

CS 683 Emerging Technologies

Fall Semester, 2005

Doc 6 Zope 3 - Basics

Contents

Reading Assignment.....	3
Zope Introduction.....	4
What is Zope?.....	4
Installing Zope.....	5
Using Zope Interface.....	8
Programing Zope.....	9
First Example & Motivation.....	9
Zope Schemas & Interfaces.....	10
Interfaces.....	13
Existing Schema.....	14
Schema Attributes.....	15
Persistence.....	16
Installing Book in a ZopeInstance.....	17

Copyright ©, All rights reserved. 2005 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

Web Component Development with Zope 3, Philipp von Weitershausen, Springer-Verlag, 2005

This document covers through Section 7.1 in the above text

The Zope 3 Developers Book - An Introduction for Python Programmers, Stephan Richter,
<http://www.zope.org/Wikis/DevSite/Projects/ComponentArchitecture/FrontPage/Zope3Book>

The Zope Book 2.6,
<http://www.zope.org/Documentation/Books/ZopeBook/>

Reading Assignment

Date	Chapters
Sept 20	2, 3, 4
Sept 22	5, 6, 7
Sept 27	8, 9, 10
Sept 29	11, 12, 13
Oct 3	14, 15, 16
Oct 6	17, 18, 19

The chapter numbers above refer to chapters in the book Web Component Development with Zope 3, von Weitershausen, Springer-Verlag, 2005

Zope Introduction

What is Zope?

Z Object Publishing Environment

Zope is a system for Web applications that

- Written in Python
- Supports/Automates much of the web application steps
 - Persistence
 - View creation
 - Data validation

Why Zope 3

Zope 3 cleans up some issues with Zope 2

Installing Zope

Instructions in Chapter 2 of von Weitershausen's book work for

- Unix
- Windows

Short Version of Instructions

Go to <http://www.zope.org/Products/Zope3/3.0.1>

Download correct version (exe for windows)

Follow the instructions in the README in the download

Creating a Zope Instance

Go to the directory that you want to contain your Zope files

Run (unix)

```
/usr/local/ZopeX3-3.0.1/bin/mkzopeinstance -d Zope3Instance -u aUsername:aPassword
```

or (windows)

```
C:\Python23\python\Scripts\mkzopeinstance -d Zope3Instance -u aUsername:aPassword
```

Note "Zope3Instance" will be a directory, so you can use another string

aUsername and aPassword are for manager of the zope instance

Multiple instance of Zope can run at the same time

Each instance uses a different port

Starting Zope Unix

```
cd Zope3Instance  
bin/runzope
```

Windows

```
cd Zope3Instance  
C:\Python23\python bin\runzope
```

Zope uses the port 8080 by default

Edit Zope3Instance/etc/zope.conf to change defaults

Using Zope Interface

Demo

Useful reading

[The new Web-based User Interface](#), Richter, ([PDF](#))

[The Zope Book](#) 2.6 ([PDF](#))

chapter [Using The Zope Management Interface](#)

Programing Zope

First Example & Motivation

Web Site for a Bookstore

For simplicity a book has

- Title
- Author
- Publication Date

```
class Book:  
    title = "  
    author = "  
    date = None
```

Issue - Data Validation

How to prevent?

```
badData = Book()  
badData.title = [1,2, 'dog', 3.234]  
badData.date = 'cat'
```

Zope Schemas & Interfaces

Schemas - used to indicate type information

Interface - used to specify schemas

```
import sys
sys.path.append('/usr/local/ZopeX3-3.0.1/lib/python/')

zope.interface import Interface
from zope.schema import TextLine, Date

class IBook(Interface):
    title = TextLine(
        title = u'Title',
        description = u'Title of the book',
        required = True)

    author = TextLine(
        title = u'author',
        description = u'Author of the book',
        required = True)

    date = Date(
        title = u'Date',
        description = u'Date book was published',
        required = True)
```

