Hacking Mediawiki

By Arthur Richards
Software Engineer, Wikimedia Foundation
arichards@wikimedia.org
IRC: awjr
[[User:awjrichards]]
What is Mediawiki anyway?

- GPL server-based Wiki software
- PHP and MySQL
- Community developed and maintained
- Powers Wikipedia and other Wikimedia projects
Welcome to Wikimedia

Wikimedia is owned and operated by the Wikimedia Foundation, a non-profit foundation dedicated to bringing free content to the world. The various Wikimedia projects are listed below:
Ok, but what is a 'wiki'?!

“A wiki (ˈwɪki/ WIK-ee) is a website that allows the creation and editing of any number of interlinked web pages via a web browser using a simplified markup language or a WYSIWYG text editor.”

Editing GNUnify

From Wikipedia, the free encyclopedia

[[File:Gnunify.png|thumb|alt-GNUnify Logo|GNUnify Logo]]

GNUnify is an annual gathering consisting of Workshops, Talks, Seminars and BOPs (Birds of a feather) held by the [[Pune Linux User Group]] [http://plug.org.in/] and the [[Symbiosis Institute of Computer Studies and Research (SICSR)|Symbiosis Institute of Computer Studies and Research]] in [[Pune]], [[India]]. It is an event held to help increase the awareness of [[FOSS]] in India especially in Pune. It's a forum to unite open minds.

GNUnify (a play on the words [[GNU]] and Unify) is generally held during the second week of February. GNUnify has featured eminent people like [[Richard Stallman]], [[David Axmark]] <ref name="richard-stallman-gnunify">[http://gnu.gds.tuwien.ac.at/www/wn/events.html FSF India's events page in 2003]</ref>, [[Brian Behlendorf]], [[David Axmark]] <ref name="technetra-gnunify-2008">http://opensourcebuzz.technetra.com/2008/02/18/gnunify-2008-it-only-gets-better Article in Feb 2008 on Technetra's Blog]</ref>, [[Danese Cooper]] <ref name="danese-blog">[http://danese.cooper.blogs.com/diva/blog/2006/02/gnunify_2006.html Danese Cooper's Blog entry on GNUnify]</ref>, [[Robert Adkins]], [[Alalita Sharma]], Prof. [[Tony Wasserman]], Prof. [[Louis Pott]], Nayan Bhusan, Sankarshan, Zabidee Bohra and Anant Narayanan. <ref name="technetra-gnunify-2009">http://opensourcebuzz.technetra.com/2009/03/08/gnunify-2009-community-support-matters Technetra's Blog Entry on GNUnify 2009]</ref> This year GNUnify is going to be held in February on 11th and 12th with the name GNUnify'11. The meetings have already started and the students of SICSR are ready with their ideas.

GNUnify 2011 is scheduled on 11 & 12 February 2011.

== GNUnify 2011 ==

As stated above, GNUnify 2011 is scheduled to be held on the 11th and 12 of February, 2011.
GNUnify

From Wikipedia, the free encyclopedia

Revision as of 20:14, 26 January 2011 (edit)
Blrajdarpriyanka22 (talk | contribs)
(del/undel)
← Previous edit

Current revision as of 08:27, 7 February 2011 (edit) (undo)
Giraffedata (talk | contribs)
("comprising of")
(del/undel)

Line 1:

GNUnify is an annual gathering comprising of Workshops, Talks, Seminars and BOFs (Birds of a feather) held by the [[Pune Linux User Group]] [http://plug.org.in/] and the [[Symbiosis Institute of Computer Studies and Research (SICS)]] in [[Pune]], [[India]]. It is an event held to help increase the awareness of [[FOSS]] in India especially in Pune. Its a forum to unite open minds.

Line 1:

GNUnify is an annual gathering consisting of Workshops, Talks, Seminars and BOFs (Birds of a feather) held by the [[Pune Linux User Group]] [http://plug.org.in/] and the [[Symbiosis Institute of Computer Studies and Research (SICS)]] in [[Pune]], [[India]]. It is an event held to help increase the awareness of [[FOSS]] in India especially in Pune. Its a forum to unite open minds.

