gives two tools in the web admin:

> **Warning**
>
> `.htaccess` **slows down every request.** When `.htaccess` files are enabled (`AllowOverride` anything but `None`), let alone used, Apache looks for them and reads and evaluates them on every request, in every directory of the path, and page loading times go up drastically (measured). The only real fix is to switch the feature off for the virtual host: `AllowOverride None` in the HTTPS virtual host, and the rules in the virtual host's own configuration as mod_rewrite rules. That is far faster in production speed tests and well worth the time it takes to optimize a production configuration. Switch over right away: copy the rules of the example HTTPS virtual host into your virtual host (with the static cache, also `staticcache_offload.conf`), set `AllowOverride None`, run `apachectl configtest`, reload Apache and test the site. The same rules for other servers: nginx, LiteSpeed, Caddy, IIS, Velocity (built in); where virtual host rules go in a control panel: Plesk, cPanel, DirectAdmin, ISPConfig. **On LiteSpeed or OpenLiteSpeed,** the server's own page cache can also answer anonymous visitors and crawlers without PHP: Page caching with LiteSpeed and OpenLiteSpeed.

- a **git dashboard** (**Setup**, `/git_manager/dashboard`): see where the installation stands against its branch on the remote, switch branches, check out a commit, pull, push, manage remotes and submodules;
- a **backup manager** (`/git_manager/backup`): timestamped backups ("captions") of the database and files, optional encryption, and sanitised SQL dumps that are safe to share.

The extension was released as 2.0.1 on 21 June 2026 (backup manager) and reached 2.0.14 on 2 October 2026.

Requirements

Exponential 6, PHP 8.0 or later (PHP 7.4 usable without support), a git binary the web server's user can run, GnuPG (`/usr/bin/gpg`) for encrypted backups, and `mysqldump` for database backups.

Quick start: make a backup before a change

1. Open **Setup > Backup** (`/git_manager/backup`).
2. Choose **DB + Files caption**, add a description such as "Before upgrade", and click create.
3. The new caption appears under **Existing Captions**, with its archives to download.

The same from the command line:

Shell

```
php extension/git_manager/bin/php/backup-create.php full -d "Before upgrade"
php extension/git_manager/bin/php/backup-list.php -v -s
```

The dashboard

Upstream card (2.0.14)

At the top of the dashboard: the remote's address (without credentials), the branch it is compared with, the time of the last fetch, a badge (**up to date**, **behind**, **ahead**, **diverged**), the commits on the remote that the installation is missing and the local ones the other way (short hash, date, author name, subject, and a link to the commit on GitHub or GitLab), and the counts of modified, staged, untracked and conflicting files.

Fetch now runs `git fetch --prune` on the remote and nothing else: no pull, merge, reset or checkout. It never waits for a password, is stopped after `FetchTimeout` seconds and shows git's output when it fails. When the site's user cannot write the remote-tracking branches, the card says which files and whose they are instead of fetching. If the SSH address does not work for the site's user and the repository is on GitHub, the card fetches anonymously over HTTPS into the same branches (a private repository then still fails). Every fetch is recorded as the audit event `system.git_manager.fetch`. The status is cached in the cache directory and recomputed when HEAD, the branch, the remote-tracking branch, `FETCH_HEAD`, `packed-refs` or the index change.

Branches, commits, pull

Switch to a local branch or a remote branch, check out a commit by hash, pull the latest, update submodules on checkout, review the commit at HEAD and its diff, and browse the commit log filtered by author or date range. The log is a fold-out that stays the way each user leaves it (the preference `admin_git_manager_commit_log`). Its summary says how many of the latest commits no remote has. A commit no remote has is highlighted orange and marked **not pushed**; one only some remotes lack is blue with **not on <remote>**; the checked out commit is green. **Only commits to push** hides the rest.

Push (2.0.6)

Push to a remote lists every remote with its address (any user name, password or token in it is left out) and how the checked out branch stands against it: up to date, commits to push, commits behind, or not on that remote yet. **Push** sends a local branch to the branch of the same name (`git push --porcelain <remote> refs/heads/x:refs/heads/x`). It is **never forced**: a remote that has commits the branch lacks refuses it and the output says so. The page asks first, naming the branch, the remote, its address and how far ahead and behind the branch is. git never waits for anyone: no password prompt, SSH in batch mode with a 15 second connect timeout, stopped after 90 seconds.

