eZ systems

Welcome!

Share your information

eZ systems

The Big Cleanup

- register_globals
- magic_quotes
- safe_mode
- zend.ze1_compatibility mode
- dl()

Share your information

New Stuff

- 64 bit integer
- goto
- late static binding
- default opcode cache (APC)
- E_STRICT on by default
- Full Unicode support

eZ systems

What is so important about Unicode anyway?

- There is more than one country in the world
- Ce n'est pas tout le monde qui parle anglais
- Tjueseks karakterer holder ikke mål
- Нот эврибади из юзин зэ сэйм скрипт ивэн
- 它变得更加复杂的与汉语语言

Share your information

eZ systems

i18n challenges

- Support for multiple encodings: conversion, detection, processing...
- Support for multiple language in different encodings and scripts

Share your information

eZ systems

Sorting Strings

How would you sort: côté (side), côte (coast), cote (dimension), coté (with dimensions)?

Logical is: cote, coté, côte, côté

But the french do it like: cote, côte, coté, côté

The french are not the only ones with "weird" sorting!

- In Lithuanian, y is sorted between i and k.
- In traditional Spanish ch is treated as a single letter, and sorted between c and d.
- In Swedish v and w are considered variant forms of the same letter.
- In German dictionaries, öf would come before of. In phone books the situation is the exact opposite.

eZ systems

Do We Need Something New?

- PHP only deals with bytes, not characters.
- PHP doesn't know anything about encodings.
- Having a binary image in a string is nice, but not if you need to deal with i18n

Share your information

eZ systems

What Do We Want for PHP?

- Native Unicode strings
- A clear separation between Binary / Native (Encoded) Strings and Unicode Strings
- Unicode string literals
- Updated language semantics
- Where possible, upgrade the existing functions
- Backwards compability
- PHP should do what most people will expect
- Make complex things possible, without making using strings in PHP complex
- Must be as good as Java's support

eZ systems

How Do We Turn It On?

- With an INI setting: unicode_semantics
- It can be turned on "per-vhost"
- (Almost) no behavioral changes when it's not enabled
- The setting does not mean you won't have any Unicode strings

Share your information

eZ systems

String Types

- string: Used to represent binary data, for example the contents of a JPEG file or a "native" string.

```
. P N G . . . . . . . . I H D R
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52


B l å     b æ     r ø     l
62 6C C3 A5 62 C3 A6 72 C3 A8 6C
```

- unicode: Strings, internally encoded in UTF-16.

```
B     l     å     b     æ     r     ø     l
62 00 6C 00 E5 00 62 00 E6 00 72 00 F8 00 6C 00
```

Share your information

eZ systems

String Literals

Unicode Semantics are off:

```
<?php // script is encoded in UTF-8
$str = "hallo daar!";
echo gettype($str), ': ', strlen($str), "\n";

$str = "привет!";
echo gettype($str), ': ', strlen($str), "\n";
?>
```

outputs:

```
string: 11
string: 13
```

Unicode Semantics are on:

```
<?php // script is encoded in UTF-8
$str = "hallo daar!";
echo gettype($str), ': ', strlen($str), "\n";

$str = "привет!";
echo gettype($str), ': ', strlen($str), "\n";
?>
```

outputs:

```
unicode: 11
unicode: 7
```

Share your information

eZ systems

Script Encoding

Interpreting an iso-8859-1 script as UTF-8:

```
<?php
    declare(encoding="iso-8859-1");
    $str = "blå = 青";
    var_inspect($str);
?>
```

Interpreting an UTF-8 script as UTF-8:

```
<?php
    declare(encoding="utf-8");
    $str = "blå = 青";
    var_inspect($str);
?>
```

Share your information

eZ systems

Characters, not Bytes!

- All functions and operators work on Code Points (characters) and not Code Units (bytes)
- Backward compatible if you only used single byte encodings before
- This does create overhead though, as we need to scan through a whole string

String Indexes:

```
<?php
    $string = "旭    "; // bytes are: 0xE9 0xBB 0x83 0xE6 0x97 0xAD

    echo $string[1];
?>
```

Share your information

ICU Locales

- ICU comes with it's own Locale information
- PHP currently uses POSIX locales for some functions only
- Those functions need to be modified

eZ systems

ICU Locales

```
<?php
    i18n_loc_set_default("nl");
    echo strtotitle("het ijsselmeer (ijsselmeer) is ßaf"), "<br/>\n";

    i18n_loc_set_default("tr");
    echo strtotitle("het ijsselmeer (ijsselmeer) is ßaf"), "\n";
?>
```

Share your information

eZ systems

ICU Locales

```
<pre># orig norm loc trad
--------------------------
<?php
    $d = $c = $b = $a = array('mapa', 'kilo', 'libro', 'llave', 'loca');
    sort($b);
    i18n_loc_set_default('es_VE');
    sort($c, SORT_LOCALE_STRING);
    i18n_loc_set_default('es_VE@collation=traditional');
    sort($d, SORT_LOCALE_STRING);

    for ($i = 0; $i < 5; ++$i) {
        echo sprintf('%d. %-5s %-5s %-5s %-5s<br/>',
            $i + 1, $a[$i], $b[$i], $c[$i], $d[$i]);
    }
?>
```

Share your information

eZ systems

i18n Extensions

- ext/unicode: always enabled, functions to set the locale, the locale aware string functions. String searching and collation API.
- i18n_regexp: Unicode aware regular expressions
- i18n_translit: PECL/translit extension, integrated with ICU's transliteration
- i18n_...: Other specialized extensions, such as normalization, break iteration...

eZ systems

When Can We Have This?

- When it is ready.
- Development version is in CVS.
- Release: hopefully at the end of 2006.

Share your information

eZ systems

Share your information

Resources

- PDM notes: http://php.net/~derick/meeting-notes.html
- This presentation: http://derickrethans.nl/talks.php
- Questions: dr@ez.no