



MediaWiki



History and overview

Tim Starling
Nairobi 2024



___TOC___

- MediaWiki origins and history
- The impacts of early decisions
- MediaWiki today
- The future

About me

- 2002: Bachelor of Science (Physics)
- October 2002: Started editing Wikipedia
- 2004-2005: Very active volunteer developer
- April 2006: Hired by WMF
- Now Principal Software Architect reporting to Runa and working with CommTech

MediaWiki

Origins and History

UseModWiki

- Initial release October 2000
- Wikipedia launched using this engine, January 2001
- Flat file database
- Camel case links

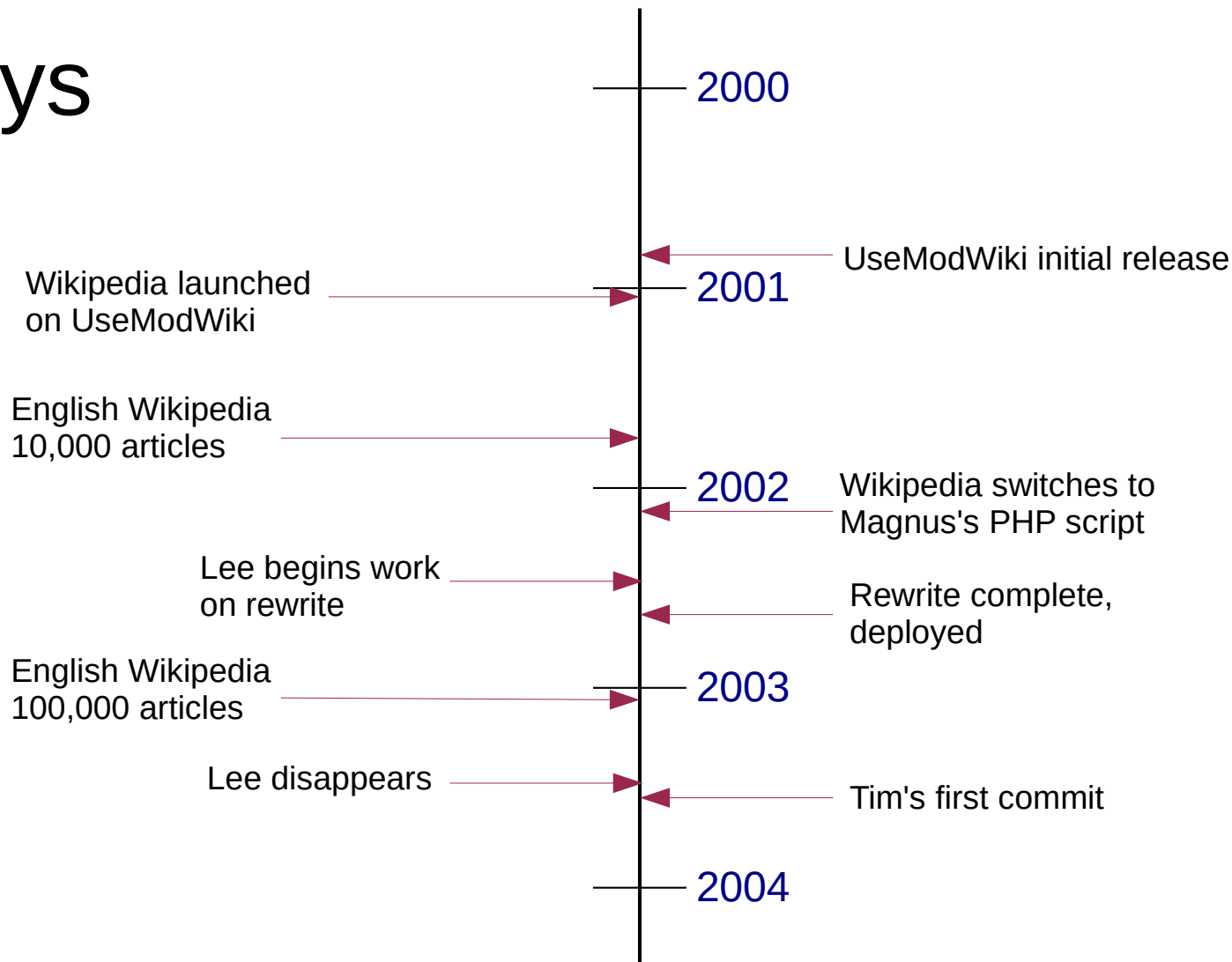
Phase II

- Magnus Manske wrote a new engine for in PHP, deployed in November 2001.
- "The Wikipedia PHP script" later referred to as Phase II
- MySQL backend
- Introduced skins, special pages

