Сергей Гусев

Ответы на вопросы пользователей SCADA-системы GENESIS32

Вопрос

После запуска утилиты лицензирования появляется сообщение:
«Ошибка утилиты лицензирования (-CKERR -8103). Дополнительная информация отсутствует.»

Ответ

Причиной возникновения этой ошибки, как правило, является установка какого-либо сервис-пакета, в результате чего некоторые файлы из состава подсистемы GenLic32 оказываются перемещенными или переписанными более старыми версиями.

Методика устранения ошибки

1. Скачать с FTP-сервера ПРОСОФТ файл ftp://ftp.prosoft.ru/pub/Software/Genesis32/Fixes/LicenseFix/CKERR1021.zip
2. Установить все последние доступные версии и пакеты обновлений Windows и GENESIS32.
   Пакеты последних обновлений GENESIS32 всегда доступны по адресу ftp://ftp.prosoft.ru/pub/Software/Genesis32/Fixes/
3. Скопировать Cks.exe и Setupex.exe в директорию Program Files\Iconics\SoftLic
4. Запустить Setupex.exe, и подождать несколько секунд (как минимум 5) перед тем как осуществлять какие-либо другие действия
5. Удалить **только** Genlic32.rst, Genlic32.41s, Genlic32.key, Genlic32.ent и Genlic32.ky2, если таковой присутствует в директории C:\Program Files\Iconics\SoftLic, а также файл C:\Windows\System32\ESNECIL.IND
6. Перезагрузиться

Вопрос

Когда подводишь курсор мыши к объекту или параметру, то видишь в жёлтеньком прямоугольнике название объекта, значение параметра и т.д.
Существует ли возможность каким-нибудь образом изменять свойства этих сообщений, заменять их сообщениями из произвольного набора, например названием параметра, временем его регистрации, отклонением от другого параметра, а также любым набором символов?

Ответ

Это несложно. Достаточно в свойствах дисплея провести следующую установку:

Далее нужно в свойствах статического объекта указать не просто имя и описание, но и добавить к нему какую-нибудь «безобидную» динамику, которая, например, будет менять цвет объекта с какого-нибудь одного цвета на точно такой же цвет. Это позволит выводить в ToolTips не только комментарии, но и значение любых параметров:

Для динамических объектов достаточно просто задать имя (или описание, если нужна вторая строка).

Вопрос

Нам в проекте необходимо вместе со значением параметра видеть еще и время его обновления.
Через DataSpy я вижу его, а вот можно ли его увидеть в GraphWorx?
Нужно ли что-нибудь дополнительно настраивать или программировать?

Ответ

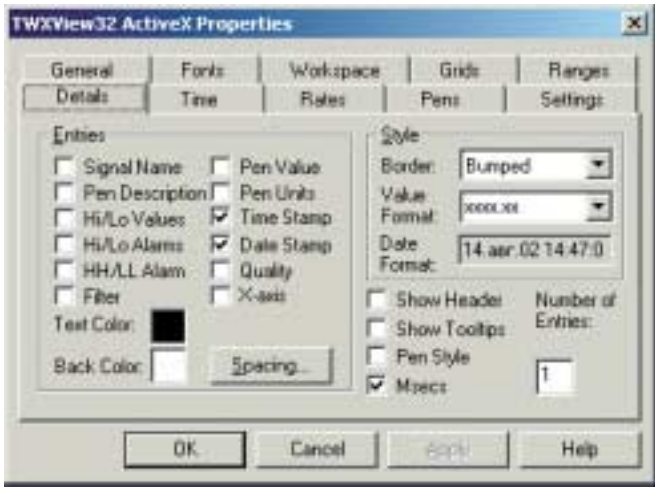
Конечно, решить эту задачу можно с помощью VBA. На диске GENESIS32 в разделе White Papers есть подобный пример доступа к значениям и атрибутам тега OPC из скрипта. Но этот код довольно объёмен и в большинстве случаев может показаться излишне сложным. Вполне удовлетворительный результат с меньшими трудозатратами можно получить, используя для доступа к таким атрибутам тега, как время происхождения, обычный ActiveX-элемент для просмотра трендов — TWXview32.

Для этого вставьте рядом с элементом, отображающим значение сигнала, стандартный TWXview32 ActiveX.

