# Zuken E3

<https://www.zuken.com/us/blog/a-beginners-guide-to-scripting-for-e3-series/>

Вся основная инф

## Запуск/Подкючение к API e3

Строчка: Set app = CreateObject("CT.Application") – позволяет подключится к запущенной Е3 и выполнить какие то действия в ней.

Set app создает указатель на класс и таким образом создаёт объект класса

CreateObject( ) – Встроенная функция VBScript, которая создает экземпляр объекта, определенного классом, указанным в скобках.

"CT.Application" - Это строка, представляющая имя класса объекта. “CT.Application” обычно относится к приложению, которое использует библиотеку Automation (OLE Automation) для взаимодействия с другими приложениями.

### Quick start

Dim app

set app = CreateObject("CT.Application")

app.PutMessage "Hello E3!!!"

set app = nothing

wscript.Quit

### Подключение к е3 и проекту

Открыт только один проект (1 процесс)

Dim app, appId

Set app = E3Connetction(appId)

app.PutMessage "Подключено к е3"

Dim job, jobId

Set job = JobConnection(jobId)

app.PutError   1, "Подключено к проекту"

Инфо

jobName = job.GetName()

jobPath = job.GetPath()

app.PutInfo 0, "path: " & jobPath & Chr(10) & "name: " & jobName

Много открытых проектов

### Запуск нового экземпляра e3 и job, если хотя бы одна е3 и job открыты

Set e3 = CreateObject( "CT.Application" )

Через диспатчер:

Set dp = CreateObject( "CT.Dispatcher" )

Set e3 = dp.OpenE3Application( <jobname> )

### Запуск нового экземпляра e3 и job, если ни одна е3 не открыта

@echo off

echo ---Подключение диса D---

if not exist D: mkdir C:\D

if not exist D: subst D: C:\D

echo ---Создание папки логов---

if not exist D:\sapr\E3\_temp mkdir D:\sapr\E3\_temp

echo ---Задаем переменные---

set bat\_path=D:\Zuken\SM\_Additional

call "%bat\_path%\e3\_env.bat"

echo ---Активация службы для взаимодействия с сервером TC---

If not exist D:\tc\_client4\_11\tccs\auto\_starttccs.bat %SystemRoot%\system32\xcopy.exe /Y /Q /R /K /H /Z "%template\_path%\E3.Teamcenter\_error.txt" "%E3\_temp%"

If exist D:\tc\_client4\_11\tccs\auto\_starttccs.bat call "D:\tc\_client4\_11\tccs\auto\_starttccs.bat"

if [%1]==[] goto normal

:advanced

echo ---Вызов E3---

call "%bat\_path%\E3.series\_2019\start\_E3\_cabel\_MU.bat" %1%

goto rest

:normal

echo ---Вызов E3---

call "%bat\_path%\E3.series\_2019\start\_E3\_cabel\_MU.bat"

goto rest

:rest

#### Описание

**1. Настройка окружения**

* **echo ---Задаем переменные---**: Выводит текст “Задаем переменные—” в консоль, чтобы указать на начало блока настройки.
* **set bat\_path=D:\Zuken\SM\_Additional**: Задает переменную bat\_path с значением пути к каталогу D:\Zuken\SM\_Additional. Этот каталог, вероятно, содержит вспомогательные файлы для работы с E3.series.
* **call "%bat\_path%\e3\_env.bat"**: Выполняет файл e3\_env.bat (находящийся в каталоге bat\_path) с помощью команды call. Этот файл, скорее всего, устанавливает переменные окружения, необходимые для запуска E3.series.