Why should I hack Mediawiki?

- Because you can
- Fix a problem with the software
- 'Scratch an itch'
- Public work record
- Mentor and be mentored
- Put the 'ww' in 'www'
- Support an awesome vision
Hacking the Software: What to Hack

- Bug fixes (http://bit.ly/geY1u0)
- Core code (http://bit.ly/geY1u0)
- Parser hooks
- External tools with the API (http://bit.ly/eBloi)
- SpecialPages (http://bit.ly/323H1o)
Diving in with VariablePage extension

http://www.flickr.com/photos/aknacer/2588798719/
VariablePage

$ svn co http://svn.wikimedia.org/svnroot/mediawiki/trunk/extensions/VariablePage

- VariablePage_bzr
- VariablePage_trunk
  - VariablePage.alias.php
  - VariablePage.body.php
  - VariablePage.i18n.php
  - VariablePage.php
Setup: ExtensionName.php

Goal:
To simplify and centralize installation and configuration

```php
<?php
require_once("$IP/extensions/ExtensionName/ExtensionName.php");
$wgExtNameFoo = true;
$wgExtNameBar = 'baz';
...
```
Setup… continued

- Possible to defer extension setup until after LocalSettings.php has been run

```php
$wgExtensionFunctions[] = 'efExtensionNameSetup';

function efExtensionNameSetup() {
    # do post-setup stuff here
}
...
Setup: database tables

```php
... 
40 # Schema updates for update.php
41 $wgHooks['LoadExtensionSchemaUpdates'][] = 'fnMyHook';
42 function fnMyHook() {
43     global $wgExtNewTables, $wgExtModifiedFields;
44     $wgExtNewTables[] = array(
45         'tablename',
46         dirname(__FILE__) . '/table.sql'
47     );
48     $wgExtModifiedFields[] = array(
49         'table',
50         'field_name',
51         dirname(__FILE__) . '/table.patch.field_name.sql'
52     );
53     return true;
54 } 
55 ...
```

Execution: VariablePage.body.php

- 'VariablePage' is a 'SpecialPage' extension
- function execute() {}
- http://path/to/mediawiki/Special:VariablePage
- For details on how to set up other types of extensions, see: http://bit.ly/eaDnLT
Coding Best Practices

- Security
- Scalability/Performance
- Security
- Concurrency
- Security
Security is Important. Really.

- Insecure extension in SVN = security risk for unwitting 3rd party admins and their users
- Insecure extension Wikipedia = potential security risk for hundreds of millions of users
Common Vulnerabilities to Avoid

- SQL Injection
- Cross-site scripting (XSS)
- Cross-site request forgery (CSRF)
SQL Injection

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.

OH, DEAR - DID HE BREAK SOMETHING? IN A WAY-

DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?

OH, YES. LITTLE BOBBY TABLES, WE CALL HIM.

WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.

AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.
SQL injection

Problem:

```php
$sql = "INSERT INTO Students VALUES ($name, ... );";
INSERT INTO Students VALUES ( 'Robert' );
DROP TABLE Students; --', ... );
```
SQL injection

Problem:

```
$sql = "INSERT INTO Students VALUES ($name, ... );"
INSERT INTO Students VALUES ( 'Robert' ); DROP TABLE Students; --', ... );
```

Fix:

```
INSERT INTO Students VALUES ( 'Robert\'); DROP TABLE Students; -', ... );
```
Prevent SQL Injection with MW functions

BAD:

```php
$dbr->query( "SELECT * FROM foo WHERE foo_id=' $id''" );
```

Acceptable:

```php
$escID = $dbr->addQuotes( $id );
$dbr->query( "SELECT * FROM foo WHERE foo_id= $escID" );
```

Correct:

