# WordPress.org Plugin Compliance Checklist

**Version:** 2.0  
**Last Updated:** October 4, 2025  
**Purpose:** A comprehensive checklist for developers and AI agents to ensure 100% WordPress.org plugin compliance

This checklist consolidates requirements from the official WordPress.org Plugin Directory Guidelines, the WordPress Plugin Check tool, PHPCS rulesets, and community best practices. Following these requirements will significantly increase the chances of first-time approval.

## Table of Contents

1. [File Structure & Initial Setup](#1-file-structure--initial-setup)
2. [Plugin Header & Metadata](#2-plugin-header--metadata)
3. [Security: Input Sanitization](#3-security-input-sanitization)
4. [Security: Output Escaping](#4-security-output-escaping)
5. [Security: Nonces & Capabilities](#5-security-nonces--capabilities)
6. [Database Queries](#6-database-queries)
7. [WordPress APIs & Libraries](#7-wordpress-apis--libraries)
8. [Scripts & Styles (Asset Enqueuing)](#8-scripts--styles-asset-enqueuing)
9. [Internationalization (i18n)](#9-internationalization-i18n)
10. [PHP Coding Standards](#10-php-coding-standards)
11. [Forbidden & Discouraged Functions](#11-forbidden--discouraged-functions)
12. [Performance Optimization](#12-performance-optimization)
13. [WordPress.org Guidelines (18 Rules)](#13-wordpressorg-guidelines-18-rules)
14. [readme.txt Requirements](#14-readmetxt-requirements)
15. [SVN & Submission Process](#15-svn--submission-process)
16. [Common Rejection Reasons](#16-common-rejection-reasons)
17. [Plugin Check Tool Requirements](#17-plugin-check-tool-requirements)

## 1. File Structure & Initial Setup

### 1.1 Main Plugin File

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Single main PHP file** | Must be located in its own folder (e.g., /my-plugin/my-plugin.php) |
| [ ] | **Standard plugin header** | Must contain at minimum: Plugin Name, Version, License, License URI, Text Domain |
| [ ] | **Unique plugin name** | Must not infringe trademarks; cannot start with "WordPress" unless authorized |
| [ ] | **GPL-compatible license** | Strongly recommended: GPLv2 or later |
| [ ] | **Text Domain matches slug** | Text Domain must exactly match the plugin folder/slug name |
| [ ] | **Requires at least** | Specify minimum WordPress version (e.g., 5.0) |
| [ ] | **Requires PHP** | Specify minimum PHP version (e.g., 7.2) |

**Example Header:**

/\*\*

\* Plugin Name: My Great Plugin

\* Description: A brief description of what my plugin does.

\* Version: 1.0.0

\* Author: Your Name

\* License: GPLv2 or later

\* License URI: https://www.gnu.org/licenses/gpl-2.0.html

\* Text Domain: my-great-plugin

\* Requires at least: 5.0

\* Requires PHP: 7.2

\*/

### 1.2 Unique Naming & Prefixes

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Unique function names** | Prefix all functions with a unique identifier (3+ characters, e.g., mgp\_ for "My Great Plugin") |
| [ ] | **Unique class names** | Prefix all classes (e.g., class MGP\_Admin) |
| [ ] | **Unique constants** | Prefix all constants (e.g., MGP\_VERSION) |
| [ ] | **Unique global variables** | Prefix all globals (e.g., $mgp\_settings) |
| [ ] | **Avoid reserved prefixes** | Do NOT use wp\_, \_, \_\_ (reserved for WordPress core) |
| [ ] | **No function\_exists() workaround** | Use truly unique names instead of wrapping in if (function\_exists()) |

### 1.3 File Organization

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Logical folder structure** | Use sub-folders: /includes, /assets, /templates, /languages |
| [ ] | **Dynamic path references** | Use plugin\_dir\_path(\_\_FILE\_\_) for file paths |
| [ ] | **Dynamic URL references** | Use plugin\_dir\_url(\_\_FILE\_\_) or plugins\_url() for URLs |
| [ ] | **No hardcoded paths** | Never hardcode /wp-content/plugins/ paths |
| [ ] | **No hidden files** | Remove .git, .svn, .DS\_Store, etc. before submission |
| [ ] | **No compressed files** | Remove .zip, .tar.gz files |
| [ ] | **No VCS directories** | Remove .git/, .svn/ directories |
| [ ] | **No dev files** | Remove node\_modules/, composer.json, package.json (or document them) |

### 1.4 Security Guards

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **ABSPATH check** | Add to ALL PHP files: if ( ! defined( 'ABSPATH' ) ) { exit; } |
| [ ] | **No direct file access** | Prevent direct URL access to plugin PHP files |
| [ ] | **No wp-load.php calls** | Never include wp-load.php or wp-config.php |
| [ ] | **No core file includes** | Use WordPress hooks instead of manually bootstrapping |

### 1.5 Activation/Deactivation/Uninstall

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Activation hook** | Use register\_activation\_hook(\_\_FILE\_\_, 'callback') for setup tasks |
| [ ] | **Deactivation hook** | Use register\_deactivation\_hook(\_\_FILE\_\_, 'callback') for cleanup |
| [ ] | **Uninstall cleanup** | Implement uninstall.php or register\_uninstall\_hook() to remove data |
| [ ] | **Capability checks** | Check current\_user\_can() in activation/deactivation callbacks |
| [ ] | **Flush rewrite rules** | If adding custom post types/taxonomies, flush rules on activation |

## 2. Plugin Header & Metadata

### 2.1 Required Header Fields

| ✓ | Field | Requirement |
| --- | --- | --- |
| [ ] | **Plugin Name** | Required. Must be unique and not infringe trademarks |
| [ ] | **Version** | Required. Must match readme.txt Stable tag |
| [ ] | **License** | Required. Must be GPLv2 or later (or GPL-compatible) |
| [ ] | **License URI** | Required. Must link to license text |
| [ ] | **Text Domain** | Required. Must exactly match plugin slug |
| [ ] | **Description** | Recommended. Brief description of functionality |
| [ ] | **Author** | Recommended. Your name or company |
| [ ] | **Author URI** | Optional. Link to your website |
| [ ] | **Requires at least** | Recommended. Minimum WordPress version |
| [ ] | **Requires PHP** | Recommended. Minimum PHP version |
| [ ] | **Domain Path** | Optional. If translations in subfolder (e.g., /languages) |

