

MySQL でサイト構築 ~ Zope との連携テクニック ~

MySQL の特徴はその快適なスピードにある。特に、Web などの応答性の要求される場面では MySQL を使うメリットは十分にあるといえるだろう。

そして、今話題のアプリケーションサーバ Zope との連携で、プログラミング技法が全くと言っていいくらい変わってしまうかもしれない。

今までのような CGI に始まり、PHP、JSP などの Web アプリケーション作成技法とは違う方法で、簡単で且つ強力な Web アプリケーションを作成する。

また、これらがオープンソースプロダクトであるということにもぜひ注目してほしい。

Why MySQL? ~MySQL を使う理由~

- 単純に応答速度を重視して作られている
- マシンへの負荷がかなり小さい
- 通常使われるファイル編成(MyISAM)ではトランザクションをサポートしないが、Web 上ではリードオンリー、もしくはセッションレスな部分がほとんどであるため、実用に支障があまり無い

Web 用途に特化した使い方をするには適してるかも。

(Slashdot、SourceForge など、OSDN サイトはほとんど MySQL が使われている。)

Why Zope? ~Zope を使う理由~

- 従来のスクリプト言語などと違い、一つのシステムとして存在
- それなりにアクの強いところを感じるかもしれないが、それに一旦なじめば、CGI を書いたり、他の環境に移るのがかたくなるくらいよくできている
- 全てのオブジェクト編集で UNDO を適用できる

ところで、Zope って何?

- オープンソースアプリケーションサーバ

Object Database (ODB)としての機能を持ち、またコンテンツ管理機能も備えている。

- ライセンスが緩やか、version 2.5.0 からは GPL 互換
- サポートされているプラットフォームが豊富 (Linux、*BSD、Mac、Windows、Solaris etc...)
- 本体のほとんどが Python で構築されている。

一言で言うと... 『**墮落のアプリケーションサーバ**』

『もう、Zope なしでサイトを管理するなんてとても考えられない』 (一件落着インターネットサービス株式会社 スタッフのコメントより~)

両方に言えること

- Free Software である

Zope をインストールするには

併せて日本 Zope ユーザ会のハウトゥードキュメント(<http://zope.jp/howto/>)や、Software Design 2002 年 2 月号を参照すると良い。

Bash の環境変数は以下にセットしてある:

```
PS1='[¥u@¥h ¥W]¥$'
```

1. 家元の <http://www.zope.org/> から tarball をダウンロード
 - x86 Linux と Windows、SPARC Solaris は Zope.org がバイナリパッケージも提供している。また、Python も同梱されている (RedHat Linux は gcc が Buggy なのでこちらを強く推奨)
 - wget を使って Zope 2.5.0 x86 Linux(libc6)用バイナリパッケージをダウンロードする例:

```
[somebody@host somewhere]$ wget http://www.zope.org/Products/Zope/2.5.0/Zope-2.5.0-linux2-x86.tgz  
(実際は一行)
```

2. Zope 用ユーザを作成(自分のアカウントで行うことも可能):

```
[root@host /root]# adduser zope  
Adding user zope...  
Adding new group zope (1003).  
Adding new user zope (1003) with group zope.  
Home directory /home/zope already exists. Not copying from /etc/skel  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
Changing the user information for zope  
Enter the new value, or press return for the default  
Full Name []: Zope  
Room Number []:  
Work Phone []:  
Home Phone []:  
Other []:  
Is the information correct? [y/n] y
```

3. 任意のディレクトリに Zope の tarball を展開 (/usr/local に作る場合):

カレントディレクトリを変更

```
[root@host /root]# cd /usr/local
```

アーカイブを展開

```
[root@host local]# tar zxvf Zope-2.5.0-linux2-x86.tgz
```

Zope 用ユーザに所有権を変更

```
[root@host local]# chown -R zope Zope-2.5.0-linux2-x86/
```

4. カレントユーザを Zope 用ユーザに変更してセットアップ:

```
[root@host local]# su zope
```

```
[zope@host local]$ cd Zope-2.5.0-linux2-x86/
```

セットアップスクリプトを起動

バイナリ版の場合、install スクリプトを起動してインストール

```
[zope@host Zope-2.5.0-linux2-x86]$ ./install
```

ソース版の場合、wo_pcgi.py を起動してコンパイル

```
[zope@host Zope-2.5.0-src]$ ./wo_pcgi.py
```

どちらも終了すると "Done!" というメッセージが出る

5. zpasswd.py を起動してスーパーユーザのパスワードを設定 (access をパスワードファイルにする場合):

```
[zope@host Zope-2.5.0-linux2-x86]$ ./bin/python zpasswd.py access
```

Username: zopeadmin (Zope のスーパーユーザ名)

Password: (パスワード)

Verify password: (パスワード再入力)

Please choose a format from:

SHA - SHA-1 hashed password

CRYPT - UNIX-style crypt password

CLEARTEXT - no protection.