Using the Schema

```
import sys
sys.path.append('/usr/local/ZopeX3-3.0.1/lib/python/')

from zope.interface import implements
from zope.schema.fieldproperty import FieldProperty

class Book(object):
    implements(IBook)

    title = FieldProperty(IBook['title'])
    author = FieldProperty(IBook['author'])
    date = FieldProperty(IBook['date'])
```

Sample Program - Major Python Magic

```
noBadData = Book()
noBadData.author = ['cat', 1]          #Runtime error
```

The schema is trapping a direct assignment to an attribute to check type usage

Some Details of the Example

Zope path in PYTHONPATH

```
import sys  
sys.path.append('/usr/local/ZopeX3-3.0.1/lib/python/')
```

When run in Zope this is not needed

Uses `sys.path.append` rather than `PYTHONPATH` to make sure path dependency is explicitly shown

Path shown is for Unix install of Zope

Interfaces

```
zope.interface import Interface
```

```
class IBook(Interface):
```

```
from zope.interface import implements
```

```
class Book(object):  
    implements(IBook)
```

Interfaces are defined by a library not part of language

Compiler does not know about interfaces

Interfaces are subclasses of Interface

By convention Interface names start with "I"

A class must subclass object when implementing an interface

object is used to make schema type checking work

implements does not force the class to have all methods

```
class Book(object):  
    implements(IBook)  
    pass
```

Existing Schema Abstract

Field	Constraint
Container	Object that supports in operator
Field	Simple field with no constraints
Iterable	Object that supports iteration

Basic Types

Bool	bool
Bytes	Useful for binary data
BytesLine	Bytes, but no newline character
Date	datetime.date
Datetime	datetime.datetime
Dict	dict
Float	float
Int	int
List	list
Set	sets.Set
Text	Unicode text
TextLine	Text, but no newline character
Tuple	tuple

Special Types

ASCII	Text with only ASCII characters
DottedName	Python dotted name
Id	URI or Id
InterfaceField	zope.interface.Interface
Password	TextLine for passwords
SourceText	Contains text of some computed output
URI	Text containing a URI

Schema Attributes

Attributes for all Fields

Name	Type	Description
title	TextLine	Label for field
description	Text	Field description
required	Bool	Is field required, True default value
readonly	Bool	Is read only
default	Field	Default value if none is given
missing_value	Field	If input is not given, use this value
order	Int	Order in which fields are defined

Specialized Attributes

min, max	Int or Float	Range of Int and Floats
min_length, max_length	Int	Required length of sequences and strings
allowed_values	Container	Applies to enumerated types

[More Details](#)

Persistence

Modifying the book so book objects are stored in a database

```
from persistent import Persistent
from zope.interface import implements
from zope.schema.fieldproperty import FieldProperty
from bookstore.interfaces import IBook

class Book(Persistent):
    implements(IBook)

    title = FieldProperty(IBook['title'])
    author = FieldProperty(IBook['author'])
    date = FieldProperty(IBook['date'])
```

Zope will now store the book in ZODB

Zope Object Database

Installing Book in a ZopeInstance

We would like to add and modify books in Zope

Tell Zope about the bookstore Package

In the directory "Zope3Instance/etc/package-includes/"

Add the file "book-configure.zcml"

book-configure.zcml

```
<include package="bookstore"/>
```

bookstore Package

In "Zope3Instance/lib/python" create the directory bookstore

Files to add to "Zope3Instance/lib/python/bookstore"

- `__init__.py`
- `book.py`
- `interfaces.py`
- `configure.zcml`

Files in Zope3Instance/lib/python/bookstore

__init__.py

```
# make this directory a package
```

book.py

```
from persistent import Persistent
from zope.interface import implements
from zope.schema.fieldproperty import FieldProperty
from bookstore.interfaces import IBook
```

```
class Book(Persistent):
    implements(IBook)
```

```
    title = FieldProperty(IBook['title'])
    author = FieldProperty(IBook['author'])
    date = FieldProperty(IBook['date'])
```

