

Mehrsprachigkeit

PHP Usergroup Würzburg / 30.11.2006
Florian Eibeck - f@eibeck.de

Mehrsprachigkeit

Warum Mehrsprachigkeit?

Die meisten Programmierer machen sich erst Gedanken um Mehrsprachigkeit, wenn ein Problem auftritt.

Das Problem sind dann Fragezeichen, Kästchen und andere komische Symbole auf der eigenen Webseite.

Mehrsprachigkeit

- Viele Fehlinformationen und Ausreden
 - Mit PHP unmöglich
 - Zu schwer
 - Aufwand lohnt sich nicht
- Begriffe unklar:
 - Internationalisierung (I18N)
 - Lokalisierung (L10N)
 - Zeichensatz / Character Set
 - Zeichenkodierung / Character Encoding
 - Unicode? ISO-8859-1? UTF-8?

Mehrsprachigkeit

Internationalisierung

Internationalisierung bedeutet in der Informatik bzw. in der Softwareentwicklung, ein Programm so zu gestalten, dass es leicht (ohne den Quellcode ändern zu müssen) an andere Sprachen angepasst werden kann.

Internationalisierung (engl. *internationalization*) wird im englischen Sprachraum oft mit **I18N** abgekürzt. Die Schreibweise I18N bezieht sich auf die Anzahl der ausgelassenen Buchstaben der englischen Schreibweise, also **18** (`nternationalizatio`) im Wort `internationalization`.

Quelle: Wikipedia

Mehrsprachigkeit

Lokalisierung

Lokalisierung steht in der Softwareentwicklung für die Anpassung von Inhalten, Prozessen, Produkten und insbesondere Computerprogrammen an die in einem bestimmten geografisch oder ethnisch umschriebenen Absatz- oder Nutzungsgebiet (Land, Region oder ethnische Gruppe) vorherrschenden "lokalen" sprachlichen und kulturellen Gegebenheiten.

Quelle: Wikipedia

Analog zu Internationalisierung wird Lokalisierung als L10N abgekürzt.

Mehrsprachigkeit

Zeichensatz

Zeichensatz (engl. Character Set) bezeichnet einen Vorrat an Symbolen zum Darstellen von Sachverhalten.

Dies sind normalerweise Buchstaben, Zahlen, Satzzeichen, Sonderzeichen.

Ein Zeichensatz ist somit eine Gruppe von Zeichen.

Beispiele:

- Lateinisches Alphabet
- Kyrillisches Alphabet
- Griechisches Alphabet

Mehrsprachigkeit

Zeichenkodierung

Zeichenkodierung (engl. Character Encoding) bezeichnet die Technik, wie ein Schriftzeichen mithilfe eines Codes gespeichert werden kann.

Beispiele:

- ASCII
- ISO-8859
- UTF-8
- UTF-16

Mehrsprachigkeit

ASCII

ASCII (*American Standard Code for Information Interchange*) ist eine 7-bittige Zeichenkodierung.

Erfunden: 1967

33 nicht druckbare und
95 druckbare Zeichen

Die Zeichen umfassen das lateinische Alphabet in Groß- und Kleinschreibung, die zehn Ziffern, sowie einige Satz- und Steuerzeichen. Der Zeichenvorrat entspricht dem Zeichensatz, der für die englische Sprache benötigt wird.

Mehrsprachigkeit

ISO-8859

Standard der *International Organization for Standardization*.

Definiert momentan 15 verschiedene 8-Bit Zeichensätze. Der bekannteste für uns ist wohl ISO-8859-1 (auch Latin-1) für westeuropäische Sprachen.

256 mögliche Zeichen, die ersten 128 sind kompatibel mit ASCII.

Windows CP1252

Von Microsoft eingeführte und verwendete Kodierung. Stimmt zum größten Teil mit ISO-8859-1 überein. Einige Details unterscheiden sich aber, was zu Interoperabilitätsproblemen führen kann.

Mehrsprachigkeit

Das Problem von allen Single-Byte Kodierungen ist die beschränkte Anzahl an kodierbaren Zeichen.

Interoperabilität zwischen verschiedenen System ist damit nicht gewährleistet.

Folge: Einführung von Zeichenkodierungen, die mehr als ein Byte für die Kodierung eines Zeichens brauchen.

Mehrsprachigkeit

Unicode

Unicode (ISO 10646), auch UCS (Universal Character Set) ist ein Zeichensatz, der alle auf dieser Welt benutzten Zeichen enthält.