**2. Активация службы взаимодействия с Teamcenter**

* **echo ---Активация службы для взаимодействия с сервером TC---**: Выводит текст “Активация службы для взаимодействия с сервером TC—” в консоль.
* **If not exist D:\tc\_client4\_11\tccs\auto\_starttccs.bat %SystemRoot%\system32\xcopy.exe /Y /Q /R /K /H /Z "%template\_path%\E3.Teamcenter\_error.txt" "%E3\_temp%"**: Проверяет, существует ли файл D:\tc\_client4\_11\tccs\auto\_starttccs.bat.
  + Если файл **не существует**, команда xcopy копирует файл "%template\_path%\E3.Teamcenter\_error.txt" в каталог "%E3\_temp%". Это, вероятно, файл для отслеживания ошибок.
  + Если файл **существует**, команда xcopy не выполняется.
* **If exist D:\tc\_client4\_11\tccs\auto\_starttccs.bat call "D:\tc\_client4\_11\tccs\auto\_starttccs.bat"**: Если файл D:\tc\_client4\_11\tccs\auto\_starttccs.bat существует, выполняет его с помощью команды call. Этот файл, скорее всего, запускает службу для взаимодействия с сервером Teamcenter (система управления данными, возможно, для инженерных данных).

**3. Запуск E3.series**

* **if [%1]==[] goto normal**: Проверяет, есть ли аргументы командной строки (%1).
  + Если аргументов нет, переходит к метке :normal.
  + Если аргументы есть, переходит к метке :advanced.

**4. Вызов E3.series с аргументами (advanced)**

* **:advanced**: Метка, определяющая блок кода для запуска E3.series с аргументами.
* **echo ---Вызов E3---**: Выводит текст “Вызов E3—” в консоль.
* **call "%bat\_path%\E3.series\_2019\start\_E3\_cabel\_MU.bat" %1%**: Выполняет файл start\_E3\_cabel\_MU.bat (в каталоге bat\_path\E3.series\_2019) с аргументами %1%. Этот файл запускает E3.series с дополнительными параметрами, переданными в командной строке.
* **goto rest**: Переходит к метке :rest.

### Функции запуска

Тело:

Dim app, appId

' Подключение к процессу

Set app = E3Connetction(appId)

' Создание объектов проекта

Dim job, jobId

' Подключение к проекту

Set job = JobConnection(jobId)

Функции:

Function E3Connetction(ByRef appId)

' Улавливатель ошибок

On Error Resume Next

' Создание объекта E3.series

Set e3 = CreateObject("CT.Application")

' Запрос идентификатора процесса

appId = e3.GetId()

' Проверка идентификатора

If (appId = 0) Then

' Вывод сообщения о точ, что COM-классы не зарегистрированы!

MsgBox "Ошибка. Не зарегистрированы COM-классы приложения E3.series!", 16, "Ошибка"

End If

' Возврат функции

Set E3Connetction = e3

' Остановка улавливателя ошибок

On Error Goto 0

End Function

Function JobConnection(ByRef jobId)

' Создание объекта проект

Set job = app.CreateJobObject()

' Идентификатор проекта

jobId = job.GetId()

' Проверка

If (jobId = 0) Then

' Вывод сообщения

app.PutError 0, "Нет запущенных проектов!"

' Очистка

Set job = Nothing

Set jobId = Nothing

' Переход к процедуре выхода из скрипта

Call RBImportExitScript(False)

End If

' Возврат функции

Set JobConnection = job

End Function

### Пример из видоса

Dim app

on error resume next

set app = CreateObject("CT.Application")

if(err.number <> 0) then

MsgBox "Error ocured"

err.clear

set app = nothing

wscript.Quit

end if

on error goto 0

app.PutMessage "Hello E3!!!"

set app = nothing

wscript.Quit

## Управление программой

При подключении к е3 создается объект класса Application (VBS его находит как “CT.Application” в системе).

Set e3 = CreateObject( "CT.Application" ) – Этот класс является главным и без него не будут работать другие, через него мы создаем Job (проект) и через job, создаем любые объекты - листы, изделия итд.

Set Prj = E3.CreateJobObject - Метод позволяющий получить ссылку на объект класса Job

Dim app, appId

Set app = E3Connetction(appId)

app.PutMessage "Подключено к е3"

Dim job, jobId

Set job = JobConnection(jobId)

app.PutError   1, "Подключено к проекту"

### Определения

Sheet (лист) – лист

Device – изделие на листе или в базе данных

Tree - ???

