

TMDC Utilities Plugin

for
REALbasic



Credits:

Main Programmers: Chris Boot

Documentation: Aram Kudurshian
Chris Boot

Disclaimer:

The Macintosh Developer Center is providing this plugin “as is”. TMDC provides no warranty that this plugin will function in the desired way but it should :-)

Copyright:

This plugin is copyrighted by The Macintosh Developer Center, 1999. If you want to distribute this plugin on a CD, BBS, or any other means, it is illegal to distribute a registered copy of the plugin, it must be an unregistered version, including all documentation and examples. Please contact us at admin.tmdc@worldnet.fr for any questions.

If you find any bugs or problems in this plugin, please do not hesitate to e-mail us at admin.tmdc@worldnet.fr or visit our web site at <http://www.tmdc.org/>.

Contents:

Items listed in **bold** are included in the unregistered version.

I.	Application & Misc	7
i.	TMDCReverseString	7
ii.	TMDCShutdownComputer	7
iii.	TMDCQuitApplication	7
iv.	TMDCHideApplication	7
v.	TMDCNotification	8
vi.	TMDCRestartComputer	8
vii.	TMDCCreateUUID	8
II.	Encryption & Hash	9
i.	TMDCKey	9
ii.	TMDCCreateNewKey	9
iii.	TMDCEncryptString	9
iv.	TMDCDecryptString	9
v.	TMDCMD5Hash	10
vi.	TMDCSHAHash	10
vii.	TMDCGetKeyFromString	10
viii.	TMDCEncryptFile	10
ix.	TMDCDecryptFile	11
III.	Files	12
i.	TMDCOpenInFinder	12
ii.	TMDCRevealInFinder	12
iii.	TMDCFinderFlags	12
a.	Invisible	12
b.	HasCustomIcon	13
c.	Alias	13
iv.	TMDCEncodeBinHexFile	13
v.	TMDCDecodeBinHexFile	14
vi.	TMDCPrintInFinder	14
vii.	TMDCCanUseNavSvcs	14
viii.	TMDCNavChooseObject	14
ix.	TMDCEncodeMacBinaryFile	15
x.	TMDCDecodeMacBinaryFile	15
xi.	TMDCEncodeMacBinaryString	15

xii.	TMDCDecodeMacBinaryString	15
xiii.	TMDCIsMacBinaryFile	16
xiv.	TMDCUUEncode	16
xv.	TMDCUUDecode	16
xvi.	TMDCFinderFlags	17
	a. HasBNDL	17
	b. Stationery	17
	c. Inited	17
	d. NoINITs	18
	e. Shareable	18
	f. NameLocked	18
xvii.	TMDCExtendedFinderFlags	18
	a. FlagsInvalid	18
	b. HasCustomBadge	19
	c. HasRoutingInfo	19
IV.	Folders	20
i.	TMDCExtensionsDisabledFolder	20
ii.	TMDCControlPanelsDisabledFolder	20
iii.	TMDCStartupItemsDisabledFolder	20
iv.	TMDCShutdownItemsDisabledFolder	20
v.	TMDCVoicesFolder	20
vi.	TMDCAppleExtrasFolder	20
vii.	TMDCThemesFolder	20
viii.	TMDCLauncherItemsFolder	20
ix.	TMDCFontsFolder	20
x.	TMDCModemScriptsFolder	20
xi.	TMDCApplicationSupportFolder	20
xii.	TMDCHelpFolder	20
xiii.	TMDCSystemExtensionsDisabledFolder	20
xiv.	TMDCApplicationsFolder	20
xv.	TMDCDocumentsFolder	20
xvi.	TMDCVolumeRootFolder	20
xvii.	TMDCChewableItemsFolder	20
xviii.	TMDCTextEncodingsFolder	20
xix.	TMDCScriptingAdditionsFolder	20
xx.	TMDCSharedLibrariesFolder	20
xxi.	TMDCControlStripModulesFolder	20
xxii.	TMDCAssistantsFolder	20
xxiii.	TMDCUtilitiesFolder	20
xxiv.	TMDCContextualMenuItemsFolder	20

xxv.	TMDCFavoritesFolder	20
xxvi.	TMDCInternetFolder	20
xxvii.	TMDCAppearanceFolder	20
xxviii.	TMDCSoundSetsFolder	20
xxix.	TMDCDesktopPicturesFolder	20
xxx.	TMDCInternetSearchSitesFolder	20
xxxi.	TMDCFindByContentFolder	20
xxxii.	TMDCScriptsFolder	20
xxxiii.	TMDCFolderActionsFolder	20
xxxiv.	TMDCRecentApplicationsFolder	20
xxxv.	TMDCRecentDocumentsFolder	20
xxxvi.	TMDCRecentServersFolder	20
xxxvii.	TMDCSpeakableItemsFolder	20
xxxviii.	TMDCKeychainFolder	20
xxxix.	TMDCQuickTimeExtensionsFolder	21
xl.	TMDCFindByContentPluginsFolder	21
V. Internet Config		22
i.	TMDCICRealName	22
ii.	TMDCICEmailAddress	22
iii.	TMDCICOrganization	22
iv.	TMDCICEmailAccount	22
v.	TMDCICEmailPassword	22
vi.	TMDCICNewsUser	23
vii.	TMDCICNewsPassword	23
viii.	TMDCICArchiePreferred	23
ix.	TMDCICUMichPreferred	23
x.	TMDCICInfoMacPreferred	23
xi.	TMDCICPhHost	23
xii.	TMDCICWhoisHost	24
xiii.	TMDCICFingerHost	24
xiv.	TMDCICFTPHost	24
xv.	TMDCICTelnetHost	24
xvi.	TMDCICSMTPHost	24
xvii.	TMDCICNNTPHost	24
xviii.	TMDCICGopherHost	25
xix.	TMDCICLDAPServer	25
xx.	TMDCICLDAPSearchbase	25
xxi.	TMDCICWWWHomePage	25
xxii.	TMDCICWAISGateway	25
xxiii.	TMDCICListFont	25

