UltimateSoftware
DESIGNS

ULTIMATE SURVEY DEVELOPER GUIDE

TABLE OF CONTENTS

Introduction

This guide provides information about how certain parts of the application are built in order to help you with common customizations.

Before making any modifications, it is a good idea to check http://www.ultimateapps.com to see if you have the latest version of the application. Also, as with any project of this type, make a backup of the working copy of the application before making any modifications.

Application Design Notes

All constants except for the connection string and database type have been written in capital letters, with a prefix of SV (Survey) or USD(Ultimate Software Designs), and are included in the file located at '/Include/Constants_inc.asp'. For example, 'SV_MESSAGE_USER_ADDED is set to 1 to tell a page that a user has been added. Most of these constants are used to limit the use of text comparisons, which are inherently much slower than comparing integers. Some of the constants are used to set things strings or other values used throughout the application in one place, such as 'SV_VERSION_NUMBER', which represents the application's version number.

All functions that are in include files are prefixed with the same prefix as the include file that the function is included in. For example, if you see the function "surveyCreation_createSurvey" called by a page, you will know that the function is located in "Include/surveyCreation_inc.asp". If a function does not have a prefix, you will know that the function is in the same page as the code that calls the function. In general, functions in include files are used to accomplish most tasks related to inserting, deleting, updating, and retrieving information from the database. Please note though, not all commands related to the database are in functions, and in turn, not all functions are related to the database.

Utility Functions

Utility functions are used to perform various general tasks that are not directly related to the business logic of the application. They are included in utility_inc.asp. The use of each function is detailed below:

utility_setCookieExpiration(string1,integer1, string2)

Takes in a cookie name as string1, an amount of time to extend the cookie as integer1, and a time period to extend the cookie by as string2. This time period can be referenced using the following constants: USD_MINUTES, USD_HOURS, USD_DAYS, USD_MONTHS, USD_YEARS

utility_SQLEncode(string1, boolean1)

Takes in a value to be added to a SQL command or query as string1 and a whether or not to change the string to NULL if it is blank as boolean1. This is used to put a SQL value in the correct syntax for the appropriate database type.

Example: strSQL = "SELECT surveyID " &_ "FROM usd_Survey " &_ "WHERE surveyTitle = " & utility_SQLEncode(strSurveyTitle, False)

utility_convertSQL(string1)

Takes in a SQL command with 'GETDATE' in it as string1, and converts the GETDATE commands to NOW if the database being used is MSAccess, and returns the converted string.

utility_executeCommand(string1)

Takes in a SQL command as string1 and executes it against the database

utility_createGUID()

Returns a globally unique identifier (GUID)

Example: strGUID = utility_createGUID()

utility_sendMail(string1, string2, string3, string4)

Takes in a "From" address as string1, an address of the intended email recipient as string2, a string of the message subject as string3, and a string of the message body as string4 and sends an email accordingly.

utility_checkForRecords(string1)

Takes in a SQL query as string1 and returns a Boolean of whether or not any records are retrieved from the database using the statement. This is useful for such things as checking to see if any surveys exist in the database. A good example of how this is used is in Include/survey_inc.asp:

```
Function survey_surveyExists(intSurveyID)
    Dim strSQL
    If not utility_isPositiveInteger(intSurveyID) Then
        survey_surveyExists = False
    Else
        strSQL = "SELECT surveyID " &_
                 "FROM usd_Survey " &_
                 "WHERE surveyID = " & intSurveyID
        survey_surveyExists = utility_checkForRecords(strSQL)
    End If
End Function
```

utility_isValidEmail(string1)

Takes in string1 and returns a boolean of whether or not the string is in the form of a valid email address

utility_getMonthText(integer1)

Takes in integer1 and returns the corresponding month name. For example, 1 = January and 12 = December

utility_checkDefault(integer1, integer2)

Takes in two integer values, integer1 and integer2 and if they match, outputs 'selected' to the screen

utility_SQLDateEncode(date1)

Takes in date1 and formats it properly to be used in a SQL string for a date field in the database

utility_isPositiveInteger(variant)

Takes in any value and returns a Boolean of whether or not it is an integer and greater than zero. Useful if you do not know if a value will exist or not, such as a page number. Here is an example from "manageSurveys.asp":

```
If utility_isPositiveInteger(intPageNumber) Then
    intPageNumber = cint(intPageNumber)
Else
    intPageNumber = 1
End If
```

utility_isMoney(variant)