#### Типы изделий

all device - все изделия (включая device, Cable, Connector, Terminal, Block, Hose, Tube, Connection, AllConnection)

device – QF1 например

Cable - провода

Connector - соединитель

Terminal

Block

Tube

Connection

AllConnection

#### Выводы и наконечники

Pin – вход или выход электротехнического изделия

dev.GetPinIds idarr – возвращает все пины изделия  
dev.GetAllPinIds idarr – тоже самое, но переходит глубже ко всем подустройствам в текущем, если это клеммная колодка. Возвращает все пины девайса и пины подустройства.

pinfit1 = pin.GetFitting – возвращает текущий наконечник, установленный на пин

pin.GetValidFittings strarr – заполняет массив всеми наконечниками установленными на всех пинах

app.PutMessage pinfit1 & " " & strarr(1)

### Разные объекты

#### Лист

'Создание нового листа

'sht.Create 0, "New", "DINA3", 0, 0

'sht.GetDrawingArea shtxmin, shtymin, shtxmax, shtymax

'sht.Display

'Подключение к активному листу

shtid = job.GetActiveSheetId ()

#### Device

###### Получить имя device-a

Name = dev.GetName

#### Символы

##### Загрузка символа из базы данных

'Получение id символа выбранного девайса

set dev = job.CreateDeviceObject

job.GetSelectedAllDeviceIds idarr

dev.SetId(idarr(1))

set sym0 = job.CreateSymbolObject()

out = dev.GetSymbolIds ( symids )

app.PutMessage(symids(1))

sym0.setid(symids(1))

'Загрузка символа из базы данных

set sym = job.CreateSymbolObject

sym.Load sym0.GetType, sym0.GetVersion

app.PutMessage sym.getid

If( sym.GetId = 0 ) Then

e3.PutError 1, "Cannot load symbol from database!!!!!!!"

wscript.Quit

End If

##### Расположение символа на новом листе

sym.GetArea symxmin, symymin, symxmax, symymax

newid = sym.Place( sht.GetId, \_

(shtxmax+shtxmin)/2 - (symxmax-symxmin)/2, \_

(shtymax+shtymin)/2 - (symymax-symymin)/2, \_

"" )

If( newid = 0 ) Then

e3.PutError 1, "Cannot place symbol"

wscript.Quit

End If

#### Pin

##### Получение атрибутов пина

##### Получение значений атрибута

##### Получение наконечников

pinfit1 = pin.GetFitting ‘ – возвращает текущий наконечник, установленный на пин

app.PutMessage "connected to nakonechnik: " & pinfit1 & " of the pin: " & pinids(43)

pin.GetValidFittings strarr ‘– заполняет массив всеми наконечниками установленными на всех

app.PutMessage "All types of ispolsovanniyh nakonechnikov on all pins of device: "

For Each str in strarr

app.PutMessage str

Next

GetFittingIds ( ids) – просто так не работает

If the setting 'Show information for option combinations' under Settings->Connection->Conductors/Wires is checked, the command returns a list of all connector pin terminals

#### Атрибуты

##### Добавление атрибутов device-а

Set con = job.CreateDeviceObject()

cntcons = job.GetConnectorIds(conids)

con.SetId(conids(1))

app.PutMessage con.GetName

Set att = job.CreateAttributeObject()

cntatts = con.GetAttributeIds(attids)

app.PutMessage cntatts

for i=1 to cntatts

att.SetId(attids(i))

app.PutMessage att.GetName & " " & i

next

att.SetId(attids(7))

name0 = att.GetName

name1 = "DeviceNote"

text = "finally"

con.SetId(conids(2))

out = con.AddAttributeValue( name1, text )

Список имён атрибутов

Примечание/линейка - "DeviceNote"

##### Удаление атрибутов device-а

'Удаляем более бесполезный атрибут прям вместе со значением

pin.DeleteAttribute choosenOne

##### Вывод id, name, value атрибута

attcnt = device.GetAttributeIds(ids)

app.PutMessage "Count of attributes: " & attcnt

app.PutMessage " "