xxiv.	TMDCICScreenFont	26
xxv.	TMDCICDocumentFont	26
xxvi.	TMDCICPrinterFont	26
xxvii.	TMDCICSignature	26
xxviii.	TMDCICPlan	26
xxix.	TMDCICQuotingString	26
xxx.	TMDCICMailHeaders	27
xxxi.	TMDCICNewsHeaders	27
xxxii.	TMDCICNewMailFlashIcon	27
xxxiii.	TMDCICNewMailDialog	27
xxxiv.	TMDCICNewMailPlaySound	28
xxxv.	TMDCICNewMailSoundName	28
xxxvi.	TMDCICWebBackgroundColor	28
xxxvii.	TMDCICUseSOCKS	28
xxxviii.	TMDCICSOCKSHost	28
xxxix.	TMDCICUseHTTPProxy	29
xl.	TMDCICHTTPProxyHost	29
xli.	TMDCICUseGopherProxy	29
xlii.	TMDCICGopherProxyHost	29
xliii.	TMDCICUseFTPProxy	29
xliv.	TMDCICFTPProxyHost	30
xlv.	TMDCICFTPProxyUser	30
xlvi.	TMDCICFTPProxyPassword	30
xlvii.	TMDCICFTPProxyAccount	30
xlviii.	TMDCICUsePassiveFTP	30
il.	TMDCICSnailMailAddress	30
l.	TMDCICMacSearchHost	31
li.	TMDCICWWWSearchPage	31
lii.	TMDCICDownloadFolder	31
liii.	TMDCICSetTypeCreator	31
VI.	GUI & Dialogs	32
i.	TMDCListViewWindowHeader	32
ii.	TMDCStdAlert	32
iii.	TMDCIconControl	32
iv.	TMDCStdConfirmYesNo	33
v.	TMDCStdConfirmOkCancel	34
vi.	TMDCStdSave	34
vii.	TMDCCustomAlert	35

VII.	Registration	36
i.	TMDCCheckRegistration	36
ii.	TMDCCreateNewRegistration	36
VIII.	Sound	37
i.	TMDCBeginThemeDragSound	37
ii.	TMDCEndThemeDragSound	37
iii.	TMDCPlayThemeSound	37
iv.	TMDCSetSystemVolume	37
v.	TMDCMADPlayer	38
IX.	Networking	39
i.	TMDCURLDownload	39
ii.	TMDCURLAccessAvaible	39
iii.	TMDCURLUpload	40
A.	Appendix A	41
i.	ThemeDragSounds	41
ii.	ThemeSounds	41

I. Application & Misc

The items within Application & Misc. are REALbasic addons which either have to do with Applications or were not able to be fit within the other categories.

i) *TMDCReverseString(text as string) As String*

Reverse string allows a body of text to be reversed so TMDC would be CDMT. The below example should display 'leM esrever'.

```
Msgbox TMDCReverseString("Reverse Me!")
```

ii) *TMDCShutdownComputer() As Boolean*

This will shutdown the computer safely causing all unsaved documents to be saved. This is the exact same method as choosing 'Special -> Shutdown' from the Finder.

```
Dim b as boolean  
b=TMDCShutdownComputer
```

iii) *TMDCQuitApplication(application as name) As Boolean*

This will quit the named application as if quit was selected from the 'File' menu.

```
Dim b as boolean  
b=TMDCQuitApplication("Finder")
```

iv) *TMDCHideApplication(application as name) As Boolean*

This does the same as TMDCQuitApplication but instead of quitting the application, it hides it. The same method is used in implementing it with the application being the name of the application you would like to hide. TMDCHide can not hide the Finder or frontmost application.

```
Dim b as boolean  
b=TMDCHideApplication("SimpleText")
```

v) *TMDCNotification*

This is a class which brings up a notification alert including a flashing icon and a bullet by the application which caused it. It has the following properties and methods:

AlertString As String

This is the string that appears in the alert.

Install()

This shows the dialog and starts the notification process.

Remove()

This removes the flashing icon and bullet. It is normally called from the Application class' Activate event so as to remove the notification when the program is brought to the front.

See the notification example for a working demonstration as it is too large to describe in this manual.

vi) *TMDCRestartComputer() As Boolean*

This will cause the computer to restart as if 'Restart' was chosen from the 'Special' menu in the Finder.

```
Dim b as boolean  
b=TMDCRestartComputer
```

vii) *TMDCCreateUUID() As String*

This will return a universal unique ID which no two computers will ever have the same, guaranteed. This unique identifier conforms with the format defined by IBM and DEC. This is mainly used for tracking internet transactions.

```
Msgbox TMDCCreateUUID
```


II. Encryption & Hash

The items within this section, encryption and hash, are used to encrypt or one-way hash data for secure transfer or storage.

i) *TMDCKey*

This class is used to store a key generated using the *TMDCCreateNewKey* method. Every key is unique and so encryption/decryption must be done through storing and retrieving the *StringValue* property of this class.

StringValue As String

Returns the generated key as a string.

```
Dim s as string  
s=Key.StringValue
```

ii) *TMDCCreateNewKey(Text as String) As TMDCKey*

This generates a new *TMDCKey* for use with encryption and decryption. The text is used as the key as a password.

```
Dim Key as TMDCKey  
Key=TMDCCreateNewKey("This is a key")
```

iii) *TMDCEncryptString(text as string, key as TMDCKey) As String*

This will encrypt a string of data in the blowfish algorithm which is 144 bit encryption.

```
Dim key as TMDCKey  
key = TMDCCreateNewKey("My Key")  
Msgbox TMDCEncryptString("Encrypt me!", key)
```

iv) *TMDCDecryptString(text as string, key as TMDCKey) As String*

This will decrypt a string of data which was encrypted with *TMDCEncryptString*.

```
Dim key as TMDCKey  
key = TMDCCreateNewKey("My Key")  
Msgbox TMDCDecryptString("Decrypt Me!", key)
```

- v) *TMDCMD5Hash(text as string) As String*

TMDCMD5Hash will one-way encrypt a string with the MD5 hash.

```
Msgbox TMDCMD5Hash("Hash Me!")
```

- vi) *TMDCSHAHash(text as string) As String*

TMDCSHAHash will one-way encrypt a string with the SHA hash.

```
Msgbox TMDCSHAHash("Hash Me!")
```