```php
$dbr->select( 'foo', '*', array( 'foo_id' => $id ) );
```
Imagine:

40  $val = $wgRequest->getVal( 'input' );
41  $wgOut->addHTML( "<input type="text" value="\$val" />");
Imagine:

40 $val = $wgRequest->getVal( 'input' );
41 $wgOut->addHTML( "<input type="text" value="$val" />");

User submits:

"/> <script> do evil stuff </script>"
Imagine:

```php
40 $val = $wgRequest->getVal( 'input' );
41 $wgOut->addHTML( "<input type="text" value=""$val" 
/>" );
```

User submits:

```
"/><script>do evil stuff</script>
```

**EVIL STUFF GETS EXECUTED!!!**
Preventing XSS

- ALWAYS escape inputs

```php
40 // better
41 $val = htmlspecialchars( $val );
42 $html = "<input type="text" name="foo" value="$.val" />";
43
44 // best, using Mediawiki functions
45 $html = Html::input( 'foo', $val );
46 ...
```

EVIL STUFF DOESN'T GET EXECUTED :D
Cross Site Request Forgery (CSRF)

Insecure extension code:

```php
40  ...
41 global $wgUser;
42 if ( $wgUser->isAllowed( 'delete' ) && isset( $_POST['delete'] ) ) {
43   $this->deleteItem( $_POST['delete'] );
44 }
45  ...
```
CSRF

Attack Vector:

'Bob' is logged in to Wikipedia. Mallory lures Bob to a website with the following HTML:

```html
```
CSRF

Attack Vector:

'Bob' is logged in to Wikipedia. Mallory lures Bob to a website with the following HTML:

```html
```

Article gets deleted by Bob, but he doesn't know!!!
CSRF Prevention

```php
... global $wgUser, $wgOut, $wgRequest;
...

$html .= Html::hidden( 'token',
    $wgUser->editToken();

$wgOut->addHthml( $html );
...
...

$token = $wgRequest->getText( 'token' )
if ( $wgUser->isAllowed( 'delete' )
    && isset( $_POST['delete'] )
    && $wgUser->matchEditToken( $token ) ) {
    $this->deleteItem( $_POST['delete'] );
}
...```
Scalability, Performance and Concurrency

Typical LAMP setup:
Scalability, Performance and Concurrency

On steroids:
Scalability, Performance and Concurrency

- Secure
- Performant
- Secure
- Scalable
- Secure
- Tolerant of concurrency
- Secure
I18n: VariablePage.i18n.php

• Translate performed by volunteers via translatewiki.net
• Putting the 'ww' in 'www'
• Even if you only know one language, your extension can be globally translingual
• Translations to your code happen automatically by SVN commits from translatewiki.net
MediaWiki:Viewdeleted_short/hi

Suggestions from translation memory
- {{नस्कर| $1 | एक संपादन न कर दिया | $1 हटा}} बंद
- {{हूस्न | नस्कर| $1| एक संपादन न कर दिया | $1 हटा}} बंद
- {{PLURAL:$1|एक हटाया हुआ|$1 हटाये हुए}} बंद

Information about message “Viewdeleted_short” (contribute)
Tab label for the undelete button when the user has permission to view the deleted history but not undelete.

Message definition (MediaWiki)
- View {{PLURAL:$1|one deleted edit|$1 deleted edits}}

Summary:

<table>
<thead>
<tr>
<th>MediaWiki</th>
<th>Current message text</th>
</tr>
</thead>
<tbody>
<tr>
<td>↓log-nocache</td>
<td><img src="%E2%86%93log-nocache" alt="FUZZY!पवे केबल में ना सर्च" /></td>
</tr>
<tr>
<td>↓log-externaleditor</td>
<td><img src="%E2%86%93log-externaleditor" alt="FUZZY!पूल स्वामित्व में बाहर संपादन पंजी का प्रयोग करें (केवल विशेषज्ञों के लिए, इसके लिए संगणक पर खाली जमा करना चाहिए होगी)" /></td>
</tr>
<tr>
<td>↓log-externaldiff</td>
<td><img src="%E2%86%93log-externaldiff" alt="FUZZY!पूल स्वामित्व में बाहर देखने के लिए बाहरी तंत्र का प्रयोग करें (केवल विशेषज्ञों के लिए, इसके लिए संगणक पर खाली जमा करना होगी)" /></td>
</tr>
<tr>
<td>↓viewdeleted_short</td>
<td>View {{PLURAL:$1</td>
</tr>
<tr>
<td>↓pool-timeout</td>
<td>Timeout waiting for the lock</td>
</tr>
<tr>
<td>↓pool-queuefull</td>
<td>Pool queue is full</td>
</tr>
<tr>
<td>↓welcomecreation</td>
<td>स्वागत स्वागत</td>
</tr>
<tr>
<td>↓php-mail-error-unknown</td>
<td>Unknown error in PHP's mail() function</td>
</tr>
</tbody>
</table>
I18n Best Practices

- 'qqq'
- \( \text{wfmsg}() \);
- Only make message changes to 'en'
- Remove unused messages (only from 'en')
I18n Best Practice Highlights

- Gender-specific, plurals, parameters all supported
- Avoid patchwork messages, but avoid message reuse
- Separate date and times in messages
- Do not include CSS/Javascript/HTML/etc
- Think about both LTR and RTL
- Avoid jargon/slang
I18n: not exactly intuitive

- Tough for new and veteran developers
- Thoroughly read the i18n guide for more: http://bit.ly/fjYtLX
- TALK TO TRANSLATORS!
  - #mediawiki-i18n (irc.freenode.net)
  - http://translatewiki.net/wiki/wiki/Support
How to Engage the Community

• Discuss, Engage, Participate
• Mailing lists: (http://bit.ly/77lNC7)
• IRC (#mediawiki on irc.freenode.net)
• Comment and document
• Commit your code (http://bit.ly/hsTaI7)
Community engagement best-practices

- Be patient...
- But don't expect patience
- RTFM
- Communicate changes
- Be concise
- Be HELPFUL
- Give credit where credit is due
- Return the favor
- CONTRIBUTE
IRC

mediawiki
Open Source Software Pro-tip

“The goal should be a solution to the problem – not simply inclusion of your code.”

- Jonathan Corbet (paraphrased from keynote address at FOSDEM 2011)
Absolutely Essential Reading

- How to become a Mediawiki Hacker: http://bit.ly/2rSaLX
Key Resources

- Http://www.mediawiki.org
- Http://wikitech.mediawiki.org
- http://svn.wikimedia.org/doc/
- http://www.mediawiki.org/wiki/API
- http://www.mediawiki.org/wiki/How_to_become_a_MediaWiki_hacker

IRC

- #mediawiki
- #mediawiki-dev
Any Questions?! 

Special thanks to:

- Ryan Lane
- Roan Kattouw
- Tomasz Finc
- Danese Cooper
- Alolita Sharma
- Harshad, the rest of the staff and all of the awesomely energetic/enthusiastic students who helped put GNUUnify on!
- SICSR

**Presentation slides can be found at: http://bit.ly/fJVTIN**
Self-introduction

- Name, where from, etc
- Time involved in FLOSS
- Time with the foundation
- What sort of stuff I work on
- Expertise (or lack of) in Mediawiki
- I'm a new Mediawiki Hacker!
What is Mediawiki anyway?

- GPL server-based Wiki software
- PHP and MySQL
- Community developed and maintained
- Powers Wikipedia and other Wikimedia projects

GPL
Software stack
Community developed/maintained
Powers Wikipedia, other projects
Runs on basic LAMP
  Available for download/install by ANYONE
Welcome to Wikimedia

Wikimedia is owned and operated by the Wikimedia Foundation, a non-profit foundation dedicated to bringing free content to the world. The various Wikimedia projects are listed below:

- Wikipedia
- Wiktionary
- Wikibooks
- Wikisource
- Wikinews
- Wikiquote
- Wikiversity
- Wikispecies
- MediaWiki
- Wikimedia Commons
- Wikimedia Incubator

Brief highlight of other projects

All powered by... Mediawiki!
Supported by the Foundation
Ok, but what is a 'wiki'?!

