

My experience installing and activating WordPress Core AI

From [Miriam Schwab](#)

AI plugin experience and issues

I installed the AI plugin on a site and tested out the functionality offered by it. Here are some of the issues I encountered:

- Category and tag suggestions work and can be useful, but the suggestions are often not relevant at all to the post.
- **Image generation requires a provider that supports it** (obviously). At first I had only connected an Anthropic API key to the site, but Claude is text-only. May be worth flagging this for users so they know to also add an API for an LLM that generates images.
- **Featured Image generation output is not great.** I asked it to create a Featured Image for a post, and it created something related, but terrible. I'm not sure what exactly the "prompt" is that is passed from the post to the AI call, but it's not effective. It could be better if the user could see the prompt first and tweak it before submission, or have the ability to tweak it after a version of the image is generated.
- **Title generation shows a button but doesn't work.** When you click into the title field in the post editor, an AI generation button appears. Clicking it doesn't do anything. It's unclear whether this is an incomplete feature or a bug, but it creates a confusing experience – users will see it and expect it to work.
- **Meta description generation doesn't persist after save.** The AI plugin generates a meta description and offers two options: copy to clipboard, or apply directly. Applying it works – the meta description appears in the sidebar. But after saving the post, it disappears. Checking the page source confirms it wasn't saved. The feature appears to work but the output is silently lost.
- **Image Generation in Media Library is better.** I tested the Image Generation capability in the Media Library, and that offered better results, probably because I was able to manually enter the prompt.
- **Saving AI-generated image doesn't complete.** After generating an AI image in the Media Library, I clicked Save to Library and it visibly starts a process which includes generating an alt-tag, but then once it completes that same window is still open and it seems like

nothing happened. It turns out it did save the image to the library each time I clicked the Save to Library button (so I ended up with three copies of it in the Library), but there was no confirmation or indication of completion.

A note about rate limits

I tried to test the Review Notes feature and it hit a 429 error immediately on my Claude API key (30,000 input tokens per minute). I'm thinking this can be managed better in one of the following ways:

1. I'm not sure what this action is doing, but it may not have to be so resource intensive. There may be room for some optimization here.
2. If one API key hits a rate limit, it would be useful if the AI offered to try one of the others.
3. This experience made me curious: why did it prioritize Claude over OpenAI from the start? How does it decide which LLM to leverage?

There is also a cost consideration worth factoring in before turning everything on. Once I saw how resource intensive this action was, it made me wonder how we can prevent LLMs from "running away" with opaque actions that can rack up token costs.

My experience connecting Claude to my site as an agent

A few things that aren't obvious when setting up the MCP connection:

Strip spaces from application passwords. When you generate your Application Password in WordPress for this (under Users), the password is displayed with spaces for readability (e.g. `xGnK JyJZ PPCa wnZ7`). If you copy and paste this as is into the WP_API_Password field above, the proxy doesn't normalize this and the spaced version fails silently. Users need to know to remove the spaces.

The core abilities aren't MCP-public by default. WordPress registered them, but they need a `meta.mcp.public = true` flag to be exposed via the MCP Adapter's default server:

```
add_filter( 'wp_register_ability_args', function( $args, $name ) {  
  
    $score = ['core/get-site-info', 'core/get-user-info', 'core/get-environment-info'];  
  
    if ( in_array( $name, $score, true ) ) {
```

```

    $args['meta']['mcp']['public'] = true;

}

return $args;

}, 10, 2 );

```

Building custom abilities

The core abilities and the three abilities the AI plugin provides are useful, but limited. For real content management work like listing posts by category, creating drafts, updating excerpts in bulk, managing tags, you need to register your own abilities, which I did.

Claude and I built a plugin that extends the default set. We started with the three core abilities which exist in WordPress but aren't MCP-public by default, and added a filter to expose them. Then we built get-posts (retrieve and filter posts by category, tag, status, and search), create-post (create drafts from conversation notes), and get-categories (list all categories with post counts).

