



Manual:Configuration settings

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

Never edit `MainConfigSchema.php`; add appropriate lines to `LocalSettings.php` instead and amend them as appropriate.

If you cannot find a configuration setting here, see if it is defined in `MainConfigSchema.php`. The variable should have some documentation there.



Settings that were removed in MediaWiki 1.39 or older versions are no longer listed here. Please use the [search box](#) to find the relevant manual page (e.g. `Manual:$wgAllowAnonymousMinor`).

[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

- `$wgAutoloadLocalClasses` – Array mapping class names to filenames, for autoloading.
- `$wgConf` – Create a site configuration object.

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.
- `$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.

- \$wgBaseDirectory (removed in 1.44) – Absolute filesystem path of the root directory of the MediaWiki installation.
- \$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 – Redirect insecure HTTP requests to HTTPS.
- \$wgHttpsPort – Port where you have HTTPS running. Supports HTTPS on non-standard ports.
- \$wgImgAuthPath – 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 (removed in 1.42) – URLs for HD versions of the site logo.
- \$wgLogos – The URLs or file paths to various wiki logos.
- \$wgLogo – The URL of the site logo.
- \$wgMainPageIsDomainRoot – 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/>) meta tag.
- \$wgRestPath – The URL path to `rest.php`.
- \$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.
- \$wgStyleDirectory – The file system path to the skins directory.
- \$wgStylePath – The URL of the skins directory.
- \$wgThumbPath – 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.

- [\\$wgAllowCopyUploads](#) – Allow uploads from URLs as well as local files.
- [\\$wgAttemptFailureEpoch](#) – Bump this to quickly retry thumbnail creation failures.
- [\\$wgCheckFileExtensions](#) – Check if the file extension is allowed on upload.
- [\\$wgCopyUploadAllowOnWikiDomainConfig](#) – If **true**, the value of [\\$wgCopyUploadsDomains](#) will be merged with the contents of [MediaWiki:Copyupload-allowed-domains](#).
- [\\$wgCopyUploadProxy](#) – HTTP proxy to use for copy upload requests.
- [\\$wgCopyUploadsDomains](#) – Limit [\\$wgAllowCopyUploads](#) to a specific list of domains
- [\\$wgCopyUploadsFromSpecialUpload](#) – Add an interface to Special:Upload to enable uploading from URLs
- [\\$wgCopyUploadTimeout](#) – HTTP timeout (in seconds) for upload by URL
- [\\$wgCSPUploadEntryPoint](#) (**introduced in 1.45**) – Controls whether `thumb.php` and `img_auth.php` send CSP headers
- [\\$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.
- [\\$wgEnableAsyncUploadsByURL](#) – Enable the async processing of upload by url in Special:Upload.
- [\\$wgEnableUploads](#) – Allow images and other files to be uploaded through the wiki.
- [\\$wgFileBackends](#) – File backend structure configuration.
- [\\$wgProhibitedFileExtensions](#) – 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](#) (**deprecated in 1.41**) – Override the default list of illegal characters in file names.

- \$wgImagePreconnect – Add a preconnect link for browsers to a remote FileRepo host..
- \$wgImgAuthDetails – Display reason access to uploaded file was denied to users (img_auth only)
- \$wgImgAuthUrlPathMap – Map of relative URL directories to match to internal `mwtore://` 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 – Whether translatable media (currently, only SVG images) should be shown in page's target language
- \$wgMinUploadChunkSize – Minimum upload chunk size, in bytes.
- \$wgNativeImageLazyLoading – Allows to toggle native image lazy loading, via the "loading" attribute in HTML.
- \$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 \$wgFileExtensions.
- \$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

- \$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.
- \$wgVerifyMimeTypeIE (**removed in 1.40**) – Determines whether extra checks for Internet Explorer type detection should be applied.
- \$wgMimeTypeExclusions – 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

- \$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 **** 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 – 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.
- \$wgDjvuTxt – Path of the djvutxt DjVu text extraction utility.
- \$wgDjvuUseBoxedCommand – Whether to use BoxedCommand or not.
- \$wgShellboxShell – The POSIX-compatible shell to use when running scripts.

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

SVG

- \$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.
- \$wgSVGNativeRendering – Whether native rendering by the browser agent is allowed.
- \$wgSVGNativeRenderingSizeLimit – Limit on an SVG file's size for it to be rendered by the client, in bytes.

