

## Manual:Configuration settings

---

This is an index of all supported **configuration settings** based on the **DefaultSettings.php** (<https://github.com/wikimedia/mediawiki-core/blob/master/includes/DefaultSettings.php>) file.

Never edit DefaultSettings.php; copy appropriate lines to *LocalSettings.php* instead and amend them as appropriate.

If you can't find a configuration setting here, see if it is defined in DefaultSettings.php. The variable should have some documentation there.



Settings removed in versions of MediaWiki prior to 1.27 are no longer listed here. Please use the search box to find the relevant manual page (e.g. Manual:\$wgAllowAnonymo

Listed alphabetically | **Listed by function** | Category:MediaWiki configuration settings

### General Settings

---

- [\\$wgSitename](#) – The name of the site.
- [\\$wgVersion](#) (**deprecated in 1.35**) – MediaWiki version number.

### Global Objects

---

## Contents

---

### General Settings

### Global Objects

### Server URLs and file paths

### Files and file uploads

#### Shared uploads

#### MIME types

#### Images

##### DjVu

##### EXIF

##### ImageMagick

##### SVG

##### Thumbnail settings

#### Antivirus

### Email settings

#### Email notification (Enotif) settings

### Database settings

#### LoadBalancer settings

#### MySQL-specific

#### PostgreSQL-specific

#### SQLite-specific

#### Shared DB settings

#### Extension fields

### Content handlers and storage

### Performance hacks and limits

### Cache

#### File Cache

#### Message Cache

#### Sidebar Cache

#### Parser Cache

#### Memcached settings

#### WAN cache

### HTTP proxy (CDN) settings

#### HTCP multicast purging

### Language, regional and character encoding settings

- [\\$wgAutoloadLocalClasses](#) – Array mapping class names to filenames, for autoloading.
- [\\$wgConf](#) – Create a site configuration object. Not used for much in a default install.

## Server URLs and file paths

---

- [\\$IP](#) – The full path to the installation directory.
- [\\$wgActionPaths](#) – Paths for various user actions. Used to make URLs prettier.
- [\\$wgAppleTouchIcon](#) – Optional URL to an icon to use when bookmarking to the home screen on Apple handheld devices
- [\\$wgArticlePath](#) – The base URL used to create article links.

[Timezone settings](#)

[Language-specific](#)

### **[Output format and skin settings](#)**

[Output](#)

[Skins](#)

### **[ResourceLoader](#)**

### **[Page titles and redirects](#)**

[Namespaces](#)

[Mobile support](#)

### **[Interwiki links and sites](#)**

[Interwiki cache](#)

### **[Parser](#)**

[HTML](#)

[Tidy](#)

### **[Statistics and content analysis](#)**

### **[User accounts, authentication](#)**

[Authentication](#)

### **[User rights, access control and monitoring](#)**

[Access](#)

### **[Security](#)**

### **[Cookies](#)**

### **[Profiling, testing and debugging](#)**

[Debug](#)

[Profiling](#)

### **[Search](#)**

### **[Edit user interface](#)**

### **[Maintenance Scripts setting](#)**

### **[Recent changes, new pages, watchlist and history](#)**

[Feed](#)

### **[Copyright](#)**

### **[Import/Export](#)**

### **[Extensions](#)**

### **[Category](#)**

### **[Logging](#)**

### **[Special pages](#)**

### **[Actions](#)**

### **[Robot policies](#)**

### **[Ajax, Action API and REST API](#)**

[Ajax](#)

[API](#)

### **[Shell and process control](#)**

## HTTP client

## Jobs

## Site customization

### Frontend

## Miscellaneous settings

## See also

- [\\$wgAssumeProxiesUseDefaultProtocolPorts](#) – When the wiki is running behind a proxy and this is set to true, assumes that the proxy exposes the wiki on the standard ports.
- [\\$wgCanonicalServer](#) – Canonical URL of the server, to use in IRC feeds, notification emails, etc.
- [\\$wgConfigRegistry](#) – Registry of factory functions to create config objects
- [\\$wgExtensionAssetsPath](#) – URL to extensions directory.
- [\\$wgExtensionDirectory](#) – The file system path to the extensions directory.
- [\\$wgFavicon](#) – The URL of the site favicon.
- [\\$wgFFmpegLocation](#) – Executable name of FFmpeg client.
- [\\$wgFFprobeLocation](#) – Executable name of FFprobe.
- [\\$wgFileCacheDirectory](#) – Directory where the cached pages will be saved when using server-side caching.
- [\\$wgForceHTTPS](#) ([introduced in 1.34.3](#)) – Redirect insecure HTTP requests to HTTPS.
- [\\$wgHttpsPort](#) – Port where you have HTTPS running. Supports HTTPS on non-standard ports.
- [\\$wgImgAuthPath](#) ([introduced in 1.35](#)) – The base path for `img_auth.php`.
- [\\$wgLoadScript](#) – The URL path to `load.php`.
- [\\$wgLocalStylePath](#) – Same as [\\$wgStylePath](#), but shouldn't point to an external domain.
- [\\$wgLogoHD](#) ([deprecated in 1.35](#)) – URLs for HD versions of the site logo.
- [\\$wgLogos](#) ([introduced in 1.35](#)) – The URL path to various wiki logos.
- [\\$wgLogo](#) – The URL of the site logo.
- [\\$wgMainPageIsDomainRoot](#) ([introduced in 1.34](#)) – Controls if the main page should be served as the domain root.
- [\\$wgReferrerPolicy](#) – Value for the [referrer policy](http://www.w3.org/TR/referrer-policy/) (<http://www.w3.org/TR/referrer-policy/>) meta tag.
- [\\$wgRestPath](#) ([introduced in 1.34](#)) – The URL path to `rest.php`.
- [\\$wgScriptExtension](#) ([removed in 1.31](#)) – Extension to append to script names by default.
- [\\$wgScriptPath](#) – The base URL path.
- [\\$wgScript](#) – The URL of `index.php`.
- [\\$wgServerName](#) – The name of the server without the protocol.
- [\\$wgServer](#) – The base URL of the server.
- [\\$wgSitesCacheFile](#) ([removed in 1.33](#)) – Specify the file location for the site's JSON cache file.
- [\\$wgStyleDirectory](#) – The file system path to the skins directory.
- [\\$wgStylePath](#) – The URL of the skins directory.
- [\\$wgStyleSheetPath](#) ([deprecated in 1.3](#)) – Do not set [\\$wgStyleSheetPath](#). Use [\\$wgStylePath](#) instead.
- [\\$wgThumbPath](#) ([introduced in 1.35.1](#)) – The base path for `thumb_handler.php`.
- [\\$wgTmpDirectory](#) – The path of the temporary directory.

- \$wgUploadBaseUrl – The base URL for uploaded images if shared upload directory isn't to be used
- \$wgUploadDirectory – The file system path of the upload directory.
- \$wgUploadPath – The path segment of the upload directory.
- \$wgUploadStashScalerBaseUrl (**deprecated in 1.36**) – To enable remote on-demand scaling, set this to the thumbnail base URL.
- \$wgUsePathInfo – Whether to use 'pretty' URLs.

See also file/image upload path settings.

See also skins path settings.

## Files and file uploads

---

Uploads have to be specially set up to be secure.

- \$wgAllowAsyncCopyUploads (**removed in 1.27**) – Allow asynchronous copy uploads.
- \$wgAllowCopyUploads – Allow uploads from URLs as well as local files.
- \$wgAllowJavaUploads – Allow Java archive uploads.
- \$wgAttemptFailureEpoch – Bump this to quickly retry thumbnail creation failures.
- \$wgCheckFileExtensions – Check if the file extension is allowed on upload.
- \$wgCopyUploadAsyncTimeout (**removed in 1.27**) – HTTP timeout (in seconds) for asynchronous upload by URL
- \$wgCopyUploadProxy – HTTP proxy to use for copy upload requests.
- \$wgCopyUploadsDomains – Limit \$wgAllowCopyUploads to a specific whitelist
- \$wgCopyUploadsFromSpecialUpload – Add an interface to Special:Upload to enable uploading from URLs
- \$wgCopyUploadTimeout – HTTP timeout (in seconds) for upload by URL
- \$wgDeletedDirectory – What directory to place deleted uploads in.
- \$wgDirectoryMode – Default chmod permission value for newly created directories.
- \$wgDisableUploadScriptChecks – Whether to disable the upload system's checks for HTML/JavaScript.
- \$wgEnableAsyncUploads – Enable deferred upload tasks that use the job queue.
- \$wgEnableUploads – Allow images and other files to be uploaded through the wiki.
- \$wgFileBackends – File backend structure configuration.
- \$wgFileBlacklist – Files with these extensions will never be allowed as uploads.
- \$wgFileExtensions – This is the list of preferred extensions for uploading files. Uploading files with extensions not in this list will trigger a warning.
- \$wgHashedUploadDirectory – If true, uses the `/a/ab/foo.jpg` directory structure.
- \$wgIllegalFileChars – Override the default list of illegal characters in file names.
- \$wgImagePreconnect (**introduced in 1.35**) – On pages containing images, tell the user agent to pre-connect to hosts from \$wgForeignFileRepos.
- \$wgImgAuthDetails – Display reason access to uploaded file was denied to users (img\_auth only)
- \$wgImgAuthUrlPathMap – Map of relative URL directories to match to internal `mwstore://` base storage paths.
- \$wgLocalFileRepo – A single repository structure for the local repository.

- \$wgLockManagers – Array of configuration arrays for each lock manager.
- \$wgMaxUploadSize – Maximum size of uploaded files (in bytes)
- \$wgMediaInTargetLanguage (introduced in 1.33) – Whether translatable media (currently, only SVG images) should be shown in page's target language
- \$wgMinUploadChunkSize – Minimum upload chunk size, in bytes.
- \$wgNativeImageLazyLoading (introduced in 1.34) – Toggles native image lazy loading, via the "loading" attribute.
- \$wgParserTestMediaHandlers – Media handler overrides for parser tests (they don't need to generate actual thumbnails, so a mock will do)
- \$wgStrictFileExtensions – Toggle whether users may upload files not covered by \$FileExtensions.
- \$wgThumbnailEpoch – If rendered thumbnail files are older than this timestamp, they will be re-rendered on demand as if the file didn't already exist.
- \$wgUploadDialog – Configuration for Upload dialog and the files uploaded through it.
- \$wgUploadMaintenance – To disable file delete/restore temporarily.
- \$wgUploadMissingFileUrl – Point wikilinks to missing files ("red file links") to a custom URL.
- \$wgUploadNavigationUrl – Point the upload navigation link in the sidebar to a custom URL.
- \$wgUploadSizeWarning – Warn if uploaded files are larger than this.
- \$wgUploadStashMaxAge – The maximum age of temporary (incomplete) uploaded files.

## Shared uploads

These settings are kept for backward compatibility, see \$wgForeignFileRepos for the new setting, or \$wgUseInstantCommons if you only need read access to images on Commons.