The process of building these abilities exposed some issues related to the "hook sequence" (see below): the silent NULL failures, the category requirement, the core category restriction. None of it is documented clearly, and it took a lot of trial and error to figure out what was going on. The pattern looks straightforward but has a few silent failure modes that cost time.

- **The hook sequence matters and is not documented clearly.** Register categories on `wp_abilities_api_categories_init` and abilities on `wp_abilities_api_init`. Using `init` at any priority returns NULL silently – the registry isn't ready and there's no error.
- **You can't add to the `core` category.** Attempting `'category' => 'core'` returns NULL. Register your own category first.
- **The category field is required.** Omitting it causes the same silent NULL.

Here's the pattern that works:

```

add_action( 'wp_abilities_api_categories_init', function() {

    wp_register_ability_category( 'myplugin', [

        'label' => 'My Plugin',
    ] );
} );

```

```

        'description' => 'Abilities for my plugin.',

    });

});

add_action( 'wp_abilities_api_init', function() {

    wp_register_ability( 'myplugin/get-posts', [

        'label'      => 'Get Posts',

        'description' => 'Retrieve posts with optional filtering.',

        'category'   => 'myplugin',

        'input_schema' => [ /* ... */ ],

        'output_schema' => [ /* ... */ ],

        'permission_callback' => fn() => current_user_can( 'edit_posts' ),

        'execute_callback' => function( $input ) { /* ... */ },

        'meta' => [ 'mcp' => [ 'public' => true ] ],

    ]);

});

```

- Requiring explicit confirmation before implementing changes.** Claude wasn't asking for approval before implementing changes on the site, which was concerning. I wanted it to require explicit confirmation, and I think in general it should work that way to protect users. So Claude and I built a two-step pattern into my custom abilities plugin: a preview-post-update ability that stages proposed changes and returns a before/after comparison, and an apply-post-update ability that applies them, but only after a server-side check confirms at least 30 seconds have passed since the preview. Two AI tool calls in the same response turn complete in milliseconds; a real person reading a preview and typing "yes" always takes longer. The time gap is the safety check.

Site Health Info

wp-core

version: 7.0-RC2
site_language: en_US
user_language: en_US
timezone: +00:00
permalink: /%postname%/
https_status: true
multisite: false
user_registration: 0
blog_public: 1
default_comment_status: undefined
environment_type: staging
user_count: 4
dotorg_communication: true

wp-paths-sizes

wordpress_path:
/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content
wordpress_size: 2.13 GB (2290796316 bytes)
uploads_path:
/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content/
wp-content/uploads
uploads_size: 96.72 MB (101416797 bytes)
themes_path:
/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content/
wp-content/themes
themes_size: 2.02 MB (2122795 bytes)
plugins_path:
/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content/
wp-content/plugins
plugins_size: 183.53 MB (192449349 bytes)
fonts_path:
/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content/
wp-content/uploads/fonts
fonts_size: directory not found
database_size: 10.11 MB (10600448 bytes)
total_size: 2.42 GB (2597385705 bytes)

wp-dropins (1)

object-cache.php: true

wp-active-theme

name: Miriam Schwab (miriamschwab-theme)

version: 1.4.66

author: Miriam Schwab

author_website: <https://miriamschwab.me>

parent_theme: none

theme_features: core-block-patterns, widgets-block-editor, title-tag, post-thumbnails, html5, custom-logo, editor-styles, menus

theme_path:

/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content/
wp-content/themes/miriamschwab-theme

auto_update: Disabled

wp-themes-inactive (1)

Hello Elementor: version: 3.4.7, author: Elementor Team, Auto-updates enabled

wp-plugins-active (21)

Activity Log: version: 2.11.2, author: Activity Log Team, Auto-updates enabled

Advanced Custom Fields: version: 6.8.0, author: WP Engine, Auto-updates enabled

AI: version: 0.7.0, author: WordPress.org Contributors, Auto-updates disabled

AI Provider for Anthropic: version: 1.0.2, author: WordPress AI Team, Auto-updates disabled

