

MainTree Systems uScript Shared Libraries Reference

Updated Tuesday, June 10, 2008

James Songer
jsonger@maintree.com

This work is licensed under the Creative Commons Attribution-ShareAlike License.
To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Table of Contents

mt-libBUI.....	1
AddSpellCheck	1
AddUndo.....	1
AltCSS	1
ChangeTitle.....	1
DisableCSS	1
DisableInput.....	1
MakeIEPrint.....	2
MakeIEPrintD.....	2
MaxDisplay.....	2
STFloat.....	2
TextDisp.....	2
TextDispNB	3
TextReplace	3
mt-libDialog.....	4
ButtonBox.....	4
FileBrowser.....	4
InputBox	4
MsgBox.....	5
mt-libGeneral	6
CheckPEConf.....	6
FindLastSpace.....	6
GenUID.....	6
GetMD5	6
RandomNumber.....	6
SplitText.....	7
mt-libHTMLArea.....	8
HTMLAreaInit.....	8
HTMLArea	8
mt-libIntelliTags	10
ITFile.....	11
IntelliTags	11
ProcessTag	11
mt-libPrint.....	13
agsd	13
DisplayFile.....	13
PrintFile.....	14
Appendix A: Library Configuration	15
mt-libBUI.....	15
mt-libDialog.....	15
mt-libHTMLArea.....	15
mt-libPrint.....	15

mt-libBUI

This library contains utility functions that are useable in any BUI based uScript.

AddSpellCheck() is null

This function adds a “Check Spelling” button to the current BUI form that invokes the CMHC Word Spell Checker and checks the spelling of all text areas on the current form.

AddUndo()(ObjectID is text, Levels is text) is null

This function adds a undo button that is attached to the form element identified in the ObjectID parameter. The Levels parameter specifies how many undo levels to maintain.

AltCSS() is null

Switches the rightFrame.css to an alternate version called rightFrameNew.css.

ChangeTitle()(NewTitle is text) is null

Changes the effective page title in the current form to the value provided in NewText. Useful for forms that are going to be printed.

DisableCSS() is null

Disables all CMHC provided CSS files on the current form.

DisableInput() is null

Disables the use of all input controls on the current form. Useful for display only forms.

PLEASE NOTE: Without the *Disable on Submit* modification, this function will blank all variables referenced on the form when it is submitted back.

MakeEPrint(LeftHeader is text, LandScape is binary, KillDate is binary) is null

Inserts a print button on the current form that allows the user to print using the Enhanced Printing Control. The LeftHeader parameter allows you to change what appears in the upper left hand corner of all pages. Using LeftHeader will move the page title to the right hand side of the page.

The Landscape parameter allows you to specify that the current form should be printed in landscape format. This is a binary variable that works as a true/false.

The KillDate parameter allows you to specify that today's date should not be printed on the current form. This is a binary variable that works as a true/false.

MakeEPrintD(LeftHeader is text, LandScape is binary, KillDate is binary) is null

The same as MakeEPrint except it is designed for forms that are disabled with DisableInput.

MaxDisplay() is null

Maximizes the work area of the current form. This method emulates the clicking of the maximize button in the lower right hand corner of the window. Using this method will result in the restoration of the work area when the form exits.

STFloat(op-code is text, script[] is text, label[] is text, nexttab[] is text, exec-once[] is text, depends-on[] is text, status[] is text, init-tab is text) is binary

This is a wrapper for the \$sessiontabs function that wraps the tabs in a floating table which follows the user as they scroll down the page. All of the parameters are the same as the ones for \$sessiontabs except for the lack of the left-side-html and right-side-html parameters. (For obvious reasons.)

For more information on the parameters of this function, please reference the \$sessiontabs function in the uScript documentation, chapter 7.

TextDisp(VarIn is text) is null

Displays the contents of VarIn inside of a table box. Useful for replacing a \$textarea on a form for printing purposes.

TextDispNB(VarIn is text) is null

Same as TextDisp except that it replaces all carriage returns with HTML
 tags.

