

Автоматическое создание листов для чертежей, оформленных в модели

Автор: [Dmitry Rudenko](#)

Тестировалось на:

- nanoCAD СПДС 4.0 сборка 512
- nanoCAD СПДС 4.0 сборка 665
- nanoCAD СПДС 5.0

Необходимые условия для работы скрипта:

1. Рамки листов должны быть начерчены в масштабе 1:100.
Готовые рамки и все скрипты сайта можно [скачать по ссылке](#)
2. Рамки для чертежей должны быть в блоке. Скрипт заточен под 4 вида листов:
 - A1 альбомный (блок "КЖ-A1")
 - A2 альбомный (блок "КЖ-A2")
 - A3 альбомный (блок "КЖ-A3")
 - A4 книжный (блок "КЖ-A4")
3. Должен быть установлен PDFCreator

Алгоритм действий скрипта:

1. Удаляем все существующие в чертеже листы
2. Ищем в модели все блоки рамок, сохраняем их идентификатор и положение
3. Сортируем листы по горизонтали, сортируем по вертикали
4. Создаём настроенные листы для всех рамок в пространстве модели

[Видео-иллюстрация работы скрипта \(2:02\)](#)

Запустить на печать все листы разом можно через Файл - Пакетная печать...

Встроенный PDF-принтер

В архиве присутствует вариант скрипта для печати на встроенный в nanoCAD принтер

Решения для AutoCAD

- [Форум dwg.ru: "C# .net Создание Layout для чертежей, оформленных в модели"](#)

[CreateLayoutsFor_PDFCreator.vbs](#)

```
on error resume next
```

```

Dim ms
Set ms = ThisDrawing.ModelSpace
Dim ut
Set ut = ThisDrawing.Utility

ut.Prompt "Автоматизация печати"

Dim myObj, XSize, YSize, PaperSize, orientation
Dim ppt0, ppt1(2), pt0, pt1
dim ptc, pptc(2)

Dim list(999), Xpos(999), Ypos(999) '999 листов должно хватить, ятд

Dim i, m
m = 0

'удаляем все листы, оставляем только пустой лист "0"
deletelayouts

'прогоняем все объекты модели. если среди них есть блоки с определёнными
названиями, сохраняем номер и точку вставки
for i=0 to ms.count-1
    set myObj = ms.Item(i)
    if (myObj.ObjectName = "AcDbBlockReference") then
        if (myobj.name = "КЖ-A1" OR myobj.name = "КЖ-A2" OR myobj.name =
"КЖ-A3" OR myobj.name = "КЖ-A4") then
            m = m+1 'счётчик листов
            nabor myobj, i 'сохраняем координаты рамок и номер элемента в
списке
        end if
    end if
next

Xsort 'запускаем функцию сортировки листов по положению в модели (слева направо)
Ysort 'сортируем листы по Y

'создаём лэйауты для отсортированного списка листов
for i=1 to m
    set myObj = ms.Item(list(i))
    if (myobj.name = "КЖ-A1") then
        CreateLayout myobj, 840, 594, "A1", 3
    else if (myobj.name = "КЖ-A2") then
        CreateLayout myobj, 594, 420, "A2", 3
    else if (myobj.name = "КЖ-A3") then
        CreateLayout myobj, 420, 297, "A3", 3
    else if (myobj.name = "КЖ-A4") then
        CreateLayout myobj, 210, 297, "ISO A4", 2
    end if
end if
end if

```

```

        end if
next

'удаляем лист "0"
deletezero

ut.Prompt "Готово, проверяй!"

' ===== КОНЕЦ. Дальше функции =====

'сохраняем координаты рамок и номер элемента в списке выбора
Function nabor(myobj, i)
    pt0 = myobj.InsertionPoint
    ppt0 = ut.CreateSafeArrayFromVector(pt0)
    list(m)=i
    Xpos(m)=int(ppt0(0))
    Ypos(m)=int(ppt0(1)/3000)*3000
End Function

'сортируем листы по X
Function Xsort()
    on error resume next
    ut.Prompt "Сортировка по горизонтали"
    Dim a, b
    for a=1 to m-1
        for b=1 to m-1
            if Xpos(b)>Xpos(b+1) then
                temp = Xpos(b)
                Xpos(b) = Xpos(b+1)
                Xpos(b+1) = temp

                temp = list(b)
                list(b) = list(b+1)
                list(b+1) = temp

                temp = Ypos(b)
                Ypos(b) = Ypos(b+1)
                Ypos(b+1) = temp
            end if
        next
    next
End Function

'сортируем листы по Y
Function Ysort()
    on error resume next
    ut.Prompt "Сортировка по вертикали"
    Dim a, b
    for a=1 to m-1
        for b=1 to m-1
            if Ypos(b)<Ypos(b+1) then