Takes in any value and returns a Boolean of whether or not it is in money format (a decimal place and with at least one digit before it and two digits after it.)

utility_getPercentage(integer1, integer2)

Takes in 2 integers. Divides integer1 by integer2 and returns the percent value with two decimal places

utility_GetRandomizedSequencerArray(integer1)

Takes in integer1 representing the size of an array and retrieves a randomly sorted array of numbers between 0 and the integer. See the function “itemDisplay_displayRadio” for a good example.

utility_GetRecordset(string1)

Takes in a SQL Select statement as string1. Queries the database with string1 and returns a disconnected recordset of the date.

Syntax:

```
Dim rsResults
strSQL = “SELECT userID FROM usd_user”
Set rsResults = utility_getRecordset(strSQL)
```

Note: For performance sake, please do the following after using the recordset:

```
rsResults.Close
Set rsResults = NOTHING
```

utility_formEncode(string1)

Takes in string1 and returns that string with quotes escaped out for use in form values

utility_javascriptEncode(string1)

Takes in string1 and returns that string formatted for use in javascript

utility_getNetworkUsername()

Returns the NT username if there is an NT authenticated user

utility_XMLEncode(string1)

Takes in string1 and returns the string in valid format to be used in an XML element or attribute

utility_displayError(string1, string2)

Takes in an error message as string1 and an error description as string2 and outputs the error to the page

Common User Interface Functions

Common user interface functions are used to display various interface items in the same format throughout the application without having to reuse similar or identical code. These functions are found in the file located at “Include/common_ui_inc.asp”. All functions are prefixed by “common_” and are described below:

common_helpLink(string1, string2)

Takes in a the path in the help directory that points to the requested document as string1, and the path of the location of the image that you want to display for the user to click to get help as string2. There are two constants that can be used for string2: SV_LARGE_HELP_IMAGE and SV_SMALL_HELP_IMAGE. The example below links to a file located at “/help/surveys/items/numberAnswerInputs.asp”

```
<td class="normalBold">
    <%=common_helpLink("surveys/items/numberAnswerInputs.asp",SV_SMALL_HELP_IMAGE)%>
</td>
```

common_helpLinkText(string1, string2)

Takes in the path in the help directory that points to the requested document as string1, and the text that you want to output for the help link as string2

common_tableRow(integer1)

Takes in integer1 representing the number of a row in a table and outputs a row with a different color based on whether integer1 is an even or an odd number

common_requiredFlag()

Outputs a red asterisk to the screen to indicate that a field is required

common_backLink()

Outputs an image that says “Back” that when clicked goes back 1 page in the browser history

common_dateSelect(string1, date1, integer1, integer2)

Takes in the name to call the form item as string1, a default date as date1, a start year in integer format as integer1, and how many years that the user can select as integer2, and creates a set of dropdown menus to allow the user to select a date

common_itemTypeDescription(integer1)

Takes in the unique ID of a survey item type as integer1 and outputs its text and description to the screen

common_allTypeDescriptions()

Outputs the text and description of all item types

common_basicTableTag()

Outputs a table tag (“<table>”) with properties that are consistent throughout the application for tables containing lists of surveys, users, etc

common_basicTableHeaderRow()

Outputs a table row tag (“<tr>”) with properties that are consistent throughout the application for the top row of tables containing lists of surveys, users, etc

User Authentication

It is very common for developers to have the need to customize the way that the user authentication works. This section of the developer guide will explain basically how user authentication works, so that it can be understood and customized.

Registration

Registration can be done in a few ways, including allowing public open registration, bulk registration by administrators, and administrators registering users one by one. No matter how registration is done, there are certain key aspects of a user that need to be understood. The following describes a few things pertinent to understanding how users are handled in the system

Users based on Username/Password: In the usd_surveyUser table, each must have a unique username. They must have a pword (password) and the loginType field in the must be set to 1

Users based on Network Authentication: In the usd_surveyUser table, each must have a username, and can optionally have a networkDomain. The username and networkDomain combination must be unique, but the same username can be used if domains are different. The loginType must be set to 2. If a user does not have a networkDomain, a user by that username will be logged in, regardless of their domain.

