

PhpSpreadsheet for WordPress - Complete Plugin Package

File Structure

Create this directory structure in your WordPress plugins folder:

```
wp-content/plugins/phpspreadsheet-wp/
├── phpspreadsheet-wp.php          # Main plugin file
├── readme.txt                    # WordPress.org readme
├── README.md                     # GitHub readme
├── uninstall.php                 # Uninstall handler
├── includes/
│   ├── class-phpspreadsheet-admin.php
│   ├── class-phpspreadsheet-loader.php
│   ├── class-phpspreadsheet-installer.php
│   └── class-phpspreadsheet-hooks.php
├── templates/
│   └── admin-page.php
├── assets/
│   ├── css/
│   │   └── admin.css
│   └── js/
│       └── admin.js
├── languages/
│   └── phpspreadsheet-wp.pot
├── vendor/                      # Auto-created during installation
├── temp/                        # Auto-created during installation
└── logs/                        # Auto-created during installation
```

File Contents

1. phpspreadsheet-wp.php (Main Plugin File)


```
<?php
```

```
/**
 * Plugin Name: PhpSpreadsheet for WordPress
 * Plugin URI: https://github.com/PHPOffice/PhpSpreadsheet
 * Description: Integrates PhpSpreadsheet library into WordPress for Excel file generation. Prc
 * Version: 1.2.0
 * Author: WordPress Development Team
 * License: MIT
 * Requires at least: 5.0
 * Tested up to: 6.4
 * Requires PHP: 7.4
 * Text Domain: phpspreadsheet-wp
 * Domain Path: /languages
 */
```

```
// Prevent direct access
```

```
if (!defined('ABSPATH')) {
    exit;
}
```

```
// Define plugin constants
```

```
define('PHPSpreadsheet_WP_VERSION', '1.2.0');
define('PHPSpreadsheet_WP_PLUGIN_DIR', plugin_dir_path(__FILE__));
define('PHPSpreadsheet_WP_PLUGIN_URL', plugin_dir_url(__FILE__));
define('PHPSpreadsheet_WP_PLUGIN_FILE', __FILE__);
```

```
/**
```

```
 * Main PhpSpreadsheet WordPress Plugin Class
 */
```

```
class PhpSpreadsheet_WordPress_Plugin {
```

```
    /**
```

```
     * Single instance of the plugin
```

```
    */
```

```
    private static $instance = null;
```

```
    /**
```

```
     * PhpSpreadsheet library loaded status
```

```
    */
```

```
    private $library_loaded = false;
```

```
    /**
```

```
     * Get single instance
```

```
    */
```

```
    public static function get_instance() {
        if (null === self::$instance) {
```

```

        self::$instance = new self();
    }
    return self::$instance;
}

/**
 * Constructor
 */
private function __construct() {
    $this->init_hooks();
}

/**
 * Initialize WordPress hooks
 */
private function init_hooks() {
    // Activation and deactivation hooks
    register_activation_hook(__FILE__, array($this, 'activate'));
    register_deactivation_hook(__FILE__, array($this, 'deactivate'));

    // Plugin Loaded hook
    add_action('plugins_loaded', array($this, 'init'));

    // Admin hooks
    add_action('admin_menu', array($this, 'add_admin_menu'));
    add_action('admin_init', array($this, 'admin_init'));
    add_action('admin_notices', array($this, 'admin_notices'));

    // AJAX hooks for installation
    add_action('wp_ajax_phpspreadsheet_install', array($this, 'ajax_install_library'));
    add_action('wp_ajax_phpspreadsheet_check_status', array($this, 'ajax_check_status'));

    // Load Library early for other plugins
    add_action('init', array($this, 'load_library'), 1);
}

/**
 * Plugin activation
 */
public function activate() {
    // Check PHP version
    if (version_compare(PHP_VERSION, '7.4', '<')) {
        deactivate_plugins(plugin_basename(__FILE__));
        wp_die('PhpSpreadsheet for WordPress requires PHP 7.4 or higher.');
```

