

pier

No Kernel

LEFT BEHIND

Nick Ager

@nickager

What is *pier*?

Pier

Magritte

Seaside

Pharo

Gemstone

VW

VA

What is *pier*3?

Pier 3

Magritte 3

Seaside 3

Pharo

Gemstone

VW

VA

Magritte 2



Magritte 3



Why Magritte 3

Issues with Magritte 2:

- Name collision with #description
- class-side description:
 - Cache-invalidation
 - dynamic descriptions
 - instance specific descriptions
 - context dependent descriptions

Magritte 3

Magritte 3

Descriptions move from class to instance.

Magritte 3

- Descriptions move from class to instance.
- Use `<magritteDescription>` pragmas rather than naming convention.

Magritte 3

- Descriptions move from class to instance.
- Use `<magritteDescription>` pragmas rather than naming convention.
- All descriptions are dynamically generated.

Magritte 3

- Descriptions move from class to instance.
- Use `<magritteDescription>` pragmas rather than naming convention.
- All descriptions are dynamically generated.
- Rename `#description` to `#magritteDescription`

Magritte 3

- Descriptions move from class to instance.
- Use `<magritteDescription>` pragmas rather than naming convention.
- All descriptions are dynamically generated.
- Rename `#description` to `#magritteDescription`
- Non-backward compatible

Magritte 3 Links

- Magritte doesn't have to look ugly
<http://twitterbootstrap.seasidehosting.st>
- An introduction to Magritte 3
<http://vimeo.com/37032840>
- Slides associated with video
<http://www.slideshare.net/nickager/magritte3>

What is *pier*3?

Pier 3

Pier Wysiwyg

Pier Admin

Pier Admin Setup

Magritte 3

Seaside 3

Pharo

Gemstone

VW

VA

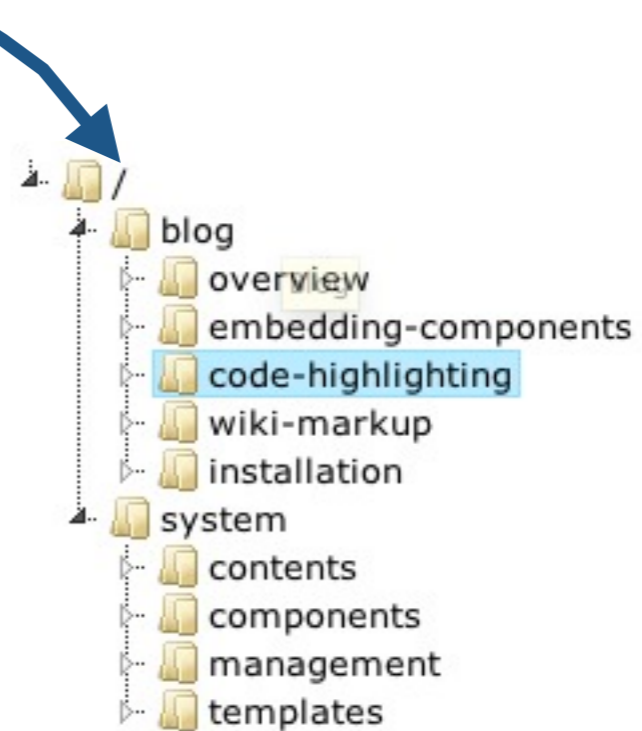
Demo



- Ideally all Pier installations will use the same Pier version so that everyone can benefit and contribute to the latest code.
- More testing of latest Pier code.
- More contributions to Pier add-ons.

What is a *pier* kernel?

1) Holds the root
to the tree of
PRStructure(s)



What is a *pier* kernel?

2) Users and groups

```
PRKernel>>users  
  ^ self propertyAt: #users ifAbsentPut: [ Set with: self adminUser ]
```

```
PRKernel>>groups  
  ^ self propertyAt: #groups ifAbsentPut: [ Set with: self adminGroup ]
```

Why the upgrade pain?