- \$wgCacheSharedUploads – Cache shared metadata in memcached.
- \$wgForeignFileRepos – An array of repository structure for foreign repositories.
- \$wgForeignUploadTargets – Configuration for foreign upload targets for the Upload dialog.
- \$wgHashedSharedUploadDirectory – Whether to hash the shared upload directory.
- \$wgRepositoryBaseUrl – Base URL for a repository wiki.
- \$wgSharedUploadDBname – DB name with metadata about shared directory.
- \$wgSharedUploadDBprefix – DB prefix for shared image uploads
- \$wgSharedUploadDirectory – Path on the file system where shared uploads can be found.
- \$wgSharedUploadPath – Full path to the web server where shared uploads can be found.
- \$wgUseSharedUploads – Whether to look in a shared repository when a file is not found locally
- \$wgFetchCommonsDescriptions – Whether to fetch commons image description pages and display them on the local wiki.

## MIME types

- \$wgLoadFileinfoExtension (removed in 1.22.0) – Switch for loading the FileInfo (<http://pecl.php.net/package/Fileinfo>) extension by PECL at runtime.
- \$wgMimeDetectorCommand – Sets an external MIME detector program.
- \$wgMimeInfoFile – Sets the MIME type info file to be used by MimeAnalyzer.php.

- \$wgMimeTypeFile – Sets the MIME type definition file to be used by MimeAnalyzer.php.
- \$wgTrivialMimeDetection – Switch for trivial MIME detection.

**Warning:** ↑ This is **not** a configuration setting, but a global state variable. It should be used solely by thumb.php!

- \$wgVerifyMimeType – Determines if the MIME type of uploaded files should be checked.
- \$wgVerifyMimeTypeE (introduced in 1.34) – Determines whether extra checks for Internet Explorer type detection should be applied.
- \$wgMimeTypeBlacklist – MIME types to disallow if \$wgVerifyMimeType is enabled.
- \$wgXMLMimeTypes – An array of XML MIME types we can detect via MimeMagic

See also: Manual:Mime type detection

## Images

- \$wgAllowImageMoving (removed in 1.36) – Enable file renaming.
- \$wgCustomConvertCommand – Use another resizing converter, e.g. GraphicMagick.
- \$wgEnableAutoRotation – Whether to enable image auto-rotation based on EXIF information
- \$wgExiftool – Path to exiftool binary. Used for lossless ICC profile swapping.
- \$wgExiv2Command – Path to the exiv2 executable.
- \$wgGalleryOptions – Default settings for the <gallery> tag.
- \$wgIgnoreImageErrors – If true, inline scaled images will still produce <img> tags ready for output instead of showing an error message.
- \$wgImageLimits – A list of user-selectable limits for the image thumbnail size on image description pages.
- \$wgJpegQuality (introduced in 1.32) – When scaling a JPEG thumbnail, this is the quality we request from the backend.
- \$wgJpegTran – Path to jpegtran, for lossless JPEG rotation.
- \$wgMaxImageArea – Don't thumbnail an image if it will use too much working memory.
- \$wgMediaHandlers – Plugins for media file type handling.
- parameters for upload path
- \$wgTrustedMediaFormats – List of trusted media-types and MIME types.

See also: Manual:Configuring file uploads

## DjVu

- \$wgDjvuDump – Path to the djvudump executable.
- \$wgDjvuOutputExtension – File extension for the DjVu post processor output.
- \$wgDjvuPostProcessor – Path of the DjVu post-processor.
- \$wgDjvuRenderer – Path of the ddjvu DjVu renderer.
- \$wgDjvuToXML – Path of the djvutoxml executable.
- \$wgDjvuTxt – Path of the djvutxt DjVu text extraction utility.

## EXIF

- \$wgShowEXIF – Show EXIF data at the end of file description pages.
- \$wgUpdateCompatibleMetadata – Whether to automatically update the `img_metadata` field if it is outdated but compatible with the current version.

## ImageMagick

- \$wgImageMagickConvertCommand – The command used to invoke ImageMagick.
- \$wgImageMagickTempDir – Temporary directory used for ImageMagick.
- \$wgMaxInterlacingAreas – Array of max pixel areas for interlacing per MIME type.
- \$wgSharpenParameter – Sharpening parameter to ImageMagick.
- \$wgSharpenReductionThreshold – Reduction in linear dimensions below which sharpening will be enabled.
- \$wgUseImageMagick – Set to true to use ImageMagick for image resizing.
- \$wgUseTinyRGBForJPGThumbnails – Swaps JPGs that use sRGB ICC profile with TinyRGB profile

Set \$wgUseImageMagick to true to use ImageMagick instead of the builtin functions.

## SVG

- \$wgAllowTitlesInSVG (removed in 1.35) – Allow `<title>` tag in SVG files.
- \$wgSVGConverter – A key from \$wgSVGConverters to use as converter.
- \$wgSVGConverterPath – Path to the directory of the SVG converter being used.
- \$wgSVGConverters – List of paths to Scalable Vector Graphics (SVG) converters.
- \$wgSVGMaxSize – Don't scale a SVG larger than this unless its native size is larger.
- \$wgSVGMetadataCutoff – Don't read SVG metadata beyond this point.

## Thumbnail settings

- \$wgUseImageResize – Whether to enable dynamic server-side image resizing (thumbnailing)
- \$wgGenerateThumbnailOnParse – Allow thumbnail rendering on page view.
- \$wgMaxAnimatedGifArea – Force thumbnailing of animated GIFs above this size to a single frame instead of an animated thumbnail.
- \$wgSharedThumbnailScriptPath – Same as \$wgThumbnailScriptPath, but for shared uploads.
- \$wgShowArchiveThumbnails – Whether to show thumbnails for old images on the image description page.
- \$wgTiffThumbnailType – The format and MIME type to convert TIFF images to.
- \$wgThumbnailScriptPath – Path to `thumb.php`, if you want thumbnail generation on the fly.
- \$wgThumbUpright – Scaling factor applied to width of images when parameter 'upright' is used.
- \$wgThumbLimits – List of image thumbnail sizes users can pick from.
- \$wgResponsiveImages – Whether or not to output high-resolution images for high-DPI displays



- \$wgThumbnailBuckets – List of intermediary thumbnail sizes to use to speed up thumbnail rendering.
- \$wgThumbnailMinimumBucketDistance – Minimal distance between intermediary thumbnail size and target thumbnail size.
- \$wgJpegPixelFormat – Sets YUV chroma subsampling mode for JPEG thumbnails
- \$wgUploadThumbnailRenderHttpCustomHost – When using the "http" \$wgUploadThumbnailRenderMethod, lets one specify a custom Host HTTP header.
- \$wgUploadThumbnailRenderHttpCustomDomain – When using the "http" \$wgUploadThumbnailRenderMethod, lets one specify a custom domain to send the HTTP request to.
- \$wgUploadThumbnailRenderMap – An array of thumbnail widths to be rendered at upload time.
- \$wgUploadThumbnailRenderMethod – The method through which the thumbnails will be prerendered for the entries in \$wgUploadThumbnailRenderMap.

## Antivirus

- \$wgAntivirus – Internal name of virus scanner.
- \$wgAntivirusRequired – Determines if a failed virus scan will cause the file to be rejected.
- \$wgAntivirusSetup – Configuration for different virus scanners.

## Email settings

---

See also User Access: [\\$wgEmailConfirmToEdit](#)

- \$wgAdditionalMailParams – Additional email parameters, will be passed as the last argument to `mail()` PHP function call.
- \$wgAllowHTMLEmail – Set to true to enable HTML emails.
- \$wgEmailAuthentication – Set to true to enable email authentication (confirmation) for this wiki. Except for password reminder emails, email functions only work for authenticated email addresses.
- \$wgEmergencyContact – Site admin email address
- \$wgEnableEmail – Set to true to enable basic email features.
- \$wgEnableSpecialMute (introduced in 1.34) (deprecated in 1.34) – Set to true to enable the Special Mute page.
- \$wgEnableUserEmailBlacklist – Set to true to enable user-to-user email blacklist.
- \$wgEnableUserEmail – Set to true to enable user-to-user email.
- \$wgNewPasswordExpiry – The time, in seconds, after which an emailed temporary password expires.
- \$wgNoReplyAddress – Reply-to address for automatically generated emails.
- \$wgPasswordExpirationDays – The number of days that a user's password is good for.
- \$wgPasswordExpireGrace – The number of seconds after a user's password expires during which they can log in without a password change.
- \$wgPasswordReminderResendTime – Minimum time, in hours, which must elapse between password reminder emails.
- \$wgPasswordSender – Password reminder email address.



- \$wgRequirePasswordforEmailChange (removed in 1.27) – Whether the user must enter their password to change their email address
- \$wgSMTP – For using a direct (authenticated) SMTP server connection.
- \$wgUserEmailConfirmationTokenExpiry – The time, in seconds, after which a email address confirmation token expires.
- \$wgUserEmailUseReplyTo – Set to true to put the sending user's email in a Reply-To header instead of From.

## Email notification (Enotif) settings

*Actual notifications for each user are defined in the options. You can change defaults with \$wgDefaultUserOptions.*

- \$wgEnotifFromEditor – Whether to make email notifications appear to come from the user who makes the edit
- \$wgEnotifImpersonal – Send a generic mail instead of a personalised mail for each user.
- \$wgEnotifMaxRecips – Maximum number of users to mail at once when using impersonal mail.
- \$wgEnotifMinorEdits – Set to true to allow sending email notification also for edits marked as minor edits
- \$wgEnotifRevealEditorAddress – Set to true to insert the page editor's email address into the email notifications Reply-To address
- \$wgEnotifUseJobQ (removed in 1.27) – Send mails via the job queue.
- \$wgEnotifUseRealName – Use real name instead of username in email "from" field.
- \$wgEnotifUserTalk – Enable email notifications for edits on users' talk pages.
- \$wgEnotifWatchlist – Set to true to allow the email notification for watched pages
- \$wgUseEnotif – Overall "use Enotif" setting
- \$wgUsersNotifiedOnAllChanges – Array of usernames who will be sent a notification email for every change which occurs on a wiki.

## Database settings

---

- \$wgAllDBsAreLocalhost (removed in 1.28) – Make all database connections secretly go to localhost.
- \$wgDataCenterUpdateStickTTL – After a state-changing request is done by a client, this determines how many seconds that client should keep using the master datacenter.
- \$wgDBAdminpassword – Password for \$wgDBAdminuser.
- \$wgDBAdminuser – Username of database administrator.
- \$wgDBcompress – Whether to use compression in DB connection.
- \$wgDBDefaultGroup (introduced in 1.32) – Default group to use when getting database connections. For use by maintenance scripts.
- \$wgDBErrorLogTZ – Timezone to use in the error log.
- \$wgDBErrorLog – File to log database errors to.
- \$wgDBmwschema – Schema that stores the MediaWiki tables in.
- \$wgDBname – Name of the database.
- \$wgDBOracleDRCP (removed in 1.34) – When set to true, enables persistent connection with DRCP on Oracle.

- \$wgDBpassword – The password for \$wgDBuser.
- \$wgDBprefix – Table name prefix.
- \$wgDBserver – Host name or IP address of database server.
- \$wgDBssl – Whether to use SSL in DB connection.
- \$wgDBtype – Type of database.
- \$wgDBuser – Database username.
- \$wgDBWindowsAuthentication (removed in 1.35) – Use Integrated Windows Authentication rather than \$wgDBuser/\$wgDBpassword combo.
- \$wgDebugDBTransactions (removed in 1.27) – Enable additional database transaction lifecycle tracing in the debug output.
- \$wgLocalDatabases – Other wikis on this site, can be administered from a single developer account.
- \$wgMultiContentRevisionSchemaMigrationStage (introduced in 1.32) (deprecated in 1.35) – RevisionStore table schema migration stage (content, slots, content\_models & slot\_roles tables).
- \$wgSearchTypeAlternatives – Alternative search types.
- \$wgSearchType – Type of search used for the wiki.