### 2.2 Version Consistency

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Header version matches readme** | Version in main file header must match Stable tag in readme.txt |
| [ ] | **Increment for each release** | Version must increase with each new release (e.g., 1.0.0 → 1.0.1) |
| [ ] | **Semantic versioning** | Follow MAJOR.MINOR.PATCH format |

## 3. Security: Input Sanitization

**Rule:** Sanitize ALL input data as soon as it's received.

### 3.1 Sanitization Functions

| ✓ | Input Type | Function to Use |
| --- | --- | --- |
| [ ] | **Text fields** | sanitize\_text\_field() |
| [ ] | **Textarea** | sanitize\_textarea\_field() |
| [ ] | **Email** | sanitize\_email() |
| [ ] | **URL** | esc\_url\_raw() |
| [ ] | **File name** | sanitize\_file\_name() |
| [ ] | **HTML class** | sanitize\_html\_class() |
| [ ] | **Key (slug)** | sanitize\_key() |
| [ ] | **Title** | sanitize\_title() |
| [ ] | **Integer** | intval() or absint() |
| [ ] | **Float** | floatval() |
| [ ] | **Boolean** | (bool) or rest\_sanitize\_boolean() |
| [ ] | **Array** | array\_map() with appropriate sanitization function |
| [ ] | **HTML content** | wp\_kses() or wp\_kses\_post() |

### 3.2 Input Sources to Sanitize

| ✓ | Source | Example |
| --- | --- | --- |
| [ ] | **$\_POST** | $name = sanitize\_text\_field( $\_POST['name'] ); |
| [ ] | **$\_GET** | $id = absint( $\_GET['id'] ); |
| [ ] | **$\_REQUEST** | $action = sanitize\_key( $\_REQUEST['action'] ); |
| [ ] | **$\_COOKIE** | $value = sanitize\_text\_field( $\_COOKIE['key'] ); |
| [ ] | **$\_SERVER** | $ip = sanitize\_text\_field( $\_SERVER['REMOTE\_ADDR'] ); |
| [ ] | **User input** | Any data from forms, AJAX, REST API |

### 3.3 Validation

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Validate data types** | Check if integer is actually integer, email is valid, etc. |
| [ ] | **Check expected values** | For dropdowns/selects, verify value is in allowed list |
| [ ] | **Reject invalid data** | Return error if data doesn't meet requirements |

**Example:**

// Sanitize and validate

$user\_id = isset( $\_POST['user\_id'] ) ? absint( $\_POST['user\_id'] ) : 0;

if ( $user\_id <= 0 ) {

wp\_die( 'Invalid user ID' );

}

## 4. Security: Output Escaping

**Rule:** Escape ALL output before sending to the browser.

### 4.1 Escaping Functions

| ✓ | Output Context | Function to Use |
| --- | --- | --- |
| [ ] | **HTML content** | esc\_html() |
| [ ] | **HTML attributes** | esc\_attr() |
| [ ] | **URLs** | esc\_url() |
| [ ] | **JavaScript** | esc\_js() |
| [ ] | **Textarea** | esc\_textarea() |
| [ ] | **SQL** | esc\_sql() (use $wpdb->prepare() instead) |
| [ ] | **HTML with allowed tags** | wp\_kses() or wp\_kses\_post() |

### 4.2 Translation + Escaping (Combined Functions)

| ✓ | Function | Use Case |
| --- | --- | --- |
| [ ] | **esc\_html\_\_()** | Translate and escape for HTML content (returns string) |
| [ ] | **esc\_html\_e()** | Translate and escape for HTML content (echoes) |
| [ ] | **esc\_attr\_\_()** | Translate and escape for HTML attributes (returns string) |
| [ ] | **esc\_attr\_e()** | Translate and escape for HTML attributes (echoes) |
| [ ] | **esc\_html\_x()** | Translate with context and escape for HTML |
| [ ] | **esc\_attr\_x()** | Translate with context and escape for attributes |

### 4.3 Output Locations to Escape

| ✓ | Location | Example |
| --- | --- | --- |
| [ ] | **Echo statements** | echo esc\_html( $variable ); |
| [ ] | **HTML attributes** | <input value="<?php echo esc\_attr( $value ); ?>"> |
| [ ] | **URLs** | <a href="<?php echo esc\_url( $link ); ?>"> |
| [ ] | **Admin notices** | echo '<div class="notice"><p>' . esc\_html( $message ) . '</p></div>'; |
| [ ] | **JavaScript variables** | var name = '<?php echo esc\_js( $name ); ?>'; |

**Example:**

// Correct: Escape on output

$title = get\_option( 'my\_plugin\_title' );

echo '<h1>' . esc\_html( $title ) . '</h1>';

// Correct: Escape in attributes

$url = get\_option( 'my\_plugin\_url' );

echo '<a href="' . esc\_url( $url ) . '">Link</a>';

## 5. Security: Nonces & Capabilities

### 5.1 Nonces (Verify User Intent)

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Create nonce in forms** | Use wp\_nonce\_field( 'action', 'nonce\_name' ) |
| [ ] | **Verify nonce on submission** | Use check\_admin\_referer( 'action', 'nonce\_name' ) |
| [ ] | **AJAX nonce creation** | Use wp\_create\_nonce( 'action' ) |
| [ ] | **AJAX nonce verification** | Use check\_ajax\_referer( 'action', 'nonce\_name' ) |
| [ ] | **URL nonce creation** | Use wp\_nonce\_url( $url, 'action' ) |
| [ ] | **URL nonce verification** | Use wp\_verify\_nonce( $\_GET['\_wpnonce'], 'action' ) |

**Example:**

// Form with nonce

<form method="post">

<?php wp\_nonce\_field( 'my\_action', 'my\_nonce' ); ?>

<input type="text" name="my\_field">

<input type="submit">

</form>

// Verify nonce on submission

if ( isset( $\_POST['my\_nonce'] ) && wp\_verify\_nonce( $\_POST['my\_nonce'], 'my\_action' ) ) {