AI Provider for Google: version: 1.0.3, author: WordPress AI Team, Auto-updates disabled

AI Provider for OpenAI: version: 1.0.2, author: WordPress AI Team, Auto-updates disabled

All-in-One WP Migration and Backup: version: 7.105, author: ServMask, Auto-updates disabled

Angie: version: 1.1.4, author: Elementor.com, Auto-updates enabled

Angie MCP for Yoast: version: 1.2.2, author: Miriam Schwab, Auto-updates disabled

Elementor: version: 4.0.2, author: Elementor.com, Auto-updates enabled

Elementor Pro: version: 4.0.2, author: Elementor.com, Auto-updates enabled

Instant Back/Forward: version: 1.3.1, author: Weston Ruter, WordPress Performance Team,
Auto-updates disabled

Manage: version: 1.0.3, author: Elementor.com, Auto-updates enabled

MCP Adapter: version: 0.4.1, author: WordPress.org Contributors, Auto-updates disabled

MS WordPress Abilities: version: 1.5.1, author: Miriam Schwab, Auto-updates disabled

Redis Object Cache: version: 2.7.0, author: Till Krüss, Auto-updates disabled

Site Kit by Google: version: 1.176.0, author: Google, Auto-updates enabled
Site Mailer - SMTP Replacement, Email API Deliverability & Email Log: version: 1.4.3, author: Elementor.com, Auto-updates enabled
WordPress Beta Tester: version: 4.0.0, author: WordPress Upgrade/Install Team, Auto-updates disabled
WP Rollback: version: 3.1.0, author: WP Rollback, Auto-updates enabled
Yoast SEO: version: 27.4, author: Team Yoast, Auto-updates enabled

wp-plugins-inactive (5)

Admin Menu Manager: version: 2.8.0, author: Custom, Auto-updates disabled
Ally - Web Accessibility & Usability: version: 4.1.0, author: Elementor.com, Auto-updates enabled
Image Optimizer - Compress, Resize and Optimize Images: version: 1.7.3, author: Elementor.com, Auto-updates enabled
Lighthouse Scanner: version: 2.2.0, author: Miriam Schwab, Auto-updates disabled
Markdown Content Negotiator for LLMs: version: 1.1.0, author: Serkan Algur, Auto-updates disabled

wp-media

image_editor: WP_Image_Editor_Imagick
imagemagick_module_version: 1809
imagemagick_version: ImageMagick 7.1.1-32 Q16-HDRI x86_64 22207 <https://imagemagick.org>
imagemagick_version: 3.7.0
file_uploads: 1
post_max_size: 256M
upload_max_filesize: 256M
max_effective_size: 256 MB
max_file_uploads: 20
imagemagick_limits:
 imagemagick::RESOURCE_TYPE_AREA: 472 GB
 imagemagick::RESOURCE_TYPE_DISK: 9.2233720368548E+18
 imagemagick::RESOURCE_TYPE_FILE: 786432
 imagemagick::RESOURCE_TYPE_MAP: 236 GB
 imagemagick::RESOURCE_TYPE_MEMORY: 118 GB
 imagemagick::RESOURCE_TYPE_THREAD: 1
 imagemagick::RESOURCE_TYPE_TIME: 0
imagemagick_file_formats: 3G2, 3GP, AAI, APNG, ART, ASHLAR, AVI, AVS, BAYER, BAYERA, BGR, BGRA, BGRO, BMP, BMP2, BMP3, BRF, CAL, CALS, CANVAS, CAPTION, CIN, CIP, CLIP, CMYK, CMYKA, CUBE, CUR, CUT, DATA, DCM, DCX, DDS, DFONT, DOT, DPX, DXT1, DXT5, EPS2, EPS3, EPT, EPT2, EPT3, FARBFELD, FAX, FF, FILE, FITS, FL32, FLV, FRACTAL, FTP, FTS, FTXT, G3, G4, GIF, GIF87, GRADIENT, GRAY, GRAYA, GV, HALD, HDR, HISTOGRAM, HRZ, HTM, HTML, HTTP, HTTPS,

