Zope and RDBMS

A beginner's guide to working with relational databases from Zope

EuroPython Conference 2007
Vilnius, Lithuania

Charlie Clark
EGENIX.COM Software GmbH
Germany
Speaker Introduction: Charlie Clark

- **Sales & Support eGenix.com Consultant**
  - More than 10 years web production experience
  - MA, DESS in Film & TV production
  - Specialist in CMS in general and Zope in particular
  - Contact: charlie@egenix.com

- **eGenix.com Software GmbH, Germany**
  - Founded in 2000
  - Core business:
    - **Consulting:** helping companies write successful Python software
    - **Product design:** professional quality Python/Zope developer tools (mxDDBC, mxDateTime, mxTextTools, etc.)
  - International customer base
Agenda

1. Introduction
2. Who is this for?
3. Zope has everything you need
4. Why ODBC?
5. ACME Training
6. Applicability
7. Lessons
This talk contains nothing new
But some things are worth repeating
Who is this presentation for?

- **Zope / Plone beginners**
  - Around since 1997 and still going
  - There are a lot of us:
    - Larger userbase than Zope 3
  - Lots of non-programmers use Zope. Plone effect

- **Content managers who need RDBMS**
  - Use Zope/Plone based CMF for content
  - Need access to other data sources
  - Anyone who needs to access an RDBMS from the web
Zope has everything you need (1)

• Pretty good documentation
  – Fully documented API but there is a lot to learn
  – Hasn't really been updated since 2001

• PageTemplates
  – Virtually enforces presentation only

• PythonScripts
  – Harness the power and simplicity of Python for application logic
  – Support modularity
Zope has everything you need (2)

• ZSQL
  – Test SQL away from the application
  – Copy & paste SQL between browser and DB app
  – Don't have to worry about SQL injection

• ExternalMethods
  – PythonScripts have some restrictions
  – Use ExternalMethods for non-Zope stuff

• Through the web (TTW)
  – Just needs a browser
  – Who needs Web 2.0?

• Immediate success, maintainable code
Why ODBC ? (1)

• Almost universal compatibility
  – Drivers for Oracle, MS SQL, DB2, PostgreSQL, Informix, Sybase, etc.
  – MS Windows, Linux, Solaris, AIX, BSD, Mac OS

• Well-maintained and up to date
  – Python 2.5, 64-bit
Why ODBC? (2)

• Setup can be complex...
  – Python ODBC driver, ODBC manager, ODBC bridge, database ODBC driver, network, permissions
  – Usually straightforward for development

• Catalog(ue) methods
  – Standardised access to database meta data
Catalog(ue) methods

- Document your application

mxODBC Database Model for CASI
Data modelling is your friend

• Data definition isn't programming
  – What not how
  – More than just storage
  – Let the database manage the data based on what you tell it

• Take time to understand your data model
  – A primer on modelling and data definition is more important than a handbook for a particular RDBMS
  – There is no O/R silver bullet

• Queries are easy
  – Once you've mastered basic syntax
  – JOINS, GROUP and ORDER
  – Application independent
ACME Trading

• Screens
  – What you expect to see and do
  – First HTML
    • Iron out HTML / CSS errors
  – Then PageTemplates
    • Tie design to application

• Control
  – Provide access to your screens through scripts
  – One place to control what happens next

• Data access
  – Individual methods for querying, adding or altering data
  – Test at anytime
What fields are required and what feedback is necessary

Our seminars will teach you all want to know about our famous RoadRunner range!

Choose a seminar  
First Name  
Surname  
Company  
Tel  
E-Mail  

Senden  Zurücksetzen
ACME Trading management

Familiar filesystem type environment
Try and fit the logic in one screen

```python
required = ['firstname', 'lastname', 'email']
errors = []
options = {}

if request.form.get('action') == 'register':
    for field in required:
        if not request.form.get(field):
            errors.append(field)
    if errors:
        return context.register(errors=errors)
# check to see if it is a customer
customer = context.validate(request.form)
if customer:
    # call method to enter the request into the seminar
    # send e-mail to customer, etc.
    # will raise an error if customer already registered
    context.SQL.iCustomer(id_customer=customer, id_event=request.form['seminar'])
    text = context.thankyou
```
ACME Trading database

Naming convention essential

```plaintext
Folder at /ACME/SQL
```

<table>
<thead>
<tr>
<th>Type Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>dCustomerEvent (Remove a customer from an event)</td>
</tr>
<tr>
<td>iCustomer (Add a customer to an event)</td>
</tr>
<tr>
<td>qCustomer (Check for an existing customer in the sales database)</td>
</tr>
<tr>
<td>qEvent (List customers by events)</td>
</tr>
<tr>
<td>qSeminar (List all seminars taking place after today)</td>
</tr>
</tbody>
</table>

(c) 2007 EGENIX.COM Software GmbH, info@egenix.com  
EuroPython Conference 2007
ACME Trading database

Develop queries in a query editor

```
SELECT id_event, event_date, title
FROM event
INNER JOIN seminar ON
    (event.id_seminar = seminar.id_seminar)
WHERE event_date > now()
ORDER BY event_date
```

![Database query and result screenshot]

(c) 2007 EGENIX.COM Software GmbH, info@egenix.com
Use same approach in management screens

List upcoming events and attendees

**Beginner** – 2007/07/31
Charlie Clark, eGenix remove
Frank Gallagher, Shameless remove

**Intermediate** – 2007/08/07
John Cleese, Monty Python remove

**Advanced** – 2007/08/14
Elmar Fudd, Fudd remove

**Evil Genius** – 2007/08/21
Bugs Bunny, Carrots, Inc. remove
Applicability

• Where TTW is desirable or requirement
  – Access to Zope instance in the file system might not be possible
  – For many browser-based environment is less intimidating
  – Text editors without syntax highlighting are worse than useless (Notepad)
  – ExternalEdit available if required

• Small-add-ons
  – Extend an existing site rather than build a new application

• Single language
  – Plain Zope2 isn't really suitable for multilingual environments
  – Fortunately this often isn't the case for small projects
Lessons (1)

• Zope encourages divide & rule
  – MVC: model, view, controller approach comes naturally in Zope
  – Delegation encouraged

• Modularity
  – Individual components have well-defined roles and relationships
  – Sensible reuse
    • ZPT macros
    • PythonScripts such as validators might be called by different controllers
    • ZSQL methods can be made flexible with dtml-if
Lessons (2)

• Extremely iterative
  – Beginners don't have a masterplan
  – Learn & adapt as you go on
  – Get immediate feedback

• Victim of your own success
  – Small is beautiful
  – Successful projects tend to grow
  – Learn the limits
  – Use when required: Products, CMF, Zope 3
    • What you've learned in your first projects will be useful

• Don't neglect beginners
eGenix.com Software, Skills and Services GmbH
Charlie Clark
Pastor-Löh-Str. 48
D-40764 Langenfeld
Germany

eMail: charlie@egenix.com
Phone: +49 211 9304112
Fax: +49 211 3005250
Web: http://www.egenix.com/