## LoadBalancer settings

- \$wgDBservers – Settings for the database load balancer (database servers and load ratios).
- \$wgLBFactoryConf – Load balancer factory configuration.
- \$wgMasterWaitTimeout (deprecated in 1.24) – How long to wait for a slave to catch up to the master.
- \$wgSlaveLagCritical (removed in 1.36) – If lag is higher than this value, show a more obvious warning than \$wgSlaveLagWarning.
- \$wgSlaveLagWarning (removed in 1.36) – If lag is higher than this value, show a warning in some special pages (like watchlist).
- \$wgDatabaseReplicaLagCritical (introduced in 1.36) – If lag is higher than this value, show a more obvious warning than \$wgDatabaseReplicaLagWarning.
- \$wgDatabaseReplicaLagWarning (introduced in 1.36) – If lag is higher than this value, show a warning in some special pages (like watchlist).

## MySQL-specific

- \$wgDBmysql5 (removed in 1.33) – Set to true to set MySQL connection into UTF-8 encoding (experimental)
- \$wgDBTableOptions – MySQL table options to use during installation or update.
- \$wgSQLMode – Value for the MySQL `sql_mode` parameter

## PostgreSQL-specific

- \$wgDBport – Database port. **Only used by PostgreSQL.**

## SQLite-specific

- \$wgSQLiteDataDir – To override default SQLite data directory.

## Shared DB settings

- \$wgSharedDB – The name of the shared database.
- \$wgSharedPrefix – Table prefix to use on the shared database tables.
- \$wgSharedTables – List of database tables (without prefixes) to share when \$wgSharedDB is enabled.
- \$wgSharedSchema – The table schema for the shared database

## Extension fields

- \$wgExtPGNewFields (removed in 1.36) – Extension defined fields to be added to existing tables
- \$wgExtNewTables (removed in 1.36) – Extension Defined tables to be added on schema change
- \$wgExtNewFields (removed in 1.36) – Extension defined fields to be added to existing tables
- \$wgExtNewIndexes (removed in 1.36) – Extension defined indexes to be added to existing tables
- \$wgExtPGAlteredFields (removed in 1.36) – Extension defined columns to be altered in tables
- \$wgExtModifiedFields (removed in 1.36) – Extension defined fields to change existing tables

## Content handlers and storage

---

- \$wgCompressRevisions – Compress new page revisions if possible
- \$wgContentHandlers – Mapping of content type handlers
- \$wgContentHandlerTextFallback (deprecated in 1.37) – How to react if a plain text version of a non-text Content object is requested using `ContentHandler::getContentText()`
- \$wgDefaultExternalStore – Default external store for new revisions.
- \$wgDiffEngine (introduced in 1.34) – Specify the difference engine to use.
- \$wgExternalDiffEngine – Name of the external diff engine to use.
- \$wgExternalServers – An array of external MySQL servers.
- \$wgExternalStores – External stores allow including content from non-database sources following URL links.
- \$wgNamespaceContentModels – Content types defined for a given namespace
- \$wgPageLanguageUseDB – Allows to change the page language for MediaWiki pages
- \$wgRevisionCacheExpiry – Cache extracted revision text in MemCached.
- \$wgTextModelsToParse – Determines which types of text are parsed as wikitext.

## Performance hacks and limits

---

- \$wgAllowSchemaUpdates – Allow database schema updates to be executed with update.php script.
- \$wgAllowSlowParserFunctions – Allow slow parser functions.
- \$wgCriticalSectionTimeLimit (introduced in 1.36) – The maximum time critical sections are allowed to stay open.
- \$wgDisableQueryPages – Disable all query pages if miser mode is on, not just some.
- \$wgLinkHolderBatchSize – LinkHolderArray batch size.
- \$wgMaxArticleSize – Maximum page size in kilobytes.
- \$wgMaximumMovedPages – Maximum number of pages to move at once when moving subpages with a page.
- \$wgMaxJobDBWriteDuration – Max time (in seconds) a job-generated transaction can spend in writes.
- \$wgMaxUserDBWriteDuration – Max time (in seconds) a user-generated transaction can spend in writes.
- \$wgMemoryLimit – The minimum amount of memory that MediaWiki "needs"; MediaWiki will try to raise PHP's memory limit if it's below this amount.
- \$wgMiserMode – Enable/disable database-intensive features
- \$wgPoolCounterConf – Configuration for processing pool control.
- \$wgQueryCacheLimit – Number of rows to cache in 'querycache' table when miser mode is on.
- \$wgQueryPages – List of Special pages that inherit from QueryPage, that should also be taken in account by the updateSpecialPages.php maintenance script.
- \$wgRequestTimeLimit (introduced in 1.36) – Set a limit on server request wall clock time.
- \$wgTransactionalTimeLimit – Throttle for timing out transactional POST requests
- \$wgWantedPagesThreshold – Minimum number of pages that must link to a page before it will appear on *Special:Wantedpages*.

## Cache

---

Main page: *[Manual:Caching](#)*

---

- \$wgCacheDirectory – Directory for per-wiki long-life cache files in the local filesystem.
- \$wgCacheEpoch – Set this to current time to invalidate all prior cached pages.
- \$wgCachePages – Allow client-side caching of pages
- \$wgChronologyProtectorStash (introduced in 1.36) – The object store type of the ChronologyProtector position store.
- \$wgClockSkewFudge (removed in 1.33) – Fudge time to add to cache timestamps to adjust for clock skew.
- \$wgExtensionInfoMTime – Time to invalidate cache of extensions loaded by the extension registration system.
- \$wgFooterLinkCacheExpiry (introduced in 1.35) – Expiry time for the footer link cache, in seconds, or 0 if disabled.
- \$wgGitInfoCacheDirectory – Directory where GitInfo will look for pre-computed cache files.
- \$wgInvalidateCacheOnLocalSettingsChange – Setting this to true will invalidate all cached pages whenever "LocalSettings.php" is changed
- \$wgLanguageConverterCacheType – The cache type for storing language conversion tables.
- \$wgMainCacheType – Object cache setting
- \$wgMainStash – Replicated object cache setting

- \$wgObjectCaches – Advanced object cache configuration.
- \$wgParserCacheUseJson (**introduced in 1.36**) (**deprecated in 1.36**) – Enable JSON serialization for ParserCache.
- \$wgPHPSessionHandling – Whether to use PHP session handling (`$_SESSION` and `session_*()` functions)
- \$wgSessionPbkdf2Iterations – Number of internal PBKDF2 iterations to use when deriving session secrets.
- \$wgSuspiciousIpExpiry (**introduced in 1.36**) – Time in seconds to remember IPs for, for the purposes of logging IP changes within the same session.
- \$wgUseETag (**removed in 1.28**) – Whether MediaWiki should send an ETag header.
- \$wgUseGzip – Use GZip to store cached pages.

See: *Interwiki for Interwiki cache settings.*

## File Cache

Settings for Server side file caching.

- \$wgFileCacheDepth – Depth of subdirectory hierarchy under \$wgFileCacheDirectory.
- \$wgUseFileCache – Create static HTML pages for unregistered viewers to reduce database traffic.

## Message Cache

- \$wgAdaptiveMessageCache – Keep track which messages are requested and load only most used messages.
- \$wgLocalisationCacheConf – Localisation cache configuration.
- \$wgMessageCacheType – Message cache setting
- \$wgUseLocalMessageCache – Make a local copy of the message cache, in addition to memcached.

## Sidebar Cache

- \$wgEnableSidebarCache – If true, cache the sidebar navigation links.
- \$wgSidebarCacheExpiry – Expiry time for the sidebar cache, in seconds.

## Parser Cache

- \$wgEnableParserCache (**removed in 1.33**) – Keep parsed pages in a cache to speed up page output.
- \$wgOldRevisionParserCacheExpiryTime (**introduced in 1.36**) – The expiry time for the parser cache for old revisions, in seconds.
- \$wgRenderHashAppend – Append a configured value to the parser cache and the sitenotice cache keys.
- \$wgParserCacheExpiryTime – Expiration time (in seconds) of cached parser information

- \$wgParserCacheType – Parser cache setting

## Memcached settings

Settings for configuring the Memcached memory-based object store (if you are using it) docs/memcached.txt (<https://gerrit.wikimedia.org/g/mediawiki/core/+HEAD/docs/memcached.txt>) has more details.

- \$wgMemCachedDebug – Will be set to false in Setup.php, if the server isn't working.
- \$wgMemCachedPersistent – Whether to use persistent connections to Memcached servers.
- \$wgMemCachedServers – Memcached servers
- \$wgMemCachedTimeout – Server connection timeout in microseconds.
- \$wgObjectCacheSessionExpiry – Expiry time to use for session storage when \$wgSessionsInObjectCache is enabled, in seconds.
- \$wgSessionsInMemcached (removed in 1.29) – Store session data in the object cache, rather than on the filesystem.
- \$wgSessionCacheType – The cache type for storing session data.
- \$wgSessionsInObjectCache (removed in 1.33) – Whether to store sessions in object cache.

## WAN cache

- \$wgEnableWANCacheReaper – Whether to verify and enforce WAN cache purges using reliable DB sources as streams.
- \$wgMainWANCache – Main wide area network cache configuration.
- \$wgWANObjectCaches – Advanced Wide-Area-Network cache configuration.

## HTTP proxy (CDN) settings

---

These settings configure MediaWiki when using a caching HTTP proxy server. They apply to caching using Varnish as well as Squid.

- \$wgCdnMaxageLagged – Cache timeout for the CDN when DB replica lag is high
- \$wgCdnMaxageStale (introduced in 1.35) – Cache timeout in seconds when delivering a stale ParserCache response due to PoolCounter contention.
- \$wgCdnMaxageSubstitute – Cache timeout for the CDN when a response is known to be wrong or incomplete (due to load)
- \$wgCdnMaxAge (introduced in 1.34) – Cache TTL for the CDN sent as s-maxage (without ESI) or Surrogate-Control (with ESI).
- \$wgCdnReboundPurgeDelay – Secondary delayed purges of URLs
- \$wgCdnServersNoPurge (introduced in 1.34) – Same as \$wgCdnServers, except that MW has no methods to send HTTP PURGE to an origin part of the configured ranges
- \$wgCdnServers (introduced in 1.34) – An array of *individual* proxy servers to help MediaWiki detect if the request has been proxied.  
But also to send back to the proxy PURGE commands on changes.
- \$wgForcedRawSMaxage – Default maximum age (in seconds) for raw CSS/JavaScript accesses.



