What’s New in the Bricolage Content Management System

David Wheeler
Kineticode

What is Bricolage?

- Enterprise-class content management system
- Written in Perl
- Powered by mod_perl
- Backed by PostgreSQL

What’s it got?

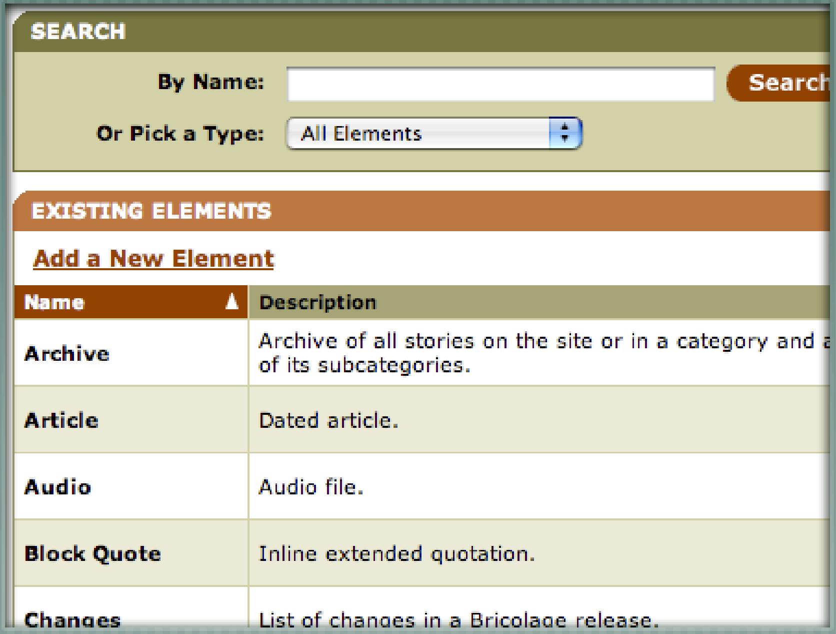
- Browser-based interface
- Complete Separation of content from presentation
- Content categorization
- Legible URLs
- Fully configurable workflows
- Management of multiple sites

What else?

- SOAP interface
- Complete document modeling in the UI
- Perl-based templating architectures
- Separates content management from content delivery
- Delivery platform-netural

New UI

- Complete overhaul of the Browser UI
- Now using Web Standards:
  - XHTML 1.0 compliant
  - Styled with CSS–skin it!
- Up to 70% smaller page sizes
- Contributed by Marshall Roch


SEARCH

By Name: Search

Or Pick a Type: All Elements

EXISTING ELEMENTS

Add a New Element

Name	Description
Archive	Archive of all stories on the site or in a category and a of its subcategories.
Article	Dated article.
Audio	Audio file.
Block Quote	Inline extended quotation.
Changes	List of changes in a Bricolage release.

Check it out!

New URI Formats

- Old style: /categories/month/year/day/slug
- New style: /%{categories}/%Y/%m/%d/%{slug}
- Why?
- **Flexibility**

Flexible URI Formats

- Use any DateTime-supported strftime format:
  /%{categories}/%W
  /reviews/books/31
- Combine parts into a directory name:
  /%y-%m-%d/%{categories}
  /2005-08-05/reviews/books
- Add arbitrary strings:
  /%{categories}/archives/%Y/week_%W
  /reviews/books/archives/2005/week_31

Revamped Database

- Originally targeted Oracle
- Migrated to PostgreSQL
- Cruft left in for 4 years:
  - NUMERIC for integers
  - NUMERIC + constraints for booleans

Boolean-kashaw!

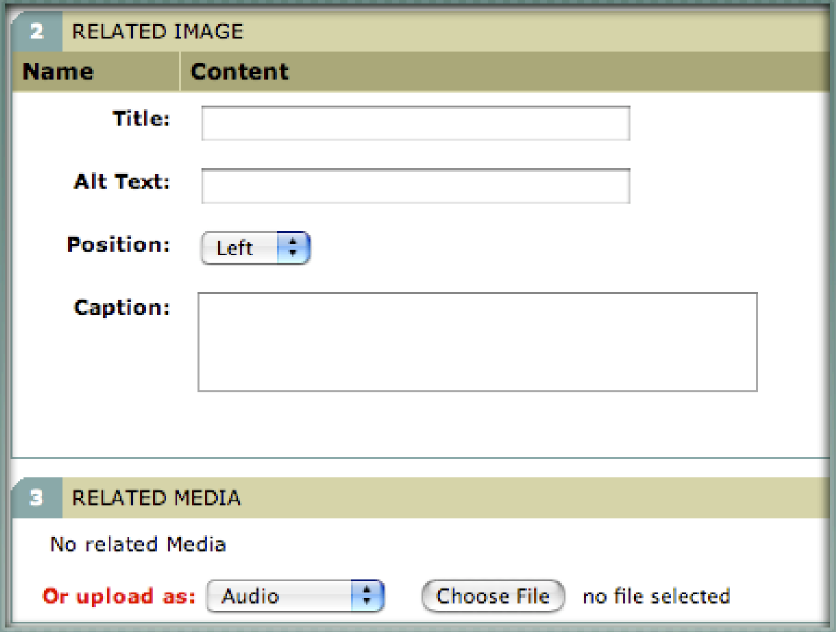
- Now integers are really integers
- Booleans are really booleans
- No more constraints for booleans
- Result: Performance gains!
- Contributed by Neil Conway

