

# content mirror

Andy McKay

clearwind consulting

# Plone content in a relational database

By Kapil <http://code.google.com/p/contentmirror/>

# confession

Alan Runyan - Entransit  
I never really bought it

# deploy

To another system

Then write something else to serve the site

What about the dynamic bits

PHP? Huh?

too complicated

resistance from me



"Mr. Osborne, may I be excused? My brain is full."

*of plone*

throw away Plone?



almost 3 years of not doing  
Plone everyday



repoze  
objective c  
iphone  
php  
prototype  
app engine  
django  
igquery  
rails  
ruby  
~~of phone~~



"Mr. Osborne, may I be excused? My brain is full."

repoze  
objective c

iphone

~~php~~  
~~prototype~~

app engine

django

igquery

~~rails~~

~~ruby~~

~~of phone~~

you

Having trouble finding Plone developers?

Plone documentation a bit confusing?

Not as smart as Martin Aspelli?

Got Plone really fast?

developing, customising  
**Plone** is hard

let **Plone** *manage* content

(it's pretty damn good at this)

let others *deliver* content

# non-profit

Create small sites quickly

Traffic is (mostly) anonymous

Simple features or features you can mash up...



# large intranet

Small number of internal editors  
Large number of internal readers  
Multi-site publishing

# setup

<http://code.google.com/p/contentmirror/wiki/Installation>  
install blah...

# zcml

(eek)

```
<configure xmlns="http://namespaces.zope.org/zope"
            xmlns:db="http://namespaces.objectrealms.net/rdb"
            xmlns:ore="http://namespaces.objectrealms.net/mirror">

    <!-- setup a database connection -->
    <db:engine url="postgres://localhost/plone"
              name="mirror-db"
              echo="True"/>

    <!-- associate the database connection to mirror db structures -->
    <db:bind engine="mirror-db"
            metadata="ore.contentmirror.schema.metadata"
            />

</configure>
```

# setup db

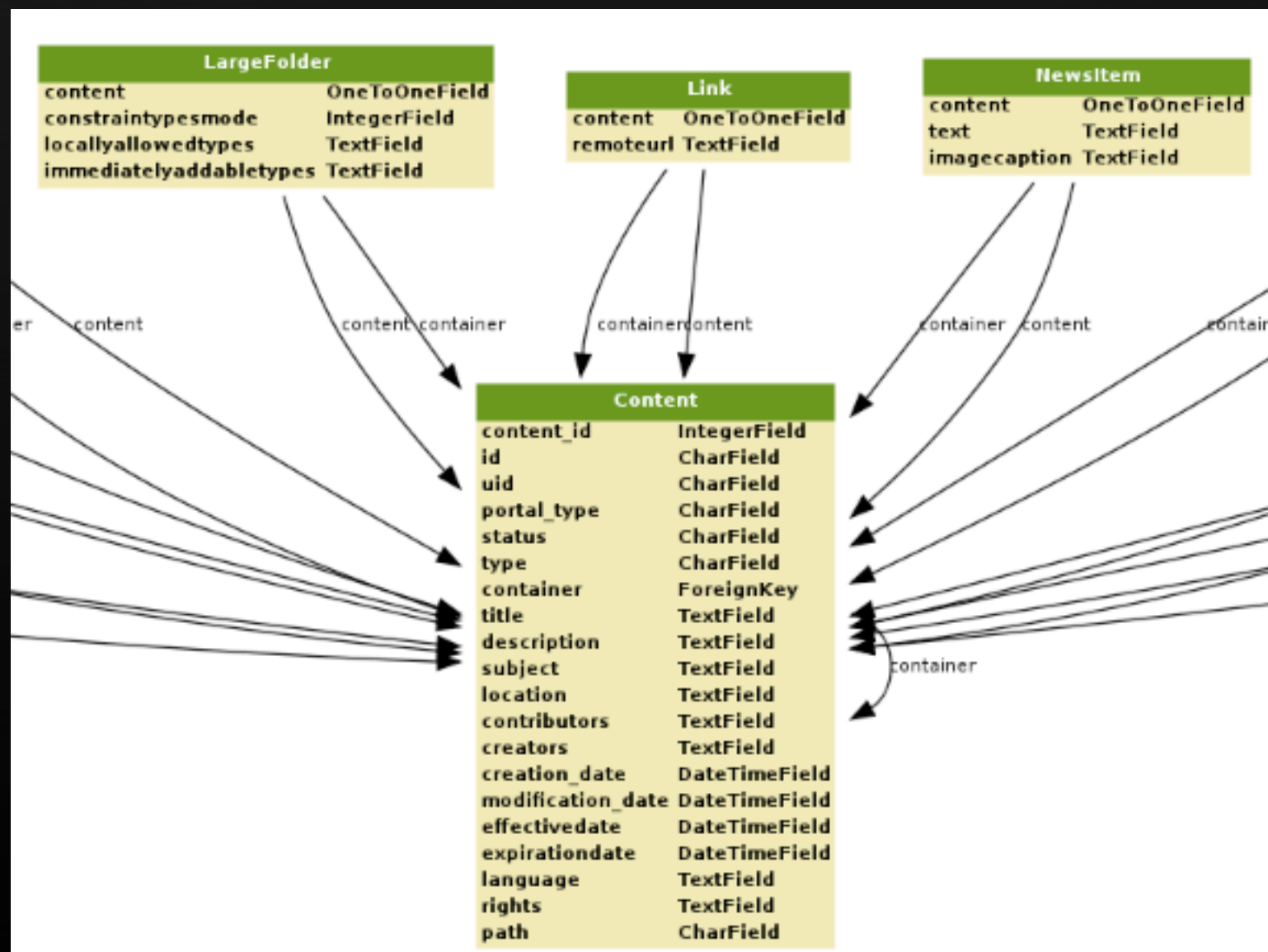
```
dropdb plone  
createdb plone --encoding=UTF-8
```

```
# sets up the db
```

```
bin/instance run parts/productdistros/ContentMirror/ddl.py postgres > mirror.sql  
psql plone < /home/andy/plone/zinstance/mirror.sql
```

```
# loads it with content
```

```
bin/instance run parts/productdistros/ContentMirror/bulk.py Plone
```



# updates on change

```
plone=# select content_id, id, title from content where id = 'front-page';
```

content_id	id	title
8	front-page	This is a test

(1 row)

# carlos

Put **repoze.bfg** in front of the database

Got 10 billion requests a second

<http://blog.delaguardia.com.mx/index.php?op=ViewArticle&articleId=120&blogId=1>

# plango

Put **django** in front of the database  
Got 60 requests a second  
(**Plone** is 3 requests a second)

<http://svn.clearwind.ca/public/django/plango>



demo bit...

# plango notes

URL publishing via middleware: plone/middleware.py

Sample templates: plone/templates/folder.html, image.html etc

Sample image handling: plone/views.py

Model: based on reflecto from Kapil

Template: <http://www.opensourcetemplates.org/>

# content mirror notes

Green workflow dropdown - doesn't cause database update  
Everything published, check workflow state in your app  
Blobs go into the database, not ideal for big blobs  
Some fields missing eg: default\_page, needed?

It looks all reasonably simple to alter

<http://code.google.com/p/contentmirror/wiki/Introduction>

how much simpler?

cost  
caching  
speed  
integration  
independence  
features

Questions?