Set attr = job.CreateAttributeObject()

'\_\_\_\_ табуляция

Dim maxLength

maxLength = 0

For i = 1 To attcnt

attr.SetId(ids(i))

If Len(attr.GetValue) > maxLength Then

maxLength = Len(attr.GetValue)

End If

Next

'app.PutMessage maxLength

'app.PutMessage ids(i) & " " & attr.GetName & " " & Space(maxLength - Len(attr.GetName))

'\_\_\_\_

app.PutMessage "List of attributes: "

Set attr = job.CreateAttributeObject()

For i = 1 To attcnt

attr.SetId(ids(i))

'app.PutMessage attr.GetName & ids(i)

app.PutMessage ids(i) & " " & attr.GetValue() & " " & attr.GetName

Next

##### Вывод value

Выше было описано, но проблема в том, что в документации нет функции att.GetValue, несмотря на то, что это работает

dev.GetAttributesIds(ids)

att = job.CreateAttributeObject()

att.SetId(id(1))

# получить имя

att.GetValue

или

name = att.GetName

att.GetAttributeValue name - не работает не смотря на то, что это в документации

Вывод: Только GetValue

#### Graph (графические объекты)

Графические объекты в E3Series это графическое отображение электрических элементов, как символ но без атрибутов и всего вообще, только графика. Представляет собой набор полигонов и точек, которые можно создавать самому в COM классах.

Символ содержит в себе информацию об графе, в виде матрицы точек для полигона, который и является графом. И по этой матрице каждый раз символ и строится

##### Группы графики на листе

nGras = sht.GetGraphIds (GraIds)

app.PutMessage nGras

##### Все элементы графики в группе

nGras = Gra.GetGraphIds (GraIds)

app.PutMessage nGras

##### Расположение графики из БД по имени символа

'Получение id символа выбранного девайса

set dev = job.CreateDeviceObject

job.GetSelectedAllDeviceIds idarr

dev.SetId(idarr(1))

set sym0 = job.CreateSymbolObject()

out = dev.GetSymbolIds ( symids )

app.PutMessage(symids(1))

sym0.setid(symids(1))

set sht = job.CreateSheetObject

// получение id листа //

'Расположение графики из БД

set gra = job.CreateGraphObject

out = gra.CreateFromSymbol( sht.getid, 100, 100, 0, 1, 0, sym0.GetType, sym0.GetVersion )

app.PutMessage out

##### Основы обращения с графикой из документации

Set e3 = CreateObject( "CT.Application" )

Set job = e3.CreateJobObject

Set Sht = job.CreateSheetObject

Set Gra = job.CreateGraphObject

Set Sym = job.CreateSymbolObject

shtid = Sht.Search (0, "1")

Sht.SetId shtid

' get all graphic elements

nGras = Sht.GetGraphIds (GraIds)

' create a group of all graphic elements

newgra = Gra.CreateGroup (GraIds)

' get id of group of graphic elements

Gra.SetId GraIds(1)

parentid = Gra.GetParentId

' add a new graphic to a group of graphic elements

newline = Gra.CreateLine (shtid, 300,100,300,200)

newgra = Gra.SetParentId (parentid)

' get all symbols

nSyms = Sht.GetSymbolIds (SymIds)

symid = SymIds(1)

' add a group of graphic elements to a symbol

Gra.SetId parentid

newgra = Gra.SetParentId (symid)

' get all graphic elements of symbol

Sym.SetId symid

nBnds = Sym.GetBoundIds (Bndids)

' ungroup symbol and group of graphic elements

Gra.SetParentId 0

' ungroup all graphic elements

nGras = Gra.GetGraphIds (GraIds)

Gra.UnGroup GraIds

### Создание объектов в проекте

#### Создание листа

Для создания объектов в документации нужно именно в класс этого объекта заходить.