// Process form

}

### 5.2 User Capabilities

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Check capabilities** | Use current\_user\_can( 'capability' ) before any action |
| [ ] | **Admin actions** | Require manage\_options or appropriate capability |
| [ ] | **Content editing** | Require edit\_posts, edit\_pages, etc. |
| [ ] | **File uploads** | Require upload\_files |
| [ ] | **Plugin management** | Require activate\_plugins |

**Common Capabilities:**

* manage\_options - Administrator
* edit\_posts - Editor, Author, Contributor
* edit\_pages - Editor
* publish\_posts - Editor, Author
* upload\_files - Editor, Author
* read - Subscriber

**Example:**

// Check capability before processing

if ( ! current\_user\_can( 'manage\_options' ) ) {

wp\_die( 'You do not have sufficient permissions.' );

}

### 5.3 Combined Nonce + Capability Check

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Always use both** | Check nonce AND capability for all actions |
| [ ] | **Check before processing** | Verify before database writes, file operations, etc. |

**Example:**

// Complete security check

if ( ! isset( $\_POST['my\_nonce'] ) || ! wp\_verify\_nonce( $\_POST['my\_nonce'], 'my\_action' ) ) {

wp\_die( 'Security check failed' );

}

if ( ! current\_user\_can( 'manage\_options' ) ) {

wp\_die( 'Insufficient permissions' );

}

// Now safe to process

$value = sanitize\_text\_field( $\_POST['my\_field'] );

update\_option( 'my\_option', $value );

## 6. Database Queries

### 6.1 Using $wpdb->prepare()

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Always use prepare()** | For ALL queries with variable data |
| [ ] | **Use placeholders** | %s for strings, %d for integers, %f for floats |
| [ ] | **No direct concatenation** | Never concatenate variables into SQL strings |
| [ ] | **Prepare before execute** | Call prepare() before query(), get\_results(), etc. |

**Example:**

// WRONG: Direct concatenation (SQL injection vulnerability)

$user\_id = $\_GET['user\_id'];

$results = $wpdb->get\_results( "SELECT \* FROM {$wpdb->prefix}my\_table WHERE user\_id = $user\_id" );

// CORRECT: Using prepare()

$user\_id = absint( $\_GET['user\_id'] );

$results = $wpdb->get\_results(

$wpdb->prepare(

"SELECT \* FROM {$wpdb->prefix}my\_table WHERE user\_id = %d",

$user\_id

)

);

### 6.2 Database Operations

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use WordPress APIs first** | Prefer get\_posts(), WP\_Query, get\_option(), etc. |
| [ ] | **Avoid direct queries** | Only use $wpdb when necessary |
| [ ] | **Use $wpdb methods** | get\_results(), get\_row(), get\_var(), insert(), update(), delete() |
| [ ] | **Table prefix** | Always use $wpdb->prefix or $wpdb->posts, $wpdb->users, etc. |
| [ ] | **Error handling** | Check for errors with $wpdb->last\_error |

### 6.3 Restricted Database Functions

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No PHP database extensions** | Don't use mysqli\_\*, PDO, pg\_\*, etc. |
| [ ] | **Use WordPress abstraction** | Always use $wpdb class |
| [ ] | **No direct MySQL** | Don't connect to MySQL directly |

## 7. WordPress APIs & Libraries

### 7.1 Use WordPress Functions

| ✓ | Instead of... | Use WordPress Function |
| --- | --- | --- |
| [ ] | file\_get\_contents() (remote) | wp\_remote\_get() |
| [ ] | curl\_\*() | wp\_remote\_get(), wp\_remote\_post() |
| [ ] | fopen() (remote) | wp\_remote\_get() |
| [ ] | mail() | wp\_mail() |
| [ ] | $\_COOKIE (direct) | wp\_set\_auth\_cookie(), wp\_clear\_auth\_cookie() |
| [ ] | header() (redirect) | wp\_redirect() or wp\_safe\_redirect() |
| [ ] | json\_encode() | Acceptable (not required to use WP function) |
| [ ] | file\_get\_contents() (local) | Acceptable for local files |
| [ ] | file\_put\_contents() (local) | Acceptable for local files |

### 7.2 Use WordPress Libraries

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **jQuery** | Use WordPress's bundled jQuery (jquery handle) |
| [ ] | **jQuery UI** | Use WordPress's bundled jQuery UI components |
| [ ] | **Underscore.js** | Use WordPress's bundled Underscore (underscore handle) |
| [ ] | **Backbone.js** | Use WordPress's bundled Backbone (backbone handle) |
| [ ] | **Moment.js** | Use WordPress's bundled Moment (moment handle) |
| [ ] | **React** | Use WordPress's bundled React (if available) |
| [ ] | **No duplicate libraries** | Never include your own copy of these libraries |

### 7.3 HTTP API

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **wp\_remote\_get()** | For GET requests |
| [ ] | **wp\_remote\_post()** | For POST requests |
| [ ] | **wp\_remote\_request()** | For custom requests |
| [ ] | **Check for errors** | Use is\_wp\_error() |
| [ ] | **Set timeout** | Use timeout parameter (default 5 seconds) |
| [ ] | **Handle response** | Use wp\_remote\_retrieve\_body(), wp\_remote\_retrieve\_response\_code() |

**Example:**

$response = wp\_remote\_get( 'https://api.example.com/data', array(

'timeout' => 10,

) );

if ( is\_wp\_error( $response ) ) {

// Handle error

$error\_message = $response->get\_error\_message();

} else {

$body = wp\_remote\_retrieve\_body( $response );

$data = json\_decode( $body );

}

## 8. Scripts & Styles (Asset Enqueuing)

### 8.1 Enqueuing Scripts

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use wp\_enqueue\_script()** | Never hardcode <script> tags |
| [ ] | **Register first (optional)** | Use wp\_register\_script() if needed |
| [ ] | **Declare dependencies** | Specify dependencies array (e.g., array('jquery')) |
| [ ] | **Version number** | Use plugin version or file modification time |
| [ ] | **Load in footer** | Set $in\_footer parameter to true |
| [ ] | **Conditional loading** | Only enqueue on pages where needed |
| [ ] | **Use handles** | Don't duplicate WordPress core script handles |

