

[Page One](#)
[Campus Computing News](#)
[Peer-to-Peer Software and You](#)
[Summer Hours](#)
[EDUCAUSE in Dallas This Year](#)
[Today's Cartoon](#)
[RSS Matters](#)
[The Network Connection](#)
[Link of the Month](#)
[WWW@UNT.EDU](#)
[Short Courses](#)
[IRC News](#)
[Staff Activities](#)
[Subscribe to Benchmarks Online](#)

# Research and Statistical Support

## University of North Texas

### RSS Matters

Link to the last RSS article here: [Publishing HTML Documents on the Web2survey Zope Server](#) - Ed.

## New Developments with Web Based Surveys on the Web2Survey Zope Server

By [Dr Rich Herrington](#), ACS Research and Statistical Support Services Consultant

**RSS** has previously published tutorials on creating HTML surveys/questionnaires on the Zope server, Web2survey ([Benchmarks Online - Dec. 04](#)). This month we are going to review some of the changes that have been implemented with the web application software QSurvey. Additionally, we will highlight some of the "Gotchas" that can be avoided when implementing HTML surveys on <https://web2survey.unt.edu>, as well as provide some helpful hints. **So what is Zope and QSurvey anyway?** To address this question, we first spend some time discussing the notion of: content management systems; content management frameworks; and web-application frameworks.

### Overview of CMS, CMF, and Web Application Frameworks

[Content Management Systems](#) (CMS) have become increasingly important in managing the rapidly increasing volumes of information available to organizations. Simply put, CMS allow the creation and maintenance of documents in a collaborative environment. The popularity of web-based collaboration has [spawned numerous CMS](#) based on web browser technology and related protocols (e.g. [HTTP](#), [WebDAV](#)). These browser based interfaces allow users to interface with content, and the functionality necessary to manage that content. Related [Content Management Frameworks](#) (CMF) are application programming interfaces (API) underlying these CMS (e.g. the [CMS Plone](#) uses the [CMF Zope](#) which is implemented in the programming language "[python](#)"). CMS are often utilized to manage [wikis](#), [learning environments](#), [online communities](#), Enterprise Content Management ([ECM](#) - e.g. record management; workflow management), and [digital libraries](#) of various sorts (e.g. [images](#)). Notably, there a number of the CMS and CMF projects are organized as [open-source](#) or free/open-source ([FLOSS](#)). Indeed, with [so many CMF/CMS](#) available (both proprietary and free), it can be difficult to choose and commit to a solution that can meet your content management needs. Making the "commit" decision even more unclear, are the many lightweight, "[web application frameworks](#)", that allow rapid pro-typing and deployment of web applications. These web application frameworks allow for database access, [session management](#), web-interface templating, and some security provisions. Less comprehensive than the CMS frameworks, the web application frameworks aim at rapid deployment through architecture templating or [design patterning](#) (e.g. general repeatable solutions to a commonly-occurring problems). Some of the more visible projects are [Turbo Gears](#), [Ruby on Rails](#), [Django](#), [CherryPy](#), [CakePHP](#), and [Java EE](#) (for a complete listing, see the Wikipedia entry on [Comparisons of Web Application Frameworks](#)). Finally, to "seal our confusion", a few of the CMF cross-over the CMS/CMF and web

application framework categories we have outlined. Two such notable products are the CMF Zope (for more details, see the free book: [The Zope Book](#)) and the related CMS Plone (for more details, see the free book: [The Definitive Guide to Plone](#)).

## Zope and the Related CMS Plone

The [CMF Zope](#) is an free/open-source object oriented web-application server that is written in the programming language [python](#). [Zope](#) comes with the [Zope Object Database \(ZODB\)](#) which allows objects (e.g. web pages, python programs, [URL's](#), folders, etc.) to be stored or persist in a [transactional database](#). As such, operations performed on objects in this database can be "undone" if needed. Basic "[object](#)" types are provided for users to create and manage through a point-and-click web interface (e.g. web-pages, [page templates](#), folders). This basic functionality is extended through add-on products that are specialized to implement more advanced functionality: [wikis](#), [blogs](#), [forms creation](#), database API connections to [RDBMS](#), etc. Hundreds of such products have been created in a highly visible [community of developers](#), most of who, have licensed their software as free/open-source projects (e.g. see <http://www.zope.org/Products/> - note: the [QSurvey](#) product is merely one such product among hundreds of add-on products). One product that deserves singling out is the open-source CMS product, [Plone](#). Plone is a CMS built on top of the CMF Zope. Plone can be used as [document publishing system](#) or as a part of a more extensive [groupware solution](#). One of the exciting characteristics of the Plone system is the rapidity with which one can bring a fully functioning CMS on-line. Our [RSS Plone site](#) (which we haven't really found the time to develop more fully, as we are mostly under-staffed in our office):