Set sheet = job.CreateSheetObject

Set newsh = job.CreateSheetObject

sheet.Search 0, "Contents"

newsh.Create 0, "NEW\_SHEEEEEEEEET!!!!!", "DINA3", sheet.GetId, 1

newsh.Display

Для подключения к существующему листу в документации через job interface – sub structures:

Set sheet = job.CreateSheetObject

ID:

Dim sheetId

sheetId = newsh.GetId

app.PutMessage sheetId

#### Создание изделия на листе

Dim app

set app = CreateObject("CT.Application")

app.PutMessage "Hello E3!!!"

##### Получение изделий из базы данных

' Запрос выбранных изделий в дереве БД

componentCnt = app.GetDatabaseTreeSelectedComponents(ComponentArray, VersionArray)

cnt = app.GetDatabaseTreeSelectedComponents ( ComponentArray, VersionArray )

ComponentArray contains the selected components from the Database Tree

VersionArray contains the version numbers

cnt number of objects found

' Задание переменных для типа изделия

Dim componentName, componentVersion

componentName = ComponentArray(1)

componentVersion = VersionArray(1)

' Вывод сообщения

app.PutInfo 0, "Выбран компонент: " & componentName & "{" & componentVersion & "}"

##### Создание объектов классов для работы с изделием

' Создание словаря символов для размещения

Dim dictSymbolIds

Set dictSymbolIds = CreateObject("Scripting.Dictionary")

' Создание объекта изделие

Dim dev, devId

Set dev = job.CreateDeviceObject()

' Создание объекта листа

Dim sht, shtId

Set sht = job.CreateSheetObject()

shtId = 0

' Создание объекта символа

Dim sym, symId

Set sym = job.CreateSymbolObject()

' Создание объекта вывода

Dim pin, pinId

Set pin = job.CreatePinObject()

' Создание объекта соединение

Dim con, conId

Set con = job.CreateConnectionObject()

' Создание объекта надпись

Dim txt, txtId

Set txt = job.CreateTextObject()

##### Инициализация класса изделий данными из БД

' Создание изделия

devId = dev.Create("", "", "", componentName, componentVersion, 0)

' Запрос символов изделия

devSymbolCnt = dev.GetSymbolIds(devSymbolIds)

' Запрос выводов изделия

devPinCnt = dev.GetPinIds(devPinIds)

' Идентификатор первого символа

symId = devSymbolIds(1)

Так мы имеем готовое изделие и можем его сохранить куда то

Dim dictSymbolIds

Set dictSymbolIds = CreateObject("Scripting.Dictionary")

dictSymbolIds.Add symId, devId

' Задание координаты для размещения символа

Dim symX, symY, symWidth

symX = symRefSourceX

symWidth = 0

##### Инициализация класса символа (изображения) изделия

sym.SetId symId

##### Размещение

Размещается именно символ, а не device

sym.Place shtId, symX, symY, "0.000000"

##### Очистка

Set txt = Nothing

Set txtId = Nothing

' Очистка

Set con = Nothing

Set conId = Nothing

' Очистка

Set pin = Nothing

Set pinId = Nothing

' Очистка

Set sym = Nothing

Set symId = Nothing

' Очистка

Set sht = Nothing

Set shtId = Nothing

' Очистка

Set dev = Nothing

Set devId = Nothing

' Очистка

Set job = Nothing

Set jobId = Nothing

##### КОД

Dim sheetId

sheetId = newsh.GetId

'Создание изделия

componentCnt = app.GetDatabaseTreeSelectedComponents(ComponentArray, VersionArray)

Dim componentName, componentVersion

componentName = ComponentArray(1)

componentVersion = VersionArray(1)

app.PutInfo 0, "Component: " & componentName & "{" & componentVersion & "}"

Dim dev, devId

Set dev = job.CreateDeviceObject()

' Создание объекта символа

Dim sym, symId

Set sym = job.CreateSymbolObject()

devId = dev.Create("", "", "", componentName, componentVersion, 0)

' Запрос символов изделия

devSymbolCnt = dev.GetSymbolIds(devSymbolIds)