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.
- \$wgThumbnailNamespaces – Defines what namespaces thumbnails will be displayed for in Special:Search.
- \$wgThumbnailScriptPath – Path to thumb.php, if you want thumbnail generation on the fly.
- \$wgThumbnailSteps (introduced in 1.44) – When defined, is an array of image widths used as steps for thumbnail sizes.
- \$wgThumbnailStepsRatio (introduced in 1.44) – Ratio of images that will use the thumbnail steps.
- \$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 (removed in 1.46) – Set to **true** to enable the Special:Mute page.
- \$wgEnableUserEmailMuteList (removed in 1.46) – Set to **true** to enable user-to-user email mutelist.
- \$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.
- \$wgSMTP – For using a direct (authenticated) SMTP server connection.
- \$wgUserEmailConfirmationTokenExpiry – The time, in seconds, after which a email

address confirmation token expires.

- \$wgUserEmailConfirmationUseHTML (introduced in 1.46) – Whether the modern email confirmation message is used
- \$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 (removed in 1.45) – Send a generic mail instead of a personalised mail for each user.
- \$wgEnotifMaxRecips (removed in 1.45) – 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
- \$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
- \$wgUsersNotifiedOnAllChanges – Array of usernames who will be sent a notification email for every change which occurs on a wiki.

Database settings

- \$wgCategoryLinksSchemaMigrationStage (introduced in 1.44) (removed in 1.45) – Migration stage for categorylinks tables
- \$wgCommentTempTableSchemaMigrationStage (removed in 1.41) – Comment temp tables schema migration stage.
- \$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 (removed in 1.45) – 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.
- \$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.
- \$wgDBstrictWarnings – Check for warnings after DB queries and throw an exception if an unacceptable warning is detected.
- \$wgDBtype – Type of database.
- \$wgDBuser – Database username.
- \$wgImageLinksSchemaMigrationStage (introduced in 1.46) – Migration stage for `imagelinks` tables.
- \$wgExternalLinksSchemaMigrationStage (removed in 1.41) – `externallinks` table schema migration stage.
- \$wgExternalLinksDomainGaps – Gaps in the `externallinks` table for certain domains.
- \$wgFileSchemaMigrationStage (introduced in 1.44) – Migration stage for file tables.
- \$wgLocalDatabases – Other wikis on this site, can be administered from a single developer account.
- \$wgMaxExecutionTimeForExpensiveQueries – Max execution time for queries of several expensive special pages such as `RecentChanges` in milliseconds.
- \$wgPageLinksSchemaMigrationStage (removed in 1.45) – The `pagelinks` table schema migration stage, for normalising `pl_namespace` and `pl_title` fields.
- \$wgSearchTypeAlternatives – Alternative search types.
- \$wgSearchType – Type of search used for the wiki.
- \$wgTemplateLinksSchemaMigrationStage (removed in 1.43) – `templatelinks` table schema migration stage, for normalizing `tl_namespace` and `tl_title` fields.
- \$wgVirtualDomainsMapping – Mapping of virtual database domains to actual databases.

LoadBalancer settings

- \$wgDBservers – Settings for the database load balancer (database servers and load ratios).
- \$wgLBFactoryConf – Configuration for the `ILBFactory` service
- \$wgMasterWaitTimeout (deprecated in 1.24) – How long to wait for a slave to catch up to the master.
- \$wgDatabaseReplicaLagCritical – If lag is higher than this value, show a more obvious

warning than `$wgDatabaseReplicaLagWarning`.

- `$wgDatabaseReplicaLagWarning` – If lag is higher than this value, show a warning in some special pages (like watchlist).

MySQL-specific

- `$wgDBTableOptions` – MySQL table options to use during installation or update.
- `$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

Content handlers and storage

- `$wgCompressRevisions` – Compress new page revisions if possible
- `$wgContentHandlers` – Mapping of content type handlers
- `$wgContentHandlerTextFallback` (**removed in 1.42**) – How to react if a plain text version of a non-text `Content` object is requested using `ContentHandler::getContentText()`
- `$wgDefaultExternalStore` – Default external store for new revisions.
- `$wgDiffEngine` – Specify the difference engine to use.
- `$wgEnableRemoteBagOStuffTests` – If this is set to **true**, phpunit will run integration tests against remote caches defined in `$wgObjectCaches`.
- `$wgExternalDiffEngine` – Name of the external diff engine to use.
- `$wgExternalServers` – Shortcut for setting `$wgLBFactoryConf["externalClusters"]`.
- `$wgExternalStores` – List of enabled `ExternalStore` protocols.