The screenshot shows a web interface for the University of North Texas (UNT) Zope Survey Server. At the top, there is a logo for the University of North Texas and the text "Computing and Information Technology Center". Below this is a search bar and a navigation menu with items: "home", "news", "members", "richherr2", "my folder", "my preferences", "undo", "plone setup", and "log out". A breadcrumb trail indicates "you are here: home".

The main content area is divided into several sections:

- Navigation:** Includes "Home" and "Members".
- Recent items:** A message stating "No items published or changed since your last log-in." with a "More..." link.
- Main Content:** A large green-bordered box containing:
  - Buttons for "contents", "view", "edit", "properties", "sharing", "add new item", and "state: visible".
  - Text: "Welcome to the UNT Zope Survey Server" with icons for email, print, and refresh.
  - Text: "This site is hosted by the Research and Statistical Support group at the University of North Texas".
  - Text: "The purpose of this website...."
  - Text: "Created by richherr2" and "Last modified 2006-03-15 11:44 AM".
- Calendar:** A calendar for May 2006 with the 13th highlighted in orange.

Through the use of [dynamic web pages](#), [templating](#), and [object code reuse](#) (e.g. using [DTML](#)), it is relatively easy to customize and maintain large scale web sites that serve dynamic content backed by databases whose content changes on a frequent basis. While dynamic templating or [server-side scripting languages](#) have been greatly utilized in web development, projects such as Zope and Plone combine the idea of a web development framework (complete with deeply integrated security provisions and session management/user management), with dynamic templating - all backed by a built-in, persistent, transactional database. (e.g. [DTML](#) and [ZPT](#) web page tags; a notable [series of Zope tutorials on DevShed](#)). Furthermore, the fundamental object scripting language that is used in implementing Zope and Plone is the [highly regarded python language](#). A recent [review of CMS systems](#) ranked the Zope/Plone combination highly in a review of 15 CMS products (on this same site there are a [number of](#)



[other CMS](#) reviews as well). [NASA's JPL \(the Object Oriented Data Technology \(OODT\) subgroup\)](#) has produced an informative (at least entertaining) comparison of a number of CMS and web-app. frameworks. The comparison is [available as a QuickTime movie](#) - careful, this is a large download of about 390 megabytes!