TextReplace(StringIn is text, PageTitle is text) is number

Searches the StringIn for all instances of \$\$ and prompts the user for an replacement value. PageTitle is displayed at the top of the form during the replacement.

The function returns a 0 if Ok is clicked or 1 if Cancel is clicked.

mt-libDialog

This library contains a series of functions for common UI prompts. Most of these functions are based off of their Visual Basic counterparts.

ButtonBox(ButtonList[] is text, DialogTitle is text) is binary

Displays a dialog with a list of buttons for the user to select.

ButtonList[] is a array of button labels.

DialogTitle is the text for the top of the dialog.

The function returns the number of the selected item or a 0 if cancel is clicked.

FileBrowser(FileNameIO is text, BasePath is text, DialogTitle is text, DialogType is binary, RelativePath is binary) is binary

Displays a file browser that allows the user to select a file from the unix file system. The file name is returned through the FileNameIO variable.

BasePath is used to restrict the user to the directory structure located below the directory specified in it.

DialogTitle is used to specify the title for the dialog.

DialogType is used to specify wither it is an open or save dialog. 1 = Open, 0 = Save

RelativePath is used to specify if the file returned is absolute or relative to the BasePath. 1 = Relative, 0 = Absolute

The function returns a 1 if Ok is clicked or 2 if Cancel is clicked.

InputDialog(DialogMessage is text, VariableIO is text, DialogTitle is text, VarLen is binary, VarEdit is text, ShowEditMsg is text, Button3 is text) is binary

Displays a box to allow for quick variable input. Similar to Visual BASIC's InputBox function.

DialogMessage contains the message to display to the user.

VariableIO is the string that contains the information provided.

DialogTitle is the title of the box; if omitted, the name of the calling script will be used.

VarLen is the length of the variable sought.

VarEdit is the MUI edit string for the variable.

ShowEditMsg is a Y/N that specifies if the edit message is shown.

Button3 is the display value for the 3rd button. If omitted, the button will not be shown.

The function returns a value from 1 to 3 based on if Ok, Cancel, or the 3rd button is clicked.

MsgBox(DialogMessage is text, Icon is binary, DialogTitle is text, Button1 is text, Button2 is text, Button3 is text) is binary

Displays a box with a message to the user with the option for up to 3 buttons. Similar to Visual BASIC's MsgBox function.

DialogMessage contains the message displayed to the user.

Icon is a binary value from 0 to 4 specifying which icon to display in the box. 0 = No Icon, 1 = Information, 2 = Question, 3 = Warning, 4 = Critical.

DialogTitle is the title of the box; if omitted, the name of the calling script will be used.

Button1 is the display value for the 1st button. If omitted, Ok will be used.

Button2 is the display value for the 2nd button. If omitted, the button will not be shown.

Button3 is the display value for the 3rd button. If omitted, the button will not be shown.

The function returns a value from 1 to 3 based on if the 1st, 2nd, or the 3rd button is clicked.

mt-libGeneral

This is a general function library that can be used by either character mode or BUI mode uScripts.

CheckPEConf (EventDate is date, EventTime is time, EventDur is time, EventStaff is text, EventKey is text, ConfPEs[] is text) is binary

Checks a provided Date, Time, Duration, and Staff for any conflicting planned events and returns a list of conflicts in an array.

EventDate, EventTime, EventDur, and EventStaff are the provided values to check in the planned event file.

If provided, EventKey is the key for the planned event that relates to the elements provided above so the function doesn't report that it is conflicting with itself.

ConfPEs[] is the array of conflicting planned event keys.

This function returns the count of conflicting planned events.

FindLastSpace(strLN is text) is binary

Returns the position of the last space next to text in the string provided in strLN

GenUID() is text

Generates and returns a Unique Identifier based on current system time in the following format:

2005-0011-151917 -> Year-Julian_Day-HourMinuteSecond

GetMD5(Textln is text) is text

Takes the text provided in Textln and generates and returns a MD5SUM checksum. This function requires that md5sum be installed on your system in /opt/freeware/bin.

RandomNumber(maxnum is integer) is integer

Returns a random number between 0 and the value of maxnum.