- vii) *TMDCGetKeyFromString(data as string) As TMDCKey*

This will recreate a TMDCKey from a string which was taken from TMDCKey.stringvalue.

```
Dim Key as TMDCKey
Key=TMDCGetKeyFromString(Socket1.Read(1042))
// 1042 is the length of the key data
```

- viii) *TMDCEncryptFile(file1 as folderitem, file2 as folderitem, key as TMDCKey) As Boolean*

This will encrypt a file's data fork using the 144 bit Blowfish encryption algorithm.

```
Dim file1, file2 as folderitem
Dim b as boolean
file1=GetOpenFolderItem("anything")
if file1 = nil then
    return
end if
file2=GetSaveFolderItem("text", "Encrypted File")
if file2 = nil then
    return
end if
b=TMDCEncryptFile(file1, file2, TMDCCreateNewKey("Key1"))
```

- ix) *TMDCDecryptFile(file 1 as folderitem, file2 as folderitem, key as TMDCKey) As Boolean*

This will decrypt a file encrypted with TMDCEncryptFile.

```
Dim file1, file2 as folderitem
Dim b as boolean
file1=GetOpenFolderItem("anything")
if file1 = nil then
    return
end if
file2=GetSaveFolderItem("text", "Decrypted File")
if file2 = nil then
    return
end if
b=TMDCDecryptFile(file1, file2, TMDCCreateNewKey("Key1"))
```

III. Files

The items within this section all deal with files and folderitems. The items in this section can extend previous REALbasic calls as well as add new ones.

i) *Folderitem->TMDCOpenInFinder*

This opens the current item through the finder whether it be: a file, folder, or window.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
f.TMDCOpenInFinder
```

ii) *Folderitem->TMDCRevealInFinder*

This will select the icon of a file, folder, or window in the and open any necessary folders to do so.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
f.TMDCRevealInFinder
```

iii) *TMDCFinderFlags*

This is a class which adds Finder flags to REALbasic.

Folderitem->TMDCFinderFlags.Invisible As Boolean

This will tell you if the file is invisible or not.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.Invisible then
    MsgBox f.name + " is invisible."
else
    MsgBox f.name + " is not invisible."
end if
```

Folderitem->TMDCFinderFlags.HasCustomIcon As Boolean

This will tell you if the file has a custom icon or not.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.HasCustomIcon then
    MsgBox f.name + " has a custom icon."
else
    MsgBox f.name + " does not have a custom icon."
end if
```

Folderitem->TMDCFinderFlags.Alias As Boolean

This will tell you if the file is an alias or not.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.Alias then
    MsgBox f.name + " is an alias."
else
    MsgBox f.name + " is not an alias."
end if
```

iv) *TMDCEncodeBinHexFile(source as Folderitem, destination as Folderitem) As Boolean*

This function will encode a file with the BinHex 4.0 method for transfer of Macintosh files over the internet.

```
Dim source, destination as folderitem
Dim b as boolean
source=GetOpenFolderItem("anything")
destination=GetSaveFolderItem("anything", "Binhexed File")
b=TMDCEncodeBinHexFile(source,destination)
```

- v) *TMDCDecodeBinHexFile(source as Folderitem, destination as Folderitem) As Boolean*

This function will remove the BinHex 4.0 encoding on a file.

```
Dim source, destination as folderitem
Dim b as boolean
source=GetOpenFolderItem("anything")
destination=GetSaveFolderItem("anything", "Decoded BinHex File")
b=TMDCDecodeBinHexFile(source,destination)
```

- vi) *Folderitem ->TMDCPrintInFinder*

This will cause the Finder to print a folderitem by opening the program, displaying the print dialog, and then quitting the application.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
f.TMDCPrintInFinder
```

- vii) *TMDCCanUseNavSvcs As Boolean*

This will return false if the computer does not have navigation services 1.1 (Mac OS 8.5 or above).

```
If TMDCCanUseNavSvcs then
    MsgBox "This computer can use navigation services!"
else
    MsgBox "This computer cannot use navigation services!"
End if
```

- viii) *TMDCNavChooseObject As Folderitem*

This will bring up a standard navigation choose (open) dialog and let you pick either a file or folder via a choose button.

```
Dim f as folderitem
f=TMDCNavChooseObject("anything")
Msgbox f.name
```

- ix) *TMDCEncodeMacBinaryFile(source as folderitem, destination as folderitem) As Boolean*

This will encode a file in Mac Binary III for internet transmission.

```
Dim source, destination as folderitem
Dim b as boolean
source=GetOpenFolderItem("anything")
destination=GetSaveFolderItem("anything", "Encoded File")
b=TMDCEncodeMacBinaryFile(source, destination)
```

- x) *TMDCDecodeMacBinaryFile(source as folderitem, destination as folderitem) As Boolean*

This will decode a Mac Binary III in to another file.

```
Dim source, destination as folderitem
Dim b as boolean
source=GetOpenFolderItem("anything")
destination=GetSaveFolderItem("anything", "Encoded File")
b=TMDCDecodeMacBinaryFile(source, destination)
```

- xi) *TMDCEncodeMacBinaryString(file as folderitem) As String*

TMDCEncodeMacBinaryString takes a file and encodes it in to a string.

```
Dim s as string
Dim f as folderitem
f=GetOpenFolderItem("anything")
s=TMDCEncodeMacBinaryString(f)
Msgbox s
```

- xii) *TMDCDecodeMacBinaryString(macbinary as string, file as folderitem) As String*