Remotes (2.0.7)

Each remote has an **Edit** fold-out to change its name and address or remove it (asking first); **Add a remote** takes a name and an address. A name is letters, digits, dot, dash and underscore, not starting with a dash or dot. An address is an `https://`, `http://`, `ssh://`, `git://` or `file://` URL, `user@host:path`, `host:path` or an absolute path; no spaces, control

characters or leading dash, and no `::`, so git's `ext::` and other transport helpers cannot be used to run a command. An address the page shows without its credential and saves unchanged is left as it is, so editing a remote never drops a token.

Submodules (2.0.8)

The Submodules section lists each submodule (path, address without credentials, the branch it follows, its commit and state) with **Update**, and an **Edit** fold-out to change the address and branch or remove it. **Add a submodule** (folded by default, remembering its state in the preference `admin_git_manager_add_submodule`) clones into a new path. Adding, changing and removing stage the change for a commit.

Command line

Shell

```
php extension/git_manager/bin/php/remote.php --help      # list, add, set-url, rename, remove, fe
tch, push
php extension/git_manager/bin/php/submodule.php --help   # list, add, set-url, set-branch, update
, remove
```

The backup manager

Open **Setup > Backup** (`/git_manager/backup`; the address before 2.0.4, `/git_manager/dump`, redirects there).

Type	Creates	Use for
Full Site Backup	SQL dump, `var/` archive and site files (`extension/`, `settings/`, `config.php`): three archives	Rebuilding the site on a new server (core files come from git)
DB + Files caption	SQL dump and `var/` archive	The daily backup
Database only	SQL dump	Before database changes or migrations
Files only (`var/`)	`var/` archive	Before bulk file changes

Choose a type, optionally add a description, optionally tick **Encrypt backup files** and enter a passphrase (AES256 through GnuPG), optionally tick **AGPL Compatible Release**, and create. Download each archive from **Existing Captions**; delete old captions one by one or with **Select All + Delete Selected**. A prominent red box appears when there are no captions at all, an orange one when the newest is older than 7 days, and each caption has an age indicator (green under 7 days, orange 7 to 90 days, red older).

Sanitised dumps you can share

The AGPL option writes an extra `sql_agpl_<time>.tar.gz` next to the private dump, with a marker file `agpl_compatible.txt` and a `licenses/` folder (GPL v2 and AGPL v3 texts) in every caption. Sanitised: password hashes, e-mail addresses, analytics IDs, payment, mail and cloud API keys, captcha keys, named settings such as `Password`, `ApiKey`, `AuthToken`, `WebhookSecret`, `SMTPPassword`, absolute disk paths, database credentials, IP addresses and SMTP credentials. Since 2.0.8 it also removes every table with secrets or personal data: password-reset and account keys (`ezforgot_password`, `ezuser_accountkey`), `ezuservisit`, notification settings,

every `ezprest_` table (OAuth clients and tokens), `ezinfocollection` (what visitors sent in forms), `ezorder`, `ezbasket`, `ezproductcollection`, `ezwishlist`, `ezpaymentobject`, `ezcollab_` and `cjwnl_` (newsletter subscribers), together with session, audit and pending-action tables. The hash pattern is applied to `ezuser` rows only, so `remote_id`s survive and the dump imports.

Backups from the command line

Shell

```
php extension/git_manager/bin/php/backup-list.php -v -s   # -v lists the files of each backup, -s
 shows sizes
php extension/git_manager/bin/php/backup-create.php fullsite -d "Before upgrade"   # fullsite | f
ull | db | var
php extension/git_manager/bin/php/backup-create.php full -e --passphrase-file=/root/backup.pass -
d "Encrypted"
php extension/git_manager/bin/php/backup-info.php 2026-06-21_20-20-49
php extension/git_manager/bin/php/backup-delete.php 2026-06-21_20-20-49 -f
```

Pass the passphrase with `--passphrase-file`, the environment variable `GIT_MANAGER_BACKUP_PASSPHRASE`, or let a terminal ask for it without echo. `-p` still works but puts it in the process list and shell history, so it prints a warning.