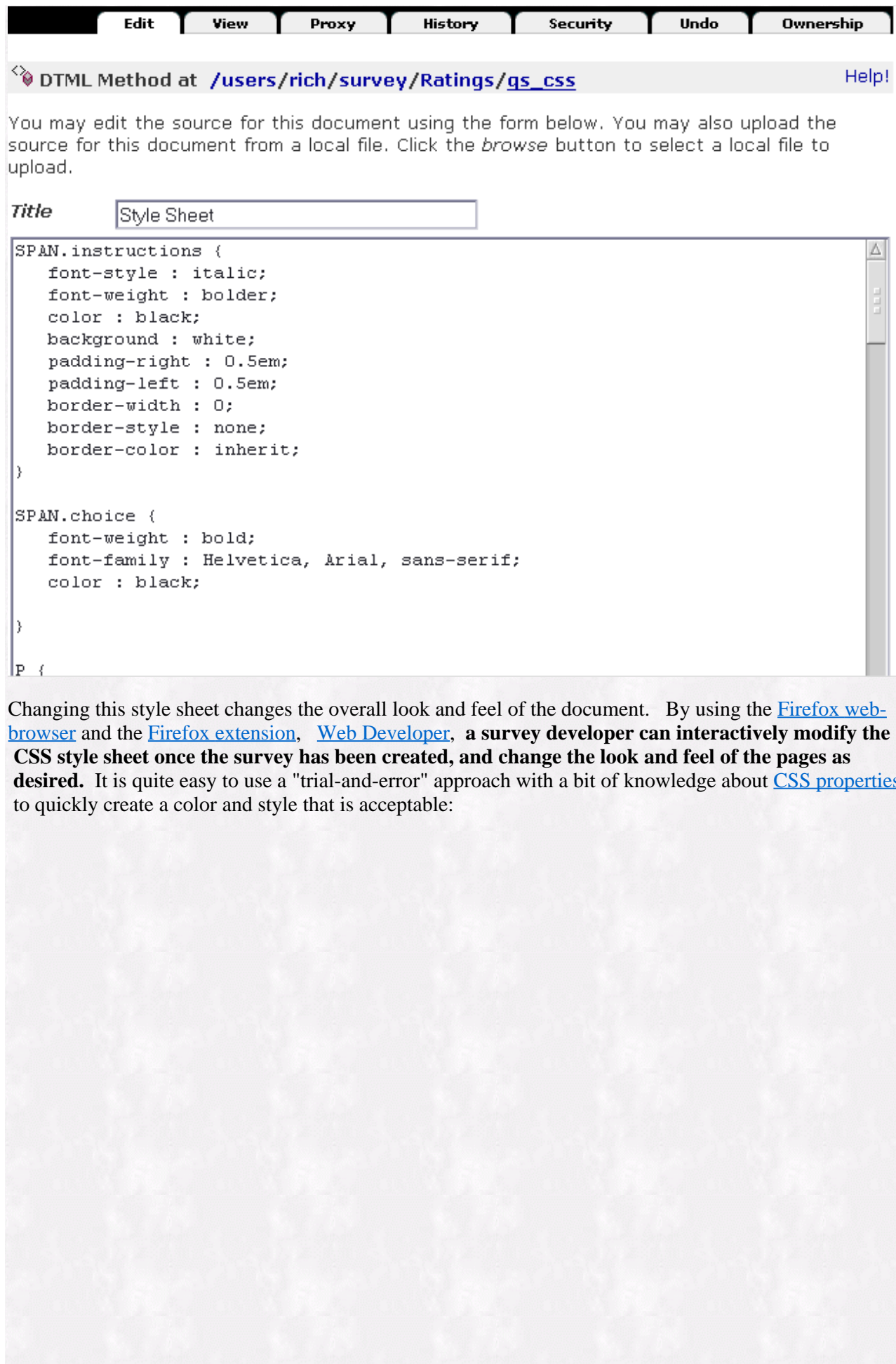
## QSurvey - A Simple Zope Application for Creating HTML Forms Based Surveys

The [QSurvey](#) product is only one of hundreds of products available at <http://www.zope.org>. The original idea behind QSurvey was to let [Zope](#) take care of the storage (e.g. [ZODB](#)), while the user focused on design issues - creating the survey with a tab-based, point-and-click interface:

Name	Value	Type
question	Mark the what you believe is the most i	string
special_instructions		string
doc_sequence	3.0	float
is_demographic	<input type="checkbox"/>	boolean

Save Changes

The design of the survey is organized such that a "QSurvey" object contains "QPages" and QPages contain question types and [DTML](#) documents and/or images (i.e. additional HTML pages). The styling of the HTML pages are accomplished by the use of default [Cascading Style Sheets \(CSS\)](#) templates that can be modified as desired:



DTML Method at [/users/rich/survey/Ratings/qs\\_css](/users/rich/survey/Ratings/qs_css) [Help!](#)

You may edit the source for this document using the form below. You may also upload the source for this document from a local file. Click the *browse* button to select a local file to upload.

*Title*

```
SPAN.instructions {
  font-style : italic;
  font-weight : bolder;
  color : black;
  background : white;
  padding-right : 0.5em;
  padding-left : 0.5em;
  border-width : 0;
  border-style : none;
  border-color : inherit;
}

SPAN.choice {
  font-weight : bold;
  font-family : Helvetica, Arial, sans-serif;
  color : black;
}

P {
```

Changing this style sheet changes the overall look and feel of the document. By using the [Firefox web-browser](#) and the [Firefox extension, Web Developer](#), a survey developer can interactively modify the **CSS style sheet once the survey has been created, and change the look and feel of the pages as desired.** It is quite easy to use a "trial-and-error" approach with a bit of knowledge about [CSS properties](#) to quickly create a color and style that is acceptable:

The screenshot shows a web browser window with a survey form and a CSS editor window. The survey form contains the following text and elements:

**What Characteristics do you find important in choosing a university to attend?**

**Which US State do you reside in?**

Texas

**Mark the what you believe is the most important characteristic in the following list of characteristics**

Reliability  Service  Quality  Price

**Please select the most appropriate response**

at Important  
Important  
Important At All

The CSS editor window shows the following code:

```

SPAN.instructions {
  font-style : italic;
  font-weight : bold;
  color : black;
  background : white;
  padding-right : 0.5em;
  padding-left : 0.5em;
  border-width : 0;
  border-style : none;
  border-color : inherit;
}

SPAN.choice {
  font-weight : bold;
  font-family : Helvetica;
  color : black;
}

P {
  font-family : Utopia;
  border-width : 0;
  border-style : none;
}

H2 {background-color:light
font-family : Utopia, Ver
border-width : 0;
border-style : none;

```

**Results from submitted surveys are available in XML format or in TAB delimited format for download, which allows easy import into other software systems such as SPSS and Excel:**

Below: a smaller section of tabs at the top of the RESULTS folder. These tabs allow download of TAB delimited files of single select, multiple select, and text responses. "Count" gives a count of the total submitted records to date:



Below: a larger window of the RESULTS folder. (notice that the username (e.g. -1497893754) is a hashed representation of an authenticated user (in this case it is "anonymous", the last one is an EUID that was authenticated):

Contents View Properties Security Undo Ownership Find tsv\_multiple tsv\_single

QSurveyResultsFolder at /users/rich/survey/Ratings/RESULTS

Accelerated HT

Type	Name	Size	Last M
<input type="checkbox"/>	-1497893754_2006_05_10_13_54_58_4543		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_16_59_32_5725		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_05_11_5511		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_05_51_1023		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_06_53_4799		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_08_24_6802		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_09_20_5535		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_12_50_7008		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_19_32_2056		2006-
<input type="checkbox"/>	-1497893754_2006_05_11_17_20_32_223		2006-
<input type="checkbox"/>	-454732910_2006_05_11_17_26_17_6366		2006-

Rename Cut Copy Delete Import/Export Select All

Because survey page presentation in **QSurvey is based on standard HTML**, it supports single and multiple select radio button and checkbox fields, and list drop downs. Additionally, "OTHER" textbox fields can be combined with the radio and checkbox type multiple choice items; of course QSurvey supports textbox only items as well. **The survey designer has control over whether items are presented vertically, horizontally, or in columns. Images can be combined in the body of the survey pages to create interesting arrangements of the anchors for the scale items, or for other purposes.** For example, below we have a "matrix" type set of single select items, with the item anchors angled above the fields:



## Please select the most appropriate response

*Important*  
*Somewhat Important*  
*Not Very Important*  
*Not Important At All*

.....Reliability

.....Quality

.....Price

.....Service

Again, since survey page presentation in **QSurvey is based on standard HTML** (with some backend Zope scripting for the data collection into the ZODB), [JavaScript](#) can be combined with the HTML pages to obtain interesting [DHTML](#) effects on survey pages. In the figure below, the first item is a TEXT box only item (ArticleFeedback); the second item is a HTML page only (Introduction - this page can contain images, JavaScript, web-links, etc; basically most valid HTML constructs); the next six items are multiple choice items (both single and multiple select types); the tenth item is an HTML page with a URL link to an external web page - in order to access that web page users have to authenticate in using their EUID username (e.g. LDAP Database); the eleventh item is a list drop down with a pre-selected entry of "Texas". The final item is a HTML page with image that has anchors for items Q2, Q3, Q4, and Q5.