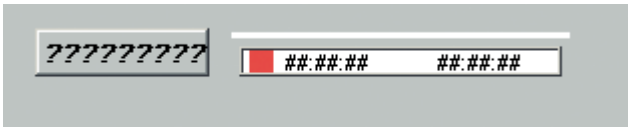
Далее оставляем в настройках рабочей области только «Детали»:

В настройке деталей оставляем только те атрибуты тега, которые нас интересуют. В данном случае – только время (с миллисекундами) и дату.

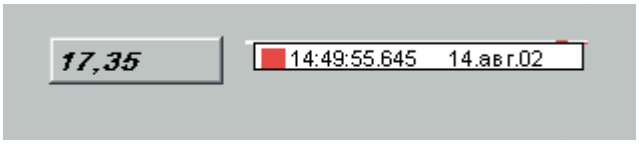
Обращаю внимание, что в поле Numbers of Entries нужно поставить «1» или другую цифру, показывающую, сколько сигналов Вам нужно отображать.



После чего уменьшаете размер TWXview32 ActiveX до тех пор, пока на экране не останется только текстовое поле с атрибутами тега, а графическое поле не выродится в одну линию.



В результате получается примерно такой вид в режиме исполнения:



Легко видеть, что, изменив немного настройки «Деталей», с помощью этого же ActiveX-элемента так же легко можно отобразить на экране и биты качества тега. Например, так:

Последние знаки вопроса относятся к несуществующему тегу.

Но нужно помнить, что технология OPC не стоит на месте, и практически во всех новых OPC-серверах, удовлетворяющих спецификации OPC 2.0 (официально принятой уже почти год назад), каждый тег, кроме простого значения, имеет теперь и множество других атрибутов, включая метку времени. Причем все эти атрибуты доступны через обычный интерфейс OPC DA (то есть как обычные старые теги). Эта поддержка OPC 2.0 есть уже у многих

производителей, включая фирму Iconics, а скоро, наверно, будет у всех.

Вопрос

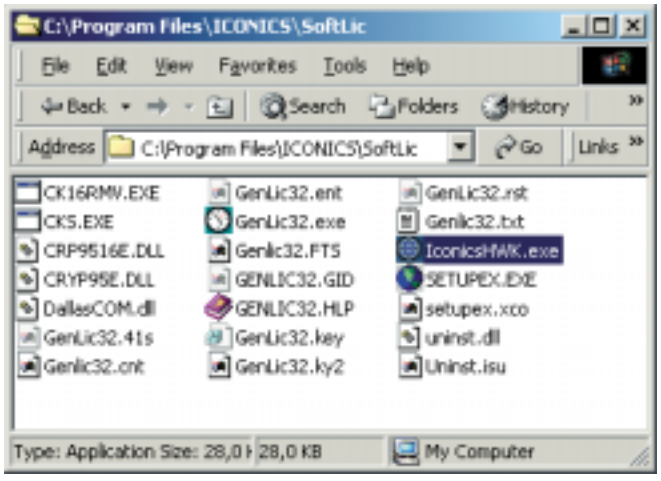
Каков механизм приобретения аппаратного ключа, если имеющиеся системы куплены без него (если лицензия уже активизирована и если еще нет)?

Ответ

Аппаратный ключ, конечно, гораздо проще приобретать в комплекте с новой лицензией. В этом случае Вам необходимо всего лишь заплатить дополнительно около 300 евро, заполнить и подписать дополнительное соглашение на использование аппаратного ключа. Всё это делается на этапе оформления заказа. После этого в коробочке, которая придет по вашему заказу, Вы найдете не только диск с последней версией GENESIS32, но и ключ аппаратной защиты. Вся необходимая информация будет уже зашита в нем.