**Example:**

function my\_plugin\_enqueue\_scripts() {

// Only on plugin's admin page

if ( isset( $\_GET['page'] ) && $\_GET['page'] === 'my-plugin' ) {

wp\_enqueue\_script(

'my-plugin-admin', // Handle

plugins\_url( 'assets/js/admin.js', \_\_FILE\_\_ ), // Source

array( 'jquery' ), // Dependencies

'1.0.0', // Version

true // In footer

);

}

}

add\_action( 'admin\_enqueue\_scripts', 'my\_plugin\_enqueue\_scripts' );

### 8.2 Enqueuing Styles

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use wp\_enqueue\_style()** | Never hardcode <link> tags |
| [ ] | **Register first (optional)** | Use wp\_register\_style() if needed |
| [ ] | **Declare dependencies** | Specify dependencies array if needed |
| [ ] | **Version number** | Use plugin version or file modification time |
| [ ] | **Media type** | Specify media type (e.g., all, screen, print) |
| [ ] | **Conditional loading** | Only enqueue on pages where needed |

**Example:**

function my\_plugin\_enqueue\_styles() {

wp\_enqueue\_style(

'my-plugin-style',

plugins\_url( 'assets/css/style.css', \_\_FILE\_\_ ),

array(),

'1.0.0',

'all'

);

}

add\_action( 'wp\_enqueue\_scripts', 'my\_plugin\_enqueue\_styles' );

### 8.3 Inline Scripts & Styles

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use wp\_add\_inline\_script()** | For small inline scripts tied to enqueued handle |
| [ ] | **Use wp\_add\_inline\_style()** | For small inline styles tied to enqueued handle |
| [ ] | **Localize scripts** | Use wp\_localize\_script() to pass PHP data to JavaScript |

**Example:**

// Enqueue script

wp\_enqueue\_script( 'my-plugin-script', plugins\_url( 'js/script.js', \_\_FILE\_\_ ), array(), '1.0.0', true );

// Add inline script

wp\_add\_inline\_script( 'my-plugin-script', 'console.log("Plugin loaded");' );

// Localize script (pass data to JS)

wp\_localize\_script( 'my-plugin-script', 'myPluginData', array(

'ajax\_url' => admin\_url( 'admin-ajax.php' ),

'nonce' => wp\_create\_nonce( 'my\_action' ),

) );

### 8.4 Asset Loading Strategy

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Set loading strategy** | Use defer or async for non-critical scripts |
| [ ] | **Use wp\_script\_add\_data()** | Set strategy: wp\_script\_add\_data( 'handle', 'strategy', 'defer' ) |
| [ ] | **Load in footer** | Prefer footer loading for better performance |

## 9. Internationalization (i18n)

### 9.1 Text Domain

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Text Domain in header** | Must be present and match plugin slug exactly |
| [ ] | **Domain Path (if needed)** | If translations in subfolder: Domain Path: /languages |
| [ ] | **Load text domain** | Use load\_plugin\_textdomain() |

**Example:**

function my\_plugin\_load\_textdomain() {

load\_plugin\_textdomain( 'my-great-plugin', false, dirname( plugin\_basename( \_\_FILE\_\_ ) ) . '/languages' );

}

add\_action( 'plugins\_loaded', 'my\_plugin\_load\_textdomain' );

### 9.2 Translation Functions

| ✓ | Function | Use Case |
| --- | --- | --- |
| [ ] | **\_\_()** | Translate string (returns) |
| [ ] | **\_e()** | Translate and echo string |
| [ ] | **\_x()** | Translate with context |
| [ ] | **\_ex()** | Translate with context and echo |
| [ ] | **\_n()** | Translate singular/plural |
| [ ] | **\_nx()** | Translate singular/plural with context |
| [ ] | **esc\_html\_\_()** | Translate and escape for HTML (returns) |
| [ ] | **esc\_html\_e()** | Translate and escape for HTML (echoes) |
| [ ] | **esc\_attr\_\_()** | Translate and escape for attributes (returns) |
| [ ] | **esc\_attr\_e()** | Translate and escape for attributes (echoes) |

### 9.3 Translation Best Practices

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Wrap all strings** | All user-facing strings must be translatable |
| [ ] | **Use text domain** | Always pass text domain as second parameter |
| [ ] | **No variables in strings** | Don't use variables inside translation functions |
| [ ] | **Use placeholders** | Use sprintf() for dynamic content |
| [ ] | **Provide context** | Use \_x() when string could be ambiguous |

**Example:**

// WRONG: Variable inside translation

echo \_\_( "Hello $name", 'my-plugin' );

// CORRECT: Use sprintf with placeholder

echo sprintf( \_\_( 'Hello %s', 'my-plugin' ), $name );

// CORRECT: With context

echo \_x( 'Post', 'noun', 'my-plugin' ); // vs. 'Post' as verb

## 10. PHP Coding Standards

### 10.1 PHP Version & Syntax

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Declare minimum PHP** | In plugin header: Requires PHP: 7.2 (or higher) |
| [ ] | **Full PHP tags** | Always use <?php ?>, never short tags <? ?> |
| [ ] | **No alternative tags** | Don't use <% %> or <script language="php"> |
| [ ] | **No closing tag** | Omit ?> at end of PHP-only files |
| [ ] | **UTF-8 encoding** | Save all files as UTF-8 without BOM |
| [ ] | **No BOM** | Byte Order Mark is not allowed |

### 10.2 Code Readability

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Human-readable code** | No obfuscation, minification, or packing of PHP |
| [ ] | **Meaningful names** | Use descriptive function/variable names |
| [ ] | **No unclear naming** | Avoid names like $z12sdf813d |
| [ ] | **Include source** | If JS/CSS is minified, include unminified source or link to repository |
| [ ] | **Document build tools** | If using build tools, document how to use them |

### 10.3 Error Handling

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No error\_reporting()** | Don't change PHP error reporting settings |
| [ ] | **No ini\_set() for errors** | Don't use ini\_set('display\_errors') |
| [ ] | **Use error\_log()** | For logging errors |
| [ ] | **No var\_dump()** | Remove all debugging statements |
| [ ] | **No print\_r()** | Remove all debugging statements |
| [ ] | **Handle exceptions** | Use try/catch for exception-prone code |

