

CMS Full Upgrade Tool Tutorial

How to update an old CMS to the latest version using upgrade-cms.php, latest-cms.zip, and database.sql.

Purpose: update CMS files and database structure while protecting local settings, server rules, images, videos, uploads, logos, and generated data.

Do not run this on a live money website without a backup. Old-school rule, still undefeated.

Safe CMS Upgrade Flow

1 Backup

Download website files and export the current database before touching anything.



2 Upload

Put upgrade-cms.php in root, then upload latest-cms.zip and database.sql.



3 Analyze

Click Analyze All. The tool checks files and database, but makes no changes.



4 Run Upgrade

If the plan looks clean, run the upgrade to update files and database structure.



5 Test Website

Check homepage, play page, admin, categories, tags, blogs, and search.

Golden rule: Analyze first, upgrade second, test third, delete the tool last.

1. What this tool does

Uploads or reads latest-cms.zip from the website root.

Uploads or reads database.sql from the website root.

Analyzes files first, showing which files will update and which protected files will be skipped.

Overwrites or adds CMS files from the ZIP when you run the real upgrade.

Updates database structure only: it creates missing tables and adds missing columns.

It does not import games, blogs, users, INSERT rows, or delete old website files. That is intentional because deleting automatically is risky for old CMS sites.

2. Files you need before starting

upgrade-cms.php - the upgrade tool placed in website root.

latest-cms.zip - the new/latest CMS files packed as a ZIP.

database.sql - the latest database structure backup. It should contain CREATE TABLE blocks.

A full backup of current website files and current database.

Example root location:

```
/public_html/upgrade-cms.php  
/public_html/latest-cms.zip  
/public_html/database.sql
```

If your hosting blocks big browser uploads, upload latest-cms.zip and database.sql with FTP/WinSCP instead. The tool will detect them in root.

3. Open the tool and upload files

Open this in browser after uploading the script:

```
https://yourdomain.com/upgrade-cms.php
```

Use the upload area to select the latest CMS ZIP and database SQL. Then click Upload Files.

CMS Full Upgrade Tool

Step 1 - Upload latest-cms.zip and database.sql

Upload latest CMS ZIP and database SQL

Latest CMS ZIP file

latest-cms.zip

Latest database SQL

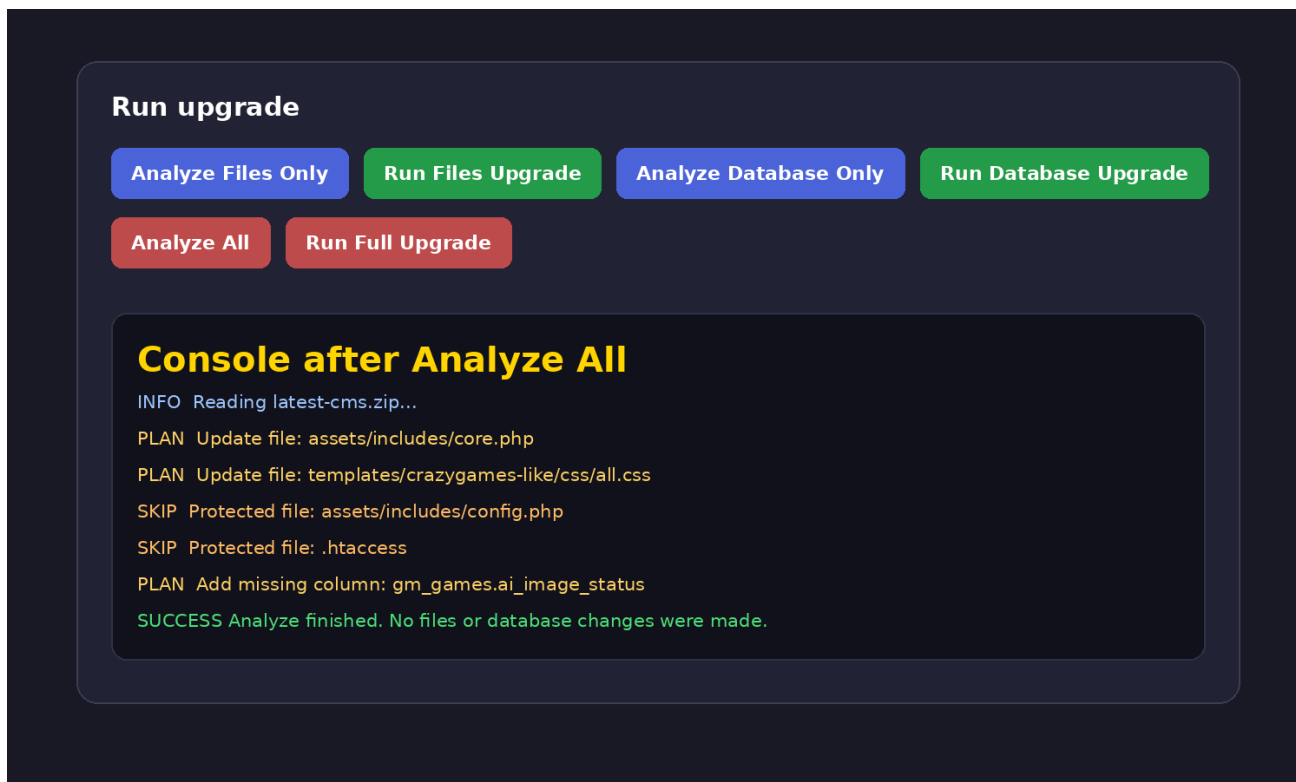
database.sql

Upload Files Files are saved in website root.