- \$wgInternalServer – Internal server name as known to Squid, if different than \$wgServer.
- \$wgLoggedOutMaxAge (introduced in 1.35) – Cache TTL for the user agent sent as max-age, for logged out users.
- \$wgMaxSquidPurgeTitles (removed in 1.27) – Maximum number of titles to purge in any one client operation
- \$wgSquidMaxage (removed in 1.35) – Cache timeout for the squid.
- \$wgSquidPurgeUseHostHeader (deprecated in 1.33) – Whether to use a Host header in purge requests sent to the proxy servers configured in \$wgSquidServers.
- \$wgSquidServersNoPurge (removed in 1.35) – Same as \$wgSquidServers, except that MW has no methods to send HTTP PURGE to an origin part of the configured ranges
- \$wgSquidServers (removed in 1.35) – An array of *individual* proxy servers to help MediaWiki detect if the request has been proxied.  
But also to send back to the proxy PURGE commands on changes.
- \$wgUseCdn (introduced in 1.34) – Enable/disable page view caching.
- \$wgUseESI (deprecated in 1.33) – If you run Squid3 with ESI support, enable this.
- \$wgUseKeyHeader (removed in 1.34) – Send the Key HTTP header for better caching.
- \$wgUsePrivateIPs – Whether private X-Forwarded-For IPs should be accepted
- \$wgUseSquid (removed in 1.35) – Enable/disable Squid.
- \$wgUseXVO (removed in 1.27) – Send X-Vary-Options header for better caching (requires patched Squid).
- \$wgVaryOnXFP – Whether to add X-Forwarded-Proto to the Vary header for API requests and RSS/Atom feeds.

## HTCP multicast purging

- \$wgHTCPMulticastTTL – HTCP multicast purging
- \$wgHTCPRouting – Routing configuration for HTCP multicast purging.

## Language, regional and character encoding settings

---

- \$wgAllUnicodeFixes – Whether to always convert certain Unicode sequences to modern ones regardless of the content language.
- \$wgBrowserBlackList (deprecated in 1.30) – Browser blacklist for non-Unicode-compliant browsers.
- \$wgDefaultLanguageVariant – Default language variant.
- \$wgDisabledVariants – Disabled variants for language variant conversion.
- \$wgDisableLangConversion – Whether to enable language variant conversion.
- \$wgDisableTitleConversion – Whether to enable language variant conversion for links.
- \$wgDummyLanguageCodes (deprecated in 1.29) – List of language codes that have been renamed to new (correct) codes, or don't correspond to an actual interface language.
- \$wgEditEncoding (removed in 1.28) – Characters not explicitly converted by output encoding
- \$wgExtraInterlanguageLinkPrefixes – Array of interwiki prefixes to be treated as language links.
- \$wgExtraLanguageCodes – List of mappings from one language code to another
- \$wgExtraLanguageNames – List of language names or overrides for default names in Names.php.



- \$wgFixArabicUnicode (removed in 1.35) – Set this to true to clean up archaic Unicode sequences in Arabic text.
- \$wgFixMalayalamUnicode (removed in 1.35) – Set this to true to clean up archaic Unicode sequences in Malayalam text.
- \$wgForceUIMsgAsContentMsg – Array of UI messages that should be treated as content messages.
- \$wgGrammarForms – Used for `{{GRAMMAR: }}` parser function.
- \$wgHideInterlanguageLinks – Hide interlanguage links from the sidebar.
- \$wgInterlanguageLinkCodeMap (introduced in 1.35) – Map of interlanguage link codes to language codes to override what is shown as the language name when the interwiki code does not match it exactly.
- \$wgInterwikiMagic – Treat language links as magic connectors, not inline links.
- \$wgLangObjCacheSize – Language cache size, or really how many languages can we handle simultaneously without degrading to crawl speed.
- \$wgLanguageCode – Site language code.
- \$wgLegacyEncoding – Set this to the character set, which you used before you updated to MediaWiki 1.5 (eg 'ISO-8859-1'). MediaWiki will then perform a temporary character set conversion when loading old revisions, which are not marked with "utf-8" flag (revisions created with MediaWiki 1.4 and earlier).
- \$wgLegacySchemaConversion – Option to speed up the conversion step for large sites when migrating the schema from 1.4 to 1.5
- \$wgLoginLanguageSelector – Show a bar of language selection links in the user login and user registration forms.
- \$wgMaxMsgCacheEntrySize – Maximum entry size in the message cache, in bytes.
- \$wgMsgCacheExpiry (removed in 1.34) – When a message in cache expires (in seconds).
- \$wgOverrideUcfirstCharacters (introduced in 1.34) – List of Unicode characters for which capitalization is overridden in `Language::ucfirst`.
- \$wgRawHtmlMessages (introduced in 1.32) – List of messages which might contain raw HTML.
- \$wgUseDatabaseMessages – Translation using MediaWiki: namespace.
- \$wgUsePigLatinVariant – Whether to enable the pig latin variant of English (en-x-piglatin), used to ease variant development work.
- \$wgVariantArticlePath – Used for links when having language's variants.
- \$wgUserFallbackLanguages – maximum number of fallback languages per user

## Timezone settings

- \$wgLocaltimezone – Fake out the timezone that the server thinks it's in.
- \$wgLocalTZoffset – Add a distance to the server timezone in Minutes.

## Language-specific

### English

- \$wgAmericanDates – Use 'mdy' instead of 'dmy' as the default date format, e.g. 'May 12' instead of '12 May'.

## Arabic/Hindi

- \$wgTranslateNumerals – For Hindi, Arabic, and other languages with their own numeral systems, use local numerals instead of Western style (0-9) numerals in some areas of the interface.

## Output format and skin settings

---

### Output

- \$wgAllowMicrodataAttributes (removed in 1.27) – Enable HTML5 microdata attributes for use in wikitext.
- \$wgAllowRdfaAttributes (removed in 1.27) – Enable RDFa attributes for use in wikitext.
- \$wgCleanupPresentationalAttributes (removed in 1.20.1) – Put as much as possible presentational HTML attributes in CSS.
- \$wgDisableOutputCompression – Disable output compression (compression enabled by default if zlib is available).
- \$wgEdititis – Display user edit counts on special page "User list".
- \$wgEditSubmitButtonLabelPublish – Vary the 'save' labels to 'publish' for public wikis.
- \$wgEnableCanonicalServerLink – Output a `<link rel="canonical">` tag on every page indicating the canonical server which should be used, i.e. \$wgServer or \$wgCanonicalServer.
- \$wgExperimentalHtmlIds (removed in 1.32) – Should we allow a broader set of characters in id attributes, per HTML5?
- \$wgExternalInterwikiFragmentMode – Determines which ID escaping mode should be used for external interwiki links
- \$wgFragmentMode – Determines how section IDs should be encoded
- \$wgHtml5Version – Defines the value of the version attribute in the `<html>` tag, if any.
- \$wgHtml5 (removed in 1.34) – Whether to output an HTML 5 doctype.
- \$wgHTMLFormAllowTableFormat – Allow form components to render tables.
- \$wgJsMimeType (removed in 1.34) – Media type (MIME type) to use for JavaScript scripts.
- \$wgMangleFlashPolicy – Whether to mangle any `<cross-domain-policy>` (Adobe cross-domain policy) tags, to prevent XSS attacks.
- \$wgMimeType – The type to send in the `$type` header.
- \$wgSend404Code – Send 404 HTTP status for not found pages.
- \$wgShowRollbackEditCount – Maximum number to count when a rollback link is shown
- \$wgSiteNotice – A message to be displayed at the top of every page.
- \$wgSiteSupportPage (removed in 1.32) – The title of the support page in your wiki.
- \$wgUseMediaWikiUIEverywhere – Applies MediaWiki UI wherever it can be supported.
- \$wgWellFormedXml (removed in 1.27) – Whether to try to make our HTML output well-formed XML.
- \$wgXhtmlNamespaces – Permit other XHTML namespaces in addition to the w3.org default.

The following setting is no longer used in supported versions of MediaWiki, but still exists for backward compatibility

- \$wgXhtmlDefaultNamespace (removed in 1.34) – Namespace to use when outputting in XHTML mode.

## Skins

- \$wgDefaultSkin – The default skin for new users.
- \$wgFallbackSkin – Fallback skin used when the skin defined by \$DefaultSkin can't be found.
- \$wgFooterIcons – Abstract list of footer icons for skins. It replaces the old `copyrightico` and `poweredbyico` code, which until MediaWiki 1.27 could be influenced with \$CopyrightIcon
- \$wgSkinMetaTags (introduced in 1.36) – An array of open graph tags which should be added by all skins.
- \$wgSkipSkins – Specify the name of skins that should not be presented in the list of available skins.
- \$wgSkipSkin (removed in 1.35) – Specify the name of a skin that should not be presented in the list of available skins.
- \$wgUseCombinedLoginLink – Login / create account link behavior when it's possible for anonymous users to create an account.
- \$wgVectorUseIconWatch – Watch and unwatch as a star icon rather than a link (for Vector skin only).
- \$wgVectorUseSimpleSearch – Search form behavior for Vector skin only.

## ResourceLoader

---

*See ResourceLoader for more information.*

- \$wgAllowSiteCSSOnRestrictedPages – Allow site-wide Cascading Style Sheets (CSS) on restricted special pages
- \$wgIncludeLegacyJavaScript (deprecated in 1.26) – Whether to include the `mediawiki.legacy` JavaScript library (old `wikibits.js`), and its dependencies.
- \$wgLegacyJavaScriptGlobals (removed in 1.37) – Whether or not to assign configuration variables to the global window object.
- \$wgResourceBasePath – Default 'remoteBasePath' value for resource loader modules.
- \$wgResourceLoaderDebug – The default debug mode (on/off) for ResourceLoader requests.
- \$wgResourceLoaderEnableJSProfiler (introduced in 1.32) – When enabled, execution of JavaScript modules is profiled client-side.
- \$wgResourceLoaderLESSImportPaths (removed in 1.32) – Default import paths for LESS modules.
- \$wgResourceLoaderLESSVars (removed in 1.32) – Global LESS variables.
- \$wgResourceLoaderMaxage – Maximum time in seconds to cache resources served by ResourceLoader.
- \$wgResourceLoaderMaxQueryLength – Maximum query string length for ResourceLoader requests.
- \$wgResourceLoaderMinifierMaxLineLength (removed in 1.31) – Maximum line length when minifying JavaScript.
- \$wgResourceLoaderMinifierStatementsOnOwnLine (removed in 1.31) – Put each statement on its own line when minifying JavaScript.
- \$wgResourceLoaderSources – Foreign module sources registry.

- \$wgResourceLoaderStorageEnabled – Whether ResourceLoader should attempt to persist modules in localStorage on browsers that support the Web Storage API.
- \$wgResourceLoaderStorageVersion – Cache version for client-side ResourceLoader module storage.
- \$wgResourceLoaderUseObjectCacheForDeps (introduced in 1.35) – Use the main stash instead of the module\_deps table for indirect dependency tracking
- \$wgResourceLoaderValidateJS – Whether to parse JavaScript module prior to minification to validate them.
- \$wgResourceLoaderValidateStaticJS (removed in 1.32) – Whether to parse statically-sourced (file-backed) JavaScript resources prior to minification to validate them.
- \$wgResourceModuleSkinStyles – Array of additional skin-provided stylesheets to existing ResourceLoader modules.
- \$wgResourceModules – Array of all extra defined modules that can later be loaded during the output.
- \$wgVueDevelopmentMode (introduced in 1.35) – Whether to use the development version of Vue.js.

## Page titles and redirects

---

- \$wgDisableHardRedirects – Disable redirects to special pages and interwiki redirects, which use a 302 and have no "redirected from" link.
- \$wgFixDoubleRedirects – Fix double redirects after a page move.
- \$wgInvalidRedirectTargets – Array of invalid page redirect targets.
- \$wgShortPagesNamespaceBlacklist – Optional array of namespaces which should be blacklisted from Special:ShortPages
- \$wgCapitalLinkOverrides – Per namespace configuration for \$wgCapitalLinks.
- \$wgCapitalLinks – Set this to false to avoid forcing the first letter of links to capitals.
- \$wgLegalTitleChars – Override the default list of illegal characters in page titles.
- \$wgMaxRedirects – Max number of redirects to follow when resolving redirects.