TMDCDecodeMacBinaryString takes a file and decodes it in to a string.

```
Dim s as string
Dim f1, f2 as folderitem
f1=GetOpenFolderItem("anything")
f2=GetSaveFolderItem("anything", "Decoded")
s=TMDCEncodeMacBinaryString(f1)
TMDCDecodeMacBinaryString(s, f2)
```

xiii) *TMDCIsMacBinaryFile(file as folderitem) As Boolean*

This will check if a file is encoded in MacBinary III and will return true if it is.

```
Dim file as folderitem
file=GetOpenFolderitem("anything")
if TMDCIsMacBinaryFile(file) then
    MsgBox "This is a MacBinary file."
else
    MsgBox "This is not a MacBinary file."
end if
```

xiv) *TMDCUUEncode(source as folderitem, destination as folderitem) As Boolean*

This will encode a file using the UUCode algorithm.

```
Dim source, destination as folderitem
Dim b as boolean
source=GetOpenFolderitem("Anything")
destination=GetSaveFolderitem("Anything","UU Encoded File")
b=TMDCUUEncode(source, destination)
```

xv) *TMDCUUDecode(source as folderitem, destination as folderitem) As Boolean*

This will decode a file which has been encoded with the UUCode algorithm.

```
Dim source, destination as folderitem
Dim b as boolean
source=GetOpenFolderitem("Anything")
destination=GetSaveFolderitem("Anything","UU Decoded File")
b=TMDCUUDecode(source, destination)
```


xvi) *TMDCFinderFlags*

This is a class which add's Finderflags to REALbasic.

Folderitem->TMDCFinderFlags.HasBNDL As Boolean

This will tell you if the file has a BNDL or not.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.HasBNDL then
    MsgBox f.name + " has a BNDL."
else
    MsgBox f.name + " does not have a BNDL."
end if
```

Folderitem->TMDCFinderFlags.Stationery As Boolean

This will tell you if the file is a stationery file or not.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.Stationery then
    MsgBox f.name + " is a stationery file."
else
    MsgBox f.name + " is not a stationery file."
end if
```

Folderitem->TMDCFinderFlags.Inited As Boolean

This will tell you if the file is inited.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.Inited then
    MsgBox f.name + " is inited."
else
    MsgBox f.name + " is not inited."
end if
```

Folderitem->TMDCFinderFlags.NoINITs As Boolean

This will tell you if the file is has no INIT resources.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.NoINITs then
    MsgBox f.name + " has no INITs."
else
    MsgBox f.name + " has INITs."
end if
```

Folderitem->TMDCFinderFlags.Shareable As Boolean

This will tell you if the file is set up to be shared.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.Shareable then
    MsgBox f.name + " is shareable."
else
    MsgBox f.name + " is not shareable."
end if
```

Folderitem->TMDCFinderFlags.NameLocked As Boolean

This will tell you if the file has its name locked.

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.NameLocked then
    MsgBox f.name + " name is locked."
else
    MsgBox f.name + " name is not locked."
end if
```

xvii) *TMDCExtendedFinderFlags*

This is a class which add's Mac OS 8.5 + Finderflags to REALbasic. The can only be viewed, not set.

Folderitem->TMDCExtendedFinderFlags.FlagsInvalid As Boolean

This will tell you if the file's flags are invalid.

```

Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCExtendedFinderFlags.FlagsInvalid then
    MsgBox f.name + " has invalid flags."
else
    MsgBox f.name + " does not have invalid flags."
end if

```

Folderitem->TMDCExtendedFinderFlags.HasCustomBadge As Boolean

This will tell you if the file has a custom badge.

```

Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCExtendedFinderFlags.HasCustomBadge then
    MsgBox f.name + " has custom badge."
else
    MsgBox f.name + " does not have custom badge."
end if

```

Folderitem->TMDCExtendedFinderFlags.HasRoutingInformation As Boolean

This will tell you if the file has routing information.

```

Dim f as folderitem
f=GetOpenFolderItem("anything")
if f.TMDCFinderFlags.HasRoutingInformation then
    MsgBox f.name + " has routing information."
else
    MsgBox f.name + " does not have routing information."
end if

```

IV. Folders

The items within this all deal with Macintosh Folders which are recognized by the Mac OS. The way REALbasic receives these are with a colon at the end like HD:Applications:. Folders which do not exist when you call for them are automatically created. Use FolderName As FolderItem.

Below are the list of Folders and what they do:

Folder	What It Does
TMDCExtensionsDisabledFolder	Goes to the Extensions Disabled Folder
TMDCControlPanelsDisabledFolder	Goes to the Control Panels Disabled Folder
TMDCStartupItemsDisabledFolder	Goes to the Startup Items Disabled Folder
TMDCShutdownItemsDisabledFolder	Goes to the Shutdown Items Disabled Folder
TMDCVoicesFolder	Goes to the Voices Folder
TMDCAppleExtrasFolder	Goes to the Apple Extras Folder
TMDCThemesFolder	Goes to the Themes Folder
TMDCLauncherItemsFolder	Goes to the Launcher Items Folder
TMDCFontsFolder	Goes to the Fonts Folder
TMDCModemScriptsFolder	Goes to the Modem Scripts Folder
TMDCApplicationSupportFolder	Goes to the Application Support Folder
TMDCHelpFolder	Goes to the Help Folder
TMDCSystemExtensionsDisabledFolder	Goes to the System Extensions Disabled Folder
TMDCApplicationsFolder	Goes to the Applications Folder
TMDCDocumentsFolder	Goes to the Documents Folder
TMDCVolumeRootFolder	Goes to the Volume's Root Folder
TMDCChewableItemsFolder	Goes to the Chewable Items Folder (??!!)
TMDCTextEncodingsFolder	Goes to the Text Encodings Folder
TMDCScriptingAdditionsFolder	Goes to the Scripting Additions Folder
TMDCSharedLibrariesFolder	Goes to the Shared Libraries Folder
TMDCControlStripModulesFolder	Goes to the Control Strip Modules Folder
TMDCAssistantsFolder	Goes to the Assistants Folder
TMDCUtilitiesFolder	Goes to the Utilities Folder
TMDCContextualMenuItemsFolder	Goes to the Contextual Menu Items Folder
TMDCFavoritesFolder	Goes to the Favorites Folder
TMDCInternetFolder	Goes to the Internet Folder
TMDCAppearanceFolder	Goes to the Appearance Folder
TMDCSoundSetsFolder	Goes to the SoundSets Folder
TMDCDesktopPicturesFolder	Goes to the Desktop Pictures Folder
TMDCInternetSearchSitesFolder	Goes to the Internet Search Sites Folder
TMDCFindByContentFolder	Goes to the Find By Content Folder
TMDCScriptsFolder	Goes to the Scripts Folder
TMDCFolderActionsFolder	Goes to the Folder Actions Folder
TMDCRecentApplicationsFolder	Goes to the Recent Applications Folder
TMDCRecentDocumentsFolder	Goes to the Recent Documents Folder
TMDCRecentServersFolder	Goes to the Recent Servers Folder
TMDCSpeakableItemsFolder	Goes to the Speakable Items Folder
TMDCKeychainFolder	Goes to the Keychain Folder

TMDCQuickTimeExtensionsFolder	Goes to the QuickTime Extensions Folder
TMDCFindByContentPluginsFolder	Goes to the Find By Content Plugins Folder

Please note that TMDCKeychainFolder will only work under Mac OS 9 as the Keychain Manager is required for this functionality.