' Идентификатор первого символа

symId = devSymbolIds(1)

Dim symX, symY, symWidth

symX = 200

symY = 200

symWidth = 0

sym.SetId symId

sym.Place sheetId, symX, symY, "0.000000"

#### Создание символа

sht.GetDrawingArea shtxmin, shtymin, shtxmax, shtymax

sym.Load "XUI05K", ""

If( sym.GetId = 0 ) Then

   e3.PutError 1, "Cannot load symbol from database"

   wscript.Quit

End If

sym.GetArea symxmin, symymin, symxmax, symymax

newid = sym.Place( sht.GetId, \_

   (shtxmax+shtxmin)/2 - (symxmax-symxmin)/2, \_

   (shtymax+shtymin)/2 - (symymax-symymin)/2, \_

   "" )

If( newid = 0 ) Then

   e3.PutError 1, "Cannot place symbol"

   wscript.Quit

End If

### Подключение к объектам

Как правило выполняется подключение к объектам так:

Dim activeObj

activeObjID = job.GetActiveObjTypeId ()

Dim obj

Set obj = job.CreateObjType()

objID = obj.SetId(activeObjID)

#### Подключение к листу

OurShtID = job.GetActiveSheetId ()

Dim sht

Set sht = job.CreateSheetObject()

sht.SetId(OurShtID)

#### Подключение к древу

' Запрос активного дерева в проекте

jobActiveTreeId = job.GetActiveTreeId()

' Создание объекта дерева

Dim tree, treeId

Set tree = job.CreateTreeObject()

' Устанавливаем идентификатор активного дерева

treeId = tree.SetId(jobActiveTreeId)

#### Подключение к device-у

'Подключение к девайсу

Set device = job.CreateDeviceObject()

jobDeviceCnt = job.GetAllDeviceIds(jobDeviceIds)

app.PutMessage "Get all devices " & jobDeviceCnt

For Each deviceID in jobDeviceIds

app.PutMessage deviceID

Next

devID = device.SetId(jobDeviceIds(1))

devname = device.GetName

app.PutMessage "Device name: " & devname

#### Подключение к атрибутам девайса

attcnt = device.GetAttributeIds(ids)

Set attr = job.CreateAttributeObject()

attr.SetId(ids(2))

app.PutMessage attr.GetName

#### Подключение к пинам device-a

Set dev = job.CreateDeviceObject()

dev.SetId(ids(1))

Set pin = job.CreatePinObject

pincnt = dev.GetAllPinIds( pinids )

app.PutMessage "pins XS1: " & pincnt & " " & pinids(1)

pin.SetId(pinids(1))

app.PutMessage pin.GetName

### Список объектов в структуре

#### Список листов в проекте

Dim shts, shtids

shts = job.GetAllSheetIds(shtids)

app.PutMessage shts

For Each sht in shtids

app.PutMessage sht

Next

GetAllSheetIds ( ids )

Return the total number and identifiers of sheets in project. Use [Sheet Object](mk:@MSITStore:D:\Zuken\E3.series_2019\help\com.chm::/COM/Classes/SheetInterface/introduction.htm) to handle them.

The determination is dependent on the current setting, which has been set with [job.SetGetterOptionHandlingMode (mode)](mk:@MSITStore:D:\Zuken\E3.series_2019\help\com.chm::/COM/Classes/JobInterface/GetGetterOptionHandlingMode,_SetGetterOptionHandlingMode_-_JobInterface.htm):

* **ret = job.GetAllSheetIds ( ids )** 
  + **ids** the ids of sheets are returned, like specified with [job.SetGetterOptionHandlingMode ( mode )](mk:@MSITStore:D:\Zuken\E3.series_2019\help\com.chm::/COM/Classes/JobInterface/GetGetterOptionHandlingMode,_SetGetterOptionHandlingMode_-_JobInterface.htm) :
    - **mode = 0** (default) the ids of sheets like [job.GetSheetIds(ids)](mk:@MSITStore:D:\Zuken\E3.series_2019\help\com.chm::/COM/Classes/JobInterface/sub-structures/GetSheetIds_idarr.htm) are returned.