## Namespaces

- \$wgContentNamespaces – Namespaces which are considered to contain real content, or articles
- \$wgCanonicalNamespaceNames – Canonical namespace names.
- \$wgExtraNamespaces – Additional namespaces.
- \$wgExtraGenderNamespaces – Same as \$wgExtraNamespaces, but for namespaces with gender distinction.
- \$wgExtraSignatureNamespaces – Controls in which subject (non-talk) namespaces the "Signature" button on the edit toolbar will be displayed.
- \$wgMetaNamespace – Name used for the meta namespace.
- \$wgMetaNamespaceTalk – Name of the project talk namespace.
- \$wgNamespaceAliases – Provide additional names for namespaces.
- \$wgNamespacesWithSubpages – Which namespaces should support subpages?

## Mobile support

## Interwiki links and sites

---

- \$wgInterwikiPrefixDisplayTypes – Mapping of interwiki index prefixes to descriptors that can be used to change the display of interwiki search results.
- \$wgLocalInterwikis – Array of interwiki prefixes for this wiki.
- \$wgLocalInterwiki (removed in 1.35) – Interwiki prefix for this wiki.
- \$wgRedirectSources – Regular expression to restrict URLs which will be displayed as "redirected from" links.
- \$wgSiteTypes – Register handlers for specific types of sites.

## Interwiki cache

See *Interwiki cache* for more information.

- \$wgInterwikiCache – Set to an associative array or file path to a CDB file for Interwiki cache.
- \$wgInterwikiExpiry – Expiry time for memcached cache of interwiki table.
- \$wgInterwikiFallbackSite – Used by maintenance scripts to dump interwiki table
- \$wgInterwikiScopes – Used by `getInterwikiCache( )` to retrieve from interwiki database file

## Parser

---

- \$wgAllowDisplayTitle – Allows use of `{{DISPLAYTITLE}}` magic word
- \$wgAllowExternalImagesFrom – If \$wgAllowExternalImages is false, you can specify exceptions here.
- \$wgAllowExternalImages – Determines whether or not MediaWiki will allow external images to be rendered inline with text.
- \$wgAllowImageTag (deprecated in 1.35) – Allow `<img>` tag to be used in wikitext.
- \$wgCleanSignatures – Substitute templates in signatures.
- \$wgEnableImageWhitelist – Enables an on-wiki external image whitelist
- \$wgEnableMagicLinks – Controls whether magic link functionality should be enabled
- \$wgEnableScaryTranscluding – Allow templates to be imported/transcluded from another wiki.
- \$wgExpensiveParserFunctionLimit – Maximum number of calls per parse to expensive parser functions such as `PAGESINCATEGORY`.
- \$wgExternalLinkTarget – Set a default target for external links.
- \$wgMaxGeneratedPPNodeCount (removed in 1.35) – A complexity limit on template expansion: the maximum number of nodes generated by `Preprocessor::processToObj( )`
- \$wgMaxPPExpandDepth – Stack depth limit for expansion with the preprocessor.
- \$wgMaxPPNodeCount – A complexity limit on template expansion: the maximum number of nodes visited by `PPFrame::expand( )`
- \$wgMaxTemplateDepth – Maximum recursion depth for templates within templates.
- \$wgMaxTocLevel – Maximum indent level of toc.

- \$wgNoFollowDomainExceptions – External links to these domain names (or any subdomains) will not be set to `rel="nofollow"`.
- \$wgNoFollowLinks – If true, external URL links in wiki text will be given the `rel="nofollow"` attribute.
- \$wgNoFollowNsExceptions – Namespaces in which Manual:\$wgNoFollowLinks doesn't apply.
- \$wgParserConf (**removed in 1.36**) – Parser configuration.
- \$wgPreprocessorCacheThreshold – Preprocessor caching threshold.
- \$wgRegisterInternalExternals – Whether to register external links pointing to the same server.
- \$wgRestrictDisplayTitle – Restrict `{ {DISPLAYTITLE} }` to titles that normalize to the same canonical database key
- \$wgTranscludeCacheExpiry – Expiry time for interwiki transclusion cache
- \$wgUrlProtocols – Defines the protocols which are supported and translated in HTML by the mediawiki parser.
- \$wgUseNewMediaStructure (**introduced in 1.36**) – Emit using the new media structure described at, Parsing/Media structure

## HTML

- \$wgRawHtml – Allow raw, unchecked HTML in `<html> . . . </html>` sections.

## Tidy



This page is **outdated**. It was written for an older version of MediaWiki and may not apply to the most recent version. If you have checked or updated this page and found the content to be suitable, please remove this notice. See the [talk page](#) for a possible discussion on this.

Tidy is an open source tool that cleans up broken HTML. You can use this to ensure that broken HTML in articles doesn't affect the layout of your wiki. While Tidy itself is going to be replaced, the configuration settings can still be used for the replacements.

- \$wgTidyConfig – Configuration of the HTML postprocessing tool to use for fixing invalid HTML code.
- \$wgDebugTidy (**removed in 1.33**) – Put tidy warnings in HTML comments, only works for internal tidy.
- \$wgTidyBin (**removed in 1.33**) – Should be set to the path of the tidy binary.
- \$wgTidyConf (**removed in 1.33**) – Should be set to the path of the tidy configuration file.
- \$wgTidyInternal (**removed in 1.33**) – Controls the use of the tidy extension to use an in-process tidy library instead of spawning a separate program.
- \$wgTidyOpts (**removed in 1.33**) – Options passed to tidy on the command line.
- \$wgUseTidy (**removed in 1.33**) – Use tidy to make sure HTML output is sane.
- \$wgValidateAllHtml (**removed in 1.31**) – Validate the overall output using tidy and refuse to display the page if it's not valid.

*See also: Manual:Build Tidy from source.*



## Statistics and content analysis

---

- \$wgActiveUserDays – The number of days within which a person must make edits to be considered an "active" user
- \$wgArticleCountMethod – Method used to determine if a page in a content namespace should be counted as a valid content page (article).
- \$wgDisableCounters (removed in 1.35) – Disable page view counters.
- \$wgExperiencedUserEdits – Determines the number of edits needed to qualify at the "Experienced user" level.
- \$wgExperiencedUserMemberSince – Determines the number of days needed since account creation to qualify at the "Experienced user" level.
- \$wgLearnerEdits – Determines the number of edits needed to qualify at the "Learner" experience level.
- \$wgLearnerMemberSince – Determines the number of days needed since account creation to qualify at the "Learner" experience level.
- \$wgManualRevertSearchRadius (introduced in 1.36) – Maximum number of revisions of a page that will be checked against every new edit made to determine whether the edit was a manual revert.
- \$wgRevertedTagMaxDepth (introduced in 1.36) – Maximum depth (revision count) of reverts that will have their reverted edits marked with the mw-reverted change tag.
- \$wgSiteStatsAsyncFactor (removed in 1.34) – Sets how often site statistics should be updated.

## User accounts, authentication

---

- \$wgAllowRequiringEmailForResets (introduced in 1.34) (deprecated in 1.34) – Temporary feature flag that controls whether users will see a checkbox allowing them to require providing email during password resets.
- \$wgDefaultUserOptions – Default user preferences.
- \$wgHiddenPrefs – Disable certain user preferences.
- \$wgInvalidUsernameCharacters – Characters to prevent during new account creations.
- \$wgMaximalPasswordLength (deprecated in 1.26) – Specifies the maximal length of a user password.
- \$wgMaxNameChars – Maximum number of bytes in username.
- \$wgMaxSigChars – Maximum wiki code length for customized signatures.
- \$wgMinimalPasswordLength (deprecated in 1.26) – Specifies the minimal length of a user password.
- \$wgPasswordPolicy – Specifies various settings related to password strength and security.
- \$wgReservedUsernames – List of account names that cannot exist.
- \$wgShowIPinHeader (removed in 1.27) – Show the IP in the user bar for anonymous users.
- \$wgSignatureValidation (introduced in 1.35) – Behavior of signature validation.
- \$wgSignatureAllowedLintErrors (introduced in 1.35) – List of lint error codes which don't cause signature validation to fail.
- \$wgUserRightsInterwikiDelimiter – Character used as the separator between username and database name on Special:UserRights.

## Authentication



- \$wgAllowSecuritySensitiveOperationIfCannotReauthenticate – Whether to allow security-sensitive operations when reauthentication is not possible
- \$wgAuthenticationTokenVersion – When changed, all existing sessions are invalidated.
- \$wgAuthManagerAutoConfig – Register authentication providers to use with AuthManager
- \$wgAuthManagerConfig – Manual override for authentication provider configuration
- \$wgCentralIdLookupProviders – Register central ID lookup providers.
- \$wgCentralIdLookupProvider – Default central id lookup implementation to use
- \$wgChangeCredentialsBlacklist – List of AuthenticationRequest class names which are not changeable through Special:ChangeCredentials and the changeauthenticationdata API.
- \$wgDisableAuthManager (removed in 1.28) – Internal feature flag used for the AuthManager rollout process
- \$wgExtendedLoginCookies (removed in 1.28) – Dictates when the extended lifetime login cookies expire.
- \$wgInvalidPasswordReset – Specifies if users should be sent to a password-reset form on login in case of an inadequate password.
- \$wgPasswordConfig – Configuration for built-in password types.
- \$wgPasswordDefault – Default password type to use when hashing user passwords.
- \$wgPasswordResetRoutes – Whether to allow password resets identified by various bits of data.
- \$wgPasswordSalt (removed in 1.34) – For compatibility with old installations set to false.
- \$wgReauthenticateTime – Time frame for re-authentication
- \$wgRememberMe (introduced in 1.36) – Configures RememberMe authentication request added by AuthManager.
- \$wgRemoveCredentialsBlacklist – List of AuthenticationRequest class names which are not removable through Special:RemoveCredentials and the removeauthenticationdata API.
- \$wgSecureLogin – Whether to let user authenticate using https when they come from http.
- \$wgSessionProviders – Session provider configuration.

## User rights, access control and monitoring

---

- \$wgAddGroups – Controls who can add specific groups using Special:Userrights.
- \$wgAutoblockExpiry – Number of seconds before autoblock entries expire
- \$wgBlockAllowsUTEdit – Allow blocked users to edit their talk pages
- \$wgDeleteRevisionsBatchSize (introduced in 1.32) – Page deletions with > this number of revisions will use the job queue.
- \$wgDnsBlacklistUrls – List of DNS blacklists to use, if \$wgEnableDnsBlacklist is true.
- \$wgEmailConfirmToEdit – Require users to confirm email address before they can edit, true to enable.
- \$wgEnableDnsBlacklist – Whether to use DNS blacklists in \$wgDnsBlacklistUrls to check for open proxies.
- \$wgGrantPermissionGroups – Used to define grant groups and assign grants to them.
- \$wgGrantPermissions – Used to define grants and assign permissions to them.
- \$wgGroupPermissions – Use this to set access rights for groups and users in these groups.
- \$wgGroupsAddToSelf – Use this to allow users to add themselves to specified groups via Special:Userrights.
- \$wgGroupsRemoveFromSelf – Use this to allow users to remove themselves from specified groups via Special:Userrights.