V. Internet Config

All of the items within this section deal with the Internet Configuration Manager and its control panel. These are useful for finding out information on the Internet Config data and the user.

i) *TMDCICRealName() As String*

This gives you the name entered in the Internet Config control panel.

Msgbox TMDCICRealName

ii) *TMDCICEmailAddress() As String*

This gives you the e-mail address entered in the Internet Config control panel.

Msgbox TMDCICEmailAddress

iii) *TMDCICOrganization() As String*

This gives you the organization entered in the Internet Config control panel.

Msgbox TMDCICOrganization

iv) *TMDCICEmailAccount() As String*

This gives you the e-mail account entered in the Internet Config control panel.

Msgbox TMDCICEmailAccount

v) *TMDCICEmailPassword() As String*

This gives you the e-mail password entered in the Internet Config control panel.

Msgbox TMDCICEmailPassword

vi) *TMDCICNewsUser() As String*

This gives you the user name for the News User entered in the Internet Config control panel.

Msgbox TMDCICNewsUser

vii) *TMDCICNewsPassword() As String*

This gives you the password for the News User entered in the Internet Config control panel.

Msgbox TMDCICNewsPassword

viii) *TMDCICArchiePreferred() As String*

This gives you the preferred Archie server entered in the Internet Config control panel.

Msgbox TMDCICArchiePreferred

ix) *TMDCICUMichPreferred() As String*

This gives you the preferred UMich server entered in the Internet Config control panel.

Msgbox TMDCICUMichPreferred

x) *TMDCICInfoMacPreferred() As String*

This gives you the preferred InfoMac server entered in the Internet Config control panel.

Msgbox TMDCICInfoMacPreferred

xi) *TMDCICPhHost() As String*

This gives you the PhHost entered in the Internet Config control panel.

Msgbox TMDCICPhHost

xii) *TMDCICWhoisHost() As String*

This gives you the Whois Host entered in the Internet Config control panel.

Msgbox TMDCICWhoisHost

xiii) *TMDCICFingerHost() As String*

This gives you the Finger Host entered in the Internet Config control panel.

Msgbox TMDCICFingerHost

xiv) *TMDCICFTPHost() As String*

This gives you the FTP Host entered in the Internet Config control panel.

Msgbox TMDCICFTPHost

xv) *TMDCICTelnetHost() As String*

This gives you the Telnet Host entered in the Internet Config control panel.

Msgbox TMDCICTelnetHost

xvi) *TMDCICSMTPHost() As String*

This gives you the SMTP Host entered in the Internet Config control panel.

Msgbox TMDCICSMTPHost

xvii) *TMDCICNNTPHost() As String*

This gives you the NNTP Host entered in the Internet Config control panel.

Msgbox TMDCICNNTPHost

xviii) *TMDCICGopherHost() As String*

This gives you the Gopher Host entered in the Internet Config control panel.

Msgbox TMDCICGopherHost

xix) *TMDCICLDAPServer() As String*

This gives you the LDAP Server entered in the Internet Config control panel.

Msgbox TMDCICLDAPServer

xx) *TMDCICLDAPSearchbase() As String*

This gives you the LDAP Searchbase entered in the Internet Config control panel.

Msgbox TMDCICLDAPSearchbase

xxi) *TMDCICWWWHomePage() As String*

This gives you the default homepage entered in the Internet Config control panel.

Msgbox TMDCICWWWHomePage

xxii) *TMDCICWAISGateway() As String*

This gives you the WAIS Gateway information entered in the Internet Config control panel.

Msgbox TMDCICWAISGateway

xxiii) *TMDCICListFont() As String*

This gives you the default list font entered in the Internet Config control panel.

Msgbox TMDCICListFont

xxiv) *TMDCICScreenFont() As String*

This gives you the default screen font entered in the Internet Config control panel.

Msgbox TMDCICScreenFont

xxv) *TMDCICDocumentFont() As String*

This gives you the default document font entered in the Internet Config control panel.

Msgbox TMDCICDocumentFont

xxvi) *TMDCICPrinterFont() As String*

This gives you the default printer font entered in the Internet Config control panel.

Msgbox TMDCICPrinterFont

xxvii) *TMDCICSignature() As String*

This gives you the default signature entered in the Internet Config control panel.

Msgbox TMDCICSignature

xxviii) *TMDCICPlan() As String*

This gives you the Finger plan entered in the Internet Config control panel.

Msgbox TMDCICPlan

xxix) *TMDCICQuotingString() As String*

This gives you the character for default quoting entered in the Internet Config control panel.