Phase 3

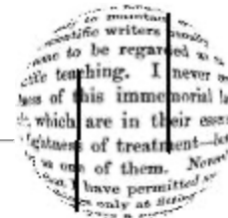
- Lee Daniel Crocker rewrote Wikipedia's engine in mid 2002
- Introduced namespaces, deprecated subpages
- Renamed "MediaWiki" in 2003

Heady Days



HomePage

[HomePage](#) | [RecentChanges](#) | [Preferences](#)



Welcome to [WikiPedia](#)! We're writing a complete encyclopedia from scratch, collaboratively. We started work in January 2001. We've got over 1,000 pages already. We want to make over 100,000. So, let's get to work! Write a little (or a lot) about what you know!

Some questions you might have are answered below, but why not first explore a few links?

[PhiloSophy](#) -- [MathematicsAndStatistics](#) -- [ScienCe](#) -- [PhysIcs](#) -- [BiologicalSciences](#) -- [AstroNomy](#) -- [ChemIstry?](#) -
- [BioChemistry](#) -- [MolecularBiology?](#) -- [BotAny?](#) -- [ZooLogy?](#) -- [EarthSciences?](#)

[CountriesOfTheWorld](#) -- [HiStory](#) -- [LinguisticS](#) & [Language](#) -- [PoLitics](#) -- [EconomicS](#) -- [AnthropOlogy?](#) --
[ArchaeOlogy?](#) -- [PsychOlogy?](#) -- [SociOlogy?](#) -- [HistoryOfScienceAndTechnology?](#) -- [AnomalousPhenomena?](#)

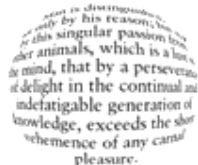
[TechnologY](#) -- [ArchitecturE](#) -- [EngineerIng](#) -- [ComputinG](#) -- [TransporT](#) -- [BusinessAndIndustry](#) --
[BusinessAndEconomics](#) -- [TheLaw](#) -- [AgriculturalScience?](#) -- [EduCation](#) -- [CommunicAtion?](#) --
[LibraryAndInformationScience?](#) -- [HealthSciences?](#) -- [FamilyAndConsumerScience?](#) -- [PublicAffairs?](#) --
[HumanComputerInteraction](#) *(this link should be moved when the Lewis and Clark class is done with it)*

[ArtsandEntertainment](#) -- [MuSic](#) -- [DanCe](#) -- [FiLm](#) -- [LiteraryArt](#) -- [ReLigion](#) -- [ReCreation](#) -- [SportS](#) -- [ClassIcs?](#) --
[CriticalTheory?](#) -- [VisualArtsAndDesign](#) -- [PaintIng](#) -- [SculpTure?](#) -- [PerformingArts?](#) -- [TheaterAndDrama?](#) --
[GameS?](#) -- [HobbieS?](#) -- [TourIsm?](#)

The above reflects only one of [WikiPedia](#)'s [CategorySchemes](#).

Want to learn about something? Add the topic to our list of [RequestedArticles](#).

You're wondering



WIKIPEDIA

The Free Encyclopedia

[Main Page](#)
[Recent changes](#)
[Random page](#)
[Watch list](#)
[Current events](#)

Protected page

[Talk page](#)
[History](#)
[What links here](#)
[Watch links](#)

[Bug reports](#)
[Special pages](#)

[Main Page](#) | [Recent changes](#) | Protected page | [History](#) | Special pages ▾

[Printable version](#)

Other languages: [Danish](#) | [German](#) | [Esperanto](#) | [Spanish](#) | [French](#) | [Dutch](#) | [Polish](#) | [Portuguese](#)

209.237.238.164

[Log in](#) | [Help](#)

Main Page

From Wikipedia, the free encyclopedia.