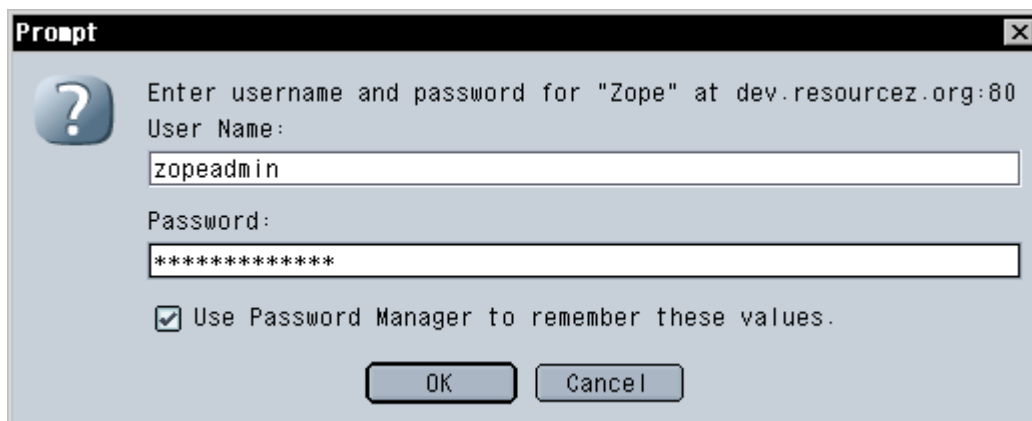
Encoding: SHA (上記のどれかを選択 上2つは暗号化方式 3番目は暗号化無し)

Domain restrictions: (ドメインの制限 必要なければそのまま Enter)

6. start スクリプトで Zope を起動
(起動後ログアウトしてもバックグラウンドで動作):

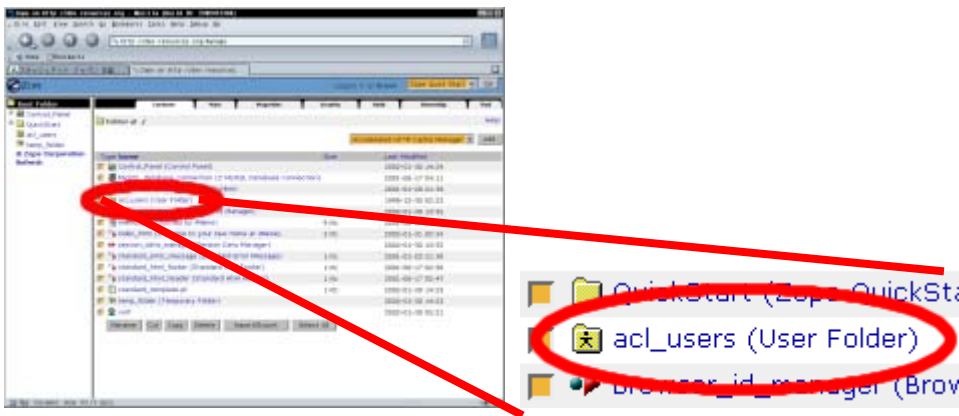
```
[zope@host Zope-2.5.0-linux2-x86]$ ./start &
-----
2002-02-04T02:27:10 INFO(0) ZODB Opening database for mounting: '142099072_1012789591.857869'
-----
2002-02-04T02:27:10 INFO(0) ZODB Mounted database '142099072_1012789591.857869' at /temp_folder
-----
2002-02-04T02:27:10 INFO(0) ZServer HTTP server started at Mon Feb  4 11:27:10 2002
      Hostname: host.resourcez.org
      Port: 8080
-----
2002-02-04T02:27:10 INFO(0) ZServer FTP server started at Mon Feb  4 11:27:10 2002
      Hostname: host
      Port: 8021
-----
2002-02-04T02:27:10 INFO(0) ZServer CGI Server started at Mon Feb  4 11:27:10 2002
      Unix socket: /usr/local/Zope-2.5.0-linux2-x86/var/pcgi.soc
```

