ScriptWorX32 – Creating VBA Wizard Applications

APPLICATIONS NOTE

December 2007

Description: Guide to create VBA Wizard Applications in ScriptWorX32.
OS Requirement: Win 2000 Pro/Server, XP Pro, Server 2003
General Requirement: Knowledge of VBA and familiarity with the application note *ScriptWorX32 - Fast Multi-threading VBA Applications*

Introduction

The main function of ScriptWorX32 is to manage and maintain scripts that perform different functions. It takes advantage of the new Visual Basic 6.3 feature – multi-threading. The use of multi-threading enables operating systems to run more efficiently because it assigns different tasks to different threads so that the CPU usage will be split up amongst these threads. This is very useful if you have a need or desire to run different tasks, which do not rely on each other, at the same time.

Script Wizards

The Script Wizards generate scripts from script templates. Each script template is stored in one script template file with a "stp" extension, which is located in the ScriptWizardTemplate directory placed in the installation directory of ScriptWorX32. Script templates support several keywords, which are replaced during script generation by values entered by user in the Wizard edit fields. Script templates also support adding of necessary Type Library references, which can be appended to a VBA container. Type Library references and optional keywords must be specified in the header section of the script.

The header section can contain optional keywords, which must be in the following format:

#KEYWORD: value

Table 1 - Available Keywords

Keywords	Value
#REFERENCE:	module1
#DESCRIPTION:	text1
#PARAMETER:	parameter
#END	

REFERENCE
The header can have one or more references specified. These references are similar to adding references manually in the VBA editor for a particular script. Each REFERENCE statement refers to one module only.

DESCRIPTION
The description is the script information to be displayed in the Script Wizard dialog. Description can be placed on several lines.

PARAMETERS
ScriptWorX32 templates allow the user to specify parameters to be parsed into the script. Parameters are exclusive, i.e. only one of them can be used per script wizard template. A parameter can be one of the following strings:

Table 2 - Parameter and Its Descriptions

Parameter	Description
par_FileName	Script template includes file name parameter, when declared, the Script Wizard displays edit field and browse button to specify the file name.
par_Tag	Script template includes OPC tag, when declared, the Script Wizard displays edit field and browse button to specify the OPC Tag.
par_None	Has no effect

END
The header section must end with this keyword.
#END marks the end of the header section and is required at all times.

Optional and Required Parameters in Scripts
Optional parameter must be defined in the header section as described above. There are also some required parameters. Required parameters can be used without being defined in the header section. Required parameters are:

Table 3 - Required Parameters

Parameter	Description
par_Name	This parameter will contain the script name
par_Key	This parameter will contain a GenRegistrar key
par_Node	This parameter will contain the GenRegistrar node.

When any optional parameters or required parameters are used in the script template code, they must be enclosed by << and >> characters. Parameters are replaced during script generation by the values specified by the user in the related edit fields.

GenRegistrar

GenRegistrar is a hidden ICONICS application holding dispatch pointers (References) to other running applications that may register there. Most ICONICS products including GraphWorX32, TrendWorX32, AlarmWorX32, DataWorX32 and their ActiveX controls register to the GenRegistrar. Each instance registers under a different keyword. The advantage of GenRegistrar is that it also works with networking, so based on key (default "*") and node name (default "*"), you can obtain instances running on another computer.

Example 1:

```
#DESCRIPTION:      Script shows the script File Name
#END
' Script <<par_Name>> was generated by ICONICS
' Script Wizard from a template ShowMessage.stp
'
Public Sub <<par_Name>>()
   MsgBoX  “The script is <<par_Name>>"
End Sub
```

Example 2:

```
#REFERENCE:        {CA84D601-E21C-11CE-97DA-00608CB87A7C}
#DESCRIPTION:      Script starts GWX32 and opens display selected
#DESCRIPTION:      in the FileName field. Gwx32 must be installed!
#PARAMETER:        par_FileName
#END

' The reference {CA84D601-E21C-11CE-97DA-00608CB87A7C} is the CLSID
' for GraphWorX32
' Script <<par_Name>> was generated by ICONICS
' Script Wizard from a template OpenDisplay.stp

Public Sub <<par_Name>>()
   ' Create an instance of ObjGraphWorX32
   Dim objGraphWorX As GWX32.GWXDisplay
   Set objGraphWorX = New GWX32.GWXDisplay
   If objGraphWorX Is Nothing then
            ' Report problem and exit
            MsgBox " ObjGraphWorX32 creation failed. Check it is installed
and registered"
            Exit Sub
   End If
   Call objGraphWorX.FileNew
   Call objGraphWorX.FileOpen("<<par_FileName>>")
   Call objGraphWorX.BringWindowToTop
   Call objGraphWorX.ShowWindow

   MsgBox "Close ObjGraphWorX32"
   call objGraphWorX.ExitApplication
   set objGraphWorX = nothing
End Sub
```