- \$wgHideUserContribLimit – The maximum number of edits a user can have and can still be hidden by users with the `hideuser` permission.
- \$wgImplicitGroups – Groups that aren't shown on `Special:Listusers` or somewhere else
- \$wgPasswordAttemptThrottle – Limit password attempts per IP per address.
- \$wgProxyList – A list of IP addresses to be banned
- \$wgProxyWhitelist – Proxy whitelist: list of addresses that are assumed to be non-proxy despite what the other methods might say.
- \$wgRateLimitLog (removed in 1.27) – Set to a filename to log rate limiter hits.
- \$wgRateLimitsExcludedIPs – Array of IPs/CIDR ranges which should never trigger the rate limiter.
- \$wgRateLimits – Simple rate limiter options to brake edit floods.
- \$wgRemoveGroups – Controls who can remove specific groups using `Special:Userrights`.
- \$wgRevokePermissions – Permission keys revoked from users in each group.
- \$wgWhitelistReadRegexp – Whitelists publicly readable titles with regular expressions.
- \$wgWhitelistRead – Pages anonymous user may see.

## Access

Blocking/banning, and some other related settings.

- \$wgAccountCreationThrottle – Number of accounts each IP address may create, 0 to disable.
- \$wgApplyIpBlocksToXff – Apply IP address blocks also to users having a blocked IP in their XFF header.
- \$wgAutoConfirmAge – Number of seconds an account is required to age before it's given the implicit 'autoconfirmed' group membership.
- \$wgAutoConfirmCount – Number of edits an account requires before it is autoconfirmed.
- \$wgAutopromoteOnceLoginRC – Whether to put user rights log entries for autopromotion in recent changes.
- \$wgAutopromoteOnce – Conditions of automatic promotion of user to specific groups that are done only once.
- \$wgAutopromote – Conditions of automatic promotion of user to specific groups
- \$wgAvailableRights – A list of available rights, in addition to the ones defined by the core.
- \$wgBlockCIDRLimit – Maximum CIDR range when blocking an IP range.
- \$wgBlockDisablesLogin – If set to true, blocked users will no longer be allowed to log in
- \$wgBotPasswordsCluster – Cluster name for the `bot_passwords` table
- \$wgBotPasswordsDatabase – Database name for the `bot_passwords` table
- \$wgCascadingRestrictionLevels – Restriction levels that can be used with cascading protection
- \$wgDeleteRevisionsLimit – Integer of how many revisions a page must have before the *bigdelete* right is required to delete it.
- \$wgEnableBotPasswords – Whether to enable bot passwords
- \$wgEnablePartialBlocks (introduced in 1.33) (removed in 1.35) – Allow partial blocks to be created
- \$wgHideIdentifiableRedirects – Protects user's privacy when doing redirects
- \$wgNamespaceProtection – Which namespaces can be edited by whom?
- \$wgNonincludableNamespaces – Use this to deny inclusion of pages from specific namespaces.

- [\\$wgPutIPinRC](#) – Log IP addresses in the recentchanges table.
- [\\$wgQueryPageDefaultLimit](#) – Integer defining default number of entries to show on query pages
- [\\$wgRestrictionLevels](#) – Set of permission keys that can be selected on the "page protection" page.
- [\\$wgRestrictionTypes](#) – Actions that can be restricted
- [\\$wgSemiprotectedRestrictionLevels](#) – Restriction levels that are considered "semi-protected"
- [\\$wgSoftBlockRanges](#) – IP ranges that should be considered soft-blocked (anon-only, account creation allowed)
- [\\$wgSpamRegex](#) – A regular expression pattern which, if the page content matched it, stops a page from being saved.
- [\\$wgSummarySpamRegex](#) – Same as [\\$wgSpamRegex](#) except for edit summaries.
- [\\$wgSysopEmailBans](#) (removed in 1.35) – Allow sysops to ban users from accessing Special:Emailuser.

See [Manual:Preventing access](#) for more methods and settings concerning access. See also [Manual:User rights](#) for more information about [\\$wgGroupPermissions](#), [\\$wgAddGroups](#), [\\$wgRemoveGroups](#), etc.

See [User Access](#) for User Access settings.

## Security

---

- [\\$wgAllowCrossOrigin](#) (introduced in 1.36) – Allow anonymous cross origin requests.
- [\\$wgAllowUserCssPrefs](#) – Allow user preferences implemented in CSS.
- [\\$wgAllowUserCss](#) – Allow user Cascading Style Sheets (CSS)
- [\\$wgAllowUserJs](#) – Allow user JavaScript pages
- [\\$wgApiFrameOptions](#) – Control framing of api pages.
- [\\$wgBreakFrames](#) – Break out of framesets. This can be used to prevent external sites from framing your site with ads.
- [\\$wgCSPFalsePositiveUrls](#) – Controls what URLs to ignore in upcoming Content-Security-Policy feature's reporting.
- [\\$wgCSPHeader](#) (introduced in 1.32) – Controls Content-Security-Policy header [Experimental]
- [\\$wgCSPReportOnlyHeader](#) (introduced in 1.32) – Controls Content-Security-Policy-Report-Only header [Experimental]
- [\\$wgEditPageFrameOptions](#) – Control framing of wiki pages globally.
- [\\$wgHKDFAlgorithm](#) – Algorithm for HMAC-based key derivation function (fast, cryptographically secure random numbers).
- [\\$wgHKDFSecret](#) – Secret for HMAC-based key derivation function (fast, cryptographically secure random numbers).
- [\\$wgRestAllowCrossOriginCookieAuth](#) (introduced in 1.36) – Allows authenticated cross-origin requests to the REST API with session cookies.
- [\\$wgSecretKey](#) – This should always be customized to a secret, unique string in [LocalSettings.php](#).
- [\\$wgSessionInsecureSecrets](#) – Allows MediaWiki to work at the cost of storing sensitive session data insecurely.
- [\\$wgSessionSecret](#) – Secret for session storage.
- [\\$wgUseSiteCss](#) – Use the site's Cascading Style Sheets (CSS)
- [\\$wgUseSiteJs](#) – Use the site's JavaScript page

# Cookies

---

- \$wgCacheVaryCookies – A list of cookies that influence the cache.
- \$wgCookieDomain – Explicit domain for login cookies.
- \$wgCookieExpiration – Number of seconds before HTTP cookies expire. You can override this for 'Keep me logged in (for up to \$1 days)' cookies with \$wgExtendedLoginCookieExpiration
- \$wgCookieHttpOnly – Set the `httpOnly` flag on all cookies set by MediaWiki (to prevent access from JavaScript).
- \$wgCookiePath – Explicitly declared "path" for cookies at the cookie domain
- \$wgCookiePrefix – Custom cookie prefix.
- \$wgCookieSameSite (**introduced in 1.34.3**) – The SameSite cookie attribute used for login cookies.
- \$wgCookieSecure – Whether cookies are secured.
- \$wgCookieSetOnAutoblock – Determines whether to set a cookie when a user is autoblocked. Doing so means that a blocked user, even after logging out and moving to a new IP address, will still be blocked.
- \$wgCookieSetOnIpBlock (**introduced in 1.32**) – Determines whether to set a cookie when an IP user is blocked. Doing so means that a blocked user, even after moving to a new IP address, will still be blocked.
- \$wgDisableCookieCheck – Disables session cookies
- \$wgExtendedLoginCookieExpiration – If non-null, duration in seconds to use for "Keep me logged in (for up to \$1 days)" sessions; 0 means until the browser is closed. If null, uses \$wgCookieExpiration
- \$wgSessionHandler (**removed in 1.33**) – Used to set PHP's `session.save_handler`.
- \$wgSessionName – Override to customize the session name.
- \$wgUseSameSiteLegacyCookies (**introduced in 1.34.3**) – Send a legacy cookie with an `ss0`-prefix when `SameSite=None` is sent.

## Profiling, testing and debugging

---

- \$wgCachePrefix – Custom value for the cache key prefix.
- \$wgEnableJavaScriptTest – Allows running of JavaScript test suites via `Special:JavaScriptTest` (such as QUnit).
- \$wgParserTestFiles (**deprecated in 1.30**) – Parser test suite files to be run by `parserTests.php`.

## Debug

*See also: Manual:How to debug*

- \$wgAggregateStatsID (**removed in 1.27**) – Allows statistics to be aggregated over more than one wiki when using udp logging.
- \$wgDebugComments – Include portions of log output in page HTML comment
- \$wgDebugDumpSqlLength (**removed in 1.27**) – Cutoff length for SQL queries dumped by \$wgDebugDumpSql
- \$wgDebugDumpSql – Dump SQL queries to debug output

- \$wgDebugLogFile – The file name of the debug log, or empty if disabled.
- \$wgDebugLogGroups – Direct \$DebugLog output for certain groups to files other than \$DebugLogFile.
- \$wgDebugLogPrefix – Prefix for debug log lines.
- \$wgDebugPrintHttpHeaders (removed in 1.34) – Print HTTP headers for every request in the debug information.
- \$wgDebugRawPage – If false, 'action=raw' and load.php hits will not result in debug output.
- \$wgDebugRedirects – Displays redirect location instead of actually redirecting to the target URL
- \$wgDebugTimestamps (removed in 1.35) – Prefix debug messages with relative timestamp.
- \$wgDebugToolbar – Display the debugging toolbar
- \$wgDeprecationReleaseLimit – Release limitation to wfDeprecat ed warnings.
- \$wgDevelopmentWarnings – Whether to throw PHP notices for some possible error conditions and for deprecated functions.
- \$wgLogExceptionBacktrace – Whether to send the exception backtrace to the error log.
- \$wgMWLoggerDefaultSpi – Default service provider for creating MWLogger instances.
- \$wgOverrideHostname – Override server hostname detection with a hardcoded value.
- \$wgPageInfoTransclusionLimit – Limit template transclusion on InfoAction
- \$wgPropagateErrors – If true, the MediaWiki error handler passes errors/warnings to the default error handler after logging them.
- \$wgShowDBErrorBacktrace (deprecated in 1.32) – If true, show a backtrace for database errors.
- \$wgShowDebug – Show debug data at bottom of content area
- \$wgShowExceptionDetails – If set to true, uncaught exceptions will print a complete stack trace to output.
- \$wgShowHostnames – If set to true, exposes host names through API and HTML comments.
- \$wgShowSQLErrors (deprecated in 1.32) – Whether to show SQL errors on "we're sorry, but there has been a database error" pages.
- \$wgSpecialVersionShowHooks – Whether to show the contents of \$wgHooks in Special:Version.
- \$wgStatsMethod (removed in 1.27) – Destination for wfIncrStats( ) data
- \$wgTrxProfilerLimits – Performance expectations for DB usage

## Profiling

- \$wgEnableProfileInfo (removed in 1.35) – Controls the availability of profileinfo.php.
- \$wgProfiler – Settings related to profiling.
- \$wgProfileCallTree (removed in 1.27) – If true, print a raw call tree instead of per-function report
- \$wgProfileLimit (removed in 1.35) – Only record profiling info for pages that took longer than this number of seconds to load.
- \$wgProfileOnly (removed in 1.35) – Don't put non-profiling info into log file
- \$wgProfilePerHost (removed in 1.27) – Should the application server hostname be stored in the profiling table?
- \$wgStatsdMetricPrefix – Prefix for metric names sent to \$wgStatsdServer
- \$wgStatsdSamplingRates – Sampling rate for statsd metrics as an associative array of patterns and rates.
- \$wgStatsdServer – A host or host:port of a statsd server.