SplitText(TextIn is text, BreakPoint is number, ArrayIO[] is text) is null

Takes the text in TextIn and splits it into segments no longer than the value of BreakPoint and returns the segments in ArrayIO[]. SplitText takes special care to make sure that no split happens in the middle of a word.

mt-libHTMLArea

This library contains the support function for the \$textarea replacement, HTMLArea. HTMLArea is a replacement for \$textarea that allows for rich HTML editing.

HTMLArealnit() is null

Inserts the JavaScript code that is necessary to support the HTMLArea control.

Execute only once per form.

HTMLArea(ObjectID is text, Height is text, Width is text, BodyStyle is text, ToolBar is text) is null

Converts a \$textarea control into a HTMLArea control. The \$countbox, AddUndo, and AddSpellCheck functions are currently incompatible with this function.

ObjectID is the Object ID of the \$textarea control being converted.

Height is a CSS style height parameter for the control. Leave blank to use the \$textarea's size

Width is a CSS style width parameter for the control. Leave blank to use the \$textarea's size

BodyStyle is the default CSS style font parameter for the contents. If left blank, it defaults to Times New Roman

ToolBar is the parameter that controls the items on the editor's toolbar.

Toolbar is in this format “[‘toolbarbutton’, ‘toolbarbutton2, ...]”

The Available options are:

BackColor – Background color for the selected text

Bold

CreateLink – Inserts a hyperlink

FontName

FontSize

ForeColor – Foreground color for the selected text

HTMLMode – Switches to HTML source mode

HorizontalRule – Inserts a Horizontal Line

Indent

InsertImage

InsertTable

Italic
JustifyCenter
JustifyLeft
JustifyRight
OrderedList – Inserts a numeric list
Outdent
PopUpEditor – Opens a editor in a pop up window
Separator
Strikethrough
Subscript
Superscript
UnOrderedList – Inserts a bullet list
Underline

The default is:

```
['fontname', 'fontsize', 'bold', 'italic', 'underline', 'separator', 'strikethrough', 'subscript', 'superscript',  
'separator', 'justifyleft', 'justifycenter', 'justifyright', 'separator', 'OrderedList', 'UnOrderedList', 'Outdent',  
'Indent', 'separator', 'forecolor', 'backcolor', 'separator', 'HorizontalRule', 'htmlmode']
```

mt-libIntelliTags

mt-libIntelliTags is a intelligent templating system that uses special tags to fill in information from relevant database registers.

This is the format of a standard IntelliTags tag:

```
<! DSTName, Option, Case !>
```

Option has different meanings depending on the data type of the DST specified.

Date types can be a number 1-3 resulting in the following:

1 or blank = date output in the MM/DD/YYYY format

2 = date output in the MON DD, YYYY format

3 = Age calculation. Useful for DOBs

Time and Duration have no options

Any other types work as follows:

X = Default edit based off of the DST's information

#number = runs the DST through the data code table specified in the *number*.

RecodeTable = runs the DST through the recode table specified in *RecodeTable*.

(blank) = returns the DST value

Case has the following options:

1 = Upper case the first character

2 = Name case the value

3 = Lower case the entire value

4 = Upper case the entire value

(blank) = Returns the value as is

This is the format of a uScript IntelliTags Tag:

```
<! USC, Script Name, Case, Script Option, Script Option, ... !>
```

Script Name is the name of the script to run. IntelliTags expects a script to use the following prototype:

```
ScriptName(OutText, DBIDs[], RegIDs[], Options[])
```

OutText is the variable that contains the results of the script.

DBIDs[] and RegIDs[] are the same values provided to IntelliTags

Options[] is a array of the options provided to IntelliTags with the Script Option parameters

The case option on the tag is the same as the one above.

ITFile(DBIDs[] is binary, RegIDs[] is text, DialogTitle is text, FileName is text, BasePath is text) is text

Prompts the user for a file then runs it through the IntelliTags processor.

DBIDs[] is a list of database numbers.

RegIDs[] is a list of corresponding register IDs.

DialogTitle is the text for the title line of the file dialog.

FileName is a pre-provided file name for the dialog.

BasePath is the path to lock the user into.

The function returns the processed text.