Welcome to Wikipedia, a collaborative project to produce a complete [encyclopedia](#) from scratch. We started in January [2001](#) and are already working on [52364 articles](#) (in the English version), with more being added and improved all the time. Anyone, including *you*, can edit any article right now, without even having to log in. You can copyedit, expand an article, write a little or write a lot. See the [Wikipedia FAQ](#) for more background information about the project, and the [help page](#) for information on how to use and contribute to Wikipedia.

The content of Wikipedia is covered by the [GNU Free Documentation License](#), which means that it is free and will remain so forever. See [wikipedia:copyrights](#) for the details and [open content](#) and [free content](#) for background.

Current events

[Current events](#) - [Israeli-Palestinian conflict](#) - [West Nile virus](#) - [Bruce Paltrow](#) - [Sputnik I](#) - [John Walker Lindh](#) - [Claus von Amsberg](#) - [Bertrand Delanoë](#) - [Quaoar](#) - [Nobel Prize](#)

Philosophy, Mathematics, and Natural Science

[Astronomy and astrophysics](#) - [Biology](#) - [Chemistry](#) - [Earth science](#) - [Mathematics](#) - [Philosophy](#) - [Physics](#) - [Statistics](#)

Line counts

UseModWiki 0.92	3,800
Phase 2 (final)	6,200
Phase 3 (May 2003)	17,600
MediaWiki core (January 2024)	800,000

cloc non-blank non-comment
excluding localisation and data files

UseModWiki

```
sub UpdateLinksList {
    my ($commandList, $doRC, $doText) = @_;

    if ($doText) {
        &BuildLinkIndex();
    }
    &RequestLock() or die "UpdateLinksList could not get main lock";
    unlink($IndexFile) if ($UseIndex);
    foreach (split(/\n/, $commandList)) {
        s/\s+$/g;
        next if (!(/^[=!|]/)); # Only valid commands.
        print "Processing $_<br>\n";
        if (/^\!(.+)/) {
            &DeletePage($1, $doRC, $doText);
        } elsif (/^=(?:\[ \] ? ([^=] +) (? : \[ \] ) ? \ = (? : \[ \] ? ([^=] +) (? : \[ \] ) ? /) {
            &RenamePage($1, $2, $doRC, $doText);
        } elsif (/^|(? : \[ \] ? ([^|] +) (? : \[ \] ) ? \ | (? : \[ \] ? ([^|] +) (? : \[ \] ) ? /) {
            &RenameTextLinks($1, $2);
        }
    }
    &NewPageCacheClear("."); # Clear cache (needs testing?)
    unlink($IndexFile) if ($UseIndex);
    &ReleaseLock();
}
```

UseModWiki

```
# Delete and rename must be done inside locks.
sub DeletePage {
    my ($page, $doRC, $doText) = @_;
    my ($fname, $status);

    $page =~ s/ /_/g;
    $page =~ s/\[+//;
    $page =~ s/\]+//;
    $status = &ValidId($page);
    if ($status ne "") {
        print "Delete-Page: page $page is invalid, error is: $status<br>\n";
        return;
    }

    $fname = &GetPageFile($page);
    unlink($fname) if (-f $fname);
    $fname = $KeepDir . "/" . &GetPageDirectory($page) . "/"$page.kp";
    unlink($fname) if (-f $fname);
    unlink($IndexFile) if ($UseIndex);
    &EditRecentChanges(1, $page, "") if ($doRC); # Delete page
    # Currently don't do anything with page text
}
```