The Clipboard sheets are also returned. These can be queried with [sht.IsClipboardPart](mk:@MSITStore:D:\Zuken\E3.series_2019\help\com.chm::/COM/Classes/SheetInterface/IsClipboardPart_-_SheetInterface.htm) .

* + - **mode = 1** only the ids of the sheets are returned, which are active according to the current option settings.
    - **mode = 2**  the ids of all sheets are returned, independent of the current option setting.

The Clipboard sheets are also returned. These can be queried with [sht.IsClipboardPart](mk:@MSITStore:D:\Zuken\E3.series_2019\help\com.chm::/COM/Classes/SheetInterface/IsClipboardPart_-_SheetInterface.htm) .

* + **ret** The return value is the number of ids.

#### Список device-ов на листе

' Запрос активного дерева в проекте

jobActiveTreeId = job.GetActiveTreeId()

' Создание объекта дерева

Dim tree, treeId

Set tree = job.CreateTreeObject()

' Устанавливаем идентификатор активного дерева

treeId = tree.SetId(jobActiveTreeId)

' Словарь для хранения идентификаторов изделий

Dim dictDeviceIds

Set dictDeviceIds = CreateObject("Scripting.Dictionary")

' Запрос изделий выбранных в дереве

jobDeviceCnt = tree.GetSelectedAllDeviceIds(jobDeviceIds)

' Перебор

For devi = 1 To jobDeviceCnt

' Текущий идентификатор

devId = jobDeviceIds(devi)

' Перебод к процедуре работы с изделием

Call DeviceWork(devId, dictDeviceIds)

Next

#### Cписок атрибутов у device-а

attcnt = device.GetAttributeIds(ids)

app.PutMessage "Count of attributes: " & attcnt

app.PutMessage " "

app.PutMessage "List of attributes: "

For Each attid in ids

app.PutMessage attid

Next

#### Cписки разных изделий во всём проекте

Count = GetDeviceIds ( ids )  
Count = GetCableIds ( ids )  
Count = GetConnectorIds ( ids )  
Count = GetTerminalIds ( ids )  
Count = GetBlockIds ( ids )  
Count = GetAllDeviceIds ( ids )  
Count = GetHoseIds ( ids )  
Count = GetTubeIds ( ids )

???

GetConnectionIds  ( ids )  
GetAllConnectionIds ( ids [,flags] )  
GetSelectedConnectionIds ( ids [,flags] )

#### Список всех device-ов в проекте

jobDeviceCnt = job.GetAllDeviceIds(jobDeviceIds)

app.PutMessage jobDeviceCnt

#### Список всех контакторов

Dim cntConn, connIds

cntConn = job.GetConnectorIds(connIds)

app.PutMessage cntConn

### Получить из БД

#### Получение данных выбранного изделия в БД

' Запрос выбранных изделий в дереве БД

componentCnt = app.GetDatabaseTreeSelectedComponents(ComponentArray, VersionArray)

' Проверка

If (componentCnt <> 1) Then

' Вывод сообщения об ошибке

app.PutError 0, "Выбирите только один компонент в дереве БД!"

' Очистка

Set job = Nothing

Set jobId = Nothing

' Процедура выхода

Call ExitScript(False)

End If

' Задание переменных для типа изделия

Dim componentName, componentVersion

componentName = ComponentArray(1)

componentVersion = VersionArray(1)

' Вывод сообщения

app.PutInfo 0, "Выбран компонент: " & componentName & "{" & componentVersion & "}"

#### Поиск компонента по БД

Set cmp = job.CreateComponentObject()

'CCCC - самая актуальная версия изделия

idConn = cmp.Search(activeConnpart, CCCC)

'Активируем изделие

cmp.SetId(idConn)

'Запрашиваем его наименование

connName = cmp.GetAttributeValue("Description")

#### Поиск девайса по БД

### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Triggered scripts

#### Подключение триггера

Set e3 = CreateObject("CT.Application")

e3.AvoidAutomaticClosing

last = e3.SetTrigger("AfterModifySymbol", "C:\Trigger.vbs", 1)

После срабатывания этого скрипта каждый раз на опредленный триггер (как здесь "AfterModifySymbol") будет выполнятся скрипт по адресу "C:\Trigger.vbs"

' Do something

'

' ...