// Create plugin directory structure

```

$this->create_directory_structure();

// Set default options
add_option('phpspreadsheet_wp_settings', array(
    'auto_load' => true,
    'installation_method' => 'composer',
    'library_version' => ''
));

// Try to install library automatically
$this->auto_install_library();

flush_rewrite_rules();
}

/**
 * Plugin deactivation
 */
public function deactivate() {
    // Clean up if needed
    flush_rewrite_rules();
}

/**
 * Initialize plugin
 */
public function init() {
    // Load text domain
    load_plugin_textdomain('phpspreadsheet-wp', false, dirname(plugin_basename(__FILE__))) .

    // Load library
    $this->load_library();
}

/**
 * Create directory structure
 */
private function create_directory_structure() {
    $dirs = array(
        PHPSpreadsheet_WP_PLUGIN_DIR . 'vendor',
        PHPSpreadsheet_WP_PLUGIN_DIR . 'temp',
        PHPSpreadsheet_WP_PLUGIN_DIR . 'logs'
    );

    foreach ($dirs as $dir) {
        if (!file_exists($dir)) {
            wp_mkdir_p($dir);
        }
    }
}

```

```

        // Add .htaccess for security
        $htaccess_content = "Order deny,allow\nDeny from all\n";
        file_put_contents($dir . '/.htaccess', $htaccess_content);
    }
}

/**
 * Auto-install library during activation
 */
private function auto_install_library() {
    // Check if Composer is available
    if ($this->is_composer_available()) {
        $this->install_via_composer();
    } else {
        // Try to download pre-compiled version
        $this->install_precompiled();
    }
}

/**
 * Check if Composer is available
 */
private function is_composer_available() {
    exec('composer --version 2>&1', $output, $return_code);
    return $return_code === 0;
}

/**
 * Install via Composer
 */
private function install_via_composer() {
    $plugin_dir = PHPSPREADSHEET_WP_PLUGIN_DIR;

    // Create composer.json
    $composer_json = array(
        'name' => 'wordpress/phpspreadsheet-wp',
        'description' => 'PhpSpreadsheet for WordPress',
        'require' => array(
            'phpoffice/phpspreadsheet' => '^1.29'
        ),
        'config' => array(
            'vendor-dir' => 'vendor',
            'optimize-autoloader' => true,
            'classmap-authoritative' => true
        ),
    ),

```

```

        'minimum-stability' => 'stable'
    );

    file_put_contents($plugin_dir . 'composer.json', json_encode($composer_json, JSON_PRETT

// Run composer install
$old_cwd = getcwd();
chdir($plugin_dir);

// Try different composer commands
$commands = array(
    'composer install --no-dev --optimize-autoloader --no-interaction',
    'php composer.phar install --no-dev --optimize-autoloader --no-interaction',
    "" . $this->find_composer_executable() . " install --no-dev --optimize-autoloader
);

$success = false;
foreach ($commands as $command) {
    exec($command . ' 2>&1', $output, $return_code);
    if ($return_code === 0) {
        $success = true;
        break;
    }
    $output = array(); // Reset output for next attempt
}

chdir($old_cwd);

if ($success && file_exists($plugin_dir . 'vendor/autoload.php')) {
    $this->log_message('PhpSpreadsheet installed successfully via Composer');
    return true;
} else {
    $this->log_message('Composer installation failed. Output: ' . implode("\n", $output
    return false;
}
}

/**
 * Find Composer executable
 */
private function find_composer_executable() {
    $possible_paths = array(
        'composer',
        'composer.phar',
        '/usr/local/bin/composer',
        '/usr/bin/composer',
        getcwd() . '/composer.phar'

```

```

);

foreach ($possible_paths as $path) {
    exec($path . ' --version 2>&1', $output, $return_code);
    if ($return_code === 0) {
        return $path;
    }
}

return 'composer';
}

/**
 * Install precompiled version
 */
private function install_precompiled() {
    // Try GitHub API first to get latest release
    $api_url = 'https://api.github.com/repos/PHPOffice/PhpSpreadsheet/releases/latest';
    $api_response = wp_remote_get($api_url, array('timeout' => 30));

    $download_url = 'https://github.com/PHPOffice/PhpSpreadsheet/archive/refs/tags/1.29.0.2';
    $version = '1.29.0';

    // Parse API response if available
    if (!is_wp_error($api_response) && wp_remote_retrieve_response_code($api_response) === 200) {
        $release_data = json_decode(wp_remote_retrieve_body($api_response), true);
        if (isset($release_data['zipball_url']) && isset($release_data['tag_name'])) {
            $download_url = $release_data['zipball_url'];
            $version = $release_data['tag_name'];
        }
    }

    $temp_file = PHPSpreadsheet_WP_PLUGIN_DIR . 'temp/phpspreadsheet.zip';
    $extract_dir = PHPSpreadsheet_WP_PLUGIN_DIR . 'vendor/phpoffice/phpspreadsheet';

    // Download with WordPress HTTP API
    $this->log_message('Downloading PhpSpreadsheet from: ' . $download_url);

    $response = wp_remote_get($download_url, array(
        'timeout' => 300,
        'redirection' => 5,
        'httpversion' => '1.1',
        'user-agent' => 'WordPress/' . get_bloginfo('version'),
        'headers' => array(
            'Accept' => 'application/zip'
        )
    ));
}