- Pier is a powerful CMS and allows rapid content creation and editing.
- Code and content are intimately intertwined.
- Pier's code-base has rapidly evolved.
- Upgrading often requires a knowledge of code changes and is non-trivial.
- Export formats are often verbose, opaque and are dumb serializers of the state (eg xml exporter etc)

```

<sixx sixx.version="1.0">
  <sixx.object sixx.id="0" sixx.type="OrderedCollection" >
    <sixx.object sixx.id="1" sixx.type="Dictionary" >
      <sixx.object sixx.id="2" sixx.type="Association" >
        <sixx.object sixx.id="3" sixx.type="Symbol" >author</sixx.object>
        <sixx.object sixx.id="4" sixx.type="String" >admin</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="5" sixx.type="Association" >
        <sixx.object sixx.id="6" sixx.name="key" sixx.type="Symbol" >name</sixx.object>
        <sixx.object sixx.id="7" sixx.name="value" sixx.type="String" >2011-05-24</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="8" sixx.type="Association" >
        <sixx.object sixx.id="9" sixx.name="key" sixx.type="Symbol" >author</sixx.object>
        <sixx.object sixx.id="10" sixx.name="value" sixx.type="String" >admin</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="11" sixx.type="Association" >
        <sixx.object sixx.id="12" sixx.name="key" sixx.type="Symbol" >uuid</sixx.object>
        <sixx.object sixx.id="13" sixx.name="value" sixx.type="String" >335262787197349761</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="14" sixx.type="Association" >
        <sixx.object sixx.id="15" sixx.name="key" sixx.type="Symbol" >title</sixx.object>
        <sixx.object sixx.id="16" sixx.name="value" sixx.type="String" >Young Inventor of the year&#33;</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="17" sixx.type="Association" >
        <sixx.object sixx.id="18" sixx.name="key" sixx.type="Symbol" >publication</sixx.object>
        <sixx.object sixx.id="19" sixx.name="value" sixx.type="TimeStamp" >24 May 2011 11:12:42</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="20" sixx.type="Association" >
        <sixx.object sixx.id="21" sixx.name="key" sixx.type="Symbol" >tags</sixx.object>
        <sixx.object sixx.id="22" sixx.name="value" sixx.type="Array" ></sixx.object>
      </sixx.object>
      <sixx.object sixx.id="23" sixx.type="Association" >
        <sixx.object sixx.id="24" sixx.name="key" sixx.type="Symbol" >comments</sixx.object>
        <sixx.object sixx.id="25" sixx.name="value" sixx.type="OrderedCollection" ></sixx.object>
      </sixx.object>
    </sixx.object>
    <sixx.object sixx.id="26" sixx.type="Dictionary" >
      <sixx.object sixx.id="27" sixx.type="Association" >
        <sixx.object sixx.name="key" sixx.idref="3" />
        <sixx.object sixx.id="28" sixx.name="value" sixx.type="String" >Designer Andy Murray is promoting cutlery with a difference with his &#42;KitchenToolbox&gt;/products/KiTh
old from Glasgow has come up with a range of cutlery which can also be used as tools. He said:“People often use knives to get fuses out of plugs so I thought I would redesign cutlery for simple every day DIY tasks
&#42;continued&gt;http://www.getitmade.com/public/blog/2011-02-01/&#42;</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="29" sixx.type="Association" >
        <sixx.object sixx.name="key" sixx.idref="6" />
        <sixx.object sixx.id="30" sixx.name="value" sixx.type="String" >2011-05-24-2</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="31" sixx.type="Association" >
        <sixx.object sixx.name="key" sixx.idref="9" />
        <sixx.object sixx.id="32" sixx.name="value" sixx.type="String" >admin</sixx.object>
      </sixx.object>
      <sixx.object sixx.id="33" sixx.type="Association" >
        <sixx.object sixx.name="key" sixx.idref="12" />

```

Sixx export

burgh, won the accolade for his designs which include the H
 pressed furniture made through his company AndThe John Logie
 hn Logie Baird invented more than the television and I think I wo

*the old method
 (opaque format)*

Kernel

Pier-Code-Exporter

I created this site as a home for a new `*blog>/blog*`. I hope you'll find something of interest within it's confines.
`+/system/components/slideshow+`.

```
structure localEnvironment: (self systemtemplatesenvironment: kernel).
structure localStyleSheet: (self systemcomponentseventCss: kernel).
structure addChild: (self blog: kernel).
structure addChild: (self projects: kernel).
structure addChild: (self links: kernel).
structure addChild: (self sailing: kernel).
structure addChild: (self about: kernel).
structure addChild: (self system: kernel).

structure securityDecoration owner: (kernel userName: 'nick.ager@gmail.com' ifNone: nil).
structure securityDecoration group: (kernel groupName: 'admin' ifNone: nil).
^ structure
```