“A wiki (ˈwɪki/ WIK-ee) is a website that allows the creation and editing of any number of interlinked web pages via a web browser using a simplified markup language or a WYSIWYG text editor.”


- WikiWikiWeb by Ward Cunningham 1994
- So named b/c of Honolulu Airport employee saying 'take “wiki wiki shuttle”'
- Wiki literally means 'quick'
- Wiki wiki sounds cooler than 'quick web'
- Intended for serious collaboration rather than just casual visitors
- Revision control – easy to CORRECT mistakes rather than making it difficult to MAKE them
- Invites ANY user to edit a web page in a web browser w/o addons
- Wiki markup for styling
- The idea is to make it easy to edit and apply styling without knowledge of HTML
• Intended for serious collaboration rather than just casual visitors
• Revision control – easy to CORRECT mistakes rather than making it difficult to MAKE them
The diffs, to see what changed
Why should I hack Mediawiki?

- Because you can
- Fix a problem with the software
- 'Scratch an itch'
- Public work record
- Mentor and be mentored
- Put the 'ww' in 'www'
- Support an awesome vision

Imagine a world in which every single human being can freely share in the sum of all knowledge.

* Global community
* Why is the community awesome?
Hacking the Software: What to Hack

- Bug fixes (http://bit.ly/geY1u0)
- Core code (http://bit.ly/geY1u0)
- Parser hooks
- External tools with the API (http://bit.ly/eBloi)
- SpecialPages (http://bit.ly/323H1o)

Template hacking/parser hooks:
parser function extensions for pimping out templates {to handle wiki text generation that involves logic that is too complex or confusing to write using normal http://www.mediawiki.org/w/index.php?title=Manual:Developing_extensions template-writing techniques.}
variable extensions extending parameters in templates extending syntax, etc
Diving in with VariablePage extension

Extension developed for the fundraiser

To facilitate A/B testing

Gives you the ability to pseudo-randomly send users to a particular page x% of the time

How we used it during the fundraiser
We have a SVN repository that is free and accessible for anyone to check out from.

To be able to commit, you need special permission

How to get the extension

Code layout
Setup: ExtensionName.php

Goal:
To simplify and centralize installation and configuration

```php
<?php
require_once("$IP/extensions/ExtensionName/ExtensionName.php");
$wgExtNameFoo = true;
$wgExtNameBar = 'baz';
...
```

Open VariablePage.php

define/validate configuration variables in global scope
over-rideable in LocalSettings
Give them GOOD UNIQUE NAME
extensive documentation = GOOD
auto-load classes
immediate/deferred setup
define hook functions

Open up LocalSettings.php
Setup... continued

- Possible to defer extension setup until after LocalSettings.php has been run

```php
<<?php
...
...
...
$wgExtensionFunctions[] = 'efExtensionNameSetup';
function efExtensionNameSetup() {
    # do post-setup stuff here
}
...
```

This is handy for using variables/scripts that might not be loaded when LocalSettings first gets to your script
Setup: database tables

```php
# Schema updates for update.php

$wgHooks['LoadExtensionSchemaUpdates'][] = 'fnMyHook';

function fnMyHook()
{
    global $wgExtNewTables, $wgExtModifiedFields;
    $wgExtNewTables[] = array('tablename',
                             dirname(__FILE__) . '/table.sql');
    $wgExtModifiedFields[] = array('table',
                                     'field_name',
                                     dirname(__FILE__) . '/table.patch.field_name.sql');
    return true;
}
```


Not covering in detail

Possible to load db schema stuffs via setup file
Execution: VariablePage.body.php

- 'VariablePage' is a 'SpecialPage' extension
- function execute() {}
- http://path/to/mediawiki/Special:VariablePage
- For details on how to set up other types of extensions, see: http://bit.ly/eaDnLT

Open VariablePage.body.php

Demonstrate

Show execution, go over code
Coding Best Practices

- Security
- Scalability/Performance
- Security
- Concurrency
- Security

Did I mention, 'Security'?
Security is Important. Really.