```



```

if (is_wp_error($response)) {
    $this->log_message('Download failed: ' . $response->get_error_message());
    return false;
}

$response_code = wp_remote_retrieve_response_code($response);
if ($response_code !== 200) {
    $this->log_message('Download failed with HTTP code: ' . $response_code);
    return false;
}

// Save downloaded content
$body = wp_remote_retrieve_body($response);
if (empty($body)) {
    $this->log_message('Downloaded file is empty');
    return false;
}

file_put_contents($temp_file, $body);

// Verify file was saved
if (!file_exists($temp_file) || filesize($temp_file) < 1000) {
    $this->log_message('Downloaded file is invalid or too small');
    return false;
}

// Extract using WordPress Filesystem API
WP_Filesystem();
global $wp_filesystem;

$temp_extract = PHPSpreadsheet_WP_Plugin_Dir . 'temp/extract/';
wp_mkdir_p($temp_extract);

// Try to extract
$result = unzip_file($temp_file, $temp_extract);

if (is_wp_error($result)) {
    $this->log_message('Extraction failed: ' . $result->get_error_message());
    return false;
}

// Find the extracted directory
$extracted_dirs = glob($temp_extract . '*', GLOB_ONLYDIR);
if (empty($extracted_dirs)) {
    $this->log_message('No directories found after extraction');
    return false;
}

```

```

    }

    $source_dir = $extracted_dirs[0];
    $this->log_message('Extracted to: ' . $source_dir);

    // Create destination directory
    wp_mkdir_p($extract_dir);

    // Copy files
    if ($this->copy_directory($source_dir, $extract_dir)) {
        // Create our custom autoloader
        $this->create_autoloader();

        // Clean up
        $wp_filesystem->delete($temp_file);
        $wp_filesystem->delete($temp_extract, true);

        $this->log_message('PhpSpreadsheet installed successfully (precompiled version: ' .
            return true;
    } else {
        $this->log_message('Failed to copy extracted files');
        return false;
    }
}

/**
 * Create simple autoloader
 */
private function create_autoloader() {
    $autoloader_content = '<?php

/**
 * Simple autoloader for PhpSpreadsheet
 * Compatible with PHP 7.4+
 */

// Prevent direct access
if (!defined("ABSPATH")) {
    exit;
}

// Register autoloader using corrected logic
spl_autoload_register(function ($class) {
    if (strpos($class, '\PhpOffice\PhpSpreadsheet\'') === 0) {
        $php_path = "PhpOffice\PhpSpreadsheet\\";
        $file = __DIR__ . "/phpoffice/phpspreadsheet/src/" . str_replace('\\', '/', $class);
        if (file_exists($file)) {
            require_once $file;
        }
    }
});

```

```

    }
}
});

// Alternative: Try to load Composer autoloader if available
$composer_autoload = __DIR__ . "/phpoffice/phpspreadsheet/vendor/autoload.php";
if (file_exists($composer_autoload)) {
    require_once $composer_autoload;
}
';

    file_put_contents(PHPSpreadsheet_WP_Plugin_Dir . 'vendor/autoload.php', $autoloader_cor
}

/**
 * Copy directory recursively
 */
private function copy_directory($src, $dst) {
    if (!file_exists($src)) {
        $this->log_message('Source directory does not exist: ' . $src);
        return false;
    }

    // Use WordPress filesystem API
    WP_Filesystem();
    global $wp_filesystem;

    if (!$wp_filesystem) {
        $this->log_message('WordPress filesystem not available, falling back to PHP function');
        return $this->copy_directory_php($src, $dst);
    }

    // Create destination directory
    if (!$wp_filesystem->is_dir($dst)) {
        if (!$wp_filesystem->mkdir($dst, FS_CHMOD_DIR)) {
            $this->log_message('Failed to create directory: ' . $dst);
            return false;
        }
    }

    // Copy files recursively
    return $this->copy_directory_recursive($src, $dst, $wp_filesystem);
}

/**
 * Copy directory recursively using WordPress filesystem
 */
private function copy_directory_recursive($src, $dst, $wp_filesystem) {