Phase 2

```
<?
# THIS IS THE MAIN FILE OF THE PHP WIKIPEDIA SCRIPT

$styleSheet = "" ;
$bodyOptions = "";
$append = "";
$headerScript = "";

# Includes the necessary parts
include_once ( "../wikiSettings.php" ) ;
include_once ( "../basicFunctions.php" ) ;
include_once ( "../databaseFunctions.php" ) ;
include_once ( "../wikiTitle.php" ) ;
include_once ( "../wikiPage.php" ) ;
include_once ( "../wikiUser.php" ) ;

# Main program

$expiration = time()+31536000 ; # Cookies set will expire in one year from now
$pageTitle = "" ;
```

```
<?
include_once ( "special_makelog.php" ) ;

function deletepage () {
    global $wikiDeleteTitle , $wikiDeleteDenied , $wikiDeleteSuccess , $wikiDeleteMsg1 , $wikiDeleteMsg2 ;
    global $THESCRIPRT , $target , $user , $iamsure , $vpage , $wikiSQLServer ;
    $target = stripslashes ( $target ) ;
    $vpage = new WikiPage ;
    $vpage->title = $title ;
    $vpage->makeSecureTitle () ;
    $ti = $vpage->secureTitle ;

    $vpage->special ( str_replace ( "$1" , $target , $wikiDeleteTitle ) ) ;
    $vpage->makeSecureTitle () ;
    if ( !in_array ( "is_sysop" , $user->rights ) ) return $wikiDeleteDenied ;
    if ( $iamsure == "yes" ) {
        $ret = "<h2>".str_replace("$1",$target,$wikiDeleteSuccess)."</h2>" ;
        $connection = getDBconnection () ;

        # Move from cur to old for potential recovery later
        $deadpage = new WikiPage ;
        $deadpage->setTitle ( $target ) ;
        $deadpage->backup () ;

        # Okay, now go for it
        $sql = "DELETE FROM cur WHERE cur_title=\"$target\" " ;
        $result = mysql_query ( $sql , $connection ) ;
```


Phase 3

```
<?
# Main wiki script; see design.doc
#
$wgRequestTime = microtime();

session_cache_limiter( "private, must-revalidate" );
session_start();
session_register( "wsUserID" );
session_register( "wsUserName" );
session_register( "wsUserPassword" );

global $IP;
include_once( "../LocalSettings.php" );
include_once( "$IP/Setup.php" );

wfProfileIn( "main-misc-setup" );
OutputPage::setEncodings(); # Not really used yet

# Query string fields
#
global $action, $title, $search, $go, $target, $printable;
global $returnto, $diff, $oldid;
```

Article.php line 982

```
function delete()
{
    global $wgUser, $wgOut, $wgTitle;
    global $wpConfirm, $wpReason, $image, $oldimage;

    # Anybody can delete old revisions of images; only sysops
    # can delete articles and current images

    if ( ( ! $oldimage ) && ( ! $wgUser->isSysop() ) ) {
        $wgOut->sysopRequired();
        return;
    }
    if ( wfReadOnly() ) {
        $wgOut->readOnlyPage();
        return;
    }

    # Better double-check that it hasn't been deleted yet!
    $wgOut->setPagetitle( wfMsg( "confirmdelete" ) );
    if ( $image ) {
        if ( "" == trim( $image ) ) {
            $wgOut->fatalError( wfMsg( "cannotdelete" ) );
            return;
        }
    }
}
```

Article.php line 1130

```
function doDeleteArticle( $title )
{
    global $wgUser, $wgOut, $wgLang, $wpReason, $wgTitle, $wgDeferredUpdateList;

    $fname = "Article::doDeleteArticle";
    $ns = $title->getNamespace();
    $t = wfStrencode( $title->getDBkey() );
    $id = $title->getArticleID();

    if ( "" == $t ) {
        $wgOut->fatalError( wfMsg( "cannotdelete" ) );
        return;
    }

    $u = new SiteStatsUpdate( 0, 1, -$this->isCountable( $this->getContent( true ) ) );
    array_push( $wgDeferredUpdateList, $u );

    # Move article and history to the "archive" table
    $sql = "INSERT INTO archive (ar_namespace,ar_title,ar_text," .
        "ar_comment,ar_user,ar_user_text,ar_timestamp,ar_minor_edit," .
        "ar_flags) SELECT cur_namespace,cur_title,cur_text,cur_comment," .
        "cur_user,cur_user_text,cur_timestamp,cur_minor_edit,0 FROM cur " .
        "WHERE cur_namespace={ $ns } AND cur_title='{ $t }'";
    wfQuery( $sql, $fname );
}
```

MediaWiki today

DeleteAction::tempDelete()

```
protected function tempDelete() {
    $article = $this->getArticle();
    $title = $this->getTitle();
    $context = $this->getContext();
    $user = $context->getUser();
    $request = $context->getRequest();
    $outputPage = $context->getOutput();

    # Better double-check that it hasn't been deleted yet!
    $article->getPage()->loadPageData(
        $request->wasPosted() ? WikiPage::READ_LATEST : WikiPage::READ_NORMAL
    );
    if ( !$article->getPage()->exists() ) {
        $outputPage->setPageTitleMsg(
            $context->msg( 'cannotdelete-title' )->plaintextParams( $title->getPrefixedText() )
        );
        $outputPage->wrapWikiMsg( "<div class=\"error mw-error-cannotdelete\">\n$1\n</div>",
            [ 'cannotdelete', wfEscapWikiText( $title->getPrefixedText() ) ]
        );
        $this->showLogEntries();

        return;
    }
}
```