### 10.4 Global State

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No timezone changes** | Don't use date\_default\_timezone\_set() |
| [ ] | **No global overrides** | Don't modify $wpdb or other core globals |
| [ ] | **Self-contained** | Plugin should not alter global WordPress environment |

## 11. Forbidden & Discouraged Functions

### 11.1 Absolutely Forbidden (Severity 7 - Rejection)

| ✓ | Function/Construct | Reason |
| --- | --- | --- |
| [ ] | **eval()** | Security risk - arbitrary code execution |
| [ ] | **create\_function()** | Deprecated and security risk |
| [ ] | **goto** | Poor code structure |
| [ ] | **Backtick operator** | Shell command execution risk |
| [ ] | **HEREDOC** | Not allowed |
| [ ] | **NOWDOC** | Not allowed |
| [ ] | **base64\_decode()** (for obfuscation) | Code obfuscation |
| [ ] | **str\_rot13()** | Code obfuscation |
| [ ] | **move\_uploaded\_file()** | Security risk |
| [ ] | **passthru()** | Shell command execution |
| [ ] | **proc\_open()** | Shell command execution |
| [ ] | **exec()** | Shell command execution (use with extreme caution) |
| [ ] | **system()** | Shell command execution |
| [ ] | **shell\_exec()** | Shell command execution |

### 11.2 Forbidden WordPress Internal Functions

| ✓ | Function | Reason |
| --- | --- | --- |
| [ ] | **\_cleanup\_header\_comment()** | Internal function |
| [ ] | **\_get\_plugin\_data\_markup\_translate()** | Internal function |
| [ ] | **\_transition\_post\_status()** | Internal function |
| [ ] | **\_wp\_post\_revision\_fields()** | Internal function |
| [ ] | **do\_shortcode\_tag()** | Internal function |
| [ ] | **get\_post\_type\_labels()** | Internal function |
| [ ] | **wp\_get\_sidebars\_widgets()** | Internal function |
| [ ] | **wp\_get\_widget\_defaults()** | Internal function |

### 11.3 Discouraged Functions

| ✓ | Function | Alternative |
| --- | --- | --- |
| [ ] | **set\_time\_limit()** | Avoid or use temporarily |
| [ ] | **ini\_set()** | Avoid changing PHP settings |
| [ ] | **ini\_alter()** | Avoid changing PHP settings |
| [ ] | **dl()** | Don't load PHP extensions |
| [ ] | **error\_reporting()** | Don't change error reporting |
| [ ] | **extract()** | Use explicit variable assignment |

### 11.4 Deprecated WordPress Functions

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No deprecated functions** | Check WordPress Codex for deprecated functions |
| [ ] | **No deprecated classes** | Use current WordPress classes |
| [ ] | **No deprecated parameters** | Check function signatures |
| [ ] | **No deprecated constants** | Use current constants |

## 12. Performance Optimization

### 12.1 Script & Style Performance

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Scripts in footer** | Load non-critical scripts in footer |
| [ ] | **Conditional loading** | Only load assets on pages where needed |
| [ ] | **Minimize file size** | Keep cumulative scripts < 293 KB |
| [ ] | **Minimize stylesheet size** | Keep cumulative styles < 293 KB |
| [ ] | **No global loading** | Don't load assets on ALL pages unless necessary |
| [ ] | **Use loading strategy** | Implement defer or async for scripts |
| [ ] | **Combine files** | Reduce number of HTTP requests |

### 12.2 Database Performance

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Avoid slow queries** | Don't use posts\_per\_page => -1 without good reason |
| [ ] | **Use pagination** | Limit query results |
| [ ] | **Index custom tables** | Add indexes to frequently queried columns |
| [ ] | **Cache results** | Use transients for expensive queries |
| [ ] | **Avoid meta queries** | Meta queries are slow; use custom tables if needed |

### 12.3 General Performance

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use transients** | Cache expensive operations with set\_transient() |
| [ ] | **Use object cache** | Use wp\_cache\_set() for frequently accessed data |
| [ ] | **Minimize external calls** | Limit HTTP requests to external services |
| [ ] | **Set timeouts** | Use short timeouts for external requests (5-10 seconds) |
| [ ] | **Lazy load** | Load resources only when needed |
| [ ] | **Use WP Cron** | Schedule heavy tasks with wp\_schedule\_event() |

### 12.4 Image Optimization

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use WordPress functions** | Use wp\_get\_attachment\_image() for images |
| [ ] | **Responsive images** | WordPress generates srcset automatically |
| [ ] | **Optimize file size** | Compress images before including |

## 13. WordPress.org Guidelines (18 Rules)

### Guideline 1: GPL Compatibility

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **GPL-compatible license** | All code, data, images must be GPLv2 or later compatible |
| [ ] | **Third-party libraries** | All included libraries must be GPL-compatible |
| [ ] | **License file** | Include LICENSE.txt or reference in header |
| [ ] | **Check library licenses** | MIT, BSD, Apache 2.0 are compatible; proprietary is not |

### Guideline 2: Developer Responsibility

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Verify all files** | Ensure all files comply with guidelines |
| [ ] | **Check third-party code** | Verify licensing and terms of use |
| [ ] | **No circumvention** | Don't intentionally write code to bypass guidelines |
| [ ] | **Maintain compliance** | Don't restore removed code |

### Guideline 3: Stable Version Available

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Keep WordPress.org updated** | Don't distribute elsewhere without updating repo |
| [ ] | **Stable version in directory** | Users download from WordPress.org, not dev environment |

### Guideline 4: Human-Readable Code

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No obfuscation** | No p,a,c,k,e,r, uglify mangle, etc. |
| [ ] | **Clear naming** | Use meaningful variable/function names |
| [ ] | **Provide source** | Include unminified source or link to repository |
| [ ] | **Document build tools** | Explain how to use development tools |

### Guideline 5: No Trialware

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No locked features** | All code in plugin must be fully functional |
| [ ] | **No trial periods** | No time-limited functionality |
| [ ] | **No quotas** | No usage limits that lock features |
| [ ] | **No sandbox-only APIs** | Must provide real functionality |
| [ ] | **Upselling allowed** | Can promote paid add-ons (see Guideline 11) |