- \$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.
- \$wgRevisionSlotsCacheExpiry (removed in 1.45) – Revision slots may be cached in the main WAN cache and/or the local server cache to reduce load on the database.
- \$wgTextModelsToParse – Determines which types of text are parsed as wikitext.
- \$wgWikidiff2Options – Options for wikidiff2

Performance hacks and limits

- \$wgAllowSchemaUpdates – Allow database schema updates to be executed with update.php script.
- \$wgAllowSlowParserFunctions – Allow slow parser functions.
- \$wgCriticalSectionTimeLimit – The maximum time critical sections are allowed to stay open.
- \$wgDisableQueryPages – Disable all query pages if miser mode is on, not just some.
- \$wgForceDeferredUpdatesPreSend – Force deferred updates to be run before sending a response to the client, instead of attempting to run them after sending the response.
- \$wgLinkHolderBatchSize – LinkHolderArray batch size.
- \$wgMaxArticleSize – Maximum page size in kibibytes.
- \$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
- \$wgMultiShardSiteStats – Whether `site_stats` table should have multiple rows.
- \$wgPoolCounterConf – Configuration for processing pool control.
- \$wgPoolCountClientConf – Configuration array for the PoolCounter client.
- \$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 – 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 (**removed in 1.43**) – The object store type of the ChronologyProtector position store.
- \$wgChronologyProtectorSecret – Secret string for HMAC hashing in ChronologyProtector (optional)
- \$wgExtensionInfoMTime – Time to invalidate cache of extensions loaded by the extension registration system.
- \$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
- \$wgMicroStashType – Central store for reliable short-lived data.
- \$wgObjectCaches – Advanced object cache configuration.
- \$wgParserCacheFilterConfig – Per-namespace configuration for the ParserCache filter.
- \$wgParsoidCacheConfig – <translate> Configuration for the caching related to parsoid output.</translate>
- \$wgPHPSessionHandling (**deprecated in 1.45**) – Whether to use PHP session handling (`$_SESSION` and `session_*()` functions)
- \$wgSessionPbkdf2Iterations – Number of internal PBKDF2 iterations to use when deriving session secrets.
- \$wgStatsCacheType (**removed in 1.42**) – The object cache type for stats storage.
- \$wgSuspiciousIpExpiry – Time in seconds to remember IPs for, for the purposes of logging IP changes within the same session.
- \$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

- \$wgOldRevisionParserCacheExpireTime – 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.
- \$wgParserCacheAsyncExpireTime (introduced in 1.44) – The expiry time for "not ready" asynchronous content in the parser cache, in seconds.
- \$wgParserCacheAsyncRefreshJobs (introduced in 1.44) – Whether to re-run the refresh links jobs when asynchronous content becomes ready.
- \$wgParserCacheExpireTime – Expiration time (in seconds) of cached parser information
- \$wgParserCacheType – Parser cache setting
- \$wgUseContentMediaStyles (removed in 1.45) – Temporary flag to stop shipping the styles for the legacy media HTML structure that has been replaced when \$wgParserEnableLegacyMediaDOM is **false**.

Memcached settings

Settings for configuring the Memcached memory-based object store (if you are using it) docs/memcached.md has more details.

- \$wgAnonSessionCacheType (introduced in 1.45) – The cache backend for storing anonymous session data.
- \$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 in seconds.
- \$wgSessionCacheType – The cache type for storing session data.
- \$wgUseSessionCookieJwt (introduced in 1.45) – Include a JWT alongside the session cookie when using cookie-based sessions.
- \$wgSessionCookieJwtExpiration (introduced in 1.45) – JWT session cookie expiration, in seconds. Used both for cookie expiry and for the expiry field inside the JWT object.

WAN cache

- \$wgMainWANCache (removed in 1.40) – Main wide area network cache configuration.
- \$wgWANObjectCaches (removed in 1.40) – Advanced Wide-Area-Network cache configuration.
- \$wgWANObjectCache – Extra parameters to the WANObjectCache constructor.

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.

- \$wgCdnMatchParameterOrder – Indicates if should take query parameters order on incoming requests into account for CDN caching
- \$wgCdnMaxageLagged – Cache timeout for the CDN when DB replica lag is high
- \$wgCdnMaxageStale – 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 – Cache TTL for the CDN sent as s-maxage (without ESI) or Surrogate-Control (with ESI).
- \$wgCdnReboundPurgeDelay – Secondary delayed purges of URLs
- \$wgCdnServersNoPurge – Same as \$wgCdnServers, except no HTTP purges will be sent and CIDR ranges are accepted
- \$wgCdnServers – 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.
- \$wgSquidPurgeUseHostHeader (removed in 1.42) – Whether to use a Host header in purge requests sent to the proxy servers configured in \$wgSquidServers.
- \$wgUseCdn – Enable/disable page view caching.
- \$wgUseESI (deprecated in 1.33) – If you run Squid3 with ESI support, enable this.
- \$wgUsePrivateIPs – Whether private X-Forwarded-For IPs should be accepted
- \$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