Contents	View	Transition	Sub-item_Sequence	Button_Text	freq
----------	------	------------	-------------------	-------------	------

QPage at </users/rich/survey/Ratings/QPage1>

DTML Document Add

Type Name	Size	Last Modified
<input type="checkbox"/> ArticleFeedback (Please add any comments that you wish to share concerning ...)		2006-05-10 14:56
<input type="checkbox"/> Introduction (Intro)	1 Kb	2006-05-10 13:39
<input type="checkbox"/> Q1 (Mark the what you believe is the most important ...)		2006-05-10 13:43
<input type="checkbox"/> Q2 ( )		2006-05-10 13:42
<input type="checkbox"/> Q3 ( )		2006-05-10 13:17
<input type="checkbox"/> Q4 ( )		2006-05-10 13:43
<input type="checkbox"/> Q5 ( )		2006-05-10 13:16
<input type="checkbox"/> Q6 (Please check all that apply)		2006-05-10 13:53
<input type="checkbox"/> Q7 (Please explain how or why you rated the above ...)		2006-05-10 13:15
<input type="checkbox"/> Ratings (Ratings)	1 Kb	2006-05-10 14:58
<input type="checkbox"/> State (Which US State do you reside in?)		2006-05-10 14:54
<input type="checkbox"/> image (image)	1 Kb	2006-05-10 13:28

Advanced functionality in QSurvey allows, multiple page surveys with branching from single questions to different pages depending on the users selection:

The Question	View	Choices	Custom Values	Options	Branching	Security
--------------	------	---------	---------------	---------	-----------	----------

**MultipleChoiceQuestion** at </users/rich/survey/Ratings/QPage1/Q1>

To branch:

1. Check branch\_on\_this\_question, below
2. Choose the next location the survey should go after each user response.

Note: Only one question on a page should do branching

See the branching for the [page](#) this question is on.

**branch\_on\_this\_question**

user response	next page selection
Reliability	<input type="text" value="QPage1"/>
Service	<input type="text" value="Finish"/>
Quality	<input type="text" value="QPage1"/>
Price	<input type="text" value="QPage1"/>

[Authentication](#) can be controlled through assigned Zope permissions (e.g. student, faculty, anonymous, manager, etc) which can be created by the survey researcher as needed:



## QSurvey at /users/rich/survey/Ratings

The listing below shows the current security settings for this item. Permissions are rows and roles are columns. Checkboxes are used to indicate where roles are assigned permissions. You can also assign **local roles** to users, which give users extra roles in the context of this object and its subobjects.

When a role is assigned to a permission, users with the given role will be able to perform tasks associated with the permission on this item. When the *Acquire permission settings* checkbox is selected then the containing object's permission settings are used. Note: the acquired permission settings may be augmented by selecting Roles for a permission in addition to selecting to acquire permissions.

Permission		Roles			
		Anonymous	Authenticated	Manager	Owner
<input type="checkbox"/>	<b>Acquire permission settings?</b>				
<input type="checkbox"/>	Access Contents Information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Access RSession Methods	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Access Transient Objects	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Access arbitrary user session data	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Access contents information	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Access future portal content	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

For example, a survey researcher can leave the survey as accessed anonymously, or can create their own username and passwords for assignment. For on-campus surveys, the Zope product [LDAPUserFolder](#), can be used with the campus [LDAP](#) database to authenticate users on their [EUID username and password](#):

Configure	LDAP Schema	Caches	Users	Groups	Log	Undo	Ownership	Security	
LDAPUserFolder at <a href="#">/users/rich/survey/Ratings/acl_users</a>								Help!	
Change the basic properties of your LDAPUserFolder on this form.									
<i>Title</i>	<input type="text" value="ldap test"/>								
<b>Login Name Attribute</b>	<input type="text" value="uid (uid)"/>								
<b>User ID Attribute</b>	<input type="text" value="uid (uid)"/>								
<b>RDN Attribute</b>	<input type="text" value="uid (uid)"/>								
<b>Users Base DN</b>	<input type="text" value="ou=people,o=unt"/>	<b>Scope</b>	<input type="text" value="SUBTREE"/>						
<b>Group storage</b>	<input type="text" value="Groups not stored on LDAP server"/>								
<b>Groups Base DN</b>	<input type="text" value="ou=people,o=unt"/>	<b>Scope</b>	<input type="text" value="SUBTREE"/>						
<b>Manager DN</b>	<input type="text" value="cn=courseeval,ou=directory users,o=unt"/>						<b>Password</b>	<input type="text" value="xxxxxxxxxx"/>	
<b>Manager DN Usage</b>	<input type="text" value="Never"/>						<b>Read-only</b>	<input checked="" type="checkbox"/>	
<b>User object classes</b>	<input type="text" value="top, person"/>								
<b>User password encryption</b>	<input type="text" value="SHA"/>								
<b>Default User Roles</b>	<input type="text" value="Anonymous"/>								
<input type="button" value="Apply Changes"/>									