20-year-old comment!



```

<?php

namespace MediaWiki\Page;

use MediaWiki\Permissions\Authority;

/**
 * Service for page delete actions.
 *
 * Default implementation is MediaWiki\Page\PageCommandFactory.
 *
 * @since 1.37
 */
interface DeletePageFactory {

    /**
     * @param ProperPageIdentity $page
     * @param Authority $deleter
     * @return DeletePage
     */
    public function newDeletePage( ProperPageIdentity $page, Authority $deleter ): DeletePage;
}

```

```
public function __construct(
    HookContainer $hookContainer,
    RevisionStore $revisionStore,
    LBFactory $lbFactory,
    JobQueueGroup $jobQueueGroup,
    CommentStore $commentStore,
    ServiceOptions $serviceOptions,
    BagOStuff $recentDeletesCache,
    string $localWikiID,
    string $webRequestID,
    WikiPageFactory $wikiPageFactory,
    UserFactory $userFactory,
    BacklinkCacheFactory $backlinkCacheFactory,
    NamespaceInfo $namespaceInfo,
    ITextFormatter $contLangMsgTextFormatter,
    ProperPageIdentity $page,
    Authority $deleter
) {
    $this->hookRunner = new HookRunner( $hookContainer );
    $this->revisionStore = $revisionStore;
    $this->lbFactory = $lbFactory;
    $this->jobQueueGroup = $jobQueueGroup;
    $this->commentStore = $commentStore;
    $serviceOptions->assertRequiredOptions( self::CONSTRUCTOR_OPTIONS );
    $this->options = $serviceOptions;
```

```

/**
 * Back-end article deletion: deletes the article with database consistency, writes logs, purges cac
 * @note This method doesn't check user permissions. Use deleteIfAllowed for that.
 *
 * @param string $reason Delete reason for deletion log
 * @return Status Status object:
 *     - If successful (or scheduled), a good Status
 *     - If a page couldn't be deleted because it wasn't found, a Status with a non-fatal 'cannotdelet
 *     - A fatal Status otherwise.
 */
public function deleteUnsafe( string $reason ): Status {
    $this->setDeletionAttempted();
    $origReason = $reason;
    $hookStatus = $this->runPreDeleteHooks( $this->page, $reason );
    if ( !$hookStatus->isGood() ) {
        return $hookStatus;
    }
    if ( $this->associatedTalk ) {
        $talkReason = $this->contLangMsgTextFormatter->format(
            MessageValue::new( 'delete-talk-summary-prefix' )->plaintextParams( $origReason )
        );
        $talkHookStatus = $this->runPreDeleteHooks( $this->associatedTalk, $talkReason );
        if ( !$talkHookStatus->isGood() ) {
            return $talkHookStatus;
        }
    }

    $status = $this->deleteInternal( $this->page, self::PAGE_BASE, $reason );

```

```

// Archive revisions. In immediate mode, archive all revisions. Otherwise, archive
// one batch of revisions and defer archival of any others to the job queue.
$explicitTrxLogged = false;
while ( true ) {
    $done = $this->archiveRevisions( $page, $id );
    if ( $done || !$this->forceImmediate ) {
        break;
    }
    $dbw->endAtomic( __METHOD__ );
    if ( $dbw->explicitTrxActive() ) {
        // Explicit transactions may never happen here in practice. Log to be sure.
        if ( !$explicitTrxLogged ) {
            $explicitTrxLogged = true;
            LoggerFactory::getInstance( 'wfDebug' )->debug(
                'explicit transaction active in ' . __METHOD__ . ' while deleting {title}', [
                    'title' => $title->getText(),
                ] );
        }
        continue;
    }
    if ( $dbw->trxLevel() ) {
        $dbw->commit( __METHOD__ );
    }
    $this->lbFactory->waitForReplication();
    $dbw->startAtomic( __METHOD__ );
}