Msgbox TMDCICQuotingString

xxx) *TMDCICMailHeaders() As String*

This gives you the mail headers entered in the Internet Config control panel.

```
Msgbox TMDCICMailHeaders
```

xxxii) *TMDCICNewsHeaders() As String*

This gives you the news headers entered in the Internet Config control panel.

```
Msgbox TMDCICNewsHeaders
```

xxxiii) *TMDCICNewMailFlashIcon() As Boolean*

This tells you whether or not the New Mail Flash Icon is active (whether or not the application's icon flashes when you have received new mail).

```
if TMDCICNewMailFlashIcon then
    MsgBox "New Mail Flash Icon is active"
else
    MsgBox "New Mail Flash Icon is not active"
end if
```

xxxiiii) *TMDCICNewMailDialog() As Boolean*

This tells you whether or not the New Mail Dialog is active (whether or not a dialog appears when you have received new mail).

```
if TMDCICNewMailDialog then
    MsgBox "New Mail Dialog is active"
else
    MsgBox "New Mail Dialog is not active"
end if
```

xxxiv) *TMDCICNewMailPlaySound() As Boolean*

This tells you whether or not the New Mail Play Sound is active (whether or not a sound plays when you have received new mail).

```
if TMDCICNewMailPlaySound then
    MsgBox "New Mail Play Sound is active"
else
    MsgBox "New Mail Play Sound is not active"
end if
```

xxxv) *TMDCICNewMailSoundName() As String*

This gives you the New Mail Sound Name entered in the Internet Config control panel (the name of the sound which plays when you have received a new message).

```
Msgbox TMDCICNewMailSoundName
```

xxxvi) *TMDCICWebBackgroundColor() As String*

This gives you the default Internet page background. The result is in RGB values, comma separating the different colors.

```
Msgbox TMDCICWebBackgroundColor
```

xxxvii) *TMDCICUseSOCKS() As Boolean*

This tells you whether or not SOCKS is active.

```
if TMDCICUseSOCKS then
    MsgBox "SOCKS is active"
else
    MsgBox "SOCKS is not active"
end if
```

xxxviii) *TMDCICSOCKSHost() As String*

This gives you the SOCKS Host that was entered in the Internet Config panel.

```
Msgbox TMDCICSOCKSHost
```

xxxix) *TMDCICUseHTTPProxy() As Boolean*

This tells you whether or not HTTP Proxy is active.

```
if TMDCICUseHTTPProxy then
    MsgBox "HTTP Proxy is active"
else
    MsgBox "HTTP Proxy is not active"
end if
```

x1) *TMDCICHTTPProxyHost() As String*

This gives you the HTTP Proxy Host that was entered in the Internet Config panel.

```
Msgbox TMDCICHTTPProxyHost
```

xli) *TMDCICUseGopherProxy() As Boolean*

This tells you whether or not Gopher Proxy is active.

```
if TMDCICUseHTTPGopher then
    MsgBox "Gopher Proxy is active"
else
    MsgBox "Gopher Proxy is not active"
end if
```

xlii) *TMDCICGopherProxyHost() As String*

This gives you the Gopher Proxy Host that was entered in the Internet Config panel.

```
Msgbox TMDCICHTTPGopherHost
```

xliii) *TMDCICUseFTPProxy() As Boolean*

This tells you whether or not FTP Proxy is active.

```
if TMDCICUseFTPProxy then
    MsgBox "FTP Proxy is active"
else
    MsgBox "FTP Proxy is not active"
end if
```

xliv) *TMDCICFTPProxyHost() As String*

This gives you the FTP Proxy Host that was entered in the Internet Config panel.

```
Msgbox TMDCICFTPProxyHost
```

xliv) *TMDCICFTPProxyUser() As String*

This gives you the FTP Proxy User that was entered in the Internet Config panel.

```
Msgbox TMDCICFTPProxyUser
```

xlvi) *TMDCICFTPProxyPassword() As String*

This gives you the FTP Proxy Password that was entered in the Internet Config panel.

```
Msgbox TMDCICFTPProxyPassword
```

xlvii) *TMDCICFTPProxyAccount() As String*

This gives you the FTP Proxy Account that was entered in the Internet Config panel.

```
Msgbox TMDCICFTPProxyAccount
```

xlviii) *TMDCICUsePassiveFTP() As Boolean*

This tells you whether or not to use FTP Passive mode.

```
if TMDCICUsePassive FTP then
    MsgBox "Passive FTP is active"
else
    MsgBox "Passive FTP is not active"
end if
```

il) *TMDCICSnailMailAddress() As String*

This gives you the postal address that was entered in the Internet Config panel.

```
Msgbox TMDCICSnailMailAddress
```

1) *TMDCICMacSearchHost() As String*

This gives you the MacSearch Host that was entered in the Internet Config panel.

```
Msgbox TMDCICMacSearchHost
```

li) *TMDCICWWWSearchPage() As String*

This gives you the default Internet search page.

```
Msgbox TMDCICWWWSearchPage
```

lii) *TMDCICDownload Folder as Folderitem*

This gives you the Downloads folder as a folderitem.

```
Msgbox TMDCICDownloadFolder.absolutePath
```

liii) *TMDCICSetTypeCreator(file as Folderitem) As Boolean*

This will set the write icon by determining it from the extension at the end.

```
Dim f as folderitem  
Dim b as boolean  
f=GetFolderItem("archive.sit")  
b=TMDCICSetTypeCreator(f)
```

VI. GUI & Dialog

All of the items within this deal with new controls added or the Macintosh graphic user interface.

i) *TMDCListViewWindowHeader*

This is a new control added to REALbasic. It is almost like the Placard but is slightly modified. It is what is seen just above a list in the Finder. It has a lighter bottom border line and can be used to reproduce Finder-like list window. There are no methods or properties associated with this control.

ii) *TMDCStdAlert(title As String, description as string) As Boolean*

This displays a Mac OS like modal alert dialog with a red border around the edges which can be seen when an error appears in the Finder.

Dim b as boolean

b=TMDCStdAlert("Error!", "The program is now forced to quit.")

iii) *TMDIconControl*

This is a control to get icons from files, applications, resources, and the Mac OS. It is split up in to three sections, No-Icon, Icon ID, and lastly, Type/Creator or file. They can all be used to retrieve an icon.

No Icon

This displays no icon.

Icon ID

This can be used to retrieve an icon from a 'cicn' resource inside the project or an icon from the system such as icon number 0 for the stop icon, icon number 1 for the note icon, or icon number 2 for the warning icon. Other system icons can also be displayed. To display an icon within a resource, make sure it is included and then type in the icon resource number, usually beginning with 128 and higher. The property IconID can

change the ID from one to another.