### Guideline 6: Software as a Service (SaaS) Permitted

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Document service** | Clearly explain external service in readme |
| [ ] | **Link to terms** | Provide link to service's Terms of Use |
| [ ] | **Substantial functionality** | Service must provide real value |
| [ ] | **No license validation only** | Service can't exist solely to validate licenses |
| [ ] | **No fake services** | Don't move code to external service to fake functionality |
| [ ] | **No storefronts** | Plugin can't be just a front-end for external products |

### Guideline 7: No Tracking Without Consent

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Opt-in required** | User must explicitly consent to tracking |
| [ ] | **Document data collection** | Explain what data is collected and why |
| [ ] | **Privacy policy** | Provide clear privacy policy |
| [ ] | **No automated collection** | Don't collect data without confirmation |
| [ ] | **No misleading consent** | Don't trick users into submitting data |
| [ ] | **No offloading assets** | Don't load images/scripts from external servers for tracking |
| [ ] | **No undocumented blocklists** | Document use of external data |
| [ ] | **No ad tracking** | No third-party ad mechanisms that track users |

**Exception:** SaaS plugins may contact their service (but must document it).

### Guideline 8: No Executable Code via Third-Party

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No remote code execution** | Don't download and execute code from external sources |
| [ ] | **No eval() of remote data** | Don't eval() content from APIs |
| [ ] | **No create\_function() of remote data** | Don't dynamically create functions from external data |

**Exception:** Webhooks that trigger existing plugin functions are OK.

### Guideline 9: No Illegal/Dishonest/Offensive Actions

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Legal compliance** | Follow all applicable laws |
| [ ] | **No spam** | Don't send unsolicited emails |
| [ ] | **No malware** | No malicious code |
| [ ] | **No deceptive practices** | Be honest about functionality |
| [ ] | **No circumvention** | Don't help users break guidelines |

### Guideline 10: No External Links Without Permission

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No auto-inserted links** | Don't add links to user's public site without permission |
| [ ] | **No hidden credits** | Don't hide attribution links in content |
| [ ] | **Ask permission** | Use opt-in setting for any external links |

### Guideline 11: Don't Hijack Admin Dashboard

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Minimal admin notices** | Don't show constant nags |
| [ ] | **Dismissible notices** | Make notices dismissible |
| [ ] | **Limit promotions** | Show upsells on plugin pages only |
| [ ] | **No large banners** | Don't take over admin with ads |
| [ ] | **Respect user experience** | Don't be intrusive |

### Guideline 12: No Spam in Readme

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No keyword stuffing** | Write naturally |
| [ ] | **No excessive links** | Limit promotional links |
| [ ] | **No misleading tags** | Use accurate tags |
| [ ] | **Professional content** | Write clear, helpful descriptions |

### Guideline 13: Use WordPress Default Libraries

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use bundled jQuery** | Don't include your own copy |
| [ ] | **Use bundled libraries** | Use WordPress's Underscore, Backbone, etc. |
| [ ] | **No duplicate libraries** | Don't bundle libraries WordPress provides |
| [ ] | **Check available libraries** | See what WordPress includes before bundling |

### Guideline 14: Avoid Frequent Commits

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Batch changes** | Don't commit every small change |
| [ ] | **Test before committing** | Ensure changes work |
| [ ] | **Meaningful commits** | Each commit should be a logical unit |

### Guideline 15: Increment Version Numbers

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Increase version** | Each release must have higher version number |
| [ ] | **Update both files** | Change version in main file AND readme.txt |
| [ ] | **Semantic versioning** | Use MAJOR.MINOR.PATCH format |
| [ ] | **Match stable tag** | Stable tag must match actual tagged version |

### Guideline 16: Complete Plugin at Submission

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Functional plugin** | Must be fully working at submission |
| [ ] | **No placeholders** | Don't submit incomplete code |
| [ ] | **No "coming soon"** | All advertised features must exist |
| [ ] | **Test thoroughly** | Ensure plugin works before submitting |

### Guideline 17: Respect Trademarks

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **No trademark infringement** | Don't use others' trademarks in name/slug |
| [ ] | **No brand confusion** | Don't imply official affiliation |
| [ ] | **Check name availability** | Search for existing plugins with similar names |
| [ ] | **Rename if needed** | Be prepared to change name if trademark issue |

### Guideline 18: WordPress.org's Right to Maintain Directory

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Accept decisions** | WordPress.org has final say |
| [ ] | **Respond to requests** | Reply to plugin team emails |
| [ ] | **Fix issues promptly** | Address security issues immediately |
| [ ] | **Follow new guidelines** | Comply with updated requirements |

## 14. readme.txt Requirements

### 14.1 Required Fields

| ✓ | Field | Requirement |
| --- | --- | --- |
| [ ] | **Plugin Name** | Must match main file header |
| [ ] | **Contributors** | WordPress.org usernames (comma-separated) |
| [ ] | **Tags** | Relevant keywords (max 12) |
| [ ] | **Requires at least** | Minimum WordPress version |
| [ ] | **Tested up to** | Latest WordPress version tested |
| [ ] | **Stable tag** | Current version number (must match tagged version in SVN) |
| [ ] | **License** | GPL-compatible license |
| [ ] | **License URI** | Link to license |

### 14.2 Recommended Sections

| ✓ | Section | Content |
| --- | --- | --- |
| [ ] | **Short Description** | Brief description (max 150 characters) |
| [ ] | **Description** | Detailed explanation of features |
| [ ] | **Installation** | Step-by-step installation instructions |
| [ ] | **Frequently Asked Questions** | Common questions and answers |
| [ ] | **Screenshots** | List of screenshots with descriptions |
| [ ] | **Changelog** | Version history with changes |
| [ ] | **Upgrade Notice** | Important notes for users upgrading |

### 14.3 Optional but Helpful

| ✓ | Field | Purpose |
| --- | --- | --- |
| [ ] | **Requires PHP** | Minimum PHP version |
| [ ] | **Donate link** | Link to donation page |

### 14.4 Formatting

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Use markdown** | Format with markdown syntax |
| [ ] | **Headers** | Use == for h2, = for h3 |
| [ ] | **Lists** | Use \* or 1. for lists |
| [ ] | **Code blocks** | Use backticks for code |
| [ ] | **No HTML** | Use markdown, not HTML tags |

**Example readme.txt structure:**

=== Plugin Name ===

Contributors: username

Tags: tag1, tag2

Requires at least: 5.0

Tested up to: 6.4

Stable tag: 1.0.0

License: GPLv2 or later

License URI: https://www.gnu.org/licenses/gpl-2.0.html

Short description here.