Jedes Zeichen hat eindeutig Nummer (oder ist zusammengesetzt aus mehreren Zeichen).

- Entwickelt vom Unicode Consortium seit 1991
- Maximal 1.114.112 mögliche Zeichen
- In Unicode 5.0 (Juli 06) sind 99.089 Zeichen definiert
- Aufteilung in mehrere sog. Codepages

Mehrsprachigkeit

UTF-8

UTF-8 ist eine Zeichenkodierung für Unicode. Mit ihr ist es möglich alle Zeichen aus Unicode zu kodieren.

Multibyte – Characters können zwischen einem und sechs Byte groß sein. Die ersten 128 Zeichen sind kompatibel mit ASCII.

UTF-8 ist von der IETF, dem Unicode Consortium und der ISO als Standard definiert:

- RFC 3629
- *The Unicode Standard, Version 4.0*, §3.9 - §3.10 (2003)
- ISO/IEC 10646-1:2000 Annex D (2000)

Mehrsprachigkeit

UTF-8

UTF-8 hat mittlerweile eine sehr weite Verbreitung und Unterstützung. Alle modernen Betriebssysteme, Browser, Editoren und andere Software bieten UTF-8 Unterstützung.

Damit ist UTF-8 die Lösung des Problems. Alle Zeichen dieser Welt können dargestellt werden, die Kästchen und Fragezeichen verschwinden von der Webseite.

Es braucht aber ein paar Schritte um UTF-8 mit PHP ohne Probleme einsetzen zu können.

Mehrsprachigkeit

Praktischer Einsatz: UTF-8 in HTML

Als Test wird der String **Iñtërnâtiônàlizætiøn** in ein HTML-Dokument eingebaut:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>Mehrsprachigkeit</title>
  </head>
  <body>
    <p>Iñtërnâtiônàlizætiøn</p>
  </body>
</html>
```

Mehrsprachigkeit

The screenshot shows a Mozilla Firefox browser window titled 'Mehrsprachigkeit - Mozilla Firefox'. The address bar shows 'http://www.sau/mehrsprachigkeit/test1.ht'. The page content is garbled text: 'IÃ±tÃ«rnÃ¶tiÃ´nÃ lizÃ!tiÃ,n'. A 'Seiteninformation' (Page Information) dialog box is open, showing the following details:

- Mehrsprachigkeit:**
- Adresse: http://www.sau/mehrsprachigkeit/test1.ht
- Typ: text/html
- Anzeigemodus: Standardkonformer Modus
- Cache-Quelle: Festplatten-Cache
- Kodierung: ISO-8859-1
- Größe: 0,57 KB (582 Bytes)
- Verweisende URL: http://www.sau/mehrsprachigkeit/
- Modifiziert: 20.10.2006 14:33:18
- Gültig bis: 30.11.2006 09:44:17

Meta:

Name	Inhalt	↕
Content-Type	text/html; charset=utf-8	

Mehrsprachigkeit

Praktischer Einsatz: UTF-8 in HTML

Das Problem:

Der Server liefert das Dokument als ISO-8859-1, obwohl es in UTF-8 kodiert ist.

Der Browser ignoriert normalerweise die Angabe im Kopf der HTML Datei.

Die Lösung:

Die Kodierung im Content Type des HTTP-Headers mitsenden.

Mehrsprachigkeit

Praktischer Einsatz: UTF-8 in HTML

PHP:

```
header('Content-Type: text/html; charset=utf-8');
```

Apache:

```
AddDefaultCharset On  
AddDefaultCharset utf-8
```

in httpd.conf oder .htaccess

Mehrsprachigkeit

The screenshot shows a Mozilla Firefox browser window with the title 'Internationälizætïon - Mozilla Firefox'. The address bar shows the URL 'http://www.sau/mehrsprachigkeit/test1.ph'. The page content displays the text 'Internationälizætïon'. An 'Seiteninformation' (Page Information) dialog box is open, showing the following details:

Internationälizætïon:

- Adresse: http://www.sau/mehrsprachigkeit/test1.ph
- Typ: text/html
- Anzeigemodus: Standardkonformer Modus
- Cache-Quelle: Festplatten-Cache
- Kodierung: UTF-8
- Größe: 0,38 KB (393 Bytes)
- Verweisende URL: Nicht angegeben
- Modifiziert: 26.11.2006 17:35:47
- Gültig bis: Nicht angegeben

Meta:

Name	Inhalt
Content-Type	text/html; charset=utf-8

The browser status bar at the bottom shows 'Fertig' and 'Adblock'.

Mehrsprachigkeit

Es gibt einiges zu Beachten:

Daten-Ausgabe

- HTML
- Webservices
- Feeds

Daten-Eingabe

- Formulare
- Webservices
- Feeds

Daten-Verarbeitung

- PHP

Daten-Speicherung

- MySQL

Mehrsprachigkeit

Ausgabe

Wie im vorherigen Beispiel gesehen ist es wichtig, immer die Kodierung bei der Ausgabe anzugeben. Dies verhindert, daß ein Client die gesendeten Daten falsch anzeigt oder interpretiert.

PHP:

z.B.:

```
header('Content-Type: text/html; charset=utf-8');
```

oder:

```
header('Content-Type: text/xml; charset=utf-8');
```

Mehrsprachigkeit

Eingabe: Formulare

Für alle Formulare sollte die erwartete Zeichenkodierung angegeben werden:

```
<form accept-charset="utf-8">...</form>
```

Falls keine Kodierung, oder kein UTF-8 angegeben ist, kann man serverseitig nicht mehr feststellen, was der Browser eigentlich gesendet hat. Zum Beispiel würde ein Formular in einer Webseite, die als ISO-8859-1 an den Browser gesendet wurde, normalerweise auch in dieser Kodierung zurückgeschickt. Probleme entstehen dann sobald ein Benutzer Zeichen eingibt, die in dieser Kodierung nicht enthalten sind.

Mehrsprachigkeit

Eingabe: Webservices, Feeds

Beim einlesen von Daten aus externen Quellen (Webservices, Feeds) muss auf die verwendete Kodierung geachtet werden.

Vorsicht ist geboten bei unterschiedlichen Angaben im Header und im Content.

Falls Daten nicht als UTF-8 gesendet werden ist es sinnvoll, diese gleich nach Empfang in UTF-8 umzuwandeln.

Mehrsprachigkeit

PHP & UTF-8

PHP: Ein Zeichen entspricht einem Byte

```
$cnt = strlen('Internâtiônâlizætïøn');
```

Die Variable `$cnt` hat nach dem Aufruf den Wert 27.

Viele der String-Funktionen von PHP liefern bei der Verarbeitung von UTF-8 falsche oder ungültige Ergebnisse. Abhilfe schafft hier die `mbstring` Extension.

Eine gute Nachricht ist, daß PHP nicht versucht, Strings in eine Kodierung zu konvertieren.

Mehrsprachigkeit

PHP & UTF-8

```
$comment = $_POST['comment'];  
echo 'Foo ' . $comment . ' bar';
```

```
$words = array('foo', 'Iñtërnâtiônàlizætiøn', 'bar');  
$words = implode(' ', $words);
```

```
$string = 'Iñtërnâtiônàlizætiøn';  
print_r(explode('i', $string));
```

```
print_r(explode('à', $string));
```

Da jedes (valides) UTF-8 Zeichen einzigartig ist, kann **à** nicht mit anderen Zeichen verwechselt werden. `explode()` funktioniert deswegen wie erwartet.

Mehrsprachigkeit

mbstring Extension

Die Multi-Byte-String Extension von PHP bringt eine Reihe von String-Funktionen, die mit Multi-Byte Zeichen korrekt umgehen können. Mbstring versteht dabei eine ganze Reihe von Zeichenkodierungen, unter anderem auch UTF-8.

Besonders Interessant dabei ist, dass per Konfiguration in der `php.ini` bestimmte String-Funktionen mit ihren Pendanten aus der mbstring Extension ausgetauscht werden können.

Leider ist mbstring nicht standardmäßig einkompiliert, was die Verwendung bei vielen Hostern unmöglich macht.

Mehrsprachigkeit

iconv Extension

Die iconv Extension ist vor allem für das Konvertieren von Daten von einer in eine andere Zeichenkodierung hilfreich.

Iconv unterstützt eine ganze Reihe von Zeichenkodierungen, welche genau verfügbar sind hängt aber vom benutzten Betriebssystem ab.

Auch iconv ist leider nicht in der Standarddistribution vorhanden.

Mehrsprachigkeit

PHP Helfer-Klassen und Funktionen

PHP UTF-8:

<http://sourceforge.net/projects/phputf8>

UTF-8 helper functions von DokuWiki:

<http://dev.splitbrain.org/view/darcs/dokuwiki/inc/utf8.php>

Mehrsprachigkeit

MySQL & UTF-8

- Ab Version 4.1 gute Unterstützung für UTF-8
- **Vorsicht:** UTF-8 heißt in MySQL UTF8

Zeichenkodierung kann für den Server, Datenbank, Tabelle oder Spalte jeweils einzeln eingestellt werden