Since QSurvey is a free/open-source product, **the complete program code listings (python and Zope DTML) are available for modification and improvement.** The following appears in the QSurveyClass folder of QSurvey Product folder:

Methods Basic Views Property Sheets Permissions Define Permissions Subo

Z Class at  
[/Control\\_Panel/Products/QSurvey/QSurveyClass/propertiesheets/methods](#)

Accelerated HTTP Cache Manager

Type	Name	Size	Last Modified
<input type="checkbox"/>	allquestions	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	aqnew	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	branchme	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	countableitems	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	dateid	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	default_css	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	default_url	1 Kb	2006-05-09 07:27
<input type="checkbox"/>	filethedata	1 Kb	2006-02-27 05:52
<input type="checkbox"/>	filethedatapersistent	3 Kb	2006-05-09 16:55
<input type="checkbox"/>	index_html	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	pagelist	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	passwordform	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	set_next_page	1 Kb	2006-02-20 19:58
<input type="checkbox"/>	standard_html_footer	1 Kb	2006-02-20 19:58

Edit Bindings Test Proxy History Undo Security

Script (Python) at  
[/Control\\_Panel/Products/QSurvey/QSurveyClass/propertiesheets/methods/filethedataper:](#)

Title

Parameter List

Bound Names context, container, script, traverse\_subpath

Last Modified 2006-05-09 16:55

```

import string

tab = '\t'
newline = '\n'
carriage = '\r'
space = ' '

data = container.session_data_manager.getSessionData()
if data:
    surveyid=data.get('surveyid')
    #see if the id is in RESULTS, if not create it
    if surveyid not in container['RESULTS'].objectIds():
        container['RESULTS'].manage_addProduct['QSurvey'].QSurveyResultsItemClass.
        resultsItem=container['RESULTS'][surveyid]
        ipaddress = ''
        if context.REQUEST.has_key('HTTP_X_FORWARDED_FOR'):
            ipaddress = context.REQUEST['HTTP_X_FORWARDED_FOR']
        if not ipaddress:
    
```

While the original product hasn't been modified by the original author since 2003 (and as it turns out, that change was prompted by me; prior to that, changes had not been made since 2001); **we here at RSS have been fixing minor design limitations and adding features as we can.** The changes have been



slow and incremental (did I mention that we are mostly understaffed?), with a list of changes being implemented in their order of importance. The first changes that we added were to **allow completely anonymous responses, even though "authentication" was being utilized**. While this may seem implausible, it is possible in that authenticated usernames are set up to be ["hashed"](#) so that the original authenticating username could not be retrieved (backtracked). However, this hashed representation is unique in that it would allow the same user to be identified upon re-authentication but without gaining the actual identity of the original username (or user). Additionally, the original product did not allow the final survey page to be customized so that upon completion (e.g. submit button being clicked), the user could be re-directed to a customizable URL (e.g. all survey researchers using QSurvey would have to use the same ending page). To change this behavior, a "qs\_url" DTML method object was added as an object in the original product folder (i.e. the root folder inaccessible by Zope account holders) so that upon the first submission of a survey researcher's survey, a corresponding "qs\_url" object is created in the survey researcher's survey folder. **This "qs\_url" file can be edited and changed to re-direct a user to a URL of choice after the "submit" button is clicked (note: there are some important considerations with regard to this file and its contents, that if not heeded, could cause potential data loss problems)**. Currently, we are experimenting with adding an "automatic form creation and submission" section of code to this qs\_url file so that collected results can be submitted to a [RDBMS](#) if desired. Other possibilities that we are experimenting with involve using **AJAX design templating procedures** to write data to persistent objects (using an [XMLHttpRequest](#) API) as data are collected, so that if the server/browser "connection" is broken that partial results can be saved to the file system. A few projects based on Zope (e.g. [Zope-React](#)) are underway to bring the ["Model-view-controller"](#) design pattern that characterizes a number of currently popular web application frameworks (e.g. [Ruby-on-Rails](#)). Zope-React provides Zope with the same Ajax actions implemented by the library used in Rails. **Another high priority on the agenda of changes we are wanting to implement, is creating the "real-time" reporting of survey results** as they are collected in the survey folder. We already have this ability to some degree, but we are still looking to improve this. Our hope is to utilize DHTML techniques to create dynamic reporting that uses the [R statistical programming environment](#) as the backend computational engine. One such possibility is to use the ["Rpad" library](#). Another is to use the python interface to R - [Rpy](#) - to call R for data processing within a Zope python script method.