Для активизации лицензии Вам достаточно только установить ключ и активизировать режим аппаратной защиты лицензии, запустив утилиту IconicsHWK.exe:

Не удивляйтесь, что эта утилита недоступна через панель задач. Ее можно найти в папке Program Files\Iconics\Softlic:



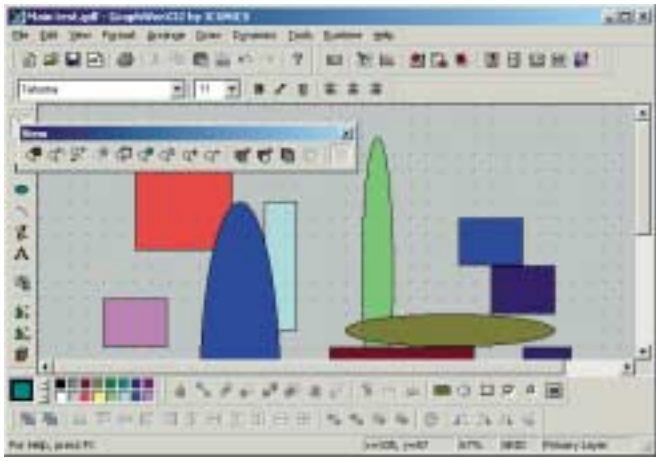
Если же Вы желаете приобрести аппаратный ключ к уже существующей программной лицензии, то Вам предстоит оплатить не только ключ, но и административный сбор за обслуживание операции переноса лицензии. Далее Вам необходимо будет удалить (выполнить операцию kill) лицензию, «вернуть» ее обратно на web-сайт Iconics и дождаться прихода аппаратного ключа.

Это займет несколько больше времени, чем поставка обычной лицензии, поскольку службе технической поддержки Iconics потребуется время на то, чтобы убедиться в правильном удалении Вами программной лицензии со своего компьютера, на удаление записи о Вашей программной лицензии с web-сайта Iconics и на внесение соответствующей записи в аппаратный ключ. Увы, но все это время (на это уходит примерно 4-6 недель) Вам придется пользоваться временной 30-суточной лицензией.

Естественно, если лицензия еще не была активизирована, то это избавит Вас от необходимости её «убивать», и Вам достаточно сообщить при заказе ключа только регистрационные данные этой лицензии.

Вопрос

Для режима разработки GraphWorX32 имеет удобно настраиваемые панели меню, в том числе и панель управления масштабом изображения, которая может быть удобно расположена в любом месте экрана. А как можно сделать такую же всплывающую панель для режима исполнения?



Ответ

Если Вы не желаете оформить подобный элемент управления внешним всплывающим окном, то необходимо создать такое окно и запустить его при переходе Вашего основного проекта в режим исполнения. Для этого удобно воспользоваться стандартным событием GwxDisplay_PostAnimateDisplay():

```
Private Sub GwxDisplay_PostAnimateDisplay()
ThisDisplay.OpenPopupWindow "zoompopup.gdf", False, True,
   False
End Sub
```

Для того чтобы сохранить возможность работы оператора с основным окном проекта, это всплывающее окно должно быть не модальным (первый параметр False), расположенным в центре экрана (второй параметр) и видимым (третий параметр).

Кроме того, очень важно, чтобы при работе с кнопками, расположенными на всплывающем окне, управление передавалось на «родительское», главное окно. Для этого в обработчике события GwxDisplay_PostAnimateDisplay() всплывающего окна нужно узнать, откуда оно было запущено, и перенаправить действия VBA-кода на управление размерами именно «родительского» окна, например так:

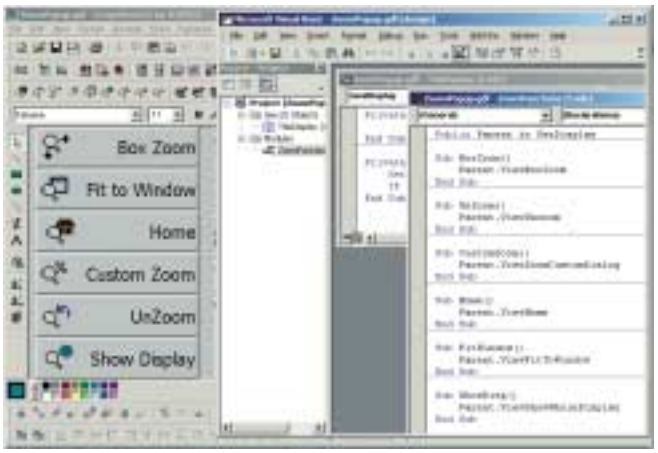
```
Private Sub GwxDisplay_PostAnimateDisplay()
  Set ZoomFunctions.Parent = ThisDisplay.GetParentDisplay
  If ZoomFunctions.Parent Is Nothing Then
ZoomFunctions.Parent = ThisDisplay
End Sub
```

Этот код присваивает объекту Parent модуля ZoomFunction значение указателя на «родительское» окно.