Related Media

- Story documents and Media documents are separate
- For relateds, have always had to:
  - Create a new media document
  - Create the story document
  - Associate the media document with the story

Related Media Uploads

- New: inline media uploads
- Must have permission to create media
- Upload new media directly from story profile:
  - In same category
  - With same cover date
- Can also auto-preview media



2 RELATED IMAGE

Name	Content
Title:	
Alt Text:	
Position:	Left
Caption:	

3 RELATED MEDIA

No related Media

Or upload as: Audio Choose File no file selected

Show me!

Pluggable Authentication

- Always had its own authentication system
- We’ve added pluggable authentication
- Provided LDAP authentication plugin
- Write your own!

```
sub authenticate {
    my ($pkg, $user, $pwd) = @_;
    my $cur = $user->_get('password');
    return $pkg unless $cur;
    return md5_hex($secret . md5_hex($pwd)) eq $cur ? $user : undef;
}
```

LDAP Configuration

```
AUTH_ENGINES      = LDAP Internal
LDAP_SERVER       = ldap.example.com
LDAP_VERSION      = 3
LDAP_USER         = 0
LDAP_PASS         = 0
LDAP_BASE         = ou=Users,dc=example,dc=com
LDAP_UID_ATTR     = uid
LDAP_FILTER       = (objectclass=*)
LDAP_GROUP        = cn=Bricolage,ou=Group,dc=example,dc=com";
LDAP_MEMBER_ATTR = uniqueMember
LDAP_TLS          = Yes
LDAP_SSL_VERSION = 3
```

Templating architectures

- HTML::Mason

```
<%perl>;
for my $e ($element->get_elements(qw(header para _pull_quote_))) {
    my $kn = $e->get_key_name;
    if ($kn eq 'para') {
        $m->print('<p>', $e->get_data, "</p>\n");
    } elsif ($kn eq 'header') {
        $m->print('<h3>', $burner->sdisplay_element($e), "</h3>\n");
    } elsif ($kn eq '_pull_quote_' && $e->get_object_order > 1) {
        $m->print($burner->sdisplay_element($e));
    } else {
        $burner->display_element($e);
    }
}
$burner->display_pages('_page_');
</%perl>
```

Templating architectures

- HTML::Template, now with:
  - Cascading category templates
  - All story attributes added as tmpl_vars

```
<tmpl_loop element_loop>
<tmpl_if is_para>
<p><tmpl_var para></p>
</tmpl_if>
<tmpl_if is_header>
<h3><tmpl_var header></h3>
</tmpl_if>
<tmpl_if is__pull_quote_>
<tmpl_var _pull_quote_>
</tmpl_if>
</tmpl_loop>
```

Templating architectures

Template Toolkit

```
[% FOREACH e = element.get_elements('header', 'para', '_pull_quote_') %]
    [% kn = e.get_key_name %]
    [% IF kn == 'para' %]
<p>[% e.get_data %]</p>
    [% ELSIF kn == 'header' %]
        [% # display_element() should just return a value. %]
<h3>[% burner.display_element(e) %]</h3>
    [% ELSIF kn == '_pull_quote_' && e.get_object_order > 1 %]
        [% PERL %]
          print $stash->get('burner')->display_element($stash->get('e'));
        [% END %]
    [% ELSE %]
        [% # Test display_element(). %]
        [% burner.display_element(e) %]
    [% END %]
[% END %]
[% burner.display_pages('_page_') %]
```

Templating architectures

And Introducing...PHP!

```
<?php
# Convenience variables.
$story   = $BRIC['story'];
$element = $BRIC['element'];
$burner  = $BRIC['burner'];
foreach ($element->get_elements('header', 'para', '_pull_quote_') as $e) {
    $kn = $e->get_key_name();
    if ($kn == 'para') {
        echo '<p>', $e->get_data(), "</p>\n";
    } else if ($kn == 'header') {
        echo '<h3>', $burner->sdisplay_element($e), "</h3>\n";
    } else if ($kn == '_pull_quote_' && $e->get_object_order() > 1) {
        echo $burner->sdisplay_element($e);
    } else {
        $burner->display_element($e);
    }
}
$burner->display_pages('_page_');
?>
```

Powered by PHP::Interpreter

- On its way to CPAN
- Use CPAN modules in PHP!

```
<?php
  perl_use('DBI');
  perl_use('DateTime');
  $dbh = perl_method("DBI->connect", "dbi:SQLite:dbname=dbfile", "", "");
  $dbh->do('CREATE TABLE foo (bar TEXT, time DATETIME)');
  $now = perl_method('DateTime->now');
  $ins = $dbh->prepare('INSERT INTO foo VALUES (?, ?)');
  $ins->execute('This is a test', $now);
  $sel = $dbh->prepare('SELECT bar, time FROM foo');
  $sel->execute();
  $a = array('foo', 'bar');
  foreach ($sel->fetch() as $val) {
      echo "$val\n";
  }
  $sel->finish();
  $dbh->do('DROP TABLE foo');
  $dbh->disconnect();
?>
```

Brought to you by...