== Description ==

Detailed description here.

== Installation ==

1. Upload plugin

2. Activate

3. Configure

== Frequently Asked Questions ==

= Question? =

Answer.

== Changelog ==

= 1.0.0 =

\* Initial release

## 15. SVN & Submission Process

### 15.1 Before Submission

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Run Plugin Check tool** | Install and run official Plugin Check plugin |
| [ ] | **Fix all errors** | Address all errors reported by Plugin Check |
| [ ] | **Review warnings** | Consider fixing warnings |
| [ ] | **Test thoroughly** | Test in clean WordPress install |
| [ ] | **Enable 2FA** | Two-Factor Authentication required on WordPress.org account |
| [ ] | **Clean package** | Remove dev files, VCS directories, etc. |
| [ ] | **Verify versions** | Ensure version numbers match everywhere |

### 15.2 Submission

| ✓ | Requirement | Details |
| --- | --- | --- |
| [ ] | **Create ZIP file** | Package plugin folder as .zip |
| [ ] | **Submit via WordPress.org** | Go to <https://wordpress.org/plugins/developers/add/> |
| [ ] | **Wait for auto-scan** | Plugin Check tool runs automatically |
| [ ] | **Fix auto-scan errors** | Address any errors before human review |
| [ ] | **Wait for human review** | Can take days to weeks |
| [ ] | **Respond to feedback** | Reply to reviewer emails promptly |

### 15.3 SVN Structure

| ✓ | Directory | Purpose |
| --- | --- | --- |
| [ ] | **/trunk/** | Development version (latest code) |
| [ ] | **/tags/** | Released versions (e.g., /tags/1.0.0/, /tags/1.0.1/) |
| [ ] | **/branches/** | Optional: experimental branches |
| [ ] | **/assets/** | Screenshots, banners, icons for WordPress.org |

### 15.4 Initial SVN Commit

| ✓ | Step | Command |
| --- | --- | --- |
| [ ] | **Checkout SVN** | svn co https://plugins.svn.wordpress.org/your-plugin-slug |
| [ ] | **Copy files to trunk** | Copy all plugin files to /trunk/ |
| [ ] | **Add files** | svn add trunk/\* |
| [ ] | **Commit trunk** | svn ci -m "Initial commit" |
| [ ] | **Create tag** | svn cp trunk tags/1.0.0 |
| [ ] | **Commit tag** | svn ci -m "Tagging version 1.0.0" |
| [ ] | **Verify stable tag** | Ensure readme.txt has Stable tag: 1.0.0 |

### 15.5 Updating Plugin

| ✓ | Step | Details |
| --- | --- | --- |
| [ ] | **Update trunk** | Make changes in /trunk/ |
| [ ] | **Update version** | Change version in main file and readme.txt |
| [ ] | **Update changelog** | Add changes to readme.txt changelog |
| [ ] | **Commit trunk** | svn ci -m "Update to 1.0.1" |
| [ ] | **Create new tag** | svn cp trunk tags/1.0.1 |
| [ ] | **Commit tag** | svn ci -m "Tagging version 1.0.1" |
| [ ] | **Update stable tag** | Change Stable tag in readme.txt to 1.0.1 |

### 15.6 Assets (Screenshots, Banners, Icons)

| ✓ | Asset | Specifications |
| --- | --- | --- |
| [ ] | **Screenshots** | screenshot-1.png, screenshot-2.png, etc. (ideally 1280x720) |
| [ ] | **Banner** | banner-772x250.png (and optionally banner-1544x500.png for retina) |
| [ ] | **Icon** | icon-128x128.png and icon-256x256.png |
| [ ] | **Location** | Place in /assets/ directory (not /trunk/) |

## 16. Common Rejection Reasons

### 16.1 Security Issues

| ✓ | Issue | Solution |
| --- | --- | --- |
| [ ] | **Missing sanitization** | Sanitize all input with appropriate functions |
| [ ] | **Missing escaping** | Escape all output with esc\_html(), esc\_attr(), etc. |
| [ ] | **No nonce checks** | Add nonce verification to all forms/actions |
| [ ] | **No capability checks** | Check current\_user\_can() before actions |
| [ ] | **SQL injection** | Use $wpdb->prepare() for all queries |
| [ ] | **No ABSPATH check** | Add if ( ! defined( 'ABSPATH' ) ) exit; to all files |

### 16.2 Guideline Violations

| ✓ | Issue | Solution |
| --- | --- | --- |
| [ ] | **Calling wp-load.php** | Use WordPress hooks instead |
| [ ] | **Hardcoded paths** | Use plugin\_dir\_path(), plugins\_url() |
| [ ] | **Not using WP libraries** | Remove custom jQuery, use WordPress's bundled version |
| [ ] | **Tracking without consent** | Add opt-in mechanism and document in readme |
| [ ] | **Obfuscated code** | Provide unminified source code |
| [ ] | **Trademark issues** | Rename plugin to avoid trademark conflicts |

### 16.3 Code Quality Issues

| ✓ | Issue | Solution |
| --- | --- | --- |
| [ ] | **Version mismatch** | Ensure version matches in header, readme, and SVN tag |
| [ ] | **Text domain mismatch** | Text domain must exactly match plugin slug |
| [ ] | **Using deprecated functions** | Replace with current WordPress functions |
| [ ] | **Short PHP tags** | Use <?php instead of <? |
| [ ] | **Dev files included** | Remove node\_modules/, .git/, etc. |

### 16.4 Documentation Issues

| ✓ | Issue | Solution |
| --- | --- | --- |
| [ ] | **Incomplete readme** | Fill out all required sections |
| [ ] | **Missing service documentation** | Document external services in readme |
| [ ] | **No privacy policy** | Add privacy policy if collecting data |
| [ ] | **Spam in readme** | Remove keyword stuffing, excessive links |

## 17. Plugin Check Tool Requirements

The official Plugin Check tool (<https://github.com/WordPress/plugin-check>) runs 22 automated checks. Ensure your plugin passes all of them.