```

```

$dir_handle = opendir($src);
if (!$dir_handle) {
    return false;
}

while (($file = readdir($dir_handle)) !== false) {
    if ($file === '.' || $file === '..') {
        continue;
    }

    $src_path = $src . '/' . $file;
    $dst_path = $dst . '/' . $file;

    if (is_dir($src_path)) {
        if (!$wp_filesystem->is_dir($dst_path)) {
            if (!$wp_filesystem->mkdir($dst_path, FS_CHMOD_DIR)) {
                closedir($dir_handle);
                return false;
            }
        }

        if (!$this->copy_directory_recursive($src_path, $dst_path, $wp_filesystem)) {
            closedir($dir_handle);
            return false;
        }
    } else {
        if (!$wp_filesystem->copy($src_path, $dst_path)) {
            // Fallback to PHP copy
            if (!copy($src_path, $dst_path)) {
                $this->log_message('Failed to copy file: ' . $src_path . ' to ' . $dst_
                closedir($dir_handle);
                return false;
            }
        }
    }
}

closedir($dir_handle);
return true;
}

/**
 * Fallback copy using PHP functions
 */
private function copy_directory_php($src, $dst) {
    $dir = opendir($src);
    if (!$dir) {

```

```

        return false;
    }

    wp_mkdir_p($dst);

    while (false !== ($file = readdir($dir))) {
        if ($file !== '.' && $file !== '..') {
            $src_path = $src . '/' . $file;
            $dst_path = $dst . '/' . $file;

            if (is_dir($src_path)) {
                $this->copy_directory_php($src_path, $dst_path);
            } else {
                copy($src_path, $dst_path);
            }
        }
    }
    closedir($dir);
    return true;
}

/**
 * Load PhpSpreadsheet library
 */
public function load_library() {
    if ($this->library_loaded) {
        return true;
    }

    $autoload_paths = array(
        PHPSPREADSHEET_WP_PLUGIN_DIR . 'vendor/autoload.php',
        ABSPATH . 'vendor/autoload.php'
    );

    foreach ($autoload_paths as $path) {
        if (file_exists($path)) {
            require_once $path;

            // Test if PhpSpreadsheet is available
            if (class_exists('PhpOffice\PhpSpreadsheet\Spreadsheet')) {
                $this->library_loaded = true;
                $this->log_message('PhpSpreadsheet loaded from: ' . $path);

                // Hook for other plugins
                do_action('phpspreadsheet_wp_loaded');

                return true;
            }
        }
    }
}

```

```

        }
    }
}

$this->log_message('PhpSpreadsheet library not found');
return false;
}

/**
 * Check if library is loaded
 */
public function is_library_loaded() {
    return $this->library_loaded;
}

/**
 * Get library version
 */
public function get_library_version() {
    if (!$this->library_loaded) {
        return false;
    }

    try {
        $reflection = new ReflectionClass('PhpOffice\PhpSpreadsheet\Spreadsheet');
        $filename = $reflection->getFileName();

        // Try to get version from composer.json
        $composer_file = dirname($filename) . '/../../composer.json';
        if (file_exists($composer_file)) {
            $composer_data = json_decode(file_get_contents($composer_file), true);
            return $composer_data['version'] ?? 'Unknown';
        }

        return 'Installed';
    } catch (Exception $e) {
        return 'Unknown';
    }
}

/**
 * Add admin menu
 */
public function add_admin_menu() {
    add_options_page(
        'PhpSpreadsheet Settings',
        'PhpSpreadsheet',