' Get trigger information

active = e3.GetTrigger("AfterModifySymbol", Filename)

' Do something

'

' ...

' Removing the trigger

last = e3.SetTrigger("AfterModifySymbol", "", -1)

#### Информация об удаленном объекте

set app = CreateObject("CT.Application")

args = app.GetScriptArguments

start

Sub Start()

For i = 0 to UBound(args)

MyArray = Split(args(i), ",", -1, 1)

app.PutMessage "Item has been deleted: "

app.PutMessage " GID: " & MyArray(0)

app.PutMessage " ObjectID: " & MyArray(1)

Next

End Sub

## Скрипты

### Создание кнопки на панели в Е3

1) ПКМ по любой панели инструментов -> Настроить… –> доп. Функции -> создать новую -> указываешь путь.

2) Обратно в Настроить –> категории -> доп функциии -> перенести свой скрипт на любую панель.

Отпуск 28 календарных дней (14 вместе0

Декабрь 4 – 31

Основные функции

### Завершение скрипта

Пример выхода из скрипта

' Процедура выхода из скрипта

Sub activeConPartExitScript (ByVal flagSuccessExit)

' Проверка флага

If (flagSuccessExit) Then

' Вывод сообщения об успешной автоматизации

appPutInfoRet = msgbox ("Формирование новых обозначений выполнено успешно!", vbMsgBoxSetForeground+vbSystemModal+vbInformation, "E3.3DRoutingBridge")

Else

' Вывод сообщения о не успешной автоматизации

Call msgbox("В ходе формирования обозначений встретилась ошибка! Формирование не выполнено!", vbMsgBoxSetForeground+vbSystemModal+vbCritical, "E3.3DRoutingBridge")

End If

' Очистка объектов

set choosenOne = Nothing

Set job = Nothing

Set jobId = Nothing

Set app = Nothing

Set appId = Nothing

Set fs = Nothing

Set showMessageMode = Nothing

Set script\_name = Nothing

Set start\_pos\_dimen = Nothing

Set logfile\_name = Nothing

Set logfile\_path = Nothing

' Выход из скрипта

WScript.Quit

End Sub

## Классы E3

При подключении к е3 создается объект класса Application (VBS его находит как “CT.Application” в системе).

Set e3 = CreateObject( "CT.Application" ) – Этот класс является главным и без него не будут работать другие, мы должны прикреплять на него другие.

Set Prj = E3.CreateJobObject - Метод позволяющий получить ссылку на объект класса Job

Примеры

Set E3 = CreateObject("CT.Application") ' (я нашел расшифровку что это "создаем объект "E3"")

Set Prj = E3.CreateJobObject ' (здесь "создаем объект "Проект"")

Set Sht = Prj.CreateSheetObject ' (здесь "создаем объект "Лист"")

Set Att = Prj.CreateAttributeObject

### Message/message box – Application.PutMessage

НЕ ВЫВОДИТ РУССКИЙ ЯЗЫК В СТАНДАРТНОМ ЗАПУСКЕ

Dim e3

set e3 = CreateObject("CT.Application")

e3.PutMessage "This is a simple message"

e3.PutInfo    0, "This is an informational message"

e3.PutWarning 0, "in case of noticeable warnings"

e3.PutError   1, "Here a message box will pop up" – message box

### Пауза

WScript.Sleep(milliseconds)

' Вывести сообщение

MsgBox "Привет!"

' Пауза на 3 секунды (3000 миллисекунд)

WScript.Sleep 3000

' Вывести другое сообщение

MsgBox "Пока!"