```



```
// For now, shunt the revision data into the archive table.
// Text is *not* removed from the text table; bulk storage
// is left intact to avoid breaking block-compression or
// immutable storage schemes.
// In the future, we may keep revisions and mark them with
// the rev_deleted field, which is reserved for this purpose.

// Lock rows in `revision` and its temp tables, but not any others.
// Note array_intersect() preserves keys from the first arg, and we're
// assuming $revQuery has `revision` primary and isn't using subtables
// for anything we care about.
$lockQuery = $revQuery;
$lockQuery['tables'] = array_intersect(
    $revQuery['tables'],
    [ 'revision', 'revision_comment_temp' ]
);
unset( $lockQuery['fields'] );
$dbw->newSelectQueryBuilder()
    ->queryInfo( $lockQuery )
    ->where( [ 'rev_page' => $id ] )
    ->forUpdate()
    ->caller( __METHOD__ )
    ->acquireRowLocks();

$deleteBatchSize = $this->options->get( MainConfigNames::DeleteRevisionsBatchSize );
```

Stack trace

1 DatabaseMySQL.php:757	Wikimedia\Rdbms\DatabaseMySQL->doSingleStatementQuery()
2 Database.php:797	Wikimedia\Rdbms\Database->attemptQuery()
3 Database.php:716	Wikimedia\Rdbms\Database->executeQuery()
4 Database.php:643	Wikimedia\Rdbms\Database->query()
5 Database.php:1734	Wikimedia\Rdbms\Database->delete()
6 DBConnRef.php:119	Wikimedia\Rdbms\DBConnRef->__call()
7 DBConnRef.php:576	Wikimedia\Rdbms\DBConnRef->delete()
8 DeleteQueryBuilder.php:229	Wikimedia\Rdbms>DeleteQueryBuilder->execute()
9 DeletePage.php:820	MediaWiki\Page\DeletePage->archiveRevisions()
10 DeletePage.php:579	MediaWiki\Page\DeletePage->deleteInternal()
11 DeletePage.php:465	MediaWiki\Page\DeletePage->deleteUnsafe()
12 DeletePage.php:367	MediaWiki\Page\DeletePage->deleteIfAllowed()
13 DeleteAction.php:183	DeleteAction->tempDelete()
14 DeleteAction.php:136	DeleteAction->show()
15 ActionEntryPoint.php:733	MediaWiki\Actions\ActionEntryPoint->performAction()
16 ActionEntryPoint.php:510	MediaWiki\Actions\ActionEntryPoint->performRequest()
17 ActionEntryPoint.php:151	MediaWiki\Actions\ActionEntryPoint->execute()
18 MediaWikiEntryPoint.php:169	MediaWiki\MediaWikiEntryPoint->run()
19 index.php:50	{main}()

Reflections

- Lee Daniel Crocker wrote 17 kloc in 2 months.
 - Professional poker player.
 - Had no prior PHP experience.
 - Credits Magnus but left no line of Magnus's code untouched.
- Good enough that we never had to rewrite.



Founder effects

- What early decisions continue to impact the product?

Human-readable URLs

`https://en.wikipedia.org/wiki/Isaac_Newton`

- Unique title is also part of the URL
- Users have to care about titles
- Normal for wikis but not for other CMSes
- Compare: Google Docs, news websites
- Even images need to have names. Naming is hard but we make our users do that.

Special pages

- Introduced in Phase 2
- Now we have 246 of them on English Wikipedia.
- An unusual design decision to make internal actions be a kind of page.

Localised and international

- Already supported 15 languages in May 2003
- Internal message identifier

```
$rctext = wfMsg( "recentchangetext" );
```

- Compare contemporary best practice: get text

```
printf(_("Hello World\n"));
```

PHP


- PHP was slow, new. But easy to learn.
- Lee knew Java. How might things have been different if he decided to port it?
- There has never been another opportunity to port.