Login

Logging into to Ultimate Survey is a fairly simple process. This occurs on the page titled 'login.asp'. The login process for a username/password based login is as follows:

1. Checks to see if the username and password combination is found in the database
2. If the username/password combination is not found, an error message is displayed
3. If the username/password combination is found, the cookie/session values are set accordingly

The application will automatically login a user based on network credentials through the following process:

1. Checks to see if a username/password user has logged in, and if so, no longer tries to login the network user
2. If there is no username/password user logged in:
    a. The NT authenticated username is retrieved if there is one
    b. The username and domain name are parsed out
    c. The database is queried to see if this domain name and username combination is found with a logintype of network user
    d. If this user is found, the session or cookie values are set accordingly

Cookie/Session Values:

Depending on the settings that the application's administrator sets, user authentication is done through either cookies or the session. All cookie value are prefixed with "(user)". For example, to set the userID value, we call "Response.cookies("user")("userID") = intUserID

1. userID – the unique integer ID of the user based on userID in the usd_surveyUser database
2. userType – the integer usertype corresponding the usertype (Administrator, Take/Create, Take Only)
3. username – the username of the logged in user
4. logintype – the integer type of the login type.
    a. 1 = Username/Password user, can be referenced by the constant SV_LOGIN_TYPE_PASSWORD

   b. 2 = Network authenticated user, can be referenced by the constant SV_LOGIN_TYPE_NETWORK
5. overridenetworkuser – if set to true, specifies that a username/password type user is logged in and network credentials should be ignored

Throughout the application, permissions are based on the userID and usertype. Here is an example code snippet from "manageIndividualSurvey.asp" that will serve to show how this is typically done:

```
Call user_loginNetworkUser()

'Get the userid and usertype out of the session or cookie
Call user_getSessionInfo(intUserID, intUserType, "","", "")

intSurveyID = cint(Request.QueryString("surveyID"))

If ((survey_getOwnerID(intSurveyID) <> intUserID) _
        and intUserType = SV_USER_TYPE_CREATOR) _
        or intUserType = SV_USER_TYPE_TAKE_ONLY or intUserType = 0 Then
    Response.Redirect("index.asp?message=" & SV_MESSAGE_NO_PERMISSION)
End If
```

Here is a step by step of what happens in this code:

1) The application logs in a network user if no username/password user is logged in. This is described earlier in this section of the Developer Guide
2) The user's session/cookie information is retrieved as necessary. In this case, only the userID and usertype are needed.
3) The survey ID is retrieved from the page URL
4) The ownerID of the survey is compared against the userID
   a. If the ownerID does not match the current userID and the usertype is a Create/Take user, if the usertype is Take Only, or if there is no usertype, the application redirects to a page telling the user that he/she does not have permission to view that page.
   b. If the user has permission, nothing happens and the page continues to load

Navigation Information

Navigation of the main tabs in Ultimate Survey is all maintained in the file located at "Include/header_inc.asp". This file was written to allow for maximum flexibility of the look and feel of the survey application.

First in header_inc.asp is a function called "header_htmlTop'. This function is called in almost every page of the application. This accomplishes a few things. First, it creates the html head tag. This includes the title that can be set through the Settings tab of the web interface. It also adds a few javascript functions that are used throughout the application. Next the body tag is created. In a call to header_htmlTop, you may specify a background color for the page (this is used in survey templates particularly) and you may also specify a string to append to the body tag that can be used to do things such as "onload" events in javascript. Finally, this function calls the default cascading stylesheet, which is located at '"Include/Stylesheets/ultimateAppsStyles.css'.

Next is a function called header_writeHeader, which is also called in almost every page. It simply takes in a usertype and pagetype, and calls other functions in the header based on the usertype. The pagetype is passed into the other functions. Although each function does many similar things, they are broken into separate functions so that you can easily make very customized HTML for each usertype. Each function does the following:

1. Title bar is created with the site name in the title color, with a background color from "Top Color" in the settings
2. A menu is created with the "Menu Color" from the settings. Menu items differ in each function. Menu subsets are displayed based on the pagetype
3. A table is opened that creates padding for the rest of the page so that there is room on the left and right of the HTML presented

At the bottom of each page, the footer is included. This footer closes the table that pads the page, and adds a grey bar, that out of the box has the version number of Ultimate Survey and Copyright Information. The file that creates this footer is "Include/footer_inc.asp".