```

```

        'manage_options',
        'phpspreadsheet-wp',
        array($this, 'admin_page')
    );
}

/**
 * Initialize admin settings
 */
public function admin_init() {
    register_setting('phpspreadsheet_wp_settings', 'phpspreadsheet_wp_settings');

    add_settings_section(
        'phpspreadsheet_wp_main',
        'PhpSpreadsheet Configuration',
        array($this, 'settings_section_callback'),
        'phpspreadsheet-wp'
    );

    add_settings_field(
        'auto_load',
        'Auto-load Library',
        array($this, 'auto_load_callback'),
        'phpspreadsheet-wp',
        'phpspreadsheet_wp_main'
    );
}

/**
 * Settings section callback
 */
public function settings_section_callback() {
    echo '<p>Configure PhpSpreadsheet library settings for WordPress.</p>';
}

/**
 * Auto-load callback
 */
public function auto_load_callback() {
    $options = get_option('phpspreadsheet_wp_settings');
    $checked = isset($options['auto_load']) && $options['auto_load'] ? 'checked' : '';
    echo "<input type='checkbox' name='phpspreadsheet_wp_settings[auto_load]' value='1' $ch
}

/**
 * Admin page
 */

```

```

public function admin_page() {
    $library_status = $this->is_library_loaded() ? true : false;
    $library_version = $this->get_library_version();

    ?>
    <div class="wrap">
        <h1>PhpSpreadsheet for WordPress</h1>

        <div class="notice notice-info">
            <p><strong>Status:</strong>
                <?php if ($library_status): ?>
                    <span style="color: green;">✓ Loaded</span>
                    (Version: <?php echo esc_html($library_version); ?>)
                <?php else: ?>
                    <span style="color: red;">X Not Loaded</span>
                <?php endif; ?>
            </p>
        </div>

        <?php if (!$library_status): ?>
            <div class="notice notice-warning">
                <p><strong>PhpSpreadsheet library is not installed.</strong></p>
                <p>
                    <button type="button" class="button button-primary" id="install-phpspre
                        Install PhpSpreadsheet
                    </button>
                    <span id="install-status" style="margin-left: 10px;"></span>
                </p>
            </div>
        <?php endif; ?>

        <form method="post" action="options.php">
            <?php
                settings_fields('phpspreadsheet_wp_settings');
                do_settings_sections('phpspreadsheet-wp');
                submit_button();
            ?>
        </form>

        <div class="postbox" style="margin-top: 20px;">
            <div class="postbox-header">
                <h2>Usage Information</h2>
            </div>
            <div class="inside">
                <p><strong>For Developers:</strong></p>
                <p>Once installed, PhpSpreadsheet is automatically available in your WordPr
                <pre><code>// Check if available

```



```

if (class_exists('PhpOffice\PhpSpreadsheet\Spreadsheet')) {
    $spreadsheet = new \PhpOffice\PhpSpreadsheet\Spreadsheet();
    // Your code here
}

```

// Or use the plugin helper

```

if (function_exists('phpspreadsheet_wp_is_loaded')) {
    if (phpspreadsheet_wp_is_loaded()) {
        // Library is ready to use
    }
}

```

Installation Paths:

- `<?php echo esc_html(PHPSPREADSHEET_WP_PLUGIN_DIR . 'vendor/au`
- `<?php echo esc_html(ABSPATH . 'vendor/autoload.php'); ?></coc`

```

jQuery(document).ready(function($) {

```

```

    $('#install-phpspreadsheet').click(function() {

```

```

        var $button = $(this);

```

```

        var $status = $('#install-status');

```

```

        $button.prop('disabled', true).text('Installing...');

```

```

        $status.html('<span style="color: orange;">Installing PhpSpreadsheet...</span>');

```

```

        $.post.ajaxurl, {

```

```

            action: 'phpspreadsheet_install',

```

```

            nonce: '<?php echo wp_create_nonce('phpspreadsheet_install'); ?>'

```

```

        }, function(response) {

```

```

            if (response.success) {

```

```

                $status.html('<span style="color: green;">✓ Installation successful! P

```

```

                setTimeout(function() {

```

```

                    location.reload();

```

```

                }, 2000);

```

```

            } else {

```

```

                $status.html('<span style="color: red;">X Installation failed: ' + res

```

```

                $button.prop('disabled', false).text('Retry Installation');

```

```

            }

```

```

        }).fail(function() {

```

```

            $status.html('<span style="color: red;">X Installation failed due to netwo