interfaces.py

```
from zope.interface import Interface
from zope.schema import TextLine, Date

class IBook(Interface):
    title = TextLine(
        title = u'Title',
        description = u'Title of the book',
        required = True)

    author = TextLine(
        title = u'author',
        description = u'Author of the book',
        required = True)

    date = Date(
        title = u'Date',
        description = u'Date book was published',
        required = True)
```

configure.zcml

```
<configure xmlns="http://namespaces.zope.org/zope">

<interface
  interface=".interfaces.IBook"
  type="zope.app.content.interfaces.IContentType"
/>

<content class=".book.Book">
  <factory
    id="bookstore.book.Book"
    title="Create a new book"
    description="This factory instantiates new books"
  />
  <require
    permission="zope.View"
    interface=".interfaces.IBook"
  />
  <require
    permission="zope.ManageContent"
    set_schema=".interfaces.IBook"
  />

</content>

<include package=".browser" />

</configure>
```

Python Package names

```
<content class=".book.Book">
```

In ".book.Book" the leading period means:

Current package (which is bookstore)

So we could write this as:

```
<content class="bookstore.book.Book">
```

Browser files

These files tell Zope

- To list books as objects that can be created
- How to generate a page to edit a book

In "Zope3Instance/lib/python/bookstore" create the directory browser

In "Zope3Instance/lib/python/bookstore/browser" add files:

- `__init__.py`
- `configure.zcml`

We will be adding more later

`__init__.py`

```
# make this directory a package
```

configure.zcml

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
>

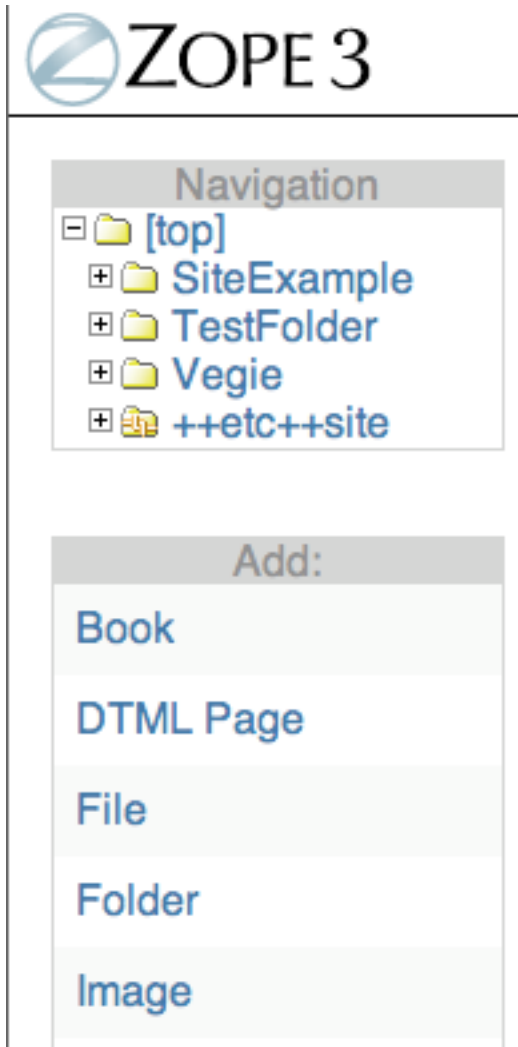
<browser:addMenuItem
  title="Book"
  class="bookstore.book.Book"
  permission="zope.ManageContent"
  view="AddBook.html"
/>

<browser:addform
  schema="bookstore.interfaces.IBook"
  content_factory="bookstore.book.Book"
  label="Add a Book"
  name="AddBook.html"
  permission="zope.ManageContent"
/>

<browser:editform
  schema="bookstore.interfaces.IBook"
  label="Edit"
  name="edit.html"
  menu="zmi_views" title="Edit"
  permission="zope.ManageContent"
/>
</configure>
```

Zope Page

Note the add box now contains "Book"



Auto-generated Form for adding a book

Add a Book

Title

author

Date

Refresh

Add

Object Name