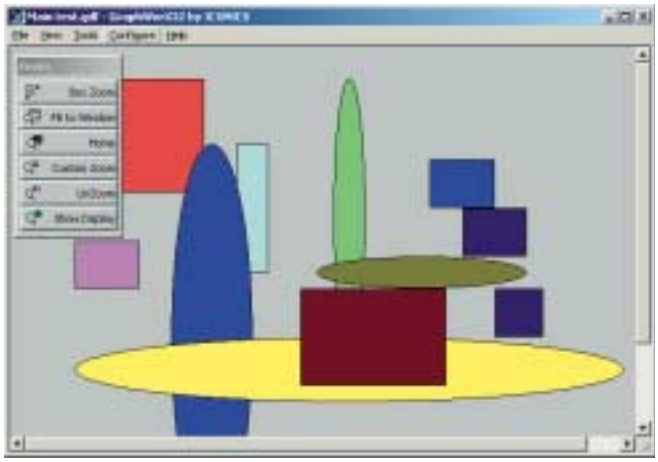
Остается только создать такой модуль с названием ZoomFunction для нашего всплывающего окна и обработать в нем нажатия на необходимые кнопки, например так:

```
Public Parent As GwxDisplay
Sub BoxZoom()
  Parent.ViewBoxZoom
End Sub
Sub UnZoom()
  Parent.ViewUnzoom
End Sub
Sub CustomZoom()
  Parent.ViewZoomCustomDialog
End Sub
Sub Home()
  Parent.ViewHome
End Sub
Sub FitWindow()
  Parent.ViewFitToWindow
End Sub
Sub ShowDisp()
  Parent.ViewShowWholeDisplay
End Sub
```

И нарисовать собственно кнопки управления, например так:



В результате получаем вот такую новую удобную панель управления, которую можем свободно перемещать по всему экрану и которая не мешает работать с основным дисплеем:



Нетрудно сделать так, чтобы пользователь не мог случайно закрыть это окошко, убрав галочку System Menu в настройках формата экранной формы.

Вопрос

Как убрать кнопку закрытия окна [X] с обычной экранной формы — понятно, а как можно убрать эту кнопку с формы VBA, которая открывается, например, по какому-нибудь событию и должна иметь возможность быть закрытой не оператором, а только из VBA?

Ответ

Да, в отличие от обычного Visual Basic, напрямую запретить кнопку [X] в своих формах VBA не позволяет. Для этого придется воспользоваться функциями API операционной системы Windows, такими как GetSystemMenu(...) и RemoveMenu(...). Для того чтобы удалить часть меню любого окна в Windows, необходимо получить указатель на это окно. Для этого существует API-функция FindWindow(...).

Таким образом, создав нужную нам форму VBA, мы прежде всего обязаны сами позаботиться о том, чтобы закрыть эту форму, например, добавив в неё кнопку с привязанным к ней скриптом:

```
Private Sub cmdExit_Click()
 ' Событие, приводящее к закрытию формы
  Unload Me
End Sub
```

Кроме того, в разделе Initialize нашей формы мы должны написать код, запрещающий в ней кнопку [X], например так:

```
Private Sub UserForm_Initialize()
  Dim lRes As Long
 ' Получаем указатель окна для нашей формы
  lRes = FindWindow(vbNullString, Me.Caption)
 ' Вызываем подпрограмму удаления кнопки для нашего
окна
  DisableCloseButton (lRes)
End Sub
```