After upload, the page should show that latest-cms.zip and database.sql are found. If the ZIP button is disabled, the ZIP is missing or was not uploaded correctly.

4. Run Analyze first

Click Analyze All first. Analyze mode is dry-run mode. It reads the ZIP and SQL, but it does not change files and does not change the database.



Good analyze output should include planned file updates, skipped protected files, existing database tables, and missing database columns that will be added.

If Analyze shows strange paths like ../, wrong project files, or tries to update many custom folders you want to keep, stop. Fix the ZIP or add more protected folders before running upgrade.

5. Protected files and folders

The tool skips important local files and generated folders by default. This is the main safety part.

Files protected by default

These are skipped even if the ZIP contains them.

Important files

LOCK assets/includes/config.php
LOCK assets/conf.php
LOCK .htaccess
LOCK robots.txt / robot.txt
LOCK ads.txt
LOCK database.sql
LOCK latest-cms.zip
LOCK upgrade-cms.php

Important folders

LOCK games-image/
LOCK games-thumb-video/
LOCK blog-img/
LOCK uploads/
LOCK cache/ logs/ tmp/
LOCK json/
LOCK sitemap/
LOCK static/logo/

Most important protected items:

Protected item	Why it is protected
assets/includes/config.php	Keeps database login and local site config safe.
.htaccess	Keeps current rewrite rules and server redirects safe.
robots.txt / robot.txt	Keeps current indexing rules safe.
games-image/	Keeps generated game thumbnails.
games-thumb-video/	Keeps local walkthrough/hover videos.
blog-img/ and uploads/	Keeps uploaded blog/media files.
json/ and sitemap/	Keeps generated local data and sitemaps.
static/logo/	Keeps custom logo files.

You can edit the protected list inside the PHP script if you have another local folder that must never be touched.

6. Run the real upgrade

Only after Analyze looks clean, click Run Full Upgrade. The tool will ask for confirmation.

Run Files Upgrade updates only files from latest-cms.zip.

Run Database Upgrade updates only missing database structure from database.sql.

Run Full Upgrade does both files and database structure.

The file upgrade overwrites/adds files from the ZIP except protected files and folders. It does not delete old files that are not in the ZIP.

The database upgrade adds missing tables and columns only. It does not import rows, so it should not replace your games, users, blogs, categories, tags, or settings.

7. What to test after upgrade

After the upgrade finishes, test these pages before touching more websites:

- Homepage loads without PHP errors.
- Game play page loads and iframe works.
- Admin panel opens and saves settings.
- Add/edit game page works.
- Categories and tags pages open correctly.
- Blog page and blog post pages load.
- Search works.
- Sitemap and robots still open.
- Check browser console for broken CSS/JS paths.

Also check PHP error_log. If one website has old custom code, the browser may look fine but PHP warnings can reveal a problem early.

8. Common problems and fixes

Problem	Most likely cause	Fix
ZipArchive missing	Server PHP does not have zip extension enabled.	Enable php-zip in hosting/server or ask hosting support.
ZIP upload fails	File is too large for upload_max_filesize/post_max_size.	Upload ZIP with FTP/WinSCP directly to root.
Database not connected	config.php uses a different connection variable.	Edit getDatabaseConnection() in the script to match config.php.
Website breaks after file upgrade	Old local custom file was overwritten.	Restore from backup or add that path to protected list, then rerun.
Analyze shows wrong root folder	ZIP contains a wrapper folder or wrong project.	The script tries to strip one wrapper folder, but still verify paths before upgrade.

9. My recommended safe workflow for many websites

- Test on one non-critical website first.
- Make backup.
- Upload upgrade-cms.php, latest-cms.zip, and database.sql.
- Click Analyze All and read the console carefully.

Run Full Upgrade only if the plan looks correct.

Test the website.

Delete upgrade-cms.php, latest-cms.zip, and database.sql from root after finishing.

Then repeat on the next website.

Do not leave upgrade-cms.php public forever. Once finished, delete it. This type of tool is powerful and should not stay online.

Quick checklist

Done?	Task
[]	Backup website files.
[]	Backup current database.
[]	Upload upgrade-cms.php to root.
[]	Upload latest-cms.zip and database.sql.
[]	Open /upgrade-cms.php in browser.
[]	Click Upload Files if not already uploaded by FTP.
[]	Click Analyze All.
[]	Confirm protected files are skipped.
[]	Confirm planned updates look normal.
[]	Click Run Full Upgrade.
[]	Test homepage, play page, admin, categories, tags, blogs, search.
[]	Check PHP error_log.
[]	Delete upgrade-cms.php, latest-cms.zip, and database.sql from root.

End note: Analyze first, upgrade second, test third, delete tool last. Simple and safe.