- Insecure extension in SVN = security risk for unwitting 3rd party admins and their users
- Insecure extension Wikipedia = potential security risk for hundreds of millions of users
Common Vulnerabilities to Avoid

- SQL Injection
- Cross-site scripting (XSS)
- Cross-site request forgery (CSRF)

There are of course others, but these are the most common ones that are easy to defend against.
SQL Injection

Haha
SQL injection

Problem:

```php
$sql = "INSERT INTO Students VALUES ($name, ... );";
INSERT INTO Students VALUES ( 'Robert' ); DROP TABLE Students; --', ... );
```

User inserts malicious query into a text input or whatevs, which can cause bad things to happen.
SQL injection

Problem:

```php
$sql = "INSERT INTO Students VALUES ($name, ... );";
INSERT INTO Students VALUES ('Robert'); DROP TABLE Students; --', ...
```

Fix:

```php
INSERT INTO Students VALUES ('Robert\'); DROP TABLE Students; --', ...
```
Prevent SQL Injection with MW functions

BAD:
```php
$dbr->query( "SELECT * FROM foo WHERE foo_id='\$id'" );
```

Acceptable:
```php
$escID = $dbr->addQuotes( \$id );
$dbr->query( "SELECT * FROM foo WHERE foo_id= \$escID" );
```

Correct:
```php
$dbr->select( 'foo', '*', array( 'foo_id' => \$id ) );
```

Considered extra nice if you use the built-in mediawiki functions (roan says Tim will like you more)
Imagine:

```php
40 $val = $wgRequest->getVal( 'input' );
41 $wgOut->addHTML( "<input type="text" value="$val" />" );
```
Imagine:

40 $val = \$wgRequest-\gt getVal( 'input' );
41 $wgOut-\gt addHTML( "<input type="text" value="$val" />" );

User submits:

"<script>do evil stuff</script>"
Exploits trust a user has in a site, or link

One of many scenarios:

* Bob's website has an XSS vulnerability
* Alice crafts a URL with malicious code in the $_GET or $_POST, and sends it to Bob
* Bob clicks link, executing Alice's malicious script in Bob's browser
* Could allow Alice to steal sensitive information otherwise only available to Bob
Preventing XSS

- ALWAYS escape inputs

```php
...
40 // better
41 $val = htmlspecialchars( $val );
42 $html = "<input type="text" name="foo" value="$val" />";
43
44 // best, using Mediawiki functions
45 $html = Html::input( 'foo', $val );
46 ...
```

EVIL STUFF DOESN'T GET EXECUTED :D

MW functions like Html::input() automagically sanitize input so you don't have to worry about it
Cross Site Request Forgery (CSRF)

Insecure extension code:

```php
40 ...  
41 global $wgUser;
42 if ($wgUser->isAllowed('delete') && isset($_POST['delete'])) {
43   $this->deleteItem($_POST['delete']);
44 }
45 ...
```

Sometimes pronounced 'Sea Surf' because it allows for 'session riding' – essentially hijacking a user's session.

Exploits the trust a site has in a browser
CSRF

Attack Vector:

'Bob' is logged in to Wikipedia. Mallory lures Bob to a website with the following HTML:

```html
```
CSRF

Attack Vector:

'Bob' is logged in to Wikipedia. Mallory lures Bob to a website with the following HTML:

```
```

Article gets deleted by Bob, but he doesn't know!!!
CSRF Prevention

```php
global $wgUser, $wgOut, $wgRequest;

$html .= Html::hidden( 'token',
    $wgUser->editToken();
$wgOut->addHthml( $html );

$token = $wgRequest->getText( 'token' )
if ( $wgUser->isAllowed( 'delete' )
    && isset( $_POST['delete'] )
    && $wgUser->matchEditToken( $token )
) {
    $this->deleteItem( $_POST['delete'] );
}
```

Easy to prevent with token checking

* token gets generated and stored in a cookie
* a hash of token (plus a salt) gets stored in a hidden form field
* on form submit, logic checks to see if the salt + token hash from the cookie matches that of the form submit

If mismatch, session is considered 'over' and request invalid
Scalability, Performance and Concurrency

Typical LAMP setup:

Mediawiki is cool because it can run on such a simple, basic set up
Scalability, Performance and Concurrency

On steroids:

But we put the setup on 'roids.