```

```

            $button.prop('disabled', false).text('Retry Installation');

```

```

        });

```

```

    });
});
</script>
<?php
}

/**
 * Admin notices
 */
public function admin_notices() {
    if (!$this->is_library_loaded()) {
        $admin_url = admin_url('options-general.php?page=phpspreadsheet-wp');
        echo '<div class="notice notice-warning is-dismissible">';
        echo '<p><strong>PhpSpreadsheet:</strong> Library not installed. <a href="' . esc_u
        echo '</div>';
    }
}

/**
 * AJAX install library
 */
public function ajax_install_library() {
    if (!wp_verify_nonce($_POST['nonce'], 'phpspreadsheet_install')) {
        wp_die('Security check failed');
    }

    if (!current_user_can('manage_options')) {
        wp_die('Insufficient permissions');
    }

    // Try Composer first, then precompiled
    $success = $this->install_via_composer();

    if (!$success) {
        $success = $this->install_precompiled();
    }

    if ($success) {
        // Try to Load the Library
        $this->library_loaded = false; // Reset status
        $loaded = $this->load_library();

        if ($loaded) {
            wp_send_json_success('PhpSpreadsheet installed and loaded successfully');
        } else {
            wp_send_json_error('Installation completed but library failed to load');
        }
    }
}

```

```

    } else {
        wp_send_json_error('Installation failed. Please check the logs or install manually.
    }
}

/**
 * AJAX check status
 */
public function ajax_check_status() {
    wp_send_json_success(array(
        'loaded' => $this->is_library_loaded(),
        'version' => $this->get_library_version()
    ));
}

/**
 * Log message
 */
private function log_message($message) {
    $log_file = PHPSPREADSHEET_WP_PLUGIN_DIR . 'logs/phpspreadsheet.log';
    $timestamp = current_time('Y-m-d H:i:s');
    $log_entry = "[ $timestamp ] $message" . PHP_EOL;
    file_put_contents($log_file, $log_entry, FILE_APPEND | LOCK_EX);
}
}

/**
 * Helper functions for other plugins/themes
 */

/**
 * Check if PhpSpreadsheet is loaded
 */
function phpspreadsheet_wp_is_loaded() {
    $plugin = PhpSpreadsheet_WordPress_Plugin::get_instance();
    return $plugin->is_library_loaded();
}

/**
 * Get PhpSpreadsheet version
 */
function phpspreadsheet_wp_get_version() {
    $plugin = PhpSpreadsheet_WordPress_Plugin::get_instance();
    return $plugin->get_library_version();
}

/**

```

```
* Force load PhpSpreadsheet
*/
function phpspreadsheet_wp_load_library() {
    $plugin = PhpSpreadsheet_WordPress_Plugin::get_instance();
    return $plugin->load_library();
}

// Initialize plugin
PhpSpreadsheet_WordPress_Plugin::get_instance();
?>
```

2. readme.txt (WordPress.org Format)

txt

=== PhpSpreadsheet for WordPress ===

Contributors: wordpressteam

Donate link: <https://example.com/donate>

Tags: excel, spreadsheet, export, phpoffice, reports, xlsx, csv, data-export

Requires at least: 5.0

Tested up to: 6.4

Requires PHP: 7.4

Stable tag: 1.2.0

License: MIT

License URI: <https://opensource.org/licenses/MIT>

Integrates PhpSpreadsheet library into WordPress for Excel file generation with seamless compat

== Description ==

PhpSpreadsheet for WordPress is a comprehensive plugin that seamlessly integrates the powerful

= Key Features =

- * **One-Click Installation** - Automatically downloads and configures PhpSpreadsheet
- * **Excel Generation** - Create professional Excel files with formatting, charts, and formulas
- * **Zero Code Changes** - Works with existing WordPress code that uses PhpSpreadsheet
- * **Admin Management** - Easy installation and status monitoring through WordPress admin
- * **Secure** - Follows WordPress security best practices
- * **Compatible** - Works with themes, plugins, and custom code
- * **RTL Support** - Full support for right-to-left languages like Hebrew and Arabic

== Installation ==

= Automatic Installation =

1. Log in to your WordPress admin panel
2. Navigate to **Plugins > Add New**
- 3.