ICB, ICO, ICON, INFO, INLINE, IPL, ISOBRL, ISOBRL6, JNG, JNX, JPE, JPEG, JPG, JPS, JSON, KERNEL, LABEL, M2V, M4V, MAC, MAP, MASK, MAT, MATTE, MIFF, MKV, MNG, MONO, MOV, MP4, MPC, MPEG, MPG, MPO, MSL, MTV, MVG, NULL, ORA, OTB, OTF, PAL, PALM, PAM, PATTERN, PBM, PCD, PCDS, PCL, PCT, PCX, PDB, PES, PFA, PFB, PFM, PGM, PGX, PHM, PICON, PICT, PIX, PJPEG, PLASMA, PNG, PNG00, PNG24, PNG32, PNG48, PNG64, PNG8, PNM, PPM, PS2, PS3, PSB, PSD, PWP, QOI, RADIAL-GRADIENT, RAS, RGB, RGB565, RGBA, RGBO, RGF, RLA, RLE, SCR, SCT, SFW, SGI, SHTML, SIX, SIXEL, SPARSE-COLOR, STEGANO, STRIMG, SUN, TEXT, TGA, THUMBNAIL, TILE, TIM, TM2, TTC, TTF, TXT, UBRL, UBRL6, UIL, UYVY, VDA, VICAR, VID, VIFF, VIPS, VST, WBMP, WEBM, WEBP, WMV, WPG, X, XBM, XC, XCF, XPM, XPS, XV, XWD, YAML, YCBCR, YCBCRA, YUV

image_format_transforms: image/heic → image/jpeg, image/heif → image/jpeg,

image/heic-sequence → image/jpeg, image/heif-sequence → image/jpeg

gd_version: bundled (2.1.0 compatible)

gd_formats: GIF, JPEG, PNG, WebP, BMP, AVIF, XPM

ghostscript_version: unknown

wp-server

server_architecture: Linux 6.6.113+ x86_64

httpd_software: nginx/1.29.3

php_version: 8.3.14 64bit

php_sapi: fpm-fcgi

max_input_variables: 10000

time_limit: 60

memory_limit: 1024M

max_input_time: 60

upload_max_filesize: 256M

php_post_max_size: 256M

curl_version: 8.14.1 OpenSSL/3.1.8

suhosin: false

imagick_availability: true

opcode_cache: not available

pretty_permalinks: true

static_robotstxt_file: true

current: 2026-04-15T15:18:47+00:00

utc-time: Wednesday, 15-Apr-26 15:18:47 UTC

server-time: 2026-04-15T15:18:46+00:00

wp-database

extension: mysqli

server_version: 8.0.43-google

client_version: mysqlnd 8.3.14
max_allowed_packet: 67109888
max_connections: 4030

wp-constants

WP_HOME: https://stg-3a6ivm.elementor.cloud
WP_SITEURL: https://stg-3a6ivm.elementor.cloud
WP_CONTENT_DIR:
/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content/
wp-content
WP_PLUGIN_DIR:
/mnt/customers/customers-bh-108dcf/76557035-bb42-4c22-b950-d0d7646b6bd6/wp-content/
wp-content/plugins
WP_MEMORY_LIMIT: 1024M
WP_MAX_MEMORY_LIMIT: 1024M
WP_DEBUG: false
WP_DEBUG_DISPLAY: true
WP_DEBUG_LOG: false
SCRIPT_DEBUG: false
WP_CACHE: false
CONCATENATE_SCRIPTS: undefined
COMPRESS_SCRIPTS: undefined
COMPRESS_CSS: undefined
WP_ENVIRONMENT_TYPE: staging
WP_DEVELOPMENT_MODE: undefined
DB_CHARSET: utf8
DB_COLLATE: undefined
EMPTY_TRASH_DAYS: 30

