

Introduzione agli eZ Components

Gaetano Giunta

PHP Day 2008

Rimini, 24/5/2008

Agenda

- Introduzione
- Architettura / design degli eZ Components
- Funzionalità delle ultime release
- Esempi di utilizzo
- Domande / altro



eZ Systems

- Società creatrice di eZ Publish e degli eZ Components
- Fondata in Norvegia nel 1999
- 85 Dipendenti - 23 nazionalità
- HQ a Skien / Norvegia, cinque uffici regionali
 - Nordics : Oslo – Copenhagen -Skien
 - Central and eastern Europe : Dortmund
 - Western Europe : Paris – Lyon – Bruxelles
 - Northern America : Vancouver - Chicago
 - A breve ... London & Tokyo
- 100% Open Source
- 100% PHP



eZ Components – di cosa si tratta?

« Una piattaforma enterprise ready di componenti php general purpose »

cioè

- Non è un framework
- ... ma un insieme di componenti debolmente accoppiati. Si possono usare insieme o singolarmente
- Non è dedicata esclusivamente allo sviluppo web
- ... ma soprattutto allo sviluppo web
- Disponibile con licenza New BSD e con supporto eZ Systems



Obiettivi della libreria

- Costruire una piattaforma solida che semplifichi lo sviluppo di applicazioni PHP 5 permettendo allo sviluppatore di non “reinventare la ruota” ogni volta
- API semplice e consistente, design OO « pulito »
- Non forzare una struttura obbligata: non è un framework
- Mantenere a lungo la compatibilità binaria e delle API
- Mantenere la stabilità e limitare le regressioni
- 100% Open Source, con chiara attribuzione dei diritti di PI
- Sviluppata, mantenuta e supportata da eZ Systems



Metodologia di sviluppo

- Pieno utilizzo delle librerie e funzionalità disponibili da PHP 5.1 (versione minima supportata: 5.1.6, a volte +)
- Discussione e documentazione di tutte le API prima della loro implementazione
- Utilizzo pervasivo di unit testing – scrittura dei test PRIMA del codice
- Documentazione sia inline (phpdoc à gogo) che offline (tutorials, reference, guida di installazione). La documentazione si scrive durante lo sviluppo, non dopo



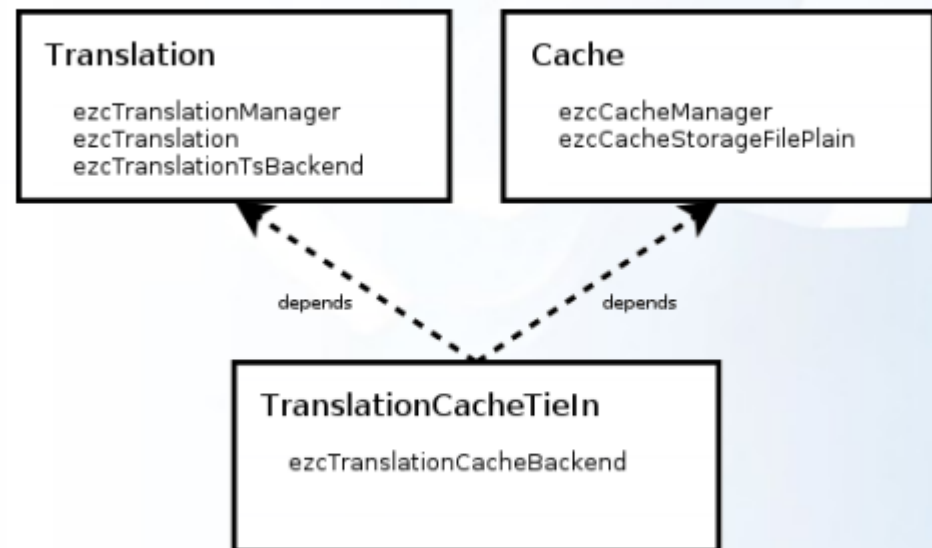
Architettura: naming delle classi

- Utilizzazione di un prefisso, a causa della mancanza di namespaces (in attesa di PHP 5.3): ezcXXX
- Leggibili: meglio nomi lunghi ma parlanti che criptici
- Talvolta « aggiustati » per una migliore leggibilità
eg: ezcMailSmtptTransport vs. EzcMailSMTPTransport
ezcTestSuite vs. PHPUnit2_Framework_TestSuite
- ...e molte altre regole su
http://ezcomponents.org/contributing/coding_standards.html