- \$wgUDPProfilerHost (removed in 1.27) – Host for UDP profiler.
- \$wgUDPProfilerPort (removed in 1.27) – Port for UDP profiler.
- \$wgUDPProfilerFormatString (removed in 1.27) – Format string for UDP profiler.

## Search

---

- \$wgAdvancedSearchHighlighting – Set to true to have more useful text snippets in search results.
- \$wgDisableInternalSearch – Disable internal search so that extensions can implement it.
- \$wgDisableSearchUpdate – If you've disabled search semi-permanently, this also disables updates to the table.
- \$wgDisableTextSearch – Disable full-text search.
- \$wgEnableOpenSearchSuggest (deprecated in 1.35) – Enable OpenSearch suggestions requested by MediaWiki.
- \$wgEnableSearchContributorsByIP – Whether searching for an IP address will be redirected to that IP's contributions page.
- \$wgNamespacesToBeSearchedDefault – Which namespaces should be searched?
- \$wgOpenSearchDefaultLimit – Integer defining default number of entries to show on OpenSearch call.
- \$wgOpenSearchDescriptionLength – Minimum length of extract in **<Description>**.
- \$wgOpenSearchTemplates – Templates for OpenSearch suggestions.
- \$wgOpenSearchTemplate (deprecated in 1.25) – Template for OpenSearch suggestions.
- \$wgSearchForwardUrl – Set this to a URL to forward search requests to some external location.
- \$wgSearchHighlightBoundaries – Regexp to match word boundaries
- \$wgSearchMatchRedirectPreference (introduced in 1.35) – Set true to allow logged-in users to set a preference whether or not matches in search results should force redirection to that page.
- \$wgSearchRunSuggestedQuery – If search returns zero results and current search engine has a "did you mean" suggestion, results for the suggestion will be shown.
- \$wgSearchSuggestCacheExpiry – Expiry time for search suggestion responses.
- \$wgSitemapNamespacesPriorities – Custom namespace priorities for sitemaps.
- \$wgSitemapNamespaces – Array of namespaces to generate a sitemap or false for all namespaces.
- \$wgSpecialSearchFormOptions (introduced in 1.34) – Options for Special:Search completion widget form created by SearchFormWidget class.
- \$wgUseTwoButtonsSearchForm (removed in 1.36) – Sidebar's search form behavior.

## Edit user interface

---

- \$wgDiff3 – Path to the GNU diff3 utility.
- \$wgDiff – Path to the GNU diff utility.
- \$wgPreviewOnOpenNamespaces – Namespaces where preview-on-first-edit is forced.
- \$wgUniversalEditButton – Enable the UniversalEditButton for browsers that support it.
- \$wgUseAutomaticEditSummaries – Whether MediaWiki should make automatic edit summaries.



## Maintenance Scripts setting

---

- [\\$wgCommandLineDarkBg](#) – Assume a dark terminal background for maintenance scripts.
- [\\$wgCommandLineMode](#) – Indicates whether MW is running in command-line mode. Should never be set manually!
- [\\$wgGitBin](#) – Path to the git binary.
- [\\$wgGitRepositoryViewers](#) – Map of Git repository URLs to their gitweb interface URLs
- [\\$wgReadOnlyFile](#) – If this file is found, the database is locked for the reason given in the contents of the file.
- [\\$wgReadOnlyWatchedItemStore](#) – Set this to true to put the wiki watchlists into read-only mode.
- [\\$wgReadOnly](#) – Disallows editing, displaying the string given as the reason.
- [\\$wgUpgradeKey](#) – Password that controls access to web-based updater.

## Recent changes, new pages, watchlist and history

---

See also [Help:Recent changes](#) and [\\$wgDefaultUserOptions](#)

- [\\$wgAllowCategorizedRecentChanges](#) **(removed in 1.31)** – Allow to filter the recentchanges by a category or one of its sub(subsubsub...)categories
- [\\$wgDisableAnonTalk](#) – Disable talk pages for anonymous users (not logged in).
- [\\$wgLogAutopatrol](#) **(removed in 1.31)** – Log autopatrol actions to the log table
- [\\$wgRCChangedSizeThreshold](#) – Highlight character count difference lower than this.
- [\\$wgRCEngines](#) – Configuration for RC feed protocol engines
- [\\$wgRCFeeds](#) – Configuration for sending RC updates
- [\\$wgRCFilterByAge](#) – Filter [\\$wgRCLinkDays](#) by [\\$wgRCMaxAge](#) to avoid showing links for numbers higher than what will be stored.
- [\\$wgRCLinkDays](#) – List days options to list in the [Special:Recentchanges](#) and [Special:Recentchangeslinked](#) pages.
- [\\$wgRCLinkLimits](#) – List changes limits options to list in the [Special:Recentchanges](#) and [Special:Recentchangeslinked](#) pages.
- [\\$wgRCMaxAge](#) – Items in the [recentchanges](#) table are periodically purged; entries older than this many seconds will go.
- [\\$wgRCShowChangedSize](#) – Show the amount of changed characters in recent changes.
- [\\$wgRCShowWatchingUsers](#) – Show watching users in recent changes, watchlist and page history views.
- [\\$wgRCWatchCategoryMembership](#) – Treat category membership changes as a [RecentChange](#).
- [\\$wgRecentChangesFlags](#) – Flags (letter symbols) shown in recent changes and watchlist to indicate certain types of edits.
- [\\$wgShowUpdatedMarker](#) – Show new changes: bold in watchlist, with "updated since my last visit" in history.
- [\\$wgSoftwareTags](#) – List of core tags to enable.
- [\\$wgStructuredChangeFiltersLiveUpdatePollingRate](#) – Polling rate, in seconds, used by the 'live update' feature of [Special:RecentChanges](#) and [Special:Watchlist](#)
- [\\$wgUnwatchedPageSecret](#) – If active watchers (per [\\$wgWatchersMaxAge](#)) are this number or less, do not disclose it.



- \$wgUnwatchedPageThreshold – Allows users without the "unwatchedpages" user right to view the number of page watchers for a specified page via the "info" action if the number of watchers is above the specified threshold.
- \$wgUseFilePatrol – Use file patrolling to check new files on Special:NewFiles
- \$wgUseNPPatrol – Use new pages patrolling to check for vandalism
- \$wgUseRCPatrol – Use RC patrolling to check for vandalism
- \$wgUseTagFilter – Allow filtering by change tag in recentchanges, history, etc.
- \$wgWatchersMaxAge – Page watchers inactive for more than this many seconds are considered inactive. Used mainly by action=info.
- \$wgWatchlistExpiryMaxDuration (introduced in 1.35) – Relative maximum duration for watchlist expiries, as accepted by strtotime().
- \$wgWatchlistExpiry (introduced in 1.35) – Enable the watchlist expiry feature.
- \$wgWatchlistPurgeRate (introduced in 1.35) – The per-edit rate at which expired watchlist items will be purged from the database.

## Feed

- \$wgFeed – Toggle syndication feeds (RSS, Atom e.g. Recentchanges, Newpages) on and off.
- \$wgAdvertisedFeedTypes – Feed types that should be provided by default.
- \$wgFeedCacheTimeout – Minimum timeout for cached Recentchanges feed, in seconds.
- \$wgFeedClasses – Available feeds objects.
- \$wgFeedDiffCutoff – When generating Recentchanges RSS/Atom feed, diffs will not be generated for pages larger than this size.
- \$wgFeedLimit – Set maximum number of results to return in syndication feeds (RSS, Atom) for e.g. Recentchanges, Newpages.
- \$wgOverrideSiteFeed – Override the site's default rss/atom feed that appears on every page.

## Copyright

---

- \$wgCopyrightIcon (removed in 1.27) – Set this to some HTML to override the rights icon with an arbitrary logo.
- \$wgMaxCredits – Set this to the number of authors that you want to be credited below an article text.
- \$wgRightsIcon – Override for copyright metadata.
- \$wgRightsPage – Override for copyright metadata.
- \$wgRightsText – Override for copyright metadata.
- \$wgRightsUrl – Override for copyright metadata.
- \$wgShowCreditsIfMax – If there are more than \$wgMaxCredits authors, show \$wgMaxCredits of them.
- \$wgUseCopyrightUpload – Set this to true if you want detailed copyright information forms on Upload.

## Import/Export

---

- \$wgExportAllowAll – Whether to allow exporting the entire wiki into a single file.

- \$wgExportAllowHistory – Enable the full-history option on Special:Export.
- \$wgExportAllowListContributors – Whether to allow to list all contributors while exporting a page.
- \$wgExportFromNamespaces – Whether to allow the "export all pages in namespace" option.
- \$wgExportMaxHistory – Special:Export requests for history of pages with more revisions than this will be rejected.
- \$wgExportMaxLinkDepth – Maximum value of the "pagelink-depth" parameter in Special:Export.
- \$wgExportPagelistLimit – Maximum number of pages returned by the GetPagesFromCategory and GetPagesFromNamespace functions.
- \$wgImportSources – List of interwiki prefixes for wikis we'll accept as sources for Special:Import
- \$wgImportTargetNamespace – Optional default target namespace for interwiki imports.
- \$wgXmlDumpSchemaVersion (introduced in 1.33) – The schema to use per default when generating XML dumps.

## Extensions

---

- \$wgAuth (removed in 1.33) – Authentication plugin. See AuthPlugin.
- \$wgAutoloadAttemptLowercase (deprecated in 1.35) – Switch controlling legacy case-insensitive classloading.
- \$wgAutoloadClasses – Array mapping class names to filenames, for autoloading.
- \$wgEnableParserLimitReporting – Controls whether the NewPP parser report is output in a HTML comment.
- \$wgExceptionHooks (removed in 1.30) – Hooks that are used for outputting exceptions.
- \$wgExtensionCredits (deprecated in 1.25) – Global list of extension credits
- \$wgExtensionEntryPointListFiles – Array of files with list(s) of extension entry points to be used in maintenance/mergeMessageFileList.php
- \$wgExtensionFunctions – A list of callback functions which are called once MediaWiki is fully initialised.
- \$wgExtensionMessagesFiles – Extension messages files.
- \$wgHooks – Global list of hooks.
- \$wgJobBackoffThrottling – Number of times work items of a job type can be run per second.
- \$wgJobClasses – Maps jobs to their handling classes.
- \$wgJobQueueIncludeInMaxLagFactor – Whether to include the number of jobs that are queued for the API's maxlag parameter.
- \$wgJobSerialCommitThreshold – Make job runners commit changes for replica DB-lag prone jobs one job at a time.
- \$wgJobTypeConf – Configuration for job storage.
- \$wgJobTypesExcludedFromDefaultQueue – Jobs that must be explicitly requested, i.e. aren't run by job runners unless special flags are set.
- \$wgMessagesDirs – Extension messages directories.
- \$wgPagePropLinkInvalidations – Page property link table invalidation lists.
- \$wgParserOutputHooks – Parser output hooks.
- \$wgServiceWiringFiles – List of service wiring files to be loaded by the default instance of MediaWikiServices.

- [\\$wgSpecialPageCacheUpdates](#) – Additional functions to be performed with [\\$updateSpecialPages](#).
- [\\$wgSpecialPages](#) – A list of custom special pages.
- [\\$wgValidSkinNames](#) – List of valid skin names. Skins can optionally use this variable to tell MediaWiki that they are available.