- \$wgAllowRawHtmlCopyrightMessages (removed in 1.45) – Whether on-wiki overrides for the `copyright` and `history_copyright` messages, which allow raw HTML, will be used.
- \$wgAllUnicodeFixes – Whether to always convert certain Unicode sequences to modern ones regardless of the content language.
- \$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 – List of language codes that have been renamed to new (correct) codes, or don't correspond to an actual interface language.
- \$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`.
- \$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 – 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.
- \$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).
- \$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.
- \$wgOverrideUcfirstCharacters – List of Unicode characters for which capitalisation is overridden in Language::ucfirst.
- \$wgRawHtmlMessages – 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.
- \$wgUseXssLanguage – Whether to enable the x-xss language code, used to make checking for XSS issues more convenient.
- \$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

- \$wgBrowserFormatDetection – Override ability of certain browsers to attempt to autodetect dataformats in pages.
- \$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.
- \$wgExternalInterwikiFragmentMode – Determines which ID escaping mode should be used for external interwiki links
- \$wgFragmentMode – Determines how section IDs should be encoded
- \$wgInterwikiLogoOverride – List of interwiki logos overrides.
- \$wgHTMLFormAllowTableFormat (removed in 1.42) – Allow form components to render tables.
- \$wgHtml5Version – Defines the value of the version attribute in the `<html>` tag, if any.
- \$wgMangleFlashPolicy (removed in 1.40) – Whether to mangle any `<cross-domain-policy>` (Adobe cross-domain policy) tags, to prevent XSS attacks.
- \$wgMimeType – The type to send in the Content-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.
- \$wgUseMediaWikiUIEverywhere (removed in 1.42) – Applies MediaWiki UI wherever it can be supported.
- \$wgXhtmlNamespaces – Permit other XHTML namespaces in addition to the w3.org default.

Skins

- \$wgDefaultSkin – The default skin for new users.
- \$wgFallbackSkin – Fallback skin used when the skin defined by \$wgDefaultSkin can't be found.
- \$wgFooterIcons – Abstract list of footer icons for skins.
- \$wgSkinMetaTags – 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.
- \$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
- [\\$wgCodexDevelopmentDir](#) – If this is set, MediaWiki will look for Codex files in this directory instead of in `resources/lib/codex/` and friends.
- [\\$wgResourceBasePath](#) – Default `remoteBasePath` value for resource loader modules.
- [\\$wgResourceLoaderClientPreferences](#) (removed in 1.41) – Enable client-side preferences for unregistered users.
- [\\$wgResourceLoaderDebug](#) – The default debug mode (on/off) for ResourceLoader requests.
- [\\$wgResourceLoaderEnableJSProfiler](#) – When enabled, execution of JavaScript modules is profiled client-side.
- [\\$wgResourceLoaderEnableSourceMapLinks](#) – Whether to include a SourceMap header in ResourceLoader responses for JavaScript modules.
- [\\$wgResourceLoaderMaxAge](#) – How long a CDN or browser may cache a ResourceLoader HTTP response.
- [\\$wgResourceLoaderMaxQueryLength](#) – Maximum query string length for ResourceLoader requests.
- [\\$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](#) (removed in 1.44) – Use the main stash instead of the `module_deps` table for indirect dependency tracking
- [\\$wgResourceLoaderValidateJS](#) – Validate JavaScript code loaded from wiki pages.
- [\\$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](#) – 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.
- \$wgShortPagesNamespaceExclusions – Optional array of namespaces which should be excluded 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 (**deprecated in 1.41**) – Override the default list of illegal characters in page titles.

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

- \$wgLocalInterwikis – Array of interwiki prefixes 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 – Interwiki cache as an associative array.

- \$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 (removed in 1.42) – Allow `` 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.
- \$wgExternalLinksIgnoreDomains (introduced in 1.46) – List of domains that will be ignored in being tracked into `externallinks` table.
- \$wgMaxPPEExpandDepth – 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 the table of contents (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.
- \$wgParserCacheNewKeySchemaRatio (introduced in 1.44) (removed in 1.44) – Ratio for use of new parser cache key schema.
- \$wgParserEnableLegacyHeadingDOM (removed in 1.45) – Enable legacy HTML structure for headings in the output from the Parser.
- \$wgParserEnableLegacyMediaDOM (removed in 1.45) – Enable legacy media HTML structure in the output from the Parser.