Zope の管理画面へログインし、通常使う Manager アカウントを発行
(デフォルトでの URL は <http://host:8080/>)



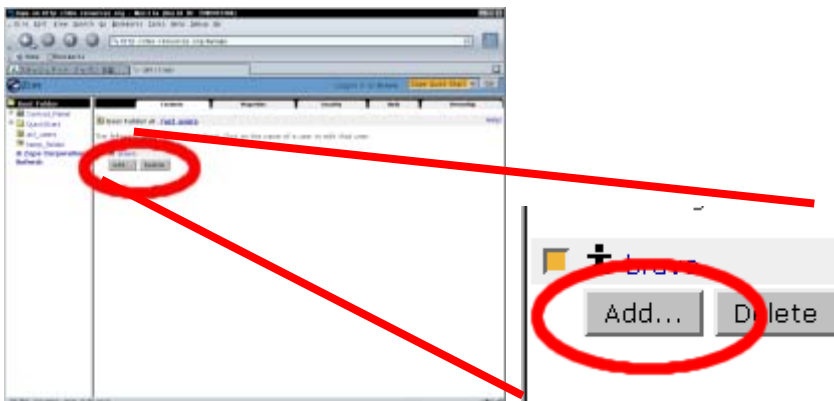
スーパーユーザでログイン

acl_users フォルダに入る

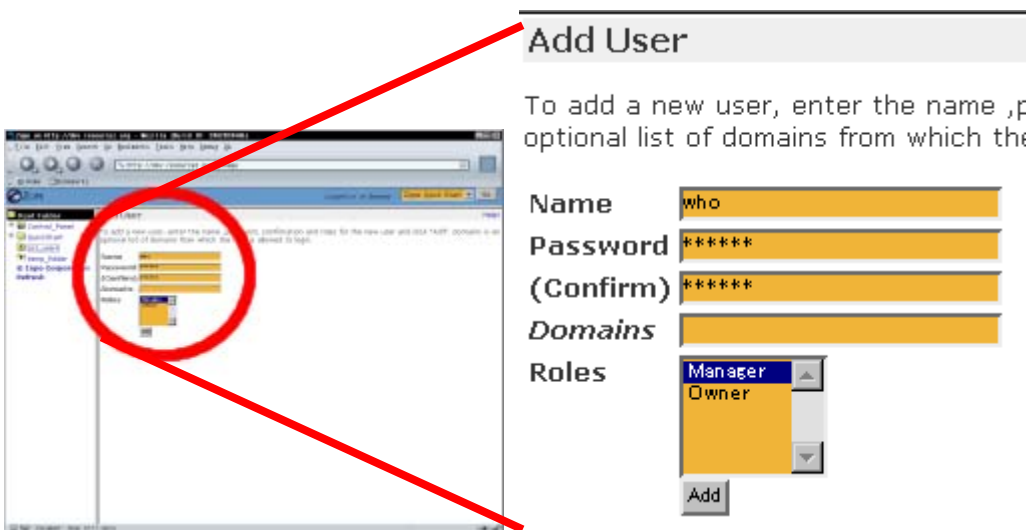


ユーザフォルダのアイコン

Add ボタンでユーザを作成、Manager 権限を使えるようにする



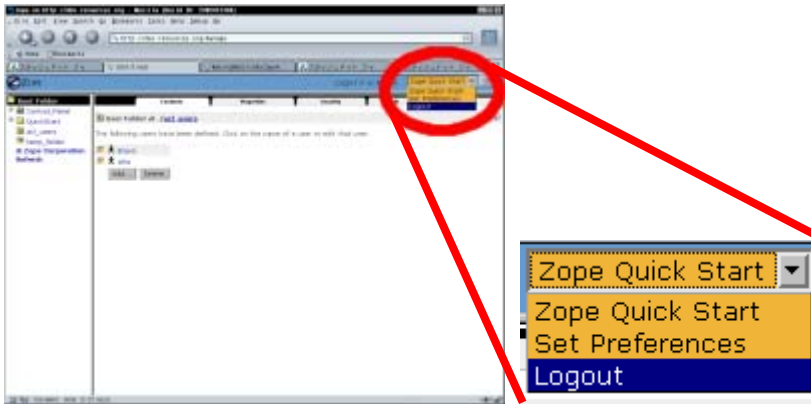
acl_users の中身



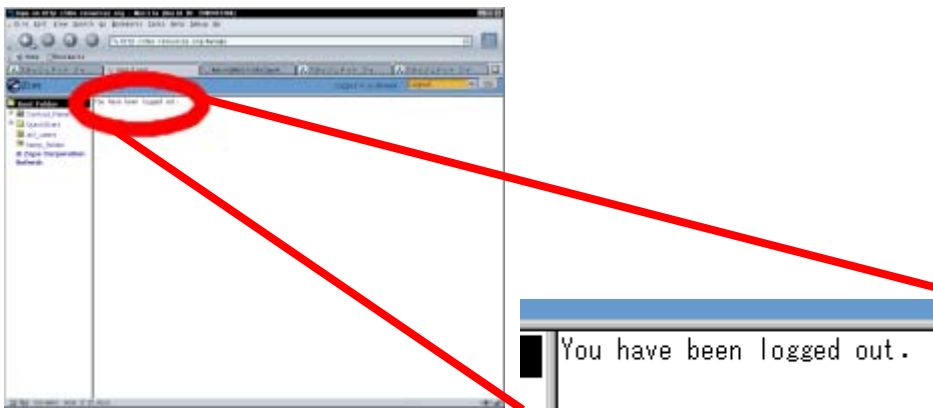