- Developed by George Schlossnagle
- With help from Sterling Hughes, Wesley Furlong, and segfaults from yours truly
- Sponsored by Portugal Telecom–SAPO.pt
- Get it from Subversion now and CPAN soon! https://svn.perl.org/modules/PHP-Sandwich/

Summer of Code

- Bricolage awarded four internships
- Marshall Roch adding Input Channels
- Temas Mazei porting to MySQL
- Scott Loyd porting to Apache 2/mod_perl 2
- Sam Strasser adding example document types and templates
- Thank you Google!

The Future

- Bricolage 1.10 in September
- Bricolage 1.12 by year’s end
- Bricolage 2.0...

The Kinetic Platform

- New platform for enterprise application development
- Bricolage 2.0 will be the port to TKP
- Features:
    - Object/Relational mapping in the database
    - REST interface layered over entire API
    - Ajax-powered browser interface

Bricolage search

```
Bric::Biz::Asset::Business::Story->list({
    cover_date_start => "$date 00:00:00",
    cover_date_end   => "$date 23:59:59",
    element_key_name => 'story',
    primary_uri      => "/$service%",
    Order            => 'cover_date',
    OrderDirection   => 'DESC',
    publish_status   => 1,
});
```

- Single attributes
- Only AND searches

Kinetic search basics

- Contained object search
  'contact.value' => 'larry@wall.org'
- String, number, and DateTime parameters
- NULL seraches using undef
- Value ranges using array references:
  'fav_number' => [1 => 100]

Kinetic search operators

NOT	Inverse comparison, can be used with:
LIKE	SQL syntax
MATCH	Regular expressions
GT	Greater Than
LT	Less Than
GE	Greater than/Equal to
LE	Less than/Equal to

Kinetic compound searches

- AND
- OR
- ANY
- BETWEEN

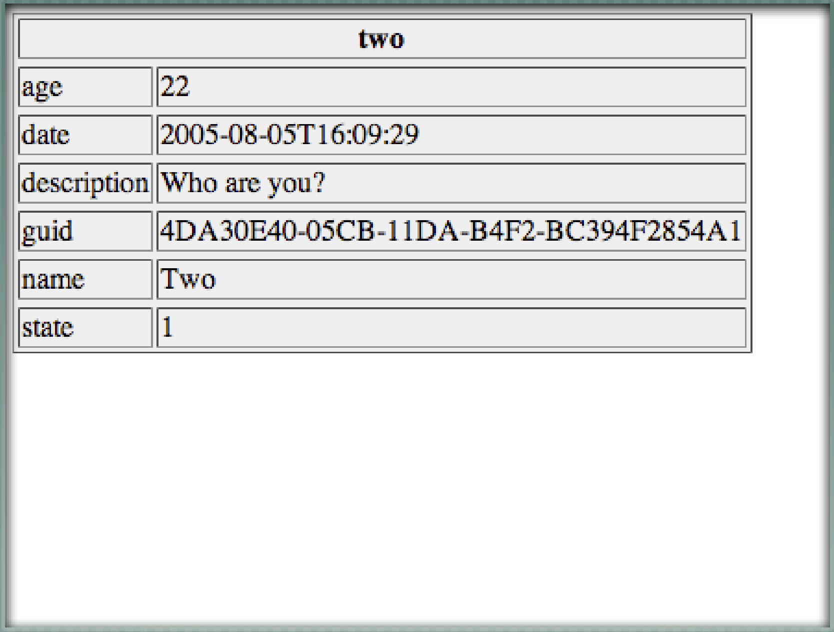
Kinetic search example

```
Kinetic::Party::Person->search(q{
    last_name                        => 'Wall',
    first_name                       => 'Larry',
    OR (bio                          => LIKE '%perl%'),
    OR ('contact.type'               => MATCH 'email$',
         AND ('contact.value' => MATCH '@cpan\.org$'),
         AND ('fav_number        => GE 42 )
       )
});
```

- Native code API
- Plain text API—REST!

When?

- Perpetually a year away
- There has been progress!
- Development of The Kinetic Platform continues apace
- It’s not just vapor!



two	
age	22
date	2005-08-05T16:09:29
description	Who are you?
guid	4DA30E40-05CB-11DA-B4F2-BC394F2854A1
name	Two
state	1

Check it out!

The Kinetic Platform

- Not much to look at yet, but...
- Pure REST server
- Transforming objects with XSLT
- Adding editing next
- Uses polymorphic database design
- Bricolage to be ported next year

Thank you

David Wheeler
Kineticode
http://www.kineticode.com/