- \$wgParserEnableUserLanguage – Set this to **true** to allow the `{{USERLANGUAGE}}` magic word to return the actual user language.
- \$wgParsoidExperimentalParserFunctionOutput (introduced in 1.44) – If set, Parsoid's HTML output for parser functions will be different from Parsoid HTML spec 2.x.x and lets us experiment with a better output that might be rolled out in a future 3.x Parsoid HTML version.
- \$wgParsoidFragmentSupport (introduced in 1.44) (removed in 1.44) – Enable fragment support in Parsoid (transclusions returning html).
- \$wgParsoidSettings – Default Parsoid configuration.
- \$wgParsoidSelectiveUpdateSampleRate – Sample rate for collecting statistics on Parsoid selective update.
- \$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.
- \$wgUseLegacyMediaStyles – Temporary flag to re-enable the old thumbnail styles provided by core for the legacy media HTML structure.

HTML

- \$wgRawHtml – Allow raw, unchecked HTML in `<html> . . . </html>` sections.
- \$wgTidyConfig – Configuration of the HTML postprocessing tool to use for fixing invalid HTML code.

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).
- \$wgManualRevertSearchRadius – 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 – Maximum depth (revision count) of reverts that will have their reverted edits marked with the mw-reverted change tag.

User accounts, authentication

- `$wgAllowRequiringEmailForResets` (removed in 1.43) – Temporary feature flag that controls whether users will see a checkbox allowing them to require providing email during password resets.
- `$wgAutoCreateTempUser` – Configuration for automatic creation of temporary accounts on page save.
- `$wgConditionalUserOptions` – Default user preferences.
- `$wgDefaultUserOptions` – Default user preferences.
- `$wgHiddenPrefs` – Disable certain user preferences.
- `$wgInvalidUsernameCharacters` – Characters to prevent during new account creations.
- `$wgMaximalPasswordLength` (removed in 1.42) – Specifies the maximal length of a user password.
- `$wgMaxNameChars` – Maximum number of bytes in username.
- `$wgMaxSigChars` – Maximum wiki code length for customized signatures.
- `$wgMinimalPasswordLength` (removed in 1.42) – 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.
- `$wgSignatureValidation` – Behavior of signature validation.
- `$wgSignatureAllowedLintErrors` – List of lint error codes which don't cause signature validation to fail.
- `$wgUserrightsInterwikiDelimiter` – Character used as the separator between username and database name on `Special:UserRights`.
- `$wgUserJsPrefLimit` (introduced in 1.46) – Maximum number of userjs preferences allowed for each user

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.

- \$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.
- \$wgReauthenticateTime – Time frame for re-authentication
- \$wgRememberMe – 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.
- \$wgUserRegistrationProviders – User registration timestamp provider classes

User rights, access control and monitoring

- \$wgAddGroups – Controls who can add specific groups using Special:UserRights.
- \$wgAutoblockExemptions (introduced in 1.44) – List of IP addresses or CIDR ranges that are exempt from autoblocks.
- \$wgAutoblockExpiry – Number of seconds before autoblock entries expire
- \$wgBlockAllowsUEdit – Allow blocked users to edit their talk pages
- \$wgDeleteRevisionsBatchSize – 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.
- \$wgGrantRiskGroups – Group grants by risk level.
- \$wgGroupInheritsPermissions – Groups that should inherit permissions from another group.
- \$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 are not shown on Special:Listusers or somewhere else
- \$wgPasswordAttemptThrottle – Limit password attempts per IP per address.
- \$wgPrivilegedGroups – List of groups which should be considered privileged (user accounts belonging in these groups can be abused in dangerous ways).
- \$wgProxyList – A list of IP addresses to be banned
- \$wgProxyWhitelist – Proxy allowlist: list of addresses that are assumed to be non-proxy despite what the other methods might say.
- \$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.
- \$wgRestrictedGroups (introduced in 1.46) – Conditions required for assignment of user to specific groups
- \$wgRevokePermissions – Permission keys revoked from users in each group.
- \$wgUserRequirementsPrivateConditions (introduced in 1.46) – List of user requirements conditions that shouldn't be evaluated unless there's a need
- \$wgWhitelistReadRegexp – Whitelists publicly readable titles with regular expressions.
- \$wgWhitelistRead – Pages users without the read right 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.
- \$wgAutopromoteOnceRCExcludedGroups (introduced in 1.44) – Defines a denylist of

group names.