Introduces a level of complexity that coder most be aware of
### Scalability, Performance and Concurrency

- Secure
- Performant
- Secure
- Scalable
- Secure
- Tolerant of concurrency
- Secure

* High performance code that's not going to crash under load

* Concurrency problems:
  ** On Wikipedia, we use many DB's – selecting which DB to use is possible in the code
  ** Database lag (use DB_MASTER for reads if data needs to be up-to-date)
  ** Attaching timestamps to things like counters is a good idea
I18n: VariablePage.i18n.php

- Translate performed by volunteers via translatewiki.net
- Putting the 'ww' in 'www'
- Even if you only know one language, your extension can be globally translingual
- Translations to your code happen automatically by SVN commits from translatewiki.net

Overview of i18n

You only need to know 1 language, but your code will work in useable by many!

SUPER COOL! Go Translators!!!

Go over the i18n file

Demonstrate on the localhost!
Example of translation interface on translate wiki.

Anyone can do it!

EN → Hindi
I18n Best Practices

- 'qqq'
- `wfmsg();`
- Only make message changes to 'en'
- Remove unused messages (only from 'en')

Be sure to explain your messages in the 'qqq' array – it is reserved for documentation

Typically, messages are initially written in English

Removing your message from English will automatically remove the corresponding messages from other langs
I18n Best Practice Highlights

- Gender-specific, plurals, parameters all supported
- Avoid patchwork messages, but avoid message reuse
- Separate date and times in messages
- Do not include CSS/Javascript/HTML/etc
- Think about both LTR and RTL
- Avoid jargon/slang
I18n: not exactly intuitive

- Tough for new and veteran developers
- Thoroughly read the i18n guide for more: http://bit.ly/fjYtLX
- TALK TO TRANSLATORS!
  - #mediawiki-i18n (irc.freenode.net)
  - http://translatewiki.net/wiki/Support
How to Engage the Community

- Discuss, Engage, Participate
- IRC (#mediawiki on irc.freenode.net)
- Comment and document
- Commit your code (http://bit.ly/hsTalT)
Community engagement best-practices

- Be patient...
- But don't expect patience
- RTFM
- Communicate changes
- Be concise
- Be HELPFUL
- Give credit where credit is due
- Return the favor
- CONTRIBUTE
See, it's neat!

Plus when you contribute or do something with a bug, the bot tells the channel. People watch this and will see what you're up to and often engage YOU
Open Source Software Pro-tip

“The goal should be a solution to the problem – not simply inclusion of your code.”
- Jonathan Corbet (paraphrased from keynote address at FOSDEM 2011)

This is paraphrased.

The point is, LET GO OF CONTROL.
Absolutely Essential Reading

- How to become a Mediawiki Hacker: http://bit.ly/2rSaLX
Key Resources

- Http://www.mediawiki.org
- Http://wikitech.mediawiki.org
- http://svn.wikimedia.org/doc/
- http://www.mediawiki.org/wiki/API
- http://www.mediawiki.org/wiki/How_to_become_a_MediaWiki_hacker

IRC

- #mediawiki
- #mediawiki-dev
Any Questions?!

Special thanks to:
- Ryan Lane
- Roan Kattouw
- Tomasz Finc
- Danese Cooper
- Alolita Sharma
- Harshad, the rest of the staff and all of the awesomely energetic/enthusiastic students who helped put GNUnify on!
- SICSR

**Presentation slides can be found at: http://bit.ly/fJVTIN**