```

```

        temp = Xpos(b)
        Xpos(b) = Xpos(b+1)
        Xpos(b+1) = temp

        temp = list(b)
        list(b) = list(b+1)
        list(b+1) = temp

        temp = Ypos(b)
        Ypos(b) = Ypos(b+1)
        Ypos(b+1) = temp
    end if
next
next
End Function

'создаём лэйауты
Function CreateLayout(myobj, XSize, YSize, PaperSize, orientation)
    on error resume next
    'левая нижняя точка - совпадает с точкой вставки блока
    pt0 = myobj.InsertionPoint
    ppt0 = ut.CreateSafeArrayFromVector(pt0)

    'правая верхняя точка
    ppt1(0) = ppt0(0) + XSize * 100 * myobj.XScaleFactor
    ppt1(1) = ppt0(1) + YSize * 100 * myobj.XScaleFactor
    ppt1(2) = 0
    'ut.CreateTypedArray pt1, 5, ppt1(0), ppt1(1), ppt1(2)

    ut.prompt "Создаём лист #" & i

    'создаём новый лист
    Dim olayt
    Set olayt = ThisDrawing.Layouts.add(i)
    ThisDrawing.ActiveLayout = olayt 'переключаемся на новый лист
    'ThisDrawing.MSpace = FALSE 'отключаем модель (хз зачем)

    'отключаем "Масштаб в единицах пространства листа" для корректного
    отображения линий
    ThisDrawing.SetVariable "PSLTSCALE", 0

    'настройки печати
    olayt.ConfigName = "PDFCreator" 'плоттер
    dim Retval, r
    Retval=ut.CreateSafeArrayFromVector(olayt.GetCanonicalMediaNames())
    For Each r In Retval
        if (olayt.GetLocaleMediaName(r) = PaperSize) Then
            olayt.CanonicalMediaName = r 'выбираем формата листа
            Exit For
        End If
    Next

```

```

olayt.PlotRotation = orientation 'поворот 2-книжная 3-альбомная
'olayt.PlotType = 4 'рамка
'olayt.SetWindowToPlot pt1, pt2 'область печати
'olayt.PlotOrigin = "0,0" 'отступы
'olayt.CenterPlot=true 'центрировать
olayt.StandardScale=16 '16 - 1:1, 0 - вписать
olayt.StyleSheet="monochrome.ctb" 'стиль печати - монохром
'цветозависимый (хз как настроить, но вроде по умолчанию стоит)
olayt.PlotWithPlotStyles = true 'учитывать стили печати
olayt.PlotWithLineweights = true 'учитывать вес линий
olayt.PaperUnits = 1 'acMillimeters 'ед. измерения
olayt.RefreshPlotDeviceInfo

'координаты центра видового экрана
pptc(0) = XSize / 2
pptc(1) = YSize / 2
pptc(2) = 0
ut.CreateTypedArray ptc, 5, pptc(0), pptc(1), pptc(2)

'создадим видовой экран
Dim PSVport
Set PSVport = ThisDrawing.PaperSpace.AddPViewport(ptc, XSize,
YSize)

'непечатный слой
Dim layer
Set layer = ThisDrawing.Layers.Add("КЖ_непечатаемый")
layer.Plottable = false
PSVport.layer = "КЖ_непечатаемый"

'масштаб отображения
PSVport.CustomScale = 1 / (100.1 * myobj.XScaleFactor)

'координаты центра видимой зоны
pptc(0) = (ppt0(0)+ppt1(0)) / 2
pptc(1) = (ppt0(1)+ppt1(1)) / 2
pptc(2) = 0
ut.CreateTypedArray ptc, 5, pptc(0), pptc(1), pptc(2)
PSVport.Target = ptc

' ThisDrawing.Regen 1
End Function

'удаляем все листы, оставляем только пустой лист "0"
Function deletelayouts()
on error resume next
Dim lay
Set lay = ThisDrawing.Layouts.add("0")
Dim a, b
b = ThisDrawing.Layouts.count-1
for a = 0 to b

```

```

        if (ThisDrawing.Layouts.Item(b-a).Name <> "Model" and
ThisDrawing.Layouts.Item(b-a).Name <> "0") then
            ThisDrawing.Utility.Prompt "Лист " &
ThisDrawing.Layouts.Item(b-a).Name & " удалён"
            ThisDrawing.Layouts.Item(b-a).delete
        end if
    next
End Function

'удаляем лист "0"
Function deletezero()
    on error resume next
    Dim b
    for b = 0 to ThisDrawing.Layouts.count-1
        if (ThisDrawing.Layouts.Item(b).Name = "0") then
            ThisDrawing.Utility.Prompt "Лист " &
ThisDrawing.Layouts.Item(b).Name & " удалён"
            ThisDrawing.Layouts.Item(b).delete
            Exit for
        end if
    next
End Function

```

swell, vba, layout, печать