Content is Code

KernelCreatorNickagercom>>blogScreenecastcreatinganewinstancebasedonpreconfiguredAMI: kernel

```
| structure |
structure := PBPost named: 'Screenecast-creating-a-new-instance-based-on-pre-configured-AMI'.
structure title: 'Screenecast: Create a new EC2 instance from a preconfigured Seaside/Gemstone AMI'.
structure tags: {'Gemstone'. 'EC2'. 'screenecast'}.
structure author: 'Nick Ager'.
structure publication: (TimeStamp year: 2011 month: 1 day: 2 hour: 15 minute: 52 second: 51).
structure contents: 'Here''s a screenecast to accompany my blog post on *deploying your Seaside application for free using Amazon EC2 and Gemstone>/blog/Create-a-free-Gemstone-server-in-the-cloud-in-10-minutes*:
```

```
{{{<iframe src="http://player.vimeo.com/video/18375790" width="640" height="480" frameborder="0"></iframe><p><a href="http://vimeo.com/18375790">watch on Vimeo</a></p>}}}}
```

```
structure securityDecoration owner: (kernel userName: 'nick.ager@gmail.com' ifNone: nil).
structure securityDecoration group: (kernel groupName: 'admin' ifNone: nil).
^ structure
```

Code is Content

Pier-Code-Exporter

**Makes upgrading old
Pier installations
possible**

Pier-Code-Exporter

**Reduce the pain of
future upgrades**

Exporting code benefits

- ☑ The exporter uses its knowledge of Pier to export in a readable format eg:
 - PRDocument(s) are exported as wiki-strings rather than serialisations of the document AST
- ☑ The output can be edited and manipulated
 - it's just code.
- ☑ You can check-in your exported kernel into your favourite VCS.

Status

- Pharo and Gemstone
- Works for Pier 1, Pier 2 and Pier 3 kernels
- Used to upgrade active Pier 1 and Pier 2 sites.
- Not currently implemented as a Kernel Persistency mechanism

Procedure

1. Install `Pier-Exporter-Code`
2. Export and test the exported Kernel
3. Export a Pier 3 kernel
4. Install Pier 3 and exported kernel

Installing Code Exporter Pier 1 - (Seaside 2.8)

Add the following repository in Monticello browser:

MCHttpRepository

location: '<http://source.lukas-renggli.ch/pier3addons>'

user: ''

password: ''

Load the following packages:

Pier-Exporter-Seaside28-Code

Pier-Exporter-Code

no Gofer in Seaside 2.8!

Installing Code Exporter Pier 2 - (Seaside 3.0)

```
Gofer it
  renggli: 'pier3addons';
  package: 'Pier-Exporter-Code';
  load.
```

Installing Code Exporter - Pier 3 (Seaside 3.0 and Magritte 3)

```
Gofer it
  renggli: 'pier3addons';
  package: 'Pier-Exporter-Code';
  load.
```

```
Gofer it
  renggli: 'pier3addons';
  package: 'Pier-Exporter-Pier3-Code';
  load.
```

Installing Code Exporter - Pier 3 (Seaside 3.0 and Magritte 3) #2

alternative with Metacello

```
Gofer it
  renggli: 'pier3addons';
  package: 'ConfigurationOfPier3Add0ns';
  load.
```

```
ConfigurationOfPier3Add0ns project stableVersion load: 'Code-Exporter-Pier3'
```

Two export methods

Export Pier 1 or Pier 2 kernel:

```
PRKernelCodeExporter exportAsPier1or2Kernel: 'pier'.
```

Export Pier 3 kernel:

```
PRKernelCodeExporter exportAsPier3Kernel: 'nasite'.
```

dispatcher name



Pier 1 or 2

```
NickagercomKernelCreator>>systemcomponentsblogarchive: kernel
| structure |
structure := PRComponent named: 'blogarchive'.
structure title: 'Blog Archive'.
structure componentClass: PBPostArchive.
structure write: 10 using: PBPostArchive descriptionMaxItems.
structure write: true using: PBPostArchive descriptionShowCount.

structure securityDecoration owner: (kernel userNamed: 'nick.ager@gmail.com'
ifNone: nil).
structure securityDecoration group: (kernel groupNamed: 'admin' ifNone: nil).
^ structure
```