Decrypt an encrypted file (`.tar.gz.gpg`):

Shell

```
gpg --batch --yes --passphrase "YOUR_PASSPHRASE" -d backup_file.tar.gz.gpg | tar -xzf -
```

There is no recovery for a lost passphrase.

Security of backups (2.0.8)

- Every file and directory a backup writes is created owner-only (umask 077, folders 0700); before they were world-readable and held the database and settings.
- `mysqldump` gets host, port, user and password from an option file readable by its owner only (`--defaults-extra-file`), removed after the dump; the password is never on a command line, and database name and output file are quoted.
- The backup folder has a deny-all `.htaccess` and an empty `index.html`.
- Downloads are served by an authenticated controller with `Cache-Control: private, no-store` and `X-Content-Type-Options: nosniff`. `VarExcludeDirs` leaves out `site/sessions`.
- The backup page escapes descriptions, timestamps, file names and messages.
- Branch names and commit hashes reach git only when valid and quoted; a hash must be 7 to 40 hexadecimal characters.

Settings

All in `git_manager.ini`, block names as shown.

Block	Key	Default	Meaning
GitManagerSettings	BackupPath	`var/site/backups/captions`	Where captions are stored

Block	Key	Default	Meaning
GitManagerSettings	MySQLDumpCommand	mysqldump --host={host} --port={port} ...	Dump command with placeholders
GitManagerSettings	VarExcludeDirs[]	cache, log, site/backups, site/sessions	Left out of the var/ archive
GitManagerSettings	MaxBackups	36	Oldest captions are removed beyond this; 0 is unlimited
GitManagerSettings	CompressBackups	enabled	gzip
GitManagerSettings	EnableEncryption	enabled	Offer GPG encryption
GitManagerSettings	DeleteUnencryptedAfterEncryption	enabled	Keep only the .gpg file
UpstreamSettings	Remote	origin	Remote the Upstream card compares with
UpstreamSettings	Branch	empty	Branch on that remote; empty uses the upstream of the checked out branch
UpstreamSettings	FetchTimeout	60	Seconds before a fetch is stopped
UpstreamSettings	HttpsFallback	enabled	Anonymous HTTPS fetch of a GitHub repository
UpstreamSettings	WebUrl	empty	Browser address for commit links; empty derives it from the remote
UpstreamSettings	MaxCommits	100	Commits listed each way
UpstreamSettings	StatusCacheTTL	60	Seconds the uncommitted-file counts are kept

Policies

Role policies in module `git_manager`:

Function	Allows
git_manager	The dashboard, commit details, fetching
backup	The backup page and downloads (called dump before 2.0.4; roles that grant the old name keep working)
push	Pushing a branch
remotes	Adding, changing and removing remotes
submodules	Adding, changing and removing submodules

Rename old role policies with:

Shell

```
php extension/git_manager/bin/php/upgrade-policy-dump-to-backup.php --dry-run
php extension/git_manager/bin/php/upgrade-policy-dump-to-backup.php
```

Languages

The extension carries translation files in `translations/<locale>/translation.ts`: eng-US and ger-DE. The German file holds 199 messages (count `<message` in `translations/ger-DE/translation.ts`). The texts are looked up in the contexts `extension/git_manager`, `kernel/navigationpart`, `design/admin/pagelayout` and `design/admin/parts/setup/menu`.

After editing a translation file, refresh the compiled translation cache with `./console exp:ezgeneratetranslationcache` and clear the template and content caches. `./console exp:ezchecktranslation ger-DE` prints statistics of the kernel's `share/translations/ger-DE/translation.ts`, not of this extension's file.

Related pages

- Audit trail: the `system.git_manager.fetch` event
- ezupdate: updates after a backup
- Extension metadata specification
- Chronicle and release notes
- Change ledger
- Behaviour changes of the extensions
- Months: 2026-06, 2026-09, 2026-10 (all extensions)

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.