- \$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
- \$wgBlockTargetMigrationStage (deprecated in 1.43) – `ipblocks` table schema migration stage, for normalizing `ipb_address` field and adding the `block_target` table.
- \$wgBotPasswordsCluster (deprecated in 1.42) – Cluster name for the `bot_passwords` table
- \$wgBotPasswordsDatabase (deprecated in 1.42) – 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
- \$wgEnableMultiBlocks – If this is **false**, the number of blocks of a given target is limited to only 1.
- \$wgEnablePartialActionBlocks (removed in 1.45) – Flag to enable partial blocks against performing certain actions.
- \$wgExternalQuerySources (introduced in 1.45) – Configuration for external query sources.
- \$wgHideIdentifiableRedirects – Protects user's privacy when doing redirects
- \$wgImplicitRights – A list of implicit rights, in addition to the ones defined by the core.
- \$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 `$wgSpamRegex` except for edit summaries.
- `$wgTempAccountCreationThrottle` – Number of temporary accounts each IP address may create per specified period(s).
- `$wgTempAccountNameAcquisitionThrottle` – Number of temporary accounts usernames each IP address may acquire per specified period(s).

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

- `$wgAllowCrossOrigin` – Allow anonymous cross origin requests to the REST API.
- `$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` – Controls Content-Security-Policy header [Experimental]
- `$wgCSPReportOnlyHeader` – Controls Content-Security-Policy-Report-Only header [Experimental]
- `$wgEditPageFrameOptions` – Control framing of wiki pages globally.
- `$wgHKDFAlgorithm` (removed in 1.43) – Algorithm for HMAC-based key derivation function (fast, cryptographically secure random numbers).
- `$wgHKDFSecret` (removed in 1.43) – Secret for HMAC-based key derivation function (fast, cryptographically secure random numbers).
- `$wgJwtPrivateKey` (introduced in 1.45) – RSA private key for issuing JWTs, as a PEM string.
- `$wgJwtPublicKey` (introduced in 1.45) – RSA private key for verifying JWTs, as a PEM string.
- `$wgRestAllowCrossOriginCookieAuth` – Allows authenticated cross-origin requests to the REST API with session cookies.
- `$wgSecretKey` – This should always be customized to a secret, unique string in [LocalSettings.php](#).

- `$wgSessionInsecureSecrets` (**removed in 1.43**) – Allows MediaWiki to work at the cost of storing sensitive session data insecurely.
- `$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.
- `$wgCookieExpiration` – Number of seconds before HTTP cookies expire. You can override this for Keep me logged in cookies with `$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` – 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.
- `$wgCookieSetOnIpBlock` – Determines whether to set a cookie when an IP user is blocked. Doing so means that a blocked user, even after moving to a new IP address, will still be blocked.
- `$wgDisableCookieCheck` (**removed in 1.42**) – Disables session cookies
- `$wgExtendedLoginCookieExpiration` – If non-null, duration in seconds to use for Keep me logged in sessions; 0 means until the browser is closed. If null, uses `$wgCookieExpiration`
- `$wgSessionName` – Override to customize the session name.
- `$wgUseSameSiteLegacyCookies` (**removed in 1.42**) – 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` (**removed in 1.40**) – Parser test suite files to be run by `parserTests.php`.

Debug

See also: [Manual:How to debug](#)

- [\\$wgDebugComments](#) – Include portions of log output in page HTML comment
- [\\$wgDebugDumpSql](#) – Dump SQL queries to debug output
- [\\$wgDebugLogFile](#) – The file name of the debug log, or empty if disabled.
- [\\$wgDebugLogGroups](#) – Direct wfDebugLog () output for certain groups to files other than [\\$wgDebugLogFile](#).
- [\\$wgDebugLogPrefix](#) – Prefix for debug log lines.
- [\\$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
- [\\$wgDebugToolbar](#) – Display the debugging toolbar
- [\\$wgDeprecationReleaseLimit](#) – Release limitation to wfDeprecated 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.
- [\\$wgShowDebug](#) – Show debug data at bottom of content area
- [\\$wgShowExceptionDetails](#) – If set to true, uncaught exceptions will print a detailed stack trace to output.
- [\\$wgShowHostnames](#) – If set to **true**, exposes host names through API and HTML comments.
- [\\$wgSpecialVersionShowHooks](#) – Whether to show the contents of \$wgHooks in Special:Version.
- [\\$wgTrxProfilerLimits](#) – Performance expectations for DB usage

Profiling

- [\\$wgMetricsTarget](#) (**removed in 1.40**) – Output target URI for the MetricsFactory service, e.g. udp://127.0.0.1:8125.
- [\\$wgMetricsPrefix](#) (**removed in 1.40**) – Service name prefix for the MetricsFactory service.
- [\\$wgMetricsFormat](#) (**removed in 1.40**) – Output format for the MetricsFactory service,

e.g. 'statsd'.

