GEN meeting on website cooperation and networking

Introduction to PHP

UNEP/DEWA/GRID-Geneva
Hy Dao - Stefan Schwarzer - Fred Vogel - Karine Zuercher
Table of content

- Introduction
- Main features
- System requirements & installation
- Examples
  - GRID data catalogue (H. Dao) http://os.grid.unep.ch/mdd
  - GEO-3 data portal (S. Schwarzer)
  - Le site Web du SIEnG (F. Vogel) http://www.sieng.ch
  - UNEP.Net Socioeconomic Portal (K. Zuercher)
  - Le catalogue de la Genève verte (K. Zuercher)
- References & resources
- Summary & Questions
Who are we?

- UNEP’s Division of Early Warning and Assessment (DEWA)
- GRID-Geneva is one of UNEP's major centres for data and information management
- Global and regional environmental data
- Support the environment assessment and "early-warning" activities of UNEP and its partners.
Foreword

- This presentation is not a programmer’s view on the subject; it reflects a semi-naive use of PHP/HTML and JavaScript
- Our background: geography, biology, earth sciences, ...
- Solutions have been found, sometimes explanations too
- Lack of specialized programming skills, but there is no need to know about mechanics to drive a car ...
Why PHP?

- The only ways to escape of this static page:
  - click on the back button of the browser
  - type a new URL in the location bar
Why PHP?

• Navigating with hyperlinks: the content of the page is still static

Interactive texts & images (HTML)

Well, you are at a crossroad. What do you want to do?

Click here to continue

Go to step 1
Why PHP?

- Using forms: controls (checkboxes, radio buttons, text areas, ...) modified by the user before submission for processing => more interactive, but only HTML code.
Why PHP?

- **JavaScript**: more interactive elements, special functions (ex: flip images)
- **PHP**: page content dynamically generated

![Dynamic page - Netscape](Dynamic page (HTML + PHP + JavaScript))

This was your choice:
Choice = `val_continue`
Transport option = 1

**Image 1**

Go to step 3

Go back (where ever you come from)
Why PHP at GRID-Geneva?

- New technology
- Separate data, applications and layout: allows for different (and unexpected) uses of the same data
- Facilitate content updating
- Improve interactivity with users/clients
- Helps structuring our data/information
- Makes people talk (internal communication)
History of PHP

- PHP first version (originally known as Personal Home Pages) released in early 1995 by Rasmus Lerdorf
- 1995 : PHP/FI (Form Interpreter)
- 1999 : PHP4
- Latest version : PHP4.1.1 (2001)

- Over two hundred regular contributors
PHP on the Internet

- In use on over 5.5 million domains

- Market share of Internet webservers
  - Apache: 60%
  - IIS servers (ASP): 30%

(from http://www.netcraft.com/survey/)
What is a PHP script?

- Code embedded within tags: jumping between HTML and PHP (like ASP and Cold Fusion)
- No heavy code to output HTML (as needed in Perl or C)
- PHP code is enclosed in special start and end tags

```html
<html><head><title>Example</title></head>
<body>
<?php
    echo "Starting to write a PHP script";
?>
</body>
</html>
```
PHP is server-side

What the web server sees (and executes on-the-fly)

```
<H1>Greetings</H1>
<P ALIGN="center">
<?php
print "Hello GEN!";
?>
</P><HR>
```

What the web browser receives (it never sees PHP code)

```
<H1>Greetings</H1>
<P ALIGN="center">
Hello GEN!
</P><HR>
```

=> it is not possible to copy the source code from the web browser
Three-tier architecture

- **Presentation tier**: Thin client
- **Business Logic tier**: Thick server
- **Data storage tier**
Architecture: an example

Client: web browser

Servers

RS6000
PHP +
web server

SQL queries

Linux
database
server

PHP/HTML editing

Windows PC
DB access +
PHP/HTML editing & test

ODBC

ftp

Servers

Client: web browser

PHP/HTML editing

Windows PC
DB access +
PHP/HTML editing & test

ODBC

ftp
What do you need to use PHP?

- PHP (now ver. 4.1.1), in one of the following flavours:
  - executable: stand-alone (PHP-GTK), CGI (php.exe)
  - module: Apache web server (php4apache.dll), ISAPI (php4isapi.dll, not in production state)

- A web server

- A database server

- Optional:
  - extensions: additional functionalities
  - an ODBC database driver for accessing to/from other databases
System requirements

• Operating systems: Linux, many Unix variants (HP-UX, Solaris and OpenBSD, ...), Microsoft Windows, Mac OS X, RISC OS, ...

• Web servers: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape, iPlanet, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, Zeus, ...

• Databases: Oracle, Informix, Sybase, InterBase, PostgreSQL, MySQL, ODBC, ...
How to install PHP?


- Let’s have the Open Source attitude: compile the program from the source code
- Get the executable binaries
- Get PHP with your operating system (ex: RedHat Linux)
- Use an installer
  - PHP installer at http://www.php.net
  - PHP + Apache + mysql installer: EasyPHP.org
- Call your beloved system manager
Installation for Windows

- Get EasyPHP 1.5 (9 MB) at http://www.easyphp.org/
- Double-click on easyphp1-5_setup.exe
- Use myphpadmin to manage your mysql database
- Install the myodbc driver (http://www.mysql.com/downloads/api-myodbc.html)
- Create a new ODBC data source
- Access the data source with MS-Access
Write a PHP file

- Buy a good PHP manual, or go to http://www.php.net/manual/
- Open your favorite text editor
- Type your PHP/HTML/Javascript code
- Save your file with the extension .php
- Put your .php file on your (local) web server
- Access your .php file with a web browser using the http protocol:
  • http://localhost/my_file.php
- Do not use the following location
  • file://d:/Program%20Files/EasyPHP/www/my_file.php
Advantages of PHP

• Modest learning curve
• Free and open development (independence, security, community)
• Native database connectivity (mysql, ODBC, ...)
• Availability for a variety of platforms (Windows, UNIX, Linux, Mac OS, RISC)
• Simple but powerful, thin client
• Many functions (databases, mail server, PDF, ...)
Disadvantages of PHP

• Immature language
• Mixing of HTML and program code: disorganization (application and layout not always well separated), bug prone
• In general, more efficient as an Apache web server module => CGI and other web servers’ versions are often weaker (speed, reliability)
Learnings so far

• Easy to install, more complex to configure properly (ex: security)
• Need some programming skills, although some developers’ tools exist (PHAKT, PHP Coder, ...)
• Access to databases not standardized (use ADODB library to turn around this problem)
• Use PHP as Apache module, CGI version not appropriate when working with databases (connection lost at each page processing => time consuming)
Next steps for GRID-Geneva

- Put our in-house PHP applications in production phase
- Take some training either in PHP or in the field of related technologies: XML, database design, Java servlets, ...
- Future collaborations with other GEN partners?
References & resources


- PHP Home page: http://www.php.net

- Web links:
  http://www.grid.unep.ch/proser/training/it/gen2002/links.html