IntelliTags(DBIDs[] is binary, RegIDs[] is text, InputText is text) is text

Takes the text provided by InputText and replaces any IntelliTags tags located inside with appropriate values.

DBIDs[] is a list of database numbers.

RegIDs[] is a list of corresponding register IDs.

InputText is the text to be scanned and updated.

The function returns the processed text.

ProcessTag(DBIDs[] is binary, RegIDs[] is text, TagIn is text) is text

Processes the IntelliTags tag and returns the result.

DBIDs[] is a list of database numbers.

RegIDs[] is a list of corresponding register IDs.

TagIn is the tag to process.

This function returns the result of processing the tag.

mt-libPrint

This library contains functions that allows for the easy display and printing of AGS specs and text files.

PLEASE NOTE: This library has only been tested on AIX based systems.

agsd(disptitle is text, tfile is text, printrname is text, agsspec is text, parm[] is text, parm[] is text, SupressBRMSG is text, SupressPrint is text, Button3 is text, Button4 is text) is binary

Processes a AGS spec and displays the results in BUI with a option to print through a CMHC Unix printer.

Disptitle is the title to display on top of the display forms.

Tfile is the temporary file for storage of the AGS output. This is required.

Printrname is the pre-provided printer for the report output. The user has the option to override this.

Parm[] is the list of parameter names for the AGS spec.

Parmv[] is the list of parameter values for the AGS spec.

SupressBRMSG suppresses the browser message box that give the users a progress report during the AGS run. This is a Y/N value.

SupressPrint tells if the Print button should be suppressed. This is a Y/N value.

Button3 is an optional 3rd button similar to the function in the mt-libDialog functions.

Button4 is a optional 4th button similar to the function in the mt-libDialog functions.

This function returns a -3 or -4 if the 3rd or 4th buttons are clicked. Otherwise it returns the error code generated for the AGS. For a reference of AGS errors, please look at the \$agsr function in chapter 7 of the uScript documentation.

DisplayFile(disptitle is text, tfile is text, printrname is text, SupressBRMSG is text, SupressPrint is text, Button3 is text, Button4 is text) is binary

Displays a Unix text file in the BUI window.

Disptitle is the title to display on top of the display forms.

Tfile is the temporary file for storage of the AGS output. This is required.

Printername is the pre-provided printer for the report output. The user has the option to override this.

SupressBRMSG suppresses the browser message box that give the users a progress report during the AGS run. This is a Y/N value.

SupressPrint tells if the Print button should be suppressed. This is a Y/N value.

Button3 is an optional 3rd button similar to the function in the mt-libDialog functions.

Button4 is a optional 4th button similar to the function in the mt-libDialog functions.

This function returns a -3 or -4 if the 3rd or 4th buttons are clicked. Otherwise it returns a 0.

PrintFile(filename is text, printername is text, title is text) is number

Prints a textfile on a CMHC Unix printer.

Filename is the file to print.

Printername is the default printer to use. The user can override this.

Title is the title to display on top of the display forms.

Appendix A: Library Configuration

mt-libBUI

Line 14:
STFjsinclude = "/cmhcbui/cmhcbui/MainTree/mt-STFloat.js"

This line contains the location and name of the JavaScript file containing the support functions for STFloat

mt-libDialog

Line 17:
TitleFGColor = "#FFFFFF"
TitleBGColor = "#008000"

IconBase = "/cmhcbui/cmhcbui/MainTree/msgbox/"

The TitleFGColor is the color for the text at the title of the dialogs

The TitleBGColor is the background color for the text at the title of the dialogs.

IconBase is the location of the pictures for the MsgBox icons

mt-libHTMLArea

Line 13:
HTMLAreaLoc = "/cmhcbui/cmhcbui/MainTree/HTMLArea/"

This line contains the location of the support files for HTMLArea

mt-libPrint

Line 30:
controlpath = "/usr/lib/mis/local/pctrl/"
prtoref = "/usr/lib/mis/local/prtadmin/prtoref"

Controlpath is the path to your MISPRINTERS control files

Prtxref is the location and name of your MISPRINTERS prtxref file