После чего останется только оформить собственно подпрограмму DisableCloseButton () и декларации API-функций в виде отдельного модуля:

```
' Декларация стандартной API-функции GetSystemMenu
Declare Function GetSystemMenu Lib "user32" _
 (ByVal hwnd As Long, ByVal bRevert As Integer) _
As Integer
```

```
' Декларация стандартной API-функции RemoveMenu
Declare Function RemoveMenu Lib "user32" _
  (ByVal hMenu As Integer, ByVal nPosition _
  As Integer, ByVal wFlags As Integer) _
As Integer

' Декларация стандартной API-функции FindWindow
Declare Function FindWindow Lib «user32» Alias _
  "FindWindowA" (ByVal lpClassName As String, ByVal
lpWindowName As String) As Long

' Стандартная константа кнопки закрытия окна
Global Const MF_BYPOSITION = &H400

Public Sub DisableCloseButton(ThehWnd As Long)
  Dim SystemMenu As Integer
  Dim iRes As Integer
  SystemMenu = GetSystemMenu(ThehWnd, 0)
  iRes = RemoveMenu(SystemMenu%, 6, MF_BYPOSI-
TION)
End Sub
```

В результате получится форма, кнопка закрытия которой будет неактивна и для закрытия которой понадобится нажать на специальную кнопку.

Вопрос

Как в программе на VBA получить значение OPC-тега?

Ответ

Ответ на этот вопрос можно разделить на две части. Первая половина относится к тегам, которые существуют в доступных серверах OPC, но **не используются** в данной экранной форме, то есть не подключены ни к одному из существующих объектов GraphWorx. В этом случае необходимо написать довольно объемный код на VBA, пример которого есть в разделе AppNotes на любом стандартном диске GENESIS32.

Если же тег OPC имеет соединение с каким-либо объектом GraphWorx, то получить его значение не представляет никакого труда.

Достаточно воспользоваться функцией ThisDisplay.GetPointObjectFromName()

с параметром, в котором необходимо указать имя тега, например, создав в проекте кнопку и привязав к этой кнопке такой VBA-скрипт:

```
Sub a(o As GwxPick)
Dim p As GwxPoint      ' Определяем объект и соединяем
с тегом
  Set p = ThisDisplay.GetPointObjectFromName("ICON-
ICS.Simulator.1\SimulatePLC.Sine")
  MsgBox p.value
End Sub
```

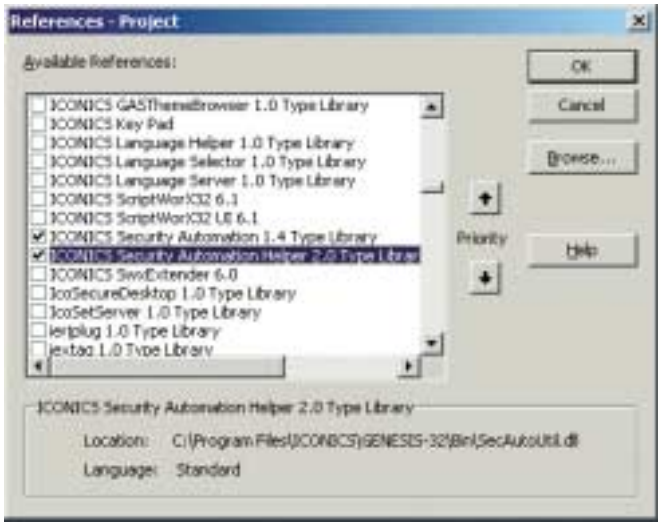
Получим:

Вопрос

Как можно в программе на VBA определить, кто именно из операторов в настоящее время зарегистрирован в системе?

Ответ

Для доступа к этой информации необходимо воспользоваться двумя библиотеками Security Automation, поставляемыми вместе с GENESIS32. Для этого в меню Tools-References в окне редактора VBA добавьте библиотеки Iconics Security Automation Library и Iconics Security Automation Helper Library:



Все, что Вам теперь необходимо сделать в Вашей программе на VBA, — это получить свойство LoggedIn у объекта класса SecDual, например так:

```
Sub a(o As GwxPick)
  Dim SecServer As SecDual     ' Определение объекта
класса Security server
  Set SecServer = New SecDual    ' Создание объекта
  SecServer.Node = «GUSEV»    ' Здесь необходимо ука-
зать наименование Вашего узла
  MsgBox SecServer.LoggedIn    ' Вывод имени зарегист-
рированного пользователя
End Sub
```

С.А. Гусев — сотрудник фирмы ПРОСОФТ
119313 Москва, а/я 81
Телефон: (095) 234-0636, факс: (095) 234-0640
E-mail: info@prosoft.ru