For example:

```
me.iconid=1
```

Type/Creator or File

This can retrieve the icon from either the given Type/Creator code or from a folderitem. If you know the type and creator, enter it in in the section where it asks. For the folderitem, use the following method:

SetIconFolderItem(item as FolderItem)

So, an example of this could be:

```
Dim f as folderitem
f=GetOpenFolderItem("anything")
if f <> nil then
    TMDCIconControl1.SetIconFolderItem(f)
end if
```

The commands *IconCreator As String* and *IconType As String* also work and can be used to change it dynamically.

iv) *TMDCStdConfirmYesNo(title As String, description As String) As Boolean*

This will display a standard Mac OS alert with two buttons, Yes and No. Returning **true** if Yes was selected and **false** if No was selected.

```
Dim b as boolean
b=TMDCStdConfirmYesNo("Yes/No", "Please press a button...")
if b then
    MsgBox "Yes was selected"
else
    MsgBox "No was selected"
end if
```

- v) *TMDCStdConfirmOkCancel(title As String, description As String) As Boolean*

This will display a standard Mac OS alert with two buttons, OK and Cancel. Returning **true** if OK was selected and **false** if Cancel was selected.

```
Dim b as boolean
b=TMDCStdConfirmOkCancel("OK/Cancel", "Please press a button...")
if b then
    MsgBox "OK was selected"
else
    MsgBox "Cancel was selected"
end if
```

- vi) *TMDCStdSave(title As String, description As String) As Integer*

This will display a standard Mac OS Save dialog with three buttons, 'Save', 'Don't Save', and lastly 'Cancel'. It will return the following integers for the button selected:

Save:	1
Cancel:	2
Don't Save:	3

This should be used now on for save dialogs as it corresponds to Apple GUI guidelines. An example to use StdSave:

```
Dim i as integer
i=TMDCStdSave("Save?", "Closing without saving will lose changes.")
if i=1 then
    MsgBox "You chose to save..."
end if
if i=2 then
    MsgBox "You have chosen to cancel closing..."
end if
if i=3 then
    MsgBox "You have chosen to discard changes..."
end if
```

vii) *TMDCCustomAlert*

This will allow you to build a Mac OS style custom alert with up to three buttons. To add them create a variable and add items to it. Below is a list of items you can change/add to the custom alert.

Item	Description
<i>Title As String</i>	Top heading
<i>Description As String</i>	Main text
<i>Button1 As String</i>	Name of button 1
<i>Button2 As String</i>	Name of button 2
<i>Button3 As String</i>	Name of button 3
<i>Default As Integer</i>	Default button (between 1-3)
<i>Cancel As Integer</i>	Cancel button (between 1-3)

To activate the alert, use **.show** on the variable. To find out which one was selected, use the command **Selected As Integer**.

For an example, please view the example file, '6) GUI & Dialog'.

VII. Registration

All of the items within this section deal with our registration system. This system is one of the most secure in the world and does not decrypt the password but create a new one from the entered information thus cannot be changed via MacsBug or an other program similar to it.

- i) *TMDCCheckRegistration(name As String, org As String, prod As String, code As String) As Boolean*

This will check the registration by creating a new registration code with the inputted information and then check the new code with the old code. It will return **true** if it is correct or **false** if the codes do not match. You can exclude 'org' if you want but including it means that the person who enters the code will NEED to enter the exactly what was written in the 'org' section.

```
Dim b as boolean
Dim name, org, prod, code as string
name= "John Doe"
org= "TMDC"
prod= "MyApp"
code= "2C2F3D15-A1DD316-88FCE81C"
b=TMDCCheckRegistration(name,org,prod,code)
if b then
    MsgBox "The code you entered was correct."
else
    MsgBox "The code you entered was incorrect."
end if
```

- ii) *TMDCCreateNewRegistration(name As String, org As String, prod As String) As String*

This will create a new code. This function is not necessary when checking the code as it would comprise the security of your program.

```
Dim code as string
code=TMDCCreateNewRegistration("John Doe", "TMDC", "MyApp")
Msgbox code
```

VIII. Sound

All of the items within this section deal with sound. 'ThemeDragSounds' are special sounds the computer makes when dragging. You can find the four letter codes in **Appendix A** as well as the 'ThemeSounds' there.

i) *TMDCBeginThemeDragSound(id As String) As Boolean*

This will start a theme drag sound, a sound which plays while moving the mouse around. In the 'id' part, enter a 4 letter code which tells which sound to play. You can find the list of codes in **Appendix A**. In the example below, you will hear a scroll bars sound when you drag it.

```
Dim b as boolean  
b=TMDCBeginThemeDragSound("slth")
```

ii) *TMDCEndThemeDragSound() As Boolean*

This will end the theme drag sound. **You must include this somewhere otherwise the sound will continue throughout the use of the machine until another application stops the drag sound.**

```
Dim b as boolean  
b=TMDCEndThemeDragSound
```

iii) *TMDCPlayThemeSound(id As String) As Boolean*

This will play a theme sound. A list of four letter codes (or IDs) can be found in **Appendix A**.

```
Dim b as boolean  
b=TMDCPlayThemeSound("wzme")
```

iv) *TMDCSetSystemVolume(vol As Integer) As Boolean*

This will set the system sound volume. Enter an integer between 0 and 256 where 0 is mute and 256 is the maximum.

```
Dim b as boolean  
b=TMDCSetSystemVolume(128)
```

v) *TMDCMADPlayer*

This will play MAD music files. This is a class with four properties.

<i>Property</i>	<i>Description</i>
File As FolderItem	The MAD file to play
Playing As Boolean	Whether the MADPlayer is playing
Position As Integer	The current position
Length As Integer	The length of this music

Example:

```
dim p as TMDCMADPlayer
dim f as FolderItem
f = GetOpenFolderItem("MAD")
if f <> nil then
    p = new TMDCMADPlayer
    p.file = f
    p.playing = true
    ProgressBar1.value = p.position
    ProgressBar1.maximum = p.length
end if
```

Please note that the TMDCMADPlayer uses MAD playing functions similar to those of PlayerPRO, by Yves Rosset. This implementation of the MAD player can only play MAD files, which can be created using PlayerPRO.