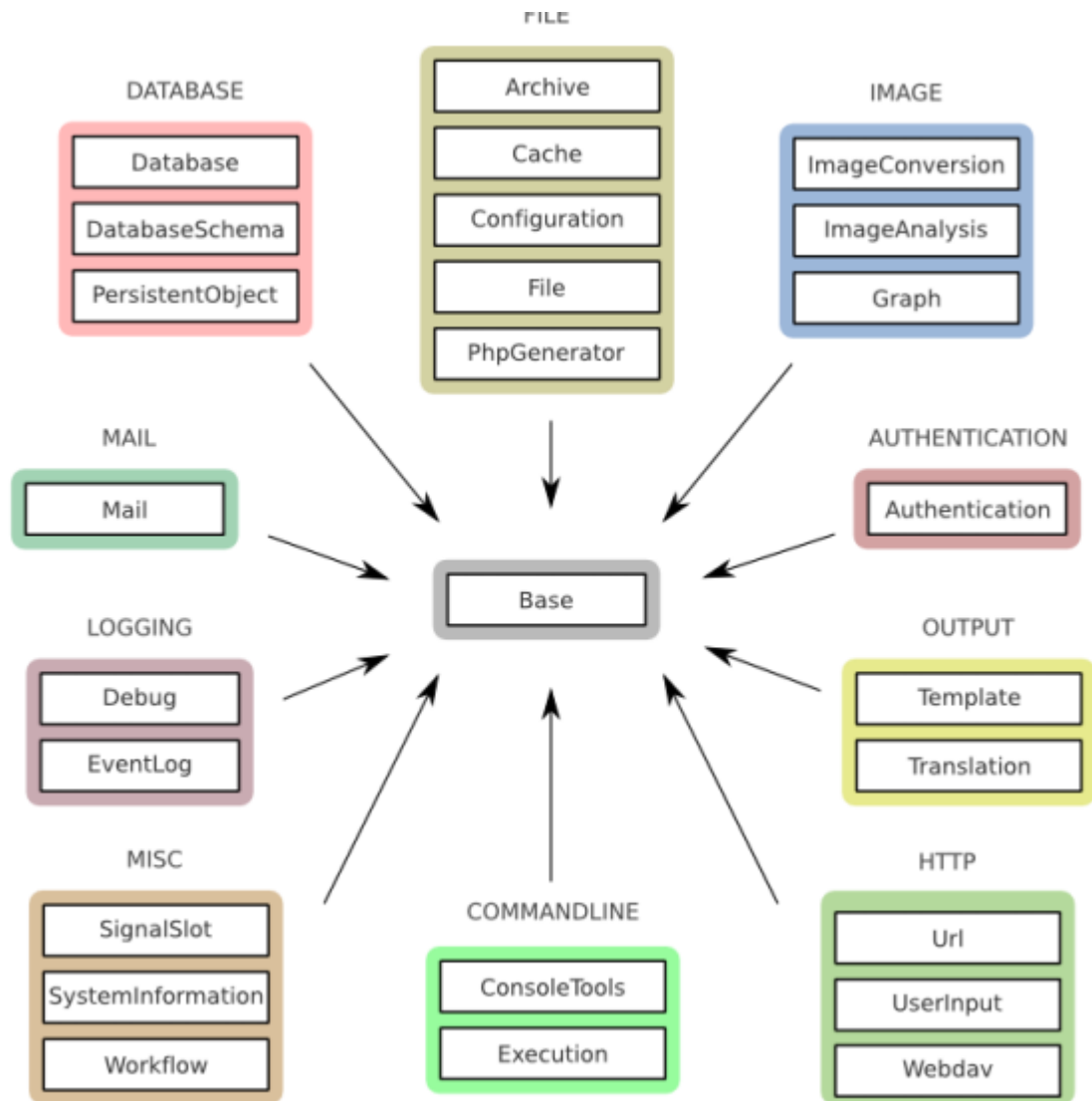
Architettura: dipendenze

- Meno ce n'è, meglio è...
- ... evitando però la duplicazione del codice
- solo dove necessario
- Componenti separati che le implementano



Schema generale

Le classi sono raggruppate in Packages



Roadmap

Release 2008.1 (expected: Giugno 2008)

Nuovi componenti:

- Document – conversione tra differenti formati di dati
- ExtendedReflection
- Feed
- Search – front-end per Solr e altri motori di ricerca



Ultime novità

- 2008.1 alpha1
 - Nuovo: Cache gerarchica multilivello
 - PersistentObject supporto per più datatypes
 - Nuovo componente: Search
 - Migliorie a: Archive, Authentication, Configuration, ConsoleTools, Database, DatabaseSchema, EventLog, Graph, Mail, Tree e Workflow
- 2008.1 alpha2
 - Nuovo componente: Document
 - Migliorie a: Base, EventLog, Search e Userinput
- 2008.1 beta: a partire dal 27 maggio



Requisiti di sistema

- PHP versione 5.1.6 (per Tree e Webdav PHP 5.2.1)
- Alcune estensioni php sono necessarie per alcuni componenti, oppure ne aumentano le capacità

es: Archive necessita di zlib (presente nell'installazione standard di PHP) e bzip2, opzionalmente può utilizzare « posix » per la gestione dei permessi dei file

- Elenco esaustivo: su

<http://ezcomponents.org/overview/requirements.html>

(c'è anche un quick reference su come compilare le estensioni mancanti)



Ambiente di sviluppo: suggerimenti

- Sempre sviluppare con `display_errors = On`, `log_errors = On`, `error_reporting = E_ALL`
- Dove si trova il log degli errori di PHP? E del webserver?
- XDebug - l'amico del PHP OOP (e non solo):
 - stack traces completi
 - Debugging step-by-step (con IDE adeguati)
 - Analisi della code coverage (con tool di visualizzazione)
 - Profiling (con `kcachegrind/wincachegrind`)



Ambiente di sviluppo: altri suggerimenti

- Imprescindibile per lo sviluppo web: Firebug
- altre estensioni Firefox utili: « Web developer », « YSlow », « LiveHTTPHeaders », « IETab », « Consolez », « Selenium IDE »
- Per gli sviluppatori windows: UNXUTILS
- In produzione:
 - Sempre utilizzare una opcode cache (APC, Eaccelerator, Xcache, Zend)
 - Utili nel troubleshooting: Phpinfo, server-status e server-info



Installazione 1: via PEAR

1. **Avere una installazione di PHP 5.1 e PEAR funzionante**
2. **Aggiungere il canale components.ez.no all'installer PEAR**

```
pear channel-discover components.ez.no
```

3. **Scaricare effettivamente i componenti**

```
pear install -a ezc/eZComponents
```

Per aggiornare una installazione esistente:

```
pear upgrade ezc/eZComponents
```

`pear help <nomecomando>` **è vostro amico!**



Installazione 2: via SVN

Per chi è interessato allo sviluppo della libreria o ha bisogno di una patch appena rilasciata

1. Avere una installazione di PHP 5.1 e un client SVN

2. Creare una directory dove installare i componenti e ivi

```
svn co http://svn.ez.no/svn/ezcomponents/trunk
```

3. Scaricare lo script che imposta l'ambiente di sviluppo

```
svn co http://svn.ez.no/svn/ezcomponents/scripts  
./scripts/setup-env.sh (o scripts\setup-env.bat)
```

E' ovviamente possibile fare un checkout limitato di solo qualche componente – attenzione alle dipendenze!



Installazione 3: old school (a.k.a. "a mano")

1. Scaricare e decomprimere la tarball

```
wget
```

```
http://ezcomponents.org/files/downloads/ezcomponents
```

```
tar -xjf ezcomponents-2007.2.1.tar.bz2
```

3. Impostare correttamente l'Include Path (manca il 2!)



Configurazione

- Include path: settato in automatico da PEAR, altrimenti va messa la directory radice in cui si è scaricato il software
- Autoload: in automatico tutte le classi degli eZComponents vengono caricate da ezcBase

```
require_once "Base/base.php";  
function __autoload( $className )  
{  
    ezcBase::autoload( $className );  
}
```

NB: per installazioni via SVN o tarball

```
require_once "Base/src/base.php";
```



E ora passiamo al codice...



Domande?

Grazie dell'attenzione!

- <http://ezcomponents.org>
- http://ez.no/community/forums/ez_components
- <http://lists.ez.no/mailman/listinfo/components>



Gaetano Giunta

Senior consultant

eZ Systems Western Europe

26 rue de la république

69002 Lyon

tel : +33 (0)4.78.37.01.56

email : gg@ez.no