## Some "Gotchas" to Avoid When Using QSurvey

Folks new to QSurvey will need to [consult the AVI tutorial series](#) on the exact steps in setting up an initial survey. So this next section is going to be more meaningful for those who have already created a QSurvey before, or have just viewed these tutorials. There are some potential problems that could occur if not addressed: **1) Blank HTML text boxes that consist of only 1 line can submit upon a carriage return, resulting in empty fields in items below that text field with the carriage return. It would be safer to not use "small text" boxes with 1 line.** It would be better to use "large" text boxes that consist of one lines. One of the first things that I changed on the default behavior of QSurvey was that large textboxes could have no smaller than 2 lines. **HOWEVER, NOTE THAT YOU CAN STILL CREATE SMALL TEXT BOXES WITH 1 LINE! THIS COULD RESULT IN MISSING DATA!** For example:

Large Text box 1 line:

Please explain how or why you rated the above characteristics.

*feel free to elaborate at length*

Small Text box 1 line:

Please explain how or why you rated the above characteristics.

*feel free to elaborate at length*

The second box will result in lost data in subsequent items on the page if the end of the line is followed here by a carriage return. Next Gotcha: **2) Make sure to edit the qs\_url file so that it has nothing in it other than an redirect to a url. The default is:**



This segment of tags causes a pause of less than a second after the "submit" button is clicked, then redirects the user to <http://www.unt.edu>. This url (<http://www.unt.edu>) can be changed to a URL that you select. **Under NO circumstances should you have "clickable URL links" on this page (e.g. href), or other submit buttons (e.g. other forms).** If the user closes the window after the "submit button" is clicked BEFORE the redirection to the URL, then data loss could occur. You might change the 0 to some other number like 2, 3, 4 (second delay before redirect), but just realize the user might close the window during this pause. **3) Never set the transition on the last page (the Final page should NOT have the transition set. In other words, do not set the "transition to the next page", on the page that has the "click to submit survey button").** Setting the transition on the last page WILL NOT ALLOW THE DATA TO BE SUBMITTED TO THE ZODB. Finally, **4) Never use "spaces" or "special characters" in the ID field's for the names of your pages or questions. TITLE fields are just strings or descriptors and can be anything.** I usually stick to "camel-hump" notation in my naming conventions: DemographicsPage1; Question1Part2; etc. Don't use spaces or underlines.

### Some Helpful Hints

Here are some helpful hints rather than Gotchas: **You can use the Zope management interface: "Rename, Cut, Copy, Delete and Paste" functions with both question pages and questions.** If your questions or pages have very similar content but only differ by the actual wording of the question, then use the "Copy" button, and then "Paste" the original template question or page however many times you need to. Then "Rename" the items as needed and edit the question content - this is a GREAT time saving trick. For example, Q1 is checked, Click the "Copy" button below and a "Paste" option will appear. Clicking the "Paste" button will give a version with the following name:

Copy that appears:



 copy\_of\_Q1 (Mark the what you believe is the most important ...)

Where the original was:

Type	Name	Size
<input type="checkbox"/> 	ArticleFeedback (Please add any comments that you wish to share concerning ...)	
<input type="checkbox"/> 	Introduction (Intro) 	1 Kb
<input checked="" type="checkbox"/> 	Q1 (Mark the what you believe is the most important ...)	
<input type="checkbox"/> 	Q2 ( )	
<input type="checkbox"/> 	Q3 ( )	
<input type="checkbox"/> 	Q4 ( )	
<input type="checkbox"/> 	Q5 ( )	
<input type="checkbox"/> 	Q6 (Please check all that apply)	
<input type="checkbox"/> 	Q7 (Please explain how or why you rated the above ...)	
<input type="checkbox"/> 	Ratings (Ratings) 	1 Kb
<input type="checkbox"/> 	State (Which US State do you reside in?)	
<input type="checkbox"/> 	image (image) 	1 Kb