Pier 3

```
NickagercomKernelCreator>>systemcomponentsblogarchive: kernel
| structure |
structure := PRComponent named: 'blogarchive'.
structure title: 'Blog Archive'.
structure prototypeInstance: (PBPostArchive new
maxItems: 10;
showCount: true;
yourself).

structure securityDecoration owner: (kernel userNamed: 'nick.ager@gmail.com'
ifNone: nil).
structure securityDecoration group: (kernel groupNamed: 'admin' ifNone: nil).
^ structure
```


Pier 1 or 2

```
NickagercomKernelCreator>>systemcomponentsblogarchive: kernel
| structure |
structure := PRCComponent named: 'blogarchive'
structure componentClass: PBPostArchive.
structure write: 10 using: PBPostArchive descriptionMaxItems.
structure write: true using: PBPostArchive descriptionShowCount.
structure securityDecoration owner: (kernel userNamed: 'nick.ager@gmail.com'
ifNone: nil).
structure securityDecoration group: (kernel groupName: 'admin' ifNone: nil).
^ structure
```

Pier 3

```
NickagercomKernelCreator>>systemcomponentsblogarchive: kernel
| structure |
structure := PRCComponent named: 'blogarchive'.
structure title: 'Blog Archive'.
structure prototypeInstance: (PBPostArchive new
maxItems: 10;
showCount: true;
yourself).
structure securityDecoration owner: (kernel userNamed: 'nick.ager@gmail.com'
ifNone: nil).
structure securityDecoration group: (kernel groupName: 'admin' ifNone: nil).
^ structure
```

Pier 3 uses prototype instances for PRCComponent(s)

Test export #1

```
PRKernelCodeExporter exportAsPier1or2Kernel: 'pier'.
```

replace with
dispatcher name



Test export #2

Remove existing kernels:

PRKernel `reset.`

**backup
first!**



Test export #3

Remove existing application:

```
PRAdmin unregister: 'pier'.
```

*replace with
dispatcher name*



Test export #4

Create kernel and register application
using exported code:

```
PRKernelCreatorForPier register.
```

exporter generates a class named:

PRKernelCreatorForXXXX

in category:

'Pier-Exported-Code'

Test export #5

Test kernel:

<http://localhost:8080/pier>

Export a Pier 3 kernel

- Export Pier 3 kernel:

```
PRKernelCodeExporter exportAsPier3Kernel: 'pier'.
```

- Add a package to Monticello named:

```
'Pier-Exported-Code'
```

- Save package into a repository using Monticello browser

Install Pier 3 (with addons)

Into a fresh Pharo image install Pier with:

```
Gofer it
  squeaksource: 'MetacelloRepository';
  package: 'ConfigurationOfPier3Add0ns';
  load.
```

```
ConfigurationOfPier3Add0ns load
```


Upgrade demo

What has changed since the Pier 1

- Verbatim markup
- Requirement for a menu component has vanished.
- Pier-Book has deprecated: `B0Chapter`, `B0Part`, `B0Section`, `B0Subsection`
- Pier 3 introduces Pier-Admin
- Not all plug-ins have been ported.
- Take care if you've derived from `PRPierFrame` or `PRContentsWidget` or `PRKernel`

Status

- All tests green on Pharo and Gemstone.
- Multibyte strings are exported in a readable fashion.
- PRDistribution & PRKernelDistributionBase kernels works
- Upgraded Pier 1 - piercms.com site
- Upgraded Pier 2 (Gemstone) nickager.com site.

Upgrading roadmap

- Make sure you have good backups.
- Install the export code - relevant for your Kernel
- Try exporting in the same format and regenerate the kernel.
- Generate a Pier 3 kernel.
- Load Pier 3 (how to load Pier3....)
- Modify markup - such as new Verbatim markup
- Are all the plugins you used ported to Pier3?
- Edit the template (environment) to remove site editing tools now replaced by Pier-Admin

Why upgrade

- Use the latest Seaside versions with access to jQuery and all the associated plug-ins.
- Pier-Admin
- Many bugs fixed
- Blogs with social sharing widgets.
-

Next steps

- Does this work for you? War-stories
- Use the refactoring infrastructure to ease migration pain? eg Book refactoring
- Use Metacello and code-exporter to be able to recreate your site.

Further information

- Pier CMS site
<http://piercms.com>
- Pier Admin screen-cast:
<http://vimeo.com/32749535>
- An introduction to Magritte 3:
<http://vimeo.com/37032840>
- Magritte/Pier mail list:
[<smallwiki@iam.unibe.ch>](mailto:smallwiki@iam.unibe.ch)