- `$wgOpenTelemetryConfig` – Configuration for OpenTelemetry instrumentation, or `null` to disable it.
- `$wgProfiler` – Settings related to profiling.
- `$wgStatsdMetricPrefix` – Prefix for metric names sent to `$wgStatsdServer`
- `$wgStatsdSamplingRates` (removed in 1.43) – Sampling rate for statsd metrics as an associative array of patterns and rates.
- `$wgStatsdServer` – A host or host:port of a statsd server.
- `$wgStatsFormat` – Output format for the StatsFactory service, e.g. 'statsd'.
- `$wgStatsPrefix` – Service name prefix for the StatsFactory service.
- `$wgStatsTarget` – Output target URI for the StatsFactory service, e.g. `udp://127.0.0.1:8125`.

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.
- `$wgEnableOpenSearchSuggest` (removed in 1.40) – Enable OpenSearch suggestions requested by MediaWiki.
- `$wgEnableSearchContributorsByIP` (removed in 1.45) – 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.
- `$wgOpenSearchTemplate` (removed in 1.42) – Template for OpenSearch suggestions.
- `$wgSearchForwardUrl` – Set this to a URL to forward search requests to an external location.
- `$wgSearchHighlightBoundaries` – Regexp to match word boundaries
- `$wgSearchMatchRedirectPreference` – 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.
- [\\$wgSitemapApiConfig](#) ([introduced in 1.45](#)) – Configuration for the sitemaps REST API endpoint `/rest.php/site/v1/sitemap/0`
- [\\$wgSitemapNamespacesPriorities](#) ([deprecated in 1.45](#)) – Custom namespace priorities for sitemaps.
- [\\$wgSitemapNamespaces](#) – Array of namespaces identified by their IDs to generate a sitemap or false for all namespaces.
- [\\$wgSpecialSearchFormOptions](#) – Options for Special:Search completion widget form created by SearchFormWidget class.

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
- [\\$wgInstallerInitialPages](#) ([introduced in 1.44](#)) – Pages to autcreate after installing MediaWiki.
- [\\$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](#).

- [\\$wgDisableAnonTalk](#) – Disable talk pages for anonymous users (not logged in).
- [\\$wgRCChangedSizeThreshold](#) – Highlight character count difference lower than this.
- [\\$wgRCEngines](#) (**deprecated in 1.38**) – Legacy mapping from URI schemes to RCFeed subclasses.
- [\\$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 your 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](#) – Relative maximum duration for watchlist expiries, as accepted by `strtotime()`.
- [\\$wgWatchlistExpiry](#) – Enable the watchlist expiry feature.
- [\\$wgWatchlistPurgeRate](#) – The per-edit rate at which expired watchlist items will be

purged from the database.

- \$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.
- \$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.
- \$wgEditWatchlistPaginate (introduced in 1.45) (removed in 1.46) – Whether to enable pagination on Special:EditWatchlist (feature flag)
- \$wgEnableChangesListQueryPartitioning (introduced in 1.45) – Allow Watchlist and RecentChangesLinked queries to be partitioned by rc_timestamp.
- \$wgEnableWatchlistLabels (introduced in 1.46) – Whether to enable the watchlist labels feature.
- \$wgWatchlistLabelsMaxPerUser (introduced in 1.46) – Maximum number of labels a user can create for their watchlist.

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

- \$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 – Controls the availability of the full-history feature on "Special:Export".
- \$wgExportAllowListContributors – Controls whether to allow to list all contributors while exporting a page using Special:Export.
- \$wgExportFromNamespaces – Whether to allow the "export all pages in namespace" option.
- \$wgExportMaxHistory – Requests using "Special:Export" for pages with a revision history higher than set 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 – The schema to use by default when generating XML dumps.

Extensions

- \$wgAutoloadAttemptLowercase (**removed in 1.40**) – 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.
- \$wgExtensionCredits – 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](#) (**removed in 1.42**) – 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](#) (**removed in 1.41**) – 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.php`.
- [\\$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.
- [\\$wgSortedCategories](#) – Whether to sort categories in `OutputPage` for display.
- [\\$wgTempCategoryCollations](#) – Additional category collations to store during `LinksUpdate`.
- [\\$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 LogFormatter derived class names.
- [\\$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](#) – 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:UnusedFiles](#), 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

- \$wgAjaxLicensePreview – Use Ajax to preview licensing for uploads.
- \$wgAjaxUploadDestCheck (removed in 1.41) – Enable Ajax check for file overwrite, pre-upload.