ユーザの追加

一旦ログアウトして、先ほど作成したアカウントでログイン

ログアウト手順

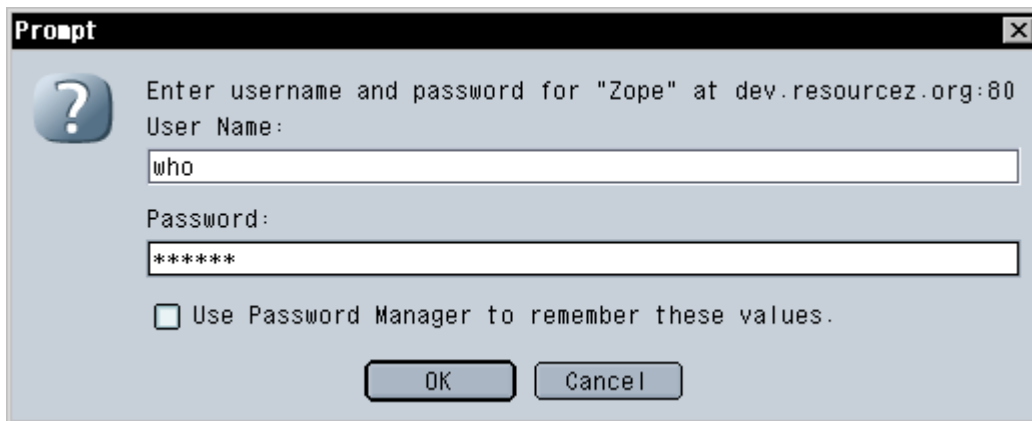


ログアウト



ログアウト時のメッセージ

再ログイン



Prompt

Enter username and password for "Zope" at dev.resourcez.org:80

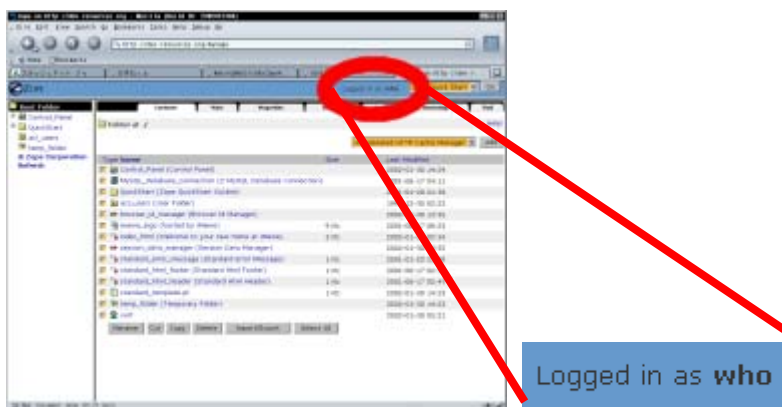
User Name:
who

Password:

Use Password Manager to remember these values.