## Category

---

- [\\$wgCategoryCollation](#) – What collation categories use to sort with
- [\\$wgCategoryMagicGallery](#) – On category pages, show thumbnail gallery for images belonging to that category instead of listing them as articles.
- [\\$wgCategoryPagingLimit](#) – Paging limit for items in categories.
- [\\$wgUseCategoryBrowser](#) – Enable/Disable experimental breadcrumb (or dmoz-style) category browsing.
- [\\$wgTrackingCategories](#) (**deprecated in 1.25**) – The list of messages representing tracking categories which are listed at [Special:TrackingCategories](#).

## Logging

---

- [\\$wgActionFilteredLogs](#) – List of log types that can be filtered by action types
- [\\$wgFilterLogTypes](#) – Show/hide links on [Special:Log](#) will be shown for these log types.
- [\\$wgLogActionsHandlers](#) – Same as [\\$wgLogActions](#), but values are functions callbacks.
- [\\$wgLogActions](#) – Lists the message key string for formatting individual events of each type and action when listed in the logs.
- [\\$wgLogHeaders](#) – Lists the message key string for descriptive text to be shown at the top of each log type.
- [\\$wgLogNames](#) – Lists the message key string for each log type.
- [\\$wgLogRestrictions](#) – Restricts log access to those who have a certain right.
- [\\$wgLogTypes](#) – List of log types.
- [\\$wgNewUserLog](#) – Whether to maintain a log of newusers, which can be viewed at [Special:Log/newusers](#).
- [\\$wgPageCreationLog](#) (**introduced in 1.32**) – Whether to maintain a log of new page creations, which can be viewed at [Special:Log/create](#).

## Special pages

---

- [\\$wgAllowSpecialInclusion](#) – Allow special page inclusions such as `{{Special:Allpages}}`.
- [\\$wgCountCategorizedImagesAsUsed](#) – On [Special:Unusedimages](#), consider images "used" if they are put into a category.
- [\\$wgDisableQueryPageUpdate](#) – Array of special pages that should not be updated.
- [\\$wgMaxRedirectLinksRetrieved](#) – Maximum number of links to a redirect page listed on [Special:Whatlinkshere/RedirectDestination](#)
- [\\$wgRangeContributionsCIDRLimit](#) – Shortest [CIDR limits](#) that can be checked in any individual range check at [Special:Contributions](#).

## Actions

---

- \$wgActions – Array of allowed values for the "action" parameter for normal pages.

## Robot policies

---

- \$wgArticleRobotPolicies – Allow customisation of robot policies on a per-page basis.
- \$wgDefaultRobotPolicy – Allows specifying the default robot policy for all pages on the wiki
- \$wgExemptFromUserRobotsControl – An array of namespace keys in which the `__INDEX__`/`__NOINDEX__` magic words will not function.
- \$wgNamespaceRobotPolicies – Allow customisation of robot policies on a per-namespace basis.

## Ajax, Action API and REST API

---

### Ajax

- \$wgAjaxEditStash (**deprecated in 1.36**) – Whether edits should be prepared via AJAX as users type edit summaries.
- \$wgAjaxExportList (**deprecated in 1.27**) – List of functions in `AjaxFunctions.php` that are callable via `action=ajax`
- \$wgAjaxLicensePreview – Use Ajax to preview licensing for uploads.
- \$wgAjaxUploadDestCheck – Enable Ajax check for file overwrite, pre-upload.
- \$wgAjaxWatch (**removed in 1.27**) – Enable watching/unwatching pages using Ajax.
- \$wgRestAPIAdditionalRouteFiles (**introduced in 1.35**) – Additional REST API Route files.
- \$wgUseAjax (**deprecated in 1.31**) – Enable AJAX support.

### API

- \$wgAllowedCorsHeaders (**introduced in 1.35**) – List of allowed headers for cross-origin API requests.
- \$wgAPICacheHelpTimeout – Set the timeout for the API help text cache. Set to 0 to disable caching of API help text.
- \$wgAPIFormatModules – API format module extensions.
- \$wgAPIListModules – API list module extensions.
- \$wgAPIMaxDBRows – Maximum number of rows to scan in API database queries.
- \$wgAPIMaxLagThreshold – Maximum amount of DB lag on a majority of DB replica DBs to tolerate before forcing bots to retry any write requests via API errors.
- \$wgAPIMaxResultSize – Maximum size (in bytes) of an API result.
- \$wgAPIMaxUncachedDiffs – The maximum number of uncached diffs that can be retrieved in one API request.
- \$wgAPIMetaModules – API meta module extensions.
- \$wgAPIModules – API module extensions.

- \$wgAPIPropModules – API prop module extensions.
- \$wgAPIRequestLog – Log file or URL (TCP or UDP) to log API requests to.
- \$wgAPIUselessQueryPages – The ApiQueryQueryPages module should skip pages that are redundant to true API queries.
- \$wgCrossSiteAJAXdomainExceptions – Domains that should not be allowed to make AJAX requests, even if they match one of the domains allowed by \$wgCrossSiteAJAXdomains.
- \$wgCrossSiteAJAXdomains – Domains that may make cross-site Ajax requests to the MediaWiki API.
- \$wgDebugAPI – Loosens API security to allow for easier debugging
- \$wgEnableAPI (removed in 1.32) – Enable direct access to the data API through api.php.
- \$wgEnableRestAPI (introduced in 1.34) (removed in 1.36) – Enable the MediaWiki REST API.
- \$wgEnableWriteAPI (removed in 1.32) – Allow the API to be used to perform write operations (page edits, rollback, etc.) when an authorized user accesses it.

## Shell and process control

---

- \$wgPhpCli – Executable name of PHP cli client (php/php5).
- \$wgMaxShellFileSize – Maximum file size created by shell processes under linux, in KB.
- \$wgMaxShellMemory – Maximum amount of virtual memory available to shell processes under Linux, in KiB.
- \$wgMaxShellTime – Maximum CPU time in seconds for shell processes under Linux.
- \$wgMaxShellWallClockTime – Limit shell commands duration using real time.
- \$wgShellboxSecretKey (introduced in 1.36) – The secret key for HMAC verification of Shellbox requests.
- \$wgShellboxUrl (introduced in 1.36) – Shell commands can be run on a remote server using Shellbox.
- \$wgShellCgroup – Limit shell commands memory using linux Cgroups.
- \$wgShellLocale – Locale for LC\_ALL.
- \$wgShellRestrictionMethod – Method to use to restrict shell commands

## HTTP client

---

- \$wgAllowExternalReqID (introduced in 1.34) – controls whether MediaWiki accepts the request ID set by the incoming request via the "X-Request-Id" header.
- \$wgAsyncHTTPTimeout – Timeout for asynchronous HTTP requests.
- \$wgHTTPConnectTimeout – Timeout for connections done internally (in seconds)
- \$wgHTTPMaxTimeout (introduced in 1.35) – The maximum HTTP request timeout in seconds.
- \$wgHTTPMaxConnectTimeout (introduced in 1.35) – The maximum HTTP connect timeout in seconds.
- \$wgHTTPImportTimeout – Timeout for HTTP requests done internally for transwiki imports, in seconds.
- \$wgHTTPProxy – Proxy to use for cURL requests
- \$wgHTTPTimeout – Timeout for HTTP requests done via cURL.
- \$wgLocalVirtualHosts – Local virtual hosts.

## Jobs

---

- \$wgJobRunRate – Number of jobs to perform per request.
- \$wgJobQueueAggregator (removed in 1.33) – Which aggregator to use for tracking which queues have jobs.
- \$wgRunJobsAsync – Controls whether on-request jobs are run asynchronously or synchronously.
- \$wgUpdateRowsPerJob – Number of rows to update per job.
- \$wgUpdateRowsPerQuery – Number of rows to update per query.

## Site customization

---

- \$wgStructuredChangeFiltersOnWatchlist (removed in 1.32) – Whether to enable RCFilters app on Special:Watchlist
- \$wgStructuredChangeFiltersShowPreference (removed in 1.32) – Whether to enable RCFilters app on Special:RecentChanges
- \$wgStyleVersion (removed in 1.32) – Appended in the query string of static CSS and JS to clear client side cache when changing this.

## Frontend

- \$wgOOUIEditPage (removed in 1.30) – Whether to use OOjs UI buttons, checkboxes and summary input in the edit form.
- \$wgPreloadJavaScriptMwUtil (removed in 1.27) – Whether to preload the mediawiki.util module as blocking module in the top queue.
- \$wgUsejQueryThree (removed in 1.31) – Whether to use jQuery 3 (with jQuery Migrate) instead of jQuery 1.

## Miscellaneous settings

---

- \$wgActorTableSchemaMigrationStage (removed in 1.34) – actor table schema migration stage.
- \$wgChangeTagsSchemaMigrationStage (introduced in 1.32) (removed in 1.33) – change\_tag table schema migration stage.
- \$wgCommentTableSchemaMigrationStage (removed in 1.33) – comment table schema migration stage.
- \$wgContentHandlerUseDB (removed in 1.35) – Whether to use the database fields introduced by the ContentHandler facility
- \$wgElementTiming (introduced in 1.33) – Enable Element Timing.
- \$wgEventRelayerConfig – Mapping of event channels (or channel categories) to EventRelayer configuration.
- \$wgFeaturePolicyReportOnly (introduced in 1.34) – List of Feature Policy Reporting types to enable.
- \$wgOriginTrials (introduced in 1.33) – Origin Trials tokens.
- \$wgPagePropsHaveSortkey (removed in 1.36) – Set to false if the page\_props table doesn't have a pp\_sortkey column

- [\\$wgPingback](#) – If true, MediaWiki will report back data about its installation from time to time to aid developers
- [\\$wgPopularPasswordFile](#) **(removed in 1.35)** – Where popular password file is located.
- [\\$wgPriorityHintsRatio](#) **(introduced in 1.34)** – Ratio of requests that should get Priority Hints when the feature is enabled.
- [\\$wgPriorityHints](#) **(introduced in 1.33)** – Enable client-side Priority Hints.
- [\\$wgRedirectOnLogin](#) – Redirect users to page on login
- [\\$wgReportToEndpoints](#) **(introduced in 1.34)** – List of endpoints for the Reporting API.
- [\\$wgReportToExpiry](#) **(introduced in 1.34)** – Expiry of the endpoint definition for the Reporting API.
- [\\$wgTagStatisticsNewTable](#) **(introduced in 1.32) (removed in 1.33)** – Temporarily flag to use `change_tag_def` table as backend of change tag statistics.
- [\\$wgVirtualRestConfig](#) – REST services configuration
- [\\$wgWikiDiff2MovedParagraphDetectionCutoff](#) **(deprecated in 1.34)** – This setting controls the maximum number of paragraphs to compare before it bails out.

## See also

---

- [Manual:Configuration for developers](#) - how to set and retrieve configuration variables.
  - [Extension registration#Configs \(Your extension/skins settings\)](#) - how to register new configuration variables for extensions and skins.
- 

Retrieved from "[https://www.mediawiki.org/w/index.php?title=Manual:Configuration\\_settings&oldid=4571083](https://www.mediawiki.org/w/index.php?title=Manual:Configuration_settings&oldid=4571083)"

---

This page was last edited on 10 May 2021, at 20:21.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.