```
(CREATE | ALTER) TABLE ... DEFAULT CHARACTER SET utf8
```

```
(CREATE | ALTER) DATABASE ... DEFAULT CHARACTER SET utf8
```

Zusätzlich kann jeweils eine Collation (Sortierreihenfolge) für jede Spalte, Tabelle und Datenbank festgelegt werden.

Mehrsprachigkeit

MySQL & PHP & UTF-8

Die Verbindung des MySQL Clients in PHP zur MySQL Datenbank ist als Standard auf die Zeichenkodierung latin-1 gesetzt.

Bevor UTF-8 Daten in die Datenbank geschrieben oder aus dieser ausgelesen werden sollen, muss die Verbindung auf UTF-8 umgestellt werden. Dies verhindert, dass Daten zerstört werden.

Dazu muss die folgende Query vor allen anderen an MySQL gesendet werden:

```
mysql_query("SET NAMES 'utf8'");
```

Mehrsprachigkeit

PHP 6

PHP 6 wird nativen Support für Unicode (UTF-8) enthalten.

Mehr Infos:

- PHP 6 and Unicode (pdf)

<http://www.gravitonic.com/downloads/talks/php-quebec-2006/php-6-and-unicode.pdf>

- PHP 6 and Unicode (pdf)

<http://derickrethans.nl/files/php6-unicode-ffm2006.pdf>

Mehrsprachigkeit

Fazit

Mit einigen einfachen Vorgehensweisen ist es möglich die eigene Webseite / -anwendung fit für den Internationalen Gebrauch zu machen. Mit der konsequenten Verwendung von UTF-8 lassen sich die Probleme mit unterschiedlichen Zeichensätzen lösen.

Ich hoffe ich konnte einen kleinen Einblick in das Thema verschaffen, der jedoch keinerlei Anspruch auf Vollständigkeit erhebt.

Für weitere Informationen, auch zum Thema Sicherheit mit UTF-8 folgen noch einige Links für das Selbststudium :).

Mehrsprachigkeit

Links

Wikipedia

- UTF-8: <http://de.wikipedia.org/wiki/UTF-8>
- ISO-8859: http://de.wikipedia.org/wiki/ISO_8859
- Unicode: <http://de.wikipedia.org/wiki/Unicode>

PHP WACT

- Character Sets / Character Encoding Issues:
<http://www.phpwact.org/php/i18n/charsets>
- Handling UTF-8 with PHP: <http://www.phpwact.org/php/i18n/utf-8>

PHP

- Iconv: <http://de.php.net/manual/de/ref.iconv.php>
- Mbstring: <http://de.php.net/manual/de/ref.mbstring.php>

Mehrsprachigkeit

Links

- The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!):
<http://www.joelonsoftware.com/articles/Unicode.html>
- Some Internâtiônâlizætiøn hints:
<http://babylon.idlevice.co.uk/phplondon/2005-05/>
- Survival guide to I18N:
<http://intertwingly.net/stories/2004/04/14/i18n.html>
- PHP, XML, and Character Encodings: a tale of sadness, rage, and (data-)loss:
<http://minutillo.com/steve/weblog/2004/6/17/php-xml-and-character-encodings-a-tale-of-sadness-rage-and-data-loss>
- php+i18n on del.icio.us: <http://del.icio.us/tag/php%2Bi18n>

Mehrsprachigkeit

Links

- FORM submission and i18n:
<http://ppewww.physics.gla.ac.uk/~flavell/charset/form-i18n.html>
- MySQL Manual:
<http://dev.mysql.com/doc/refman/5.0/en/charset-general.html>
- Google's XSS Vulnerability:
<http://shiflett.org/archive/177> und <http://shiflett.org/archive/178>
- How to develop multilingual, Unicode applications with PHP:
http://www.randomchaos.com/documents/?source=php_and_unicode

Mehrsprachigkeit

Links

- Unicode Transformation Formats: UTF-8 & Co:
<http://czyborra.com/utf/index.html#UTF-8>
- Decodeunicode: <http://decodeunicode.org/>