wp-filesystem

wordpress: writable
wp-content: writable
uploads: writable
plugins: writable
themes: writable
fonts: does not exist

acf

version: 6.8.0

plugin_type: Free
update_source: wordpress.org
ui_field_groups: 0
php_field_groups: 0
json_field_groups: 0
rest_field_groups: 0
post_types_enabled: true
ui_post_types: 0
json_post_types: 0
ui_taxonomies: 0
json_taxonomies: 0
rest_api_format: light
admin_ui_enabled: true
field_type-modal_enabled: true
field_settings_tabs_enabled: false
shortcode_enabled: false
registered_acf_forms: 0
json_save_paths: 1
json_load_paths: 1
ai_enabled: false
schema_support: false
schema_ready_objects:
 blocks: 0
 post_types: 0

google-site-kit

version: 1.176.0
php_version: 8.3.14
wp_version: 7.0-RC2
reference_url: https://stg-3a6ivm.elementor.cloud
amp_mode: no
site_status: not-connected
user_status: not authenticated
verification_status: not-verified
connected_user_count: none
active_modules: site-verification, search-console, analytics-4, pagespeed-insights
recoverable_modules: none
required_scopes:
 openid: 
 https://www.googleapis.com/auth/userinfo.profile: 
 https://www.googleapis.com/auth/userinfo.email: 

https://www.googleapis.com/auth/siteverification: 
https://www.googleapis.com/auth/webmasters: 
https://www.googleapis.com/auth/analytics.readonly: 
https://www.googleapis.com/auth/tagmanager.readonly: 

capabilities:

googlesitekit_authenticate: 
googlesitekit_setup: 
googlesitekit_view_posts_insights: 
googlesitekit_view_dashboard: 
googlesitekit_manage_options: 
googlesitekit_update_plugins: 
googlesitekit_view_splash: 
googlesitekit_view_authenticated_dashboard: 
googlesitekit_view_wp_dashboard_widget: 
googlesitekit_view_admin_bar_menu: 
googlesitekit_view_shared_dashboard: 
googlesitekit_read_shared_module_data::["search-console"]: 
googlesitekit_read_shared_module_data::["analytics-4"]: 
googlesitekit_read_shared_module_data::["pagespeed-insights"]: 
googlesitekit_manage_module_sharing_options::["search-console"]: 
googlesitekit_manage_module_sharing_options::["analytics-4"]: 
googlesitekit_manage_module_sharing_options::["pagespeed-insights"]: 
googlesitekit_delegate_module_sharing_management::["search-console"]: 
googlesitekit_delegate_module_sharing_management::["analytics-4"]: 
googlesitekit_delegate_module_sharing_management::["pagespeed-insights"]: 

enabled_features:

googleTagGateway: 
gtagUserData: 
proactiveUserEngagement: 
setupFlowRefresh: 

active_conversion_event_providers: none

consent_mode: disabled

consent_api: not-detected

search-console_shared_roles: none

search-console_management: owner

analytics-4_shared_roles: none

analytics-4_management: owner

pagespeed-insights_shared_roles: none

pagespeed-insights_management: all_admins

key_metrics_status: Setup and Enabled

key_metrics_source: Tailored Metrics

search_console_property: https://stg-3a6ivm.elementor.cloud/

analytics_4_account_id: 3878.....
analytics_4_property_id: 5287793..
analytics_4_web_data_stream_id: 1391.....
analytics_4_measurement_id: G-C3.....
analytics_4_use_snippet: yes
analytics_4_available_custom_dimensions: googlesitekit_post_type, googlesitekit_post_author
analytics_4_ads_linked: false
analytics_4_ads_linked_last_synced_at: 1775764049
analytics_4_site_kit_audiences: New visitors, Returning visitors

,