Rename   Cut   Copy   Delete   Import/Export   Select All

For that matter, **you can copy an ENTIRE survey and paste it again to create a copy of the original survey.** This is useful for creating slightly altered versions of the entire survey; or for backing up a survey that you don't want to risk losing or corrupting. **Additionally, by using [WebDAV folders](#) or [Netdrive](#) ([see last month's column](#)), you can upload, copy, and replicate items using WebDAV clients.** The image that appears at the top of every QPage:



can be customized or removed entirely. The reference to this image, unt\_logo (which is in the main QSurvey product folder), can be removed from the qs\_css file:

```

BODY {
    color : black;

    background-repeat: no-repeat;
    background-color : rgb(17,114,67);
    background-image: url(unt_logo);
    background-color : white;
    margin-top : 5em;
    margin-right : 11em;
    margin-left : 3em;
}
    
```

Just remove the entire line: "background-image: url(unt\_log);". Or you can upload a different image into the QSurvey object folder and change the reference to something like: "background-image: url(my\_image);".

### The Final Product (the survey)

The use of the QSurvey product and the [Zope Survey Server](#) in general has increased on campus.



QSurvey is used in student dissertations, faculty projects, administrative projects (e.g. Center For Distributed Learning (CDL) web- based evaluations). Last spring (2005), I taught a survey research design course for the SLIS department, and Zope/QSurvey was utilized quite a bit. Recently, RSS was involved in working with the UNT Faculty Senate in implementing a university-wide faculty evaluation of administrators at UNT (the results have been reported in [UNT's InHouse publication](#)). To give readers a feeling for how the overall product feels, I have assembled a mock survey, based on a marketing theme. The live, assembled version of the survey can be found at this address: <https://web2survey.unt.edu/users/rich/survey/Ratings/> Note that this address is on an [SSL](#) enabled website so that all HTTP calls are directed to the HTTPS port. In the end, QSurvey is only a small part of a number of useful free/open-source tools available to researchers on <http://web2survey.unt.edu>. Researchers at UNT not only have access to an enterprise level web application server that is backed by a built in object database, but have access to RDBMS (e.g. MySQL and PostgreSQL). In addition to having access to <https://web2survey.unt.edu> (contact Rich Herrington at 940-565-2140), researchers have access to <http://zope.unt.edu/>. This Zope server is supported by Web Support Services (contact: Shannon Peevey at 940-369-8876)

**One last thing to note:** [RSS](#) currently teaches a set of two [short-courses](#) - two 3 hour classes - on creating surveys using Zope and QSurvey.

## References (Warning: these references are not thorough, and to some degree represent my personal biases!)

### Books & Articles

[Please Teach Me Web Frameworks For Python!](#) - Guido van Rossum (BDFL)

[The Zope Book](#) - Amos Latteier, Michel Pelletier, Chris McDonough, Peter Sabaini

[The Definitive Guide To Plone](#) - Andy McKay

[Content Management Systems](#) - Paul Browning, University of Bristol  
Mike Lowndes, Natural History Museum, London

[Open Source Content Management Systems](#) - James Robertson

[The Object Oriented Thought Process](#) - Matt Weisfeld (article)

[The Objected Oriented Thought Process, 2nd Edition](#) - Matt Weisfeld (book)

[Web Design Frameworks: An Approach to Improve Reuse in Web Applications](#) - Schwabe, Rossi, Esmeraldo, Lyardet

### Related Websites

[OSCOM](#) - The International Association for Open Source Content Management

[CMS Watch](#) - Provides Independent Evaluations of Content Management Systems

[Python Org](#) - Official Python Website

[Django](#) - Python Based Web Framework

[Django Intro. Video](#) - Hosted by Google's Tech Talk

### Zope/Plone Communities

[Zope Org](#)

[Plone Org](#)

[Zope Labs](#)

Please note that information published in *Benchmarks Online* is likely to degrade over time, especially links to various Websites. To make sure you have the most current information on a specific topic, it may be best to search the UNT Website - <http://www.unt.edu> . You can also search *Benchmarks Online* - <http://www.unt.edu/benchmarks/archives/back.htm> as well as consult the UNT Helpdesk - <http://www.unt.edu/helpdesk/> Questions and comments should be directed to [benchmarks@unt.edu](mailto:benchmarks@unt.edu)

[Return to top](#)