API

- \$wgAllowedCorsHeaders – 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 (removed in 1.46) – 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
- \$wgEnableUnsafeXsltOption (introduced in 1.45) – Enable the deprecated xslt option in

the Action API.

- [\\$wgRestAPIAdditionalRouteFiles](#) – Additional REST API Route files.
- [\\$wgRestSandboxSpecs](#) (**removed in 1.43.2**) – A list of OpenAPI specs to be made available for exploration on Special:RestSandbox.

Shell and process control

- [\\$wgPhpCli](#) – Executable path of PHP cli client (php/php5).
- [\\$wgMaxShellFileSize](#) – Maximum file size created by shell processes under linux, in KiB.
- [\\$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](#) – The secret key for HMAC verification of [Shellbox](#) requests.
- [\\$wgShellboxUrl](#) (**removed in 1.40**) – Shell commands can be run on a remote server using Shellbox.
- [\\$wgShellboxUrls](#) – Shell commands can be run on a remote server using [Shellbox](#).
- [\\$wgShellCgroup](#) – Limit shell commands memory using linux Cgroups.
- [\\$wgShellRestrictionMethod](#) – Method to use to restrict shell commands

HTTP client

- [\\$wgAllowExternalReqID](#) – 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](#) – The maximum HTTP request timeout in seconds.
- [\\$wgHTTPMaxConnectTimeout](#) – 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.
- [\\$wgLocalHTTPProxy](#) – Proxy to use for requests to domains in [\\$wgLocalVirtualHosts](#).
- [\\$wgLocalVirtualHosts](#) – A list of URL domains that will be routed to the proxy specified by [\\$wgLocalHTTPProxy](#).

Jobs

- \$wgJobRunRate – Number of jobs to perform per request.
- \$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.

Wiki farm

- \$wgWikiFarmSettingsDirectory – A directory that contains site-specific configuration files.
- \$wgWikiFarmSettingsExtension – The file extension to be used when looking up site-specific settings files in \$wgWikiFarmSettingsDirectory, such as `'json'` or `'yaml'`.

Miscellaneous settings

- \$wgCloneArticleParserOutput (introduced in 1.46) – Whether Article should clone the ParserOutput before postprocessing.
- \$wgEditRecoveryExpiry – Number of seconds to keep Edit Recovery data after the edit is stored.
- \$wgEnableEditRecovery – Whether to enable the client-side Edit Recovery feature.
- \$wgEnableProtectionIndicators – Whether to show indicators on a page when it is protected.
- \$wgEventRelayerConfig – Mapping of event channels (or channel categories) to EventRelayer configuration.
- \$wgFeaturePolicyReportOnly – List of Feature Policy Reporting types to enable.
- \$wgFeatureShutdown (introduced in 1.44) – Allow temporarily disabling use of certain features.
- \$wgOutputPipelineStages – OutputPipelineStages to add to the DefaultOutputPipeline.
- \$wgOriginTrials – Origin Trials tokens.
- \$wgPingback – If true, MediaWiki will report back data about its installation from time to time to aid developers
- \$wgRedirectOnLogin – Redirect users to page on login
- \$wgReportToEndpoints – List of endpoints for the Reporting API.
- \$wgReportToExpiry – Expiry of the endpoint definition for the Reporting API.
- \$wgShowDiffToggleSwitch (deprecated in 1.41) – Specifies if users should see the inline switch in diff pages.
- \$wgShowLogoutConfirmation – Whether to display a confirmation screen during user log out.
- \$wgSkinsPreferred – List of preferred skins to be listed higher in Special:Preferences

- [\\$wgSpecialContributeSkinsDisabled](#) (**removed in 1.40**) – List of skins to not show the Special:Contribute page
- [\\$wgSpecialContributeSkinsEnabled](#) – List of skins that show a link to the Special:Contribute page
- [\\$wgSpecialContributeNewPageTarget](#) (**introduced in 1.45**) – Allow redirection to another page when a user selects New Page on Special:Contribute page
- [\\$wgTranslationAliasesDirs](#) – Message directories containing JSON files for localisation of special page aliases.
- [\\$wgUseCodexSpecialBlock](#) – Whether to use the new [manage blocks](#) interface at Special:Block.
- [\\$wgUsePostprocCache](#) (**introduced in 1.45**) – Whether to use the post-OutputTransform cache
- [\\$wgUseLeximorph](#) (**introduced in 1.45**) – Whether parser functions should use Leximorph handlers instead of Language methods.
- [\\$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=8188264"