### 17.1 Plugin Repository Checks

| ✓ | Check | What It Tests |
| --- | --- | --- |
| [ ] | **i18n\_usage** | Internationalization best practices |
| [ ] | **code\_obfuscation** | Detects obfuscation tools usage |
| [ ] | **file\_type** | Detects hidden files, compressed files, VCS directories |
| [ ] | **plugin\_header\_fields** | Validates required header fields |
| [ ] | **late\_escaping** | Ensures output is escaped |
| [ ] | **plugin\_updater** | Prevents custom updaters |
| [ ] | **plugin\_review\_phpcs** | Runs PHP\_CodeSniffer with WordPress rules |
| [ ] | **direct\_db\_queries** | Checks for direct database queries |
| [ ] | **enqueued\_resources** | Validates proper script/style enqueuing |
| [ ] | **plugin\_readme** | Validates readme.txt format and content |
| [ ] | **localhost** | Detects localhost/127.0.0.1 references |
| [ ] | **no\_unfiltered\_uploads** | Checks for ALLOW\_UNFILTERED\_UPLOADS |
| [ ] | **trademarks** | Checks for trademark usage in slug |
| [ ] | **offloading\_files** | Detects unnecessary remote services |

### 17.2 Security Checks

| ✓ | Check | What It Tests |
| --- | --- | --- |
| [ ] | **late\_escaping** | Output escaping |
| [ ] | **direct\_db\_queries** | SQL injection prevention |

### 17.3 Performance Checks

| ✓ | Check | What It Tests |
| --- | --- | --- |
| [ ] | **performant\_wp\_query\_params** | Slow WP\_Query parameters |
| [ ] | **enqueued\_scripts\_in\_footer** | Scripts loaded in footer |
| [ ] | **enqueued\_scripts\_size** | Total script size < 293 KB |
| [ ] | **enqueued\_styles\_size** | Total stylesheet size < 293 KB |
| [ ] | **enqueued\_styles\_scope** | Stylesheets not loaded on all pages |
| [ ] | **enqueued\_scripts\_scope** | Scripts not loaded on all pages |
| [ ] | **non\_blocking\_scripts** | Non-blocking loading strategy |
| [ ] | **image\_functions** | Proper image insertion functions |

### 17.4 PHPCS Rules (plugin-check.ruleset.xml)

| ✓ | Rule Category | Key Rules |
| --- | --- | --- |
| [ ] | **Database** | PreparedSQL, PreparedSQLPlaceholders, RestrictedClasses, RestrictedFunctions |
| [ ] | **Security** | NonceVerification, ValidatedSanitizedInput, PluginMenuSlug |
| [ ] | **PHP Restrictions** | No backticks, HEREDOC, goto, short tags, alternative tags, BOM |
| [ ] | **WordPress Best Practices** | No deprecated functions/classes, use alternative functions |
| [ ] | **Forbidden Functions** | No eval, create\_function, passthru, proc\_open, etc. |

## Final Pre-Submission Checklist

### Critical Items

| ✓ | Item |
| --- | --- |
| [ ] | All security checks pass (sanitization, escaping, nonces, capabilities) |
| [ ] | All database queries use $wpdb->prepare() |
| [ ] | All files have ABSPATH check |
| [ ] | No wp-load.php or wp-config.php includes |
| [ ] | Text domain matches plugin slug exactly |
| [ ] | Version numbers match in header, readme, and SVN tag |
| [ ] | GPL-compatible license declared |
| [ ] | No obfuscated or minified PHP code |
| [ ] | No forbidden functions (eval, create\_function, goto, etc.) |
| [ ] | Scripts and styles properly enqueued |
| [ ] | Using WordPress bundled libraries (jQuery, etc.) |
| [ ] | readme.txt complete and properly formatted |
| [ ] | Plugin Check tool shows no errors |
| [ ] | 2FA enabled on WordPress.org account |
| [ ] | All dev files removed from package |
| [ ] | Tested in clean WordPress install |
| [ ] | All 18 WordPress.org guidelines followed |

## References

1. David Bryan. (2025, June 18). *How to Build & Publish a Plugin to WordPress.Org by following WordPress Plugin Coding Standards*. Retrieved from attached document.
2. WordPress. (2025). *WordPress Plugin Check - GitHub Repository*. <https://github.com/WordPress/plugin-check>
3. WordPress. (2025). *Plugin Check Documentation - Checks*. <https://github.com/WordPress/plugin-check/blob/trunk/docs/checks.md>
4. WordPress. (2024, March 15). *Detailed Plugin Guidelines*. <https://developer.wordpress.org/plugins/wordpress-org/detailed-plugin-guidelines/>
5. WordPress. (2025). *plugin-check.ruleset.xml - PHPCS Ruleset*. <https://raw.githubusercontent.com/WordPress/plugin-check/refs/heads/trunk/phpcs-rulesets/plugin-check.ruleset.xml>
6. WordPress. (2025). *plugin-review.xml - PHPCS Ruleset*. <https://raw.githubusercontent.com/WordPress/plugin-check/refs/heads/trunk/phpcs-rulesets/plugin-review.xml>
7. WordPress. (n.d.). *Plugin Handbook*. <https://developer.wordpress.org/plugins/>
8. WordPress. (n.d.). *Plugin Readme File Standard*. <https://developer.wordpress.org/plugins/wordpress-org/how-your-readme-txt-works/>
9. WordPress. (n.d.). *How to use Subversion (SVN)*. <https://developer.wordpress.org/plugins/wordpress-org/how-to-use-subversion/>

**Document Version:** 2.0  
**Last Updated:** October 4, 2025  
**Compiled by:** Manus AI  
**License:** This checklist is provided as-is for educational purposes. WordPress and WordPress.org are trademarks of the WordPress Foundation.