Coding style

- "m" prefixes were a mistake and just won't die

C++


```
class ViewCountUpdate {  
public:  
    int mPageID;  
  
    ViewCountUpdate( int pageId )  
    {  
        mPageID = pageId;  
    }  
}
```



Property access is ambiguous
Local variables conflict with properties

PHP

```
class ViewCountUpdate {  
  
    var $mPageID;  
  
    function ViewCountUpdate( $pageId )  
    {  
        $this->mPageID = $pageId;  
    }  
}
```



Property access prefixed by \$this
"m" is redundant

Classes

- `User`, `Title`, `Article`, `OutputPage`, `Skin`: created in 2002 and were large at the time.
- Still exist with similar responsibilities.
- `Skin` still tightly integrated.
- Class size >1000 lines continues to be normal.

Extensions

- In May 2004 I introduced the idea of extensions as separate modules, registered in config, strictly separated from core.
- The first extension in this sense was BoardVote.
- There were a couple of things called extensions before this, but core knew about them.
- Evan Prodromou introduced hooks in November. I favoured subclasses.

Why extensions?

- Encouraging MediaWiki as an open source product with an ecosystem of independent developers.
- Reducing download size.
- Introducing strict boundaries between modules.

Multi-repo Git

- In 2012 we moved from Subversion to Git
- In Subversion we often changed both core and extensions in a single commit.
- With a single commit we could update all extension callers of a deprecated function.
- We split it up in the move to Git but now have some regrets, did we do the right thing?

MediaWiki Today

Incomplete migrations

- TitleParser introduced in 2014, but still hundreds of usages of Title::newFromText()
- LoggerFactory introduced in 2014, but still hundreds of callers of wfDebug()
- packageFiles
- Config globals
- Be careful about which examples you follow!

Data lifetime

Lifetime	Where	When to use
Process	global variable, class static, local static	Rarely or never.
Service container	Non-static member of service class	Persistent caches. Factory-managed class instances.
Request	Method parameter. Ephemeral object.	Form inputs, user preferences.

- Use the service container rather than class static members so that data will be wiped between tests.
- Multiple requests may be handled by a single service container, for example if FauxRequest is used.

Value objects

- Value objects store data but have few methods.
- Value objects may be created with "new"

```
$user = new UserIdentityValue( $userId, $userName );
```

Service classes

- Service classes are singletons. There is only one instance at any given time.
- The instance is stored in the service container.
- Service classes store long-lived (process lifetime) cache data.
- Service classes receive required services in their constructors, or later by a mutator method (dependency injection).
- Use ServiceWiring.php (in core or extensions) to declare service classes.

ServiceOptions

- Bundle a subset of the global config into a ServiceOptions object.

ServiceOptions

```
class MyService {
    public const CONSTRUCTOR_OPTIONS = [ MainConfigNames::MyServiceFlavour ];
    private $flavour;

    public function __construct( ServiceOptions $options ) {
        $options->assertRequiredOptions( MyService::CONSTRUCTOR_OPTIONS );
        $this->flavour = $options->get( MainConfigNames::MyServiceFlavour );
    }
}

// ServiceWiring.php
return [
    'MyService' => static function ( MediaWikiServices $services ): MyService {
        return new MyService(
            new ServiceOptions(
                MyService::CONSTRUCTOR_OPTIONS,
                $services->getMainConfig()
            )
        );
    }
];
```

Current style hints

- Use namespaces and PSR-4
- Use hook interfaces
- Use `SelectQueryBuilder`
- JSON serialization
- `packageFiles`

RequestContext

- A kitchen-sink bundle of request-related data.
- Static singleton `RequestContext::getMain()`
- Better than what came before (`$wgUser`, `$wgTitle`)

MediaWiki Future

MediaWiki future

Time wounds all heels

— Bryan Davis, 2014, regarding LoggerFactory
(also Groucho Marx etc.)

Near future

- MediaWikiEntryPoint
- Query builders
- Codex and Vue
- Parsoid read views

Murky medium

- Another hook redesign (T275847, T212482)
- Global config variable deprecation, removal
- Skin isolation (T140664)
- MessageValue

Technical hopes

- Completed migrations
- One way to do things
- Upstream engagement

Product hopes

- Wikitext 2.0
- Global templates
- Structured sister projects
- Rich content and media embedding
 - Maps, graphs, diagrams, animations
- Help the world learn

Not going to happen

- Rewrite in some other language.
 - It just keeps getting harder the more PHP code we write. The goal is receding.
- Split the monolith into many microservices.
 - High costs.
 - Not the way things are going, e.g. Parsoid.

Any questions?