IX. Networking

All of the items within this section deal with uploading and downloading from and to a server.

i) *TMDCURLDownload(url As String, file As FolderItem) As Boolean*

This will download a file to the folderitem selected. This can download off both HTTP and FTP servers and will display a Mac OS dialog which looks like a Finder copy window but is for downloading from servers.

```
Dim b as boolean
Dim f as folderitem
Dim url as string
f=GetSaveFolderItem("anything", "Downloaded File")
if f <> nil then
    url="http://www.tmdc.org/programs/tmdcbugreporter.sit.bin"
    b=TMDCURLDownload(url, f)
end if
```

ii) *TMDCURLAccessAvailable() As Boolean*

This will tell you if the technology to do the Upload/Download functions is available on the machine. Mac OS 8.6 or higher is required to use URLAccess.

```
Dim b as boolean
b=TMDCURLAccessAvailable
if b then
    MsgBox "URL Access is available."
else
    MsgBox "URL Access is not available."
end if
```

iii) *TMDCURLUpload(url As String, user As String, pass As String, file As FolderItem) As Boolean*

This will upload only to FTP servers a file you have on your hard disk. Here is an example which is supposed to (but does not work) upload a file on to ftp://ftp.apple.com.

```
Dim b as boolean
Dim url, user, pass as string
Dim f as folderitem
url= "ftp://ftp.apple.com"
user= "sjobs"
pass= "ibook"
f=GetOpenFolderItem("anything")
if f <> nil then
    b=TMDCURLUpload(url,user,pass,f)
end if
```


A. Appendix A

All the items in Appendix A deal with with the sound section and the four letter codes need to use the *TMDCBeginThemeDragSound* and *TMDCPlayThemeSound* functions.

- i) Below is a list of four letter codes which can be used with the function *TMDCBeginThemeDragSound*. These codes are case sensitive.

<i>Item</i>	<i>Code</i>
MoveWindow	wmov
GrowWindow	wgro
MoveUtilWindow	umov
GrowUtilWindow	ugro
MoveDialog	dmov
MoveAlert	amov
MoveIcon	imov
SliderThumb	slth
SliderGhost	slgh
ScrollBarThumb	sbth
ScrollBarGhost	sbgh
ScrollBarArrowDecreasing	sbad
ScrollBarArrowIncreasing	sbai
Dragging	drag

- ii) Below is a list of four letter codes which can be used with the function *TMDCPlayThemeSound*. These codes are case sensitive.

<i>Item</i>	<i>Code</i>
MenuOpen	mnuo
MenuClose	mnuc
MenuItemHilite	mnui
MenuItemRelease	mnus
WindowClosePress	wclp
WindowCloseEnter	wcle
WindowCloseExit	wclx
WindowCloseRelease	wclr
WindowZoomPress	wzmp
WindowZoomEnter	wzme
WindowZoomExit	wzmx
WindowZoomRelease	wzmr
WindowCollapsePress	wcop
WindowCollapseEnter	wcoe
WindowCollapseExit	wcox
WindowCollapseRelease	wcor

WindowDragBoundary	wdbd
UtilWinClosePress	uclp
UtilWinCloseEnter	ucle
UtilWinCloseExit	uclx
UtilWinCloseRelease	uclr
UtilWinZoomPress	uzmp
UtilWinZoomEnter	uzme
UtilWinZoomExit	uzmx
UtilWinZoomRelease	uzmr
UtilWinCollapsePress	ucop
UtilWinCollapseEnter	ucoe
UtilWinCollapseExit	ucox
UtilWinCollapseRelease	ucor
UtilWinDragBoundary	dbd
WindowOpen	wopn
WindowClose	wcls
WindowZoomIn	wzmi
WindowZoomOut	wzmo
WindowCollapseUp	wcol
WindowCollapseDown	wexp
WindowActivate	wact
UtilWindowOpen	uopn
UtilWindowClose	ucls
UtilWindowZoomIn	uzmi
UtilWindowZoomOut	uzmo
UtilWindowCollapseUp	ucol
UtilWindowCollapseDown	uexp
UtilWindowActivate	uact
DialogOpen	dopn
DialogClose	dlgc
AlertOpen	aopn
AlertClose	altc
PopupWindowOpen	pwop
PopupWindowClose	pwcl
ButtonPress	btnp
ButtonEnter	btne
ButtonExit	btnx
ButtonRelease	btnr
DefaultButtonPress	dbtp
DefaultButtonEnter	dbte
DefaultButtonExit	dbtx
DefaultButtonRelease	dbtr
CancelButtonPress	cbtp
CancelButtonEnter	cbte
CancelButtonExit	cbtx
CancelButtonRelease	cbtr
CheckboxPress	chkp
CheckboxEnter	chke
CheckboxExit	chkx
CheckboxRelease	chkr
RadioPress	radp
RadioEnter	rade

RadioExit	radx
RadioRelease	radr
ScrollArrowPress	sbap
ScrollArrowEnter	sbae
ScrollArrowExit	sbax
ScrollArrowRelease	sbar
ScrollEndOfTrack	sbte
ScrollTrackPress	sbtp
SliderEndOfTrack	slte
SliderTrackPress	sltp
BalloonOpen	blno
BalloonClose	blnc
BevelPress	bevp
BevelEnter	beve
BevelExit	bevx
BevelRelease	bevr
LittleArrowUpPress	laup
LittleArrowDnPress	ladp
LittleArrowEnter	lare
LittleArrowExit	larx
LittleArrowUpRelease	laur
LittleArrowDnRelease	ladr
PopupPress	popp
PopupEnter	pope
PopupExit	popx
PopupRelease	popr
DisclosurePress	dscp
DisclosureEnter	dsce
DisclosureExit	dscx
DisclosureRelease	dscr
TabPressed	tabp
TabEnter	tabe
TabExit	tabx
TabRelease	tabr
DragTargetHilite	dthi
DragTargetUnhilite	dtuh
DragTargetDrop	dt dr
EmptyTrash	ft rs
SelectItem	fsel
NewItem	fnew
ReceiveDrop	fdrp
CopyDone	fcpd
ResolveAlias	fral
LaunchApp	flap
DiskInsert	dski
DiskEject	dske
FinderDragOnIcon	fdon
FinderDragOffIcon	fdof