OK Cancel

作成したユーザで再ログイン



ユーザ名が作成したユーザ名になっている

MySQL に Zope 用ユーザの作成

1. mysql.user に Zope 用アカウント(閲覧用、管理用)を用意する

閲覧用:

```
mysql> INSERT INTO mysql.user (host, user, select_priv)
-> values ('localhost', 'zope', 'Y');
```

管理用:

```
mysql> INSERT INTO mysql.user
-> (host, user, password,
-> select_priv, insert_priv, update_priv, delete_priv, create_priv)
-> VALUES ('localhost', 'zopeadmin', password('hogehoge'),
-> 'Y', 'Y', 'Y', 'Y', 'Y');
```

2. mysqladmin で権限テーブルをリロードする:

```
[somebody@host somewhere]$ mysqladmin reload -p -u root
Password: (パスワード)
```

ZMySQLDA をインストールする

1. Zope で使われている Python に MySQL ライブラリを追加する

ファイルの展開:

```
[somebody@host somewhere]$ tar zxf MySQL-python-0.8.0.tar.gz
```

Zope がバイナリパッケージの場合:

```
[somebody@host somewhere]$ Zope-2.5.0-linux-x86/bin/python MySQL-python-0.8.0/setup.py install (実際は1行)
```

Python は既存のもので、Zope はソースからインストールしてある場合:

```
[somebody@host somewhere]$ su
[root@host somewhere]# python MySQL-python-0.8.0/setup.py install
```

Zope をビルドした Python(バイナリパッケージでは Zope に同梱されている Python)を必ず使うこと

2. ZMySQLDA を Zope のプロダクトに追加:

```
[root@host somewhere]# su zope
[zope@host somewhere]$ cd Zope-2.5.0-linux-x86 (Linux バイナリ版の場合)
[zope@host Zope-2.5.0-linux-x86]$ tar zxf ZMySQLDA-0.8.0.tar.gz
```

Z MySQL Database Connection と Z SQL Method を使う

Z MySQL Database Connection とは

- データベースとのやりとりを行うためのコアオブジェクト
- 次の Z SQL Method や、Z Data Query Kit を使ってデータベースを操作する

Z SQL Method とは

- SQL で制御するためのオブジェクト
- 単体で使うこともできるが、実際は DTML や PageTemplate、PythonScript から呼び出して使う

Z MySQL Database Connection の作成とテスト及びブラウザ Select 文などの Z SQL Method の作成とテスト

1. Z MySQL Connection を作成

- Zope Management Interface(以下 ZMI)のプルダウンメニューから「Z MySQL Connection」を選択
- Id、Title、接続パラメータを入力して作成

Id: osdn Title: OSDN プレゼンテーション DB Param: presentation
zopeadmin hogegege

2. それぞれのコネクションオブジェクトからテーブルが閲覧できるかを調べる

- Browse タブでテーブルを参照する
- Test タブから直接 SQL を使う

3. Z SQL Method の作成

- 先ほど作成した閲覧用 Z MySQL Connection オブジェクトから下位のディレクトリならどこでも良い
- ZMI のプルダウンメニューから「Z SQL Method」を選択
- Id、Title、Connection Id、Arguments、Query Template をそれぞれ入力
- Query Template には Database Method DTML が使われる

Database Method DTML は、DTML (Zope で使用するマークアップ言語) を SQL に埋め込むことで、動的に SQL を生成する

閲覧するための SQL:

Id: select
Connection Id: osdn
Arguments: number
Query Template:

```
SELECT
    slide_title,
    slide_read,
    slide_number,
    slide_date
FROM
    slide_article
WHERE
    slide_number = <dtml-sqlvar number type=int>
```

挿入するための SQL:

Id: insert
Connection Id: osdn
Arguments: slide_title slide_read slide_number slide_date
Query Template:

```
INSERT INTO
    slide_article
VALUES
    (<dtml-sqlvar slide_title type=string>,
    <dtml-sqlvar slide_read type=string>,
    <dtml-sqlvar slide_number type=int>,
    <dtml-sqlvar slide_date type=string>)
```

更新するための SQL:

Id: update
Connection Id: osdn
Arguments: slide_title slide_read slide_number slide_date
Query Template:

```
UPDATE
    slide_article
SET
    slide_title = <dtml-sqlvar slide_title type=string>,
    slide_read = <dtml-sqlvar slide_read type=string>,
    slide_number = <dtml-sqlvar slide_number type=int>,
    slide_date = <dtml-sqlvar slide_date type=string>
WHERE
    slide_date = <dtml-sqlvar slide_date type=string>
```

これらの他にも細かいことをするためにテーブルをカウントするオブジェクトなどを作る

Database Method DTML について

- Z SQL Method では、引数が全て文字列に変換され、Query Template に与えられる
- データベースでは文字データと数値データを扱うため、引数が文字列である場合はクオート文字『 ' 』で挟む必要になる。逆に数値データの場合ではそのまま数値を与えるため、クオート文字が不要になる。
- 文字データの場合に単にクオート文字を挟めばいいようだが、文字データ中にクオート文字が入ってくると厄介
- Z SQL Method 用に拡張された dtml-sqlvar タグを使うことで、これらの問題を解決する

PageTemplate と PythonScript による作成

PageTemplate とは

- テンプレート記述言語の一種
- XHTML ベースで、属性を拡張することによって、動的なページを生成する
- Zope2.5.0 から標準になり、DTML から移行することを推奨している

PythonScript とは

- Zope の API を Python を使って呼び出せるようにしたもの
- DTML、PageTemplate でも Zope API を使うことが可能だが、長くなるにつれ可読性に欠けてくるのでこちらがオススメ
- 独自にセキュリティポリシーを組み込んであるので安全に使える（無限ループに制限がある等）

表示用の画面を作る

スライド用の PageTemplate

Id: slide

```
<html>
<head>
<title><span tal:replace="here/title" /> <span
  tal:replace="here/totalpage" /></title>
<span tal:replace="structure here/presentation" />
<span tal:replace="structure here/script" />
</head>
<body>
<div style="height:600; margin:0px; padding:0px"
  tal:define="item python:container.select_logic(request.form); undef
string:Undefined">
<h1 tal:content="structure item/slide_title | undef">title</h1>
<hr size="5" width="100%" />
<p tal:content="structure item/slide_read | undef">read</p>

<form name="slide" action="forward" method="get">
<input type="hidden" name="page"
  tal:attributes="value request/form/page | undef" />
</form>
</div>
</body>
</html>
```

名前空間「tal」に含まれる属性について

- content: 要素の間に含まれるオブジェクトを指す
- replace: 要素そのものを指定したオブジェクトにすり替える
- define: 要素に囲まれたオブジェクトの別名を定義する
- repeat: リストのようなオブジェクトの繰り返し

他 Python の構文を埋め込んだりすることも可能

データ加工用の PythonScript

Id: select_logic

Parameter List: form=None

```
# 送信されてきたデータに page が入っていたら値を取得して整数型に変換
if form.has_key('page'):
    page = form['page']
    try:
        page = int (page)
    except:
        page = 1
else:
    page = 1

# Z SQL Method を介してデータを取得
result = container.select (number = page)

# 1 つのデータを返すだけなので、リストの最初のデータを返す
if result:
    return result.dictionaries ()[0]
else:
    return None
```

PageTemplate から直接 Z SQL Method を呼び出すことも可能でも、スクリプトを仲介させればエラー処理などが比較的楽に作成することが可能となる

PythonScript を介すメリット

- Python はもともとプログラミング言語なのでロジックを記述しやすい
- PageTemplate などの、表示能力に強いオブジェクトが行うタスクを減らすことができ、構文を整然とさせることができる

参考リソース

英語

- Zope Community <http://zope.org/>
- ZPT - Zope Page Templates <http://www.zope.org/Wikis/DevSite/Projects/ZPT/FrontPage>
- TAL Specification
<http://www.zope.org/Wikis/DevSite/Projects/ZPT/TAL%20Specification%201.4>

日本語

- 日本 Zope ユーザ会 <http://zope.jp/>
- 日本 Python ユーザ会 <http://python.jp/>
- ZopeBook 日本語翻訳プロジェクト http://z2.dns.cx/ZopeBook_JP/FrontPage
- SoftwareDesign 2000 年 9 月号 『Zope で実現 , オープンソース Web アプリケーション』
- SoftwareDesign 2002 年 2 月号 『Zope が開く Web アプリケーションサーバ新時代』