

Программирование в nanoCAD: как зарегистрировать свою команду в среде nanoCAD?



Год назад в статье, посвященной использованию скриптов в среде nanoCAD на примере перевода LISP-программы на Visual Basic Script¹, я достаточно подробно рассказал, как организовывать взаимодействие с пользователем, создавать новые объекты, раскладывать их по слоям и вызывать скрипт на исполнение в среде nanoCAD. Надеюсь, вы воспользовались этой замечательной возможностью простейшей автоматизации.

Данная статья продолжает цикл публикаций, связанных с программированием в среде nanoCAD. В частности, пришло время узнать, как вашему скрипту назначить команду и включить ее в интерфейс nanoCAD (пункт меню, кнопка на панели инструментов и сочетания клавиш). Добро пожаловать в мир безграничного программирования под nanoCAD!

Инициализация

Чтобы практически выполнить поставленные задачи, необходимо предварительно предпринять следующие действия:

- установить систему автоматизированного проектирования nanoCAD;
- загрузить любой текстовый редактор для написания скриптов (я использую Notepad++, но вполне можно ограничиться и обычным Блокнотом Windows).

Конечно, вы вправе задать резонный вопрос: "Зачем? Ведь можно все изучить и теоретически!" Однако согласитесь: осваивать новые знания на практике не только продуктивнее, но и гораздо интереснее! Поэтому — поехали...

Собственно интеграция скриптов в среде nanoCAD состоит из трех шагов: регистрация новых команд, загрузка их в среду nanoCAD и привязка к элементам интер-

фейса. Последний шаг необязателен: вы можете вызывать свои команды непосредственно с командной строки. Рассмотрим каждый шаг подробнее.

Шаг 1. Регистрация команды в nanoCAD

Регистрация новой команды в nanoCAD осуществляется посредством специализированного NSF-файла, который, по сути, является XML-файлом. Его структура хорошо разъяснена в разделе "Регистрация скриптов в качестве команд" справочного руководства по ActiveX API. Если говорить кратко, то команда описывается в файле в рамках тегов `<command></command>` и имеет четыре атрибута: "name", "weight", "cmdtype" и "capsdisable". Если с "name" и "weight", надеюсь, все понятно ("name" — это собственно имя нашей команды, которое мы будем набирать в командной строке, а "weight" — вес команды, параметр опциональный, по умолчанию равный 30), то с двумя другими надо познакомиться поближе.

Атрибут "cmdtype" определяет область действия скрипта и может иметь два значения:

0 — скрипт приложения: из скрипта есть доступ к глобальному имени `ThisApplication`;

1 — скрипт документа: из скрипта есть доступ только к глобальному имени `ThisDrawing`.

Атрибут "capsdisable" управляет выбором объектов при запуске скрипта:

0 — сохранить селекцию (`PickfirstSelectionSet`) перед началом команды;

1 — сохранить селекцию после исполнения команды.

Понятно, что имя команды должно быть уникальным — по неосторожности мож-

но переопределить базовые команды платформы. Например, конструкция `<command name="save" weight="30" cmdtype="1" capsdisable="0">`

...
`</command>`

переопределит команду сохранения документа.

В одном NSF-файле можно определять несколько команд. И все это структурируется следующим образом:

`<?xml version="1.0" encoding="utf-8"?>`
`<package>`

`<command name="cmd1" weight="30"`
`cmdtype="1" capsdisable="0">`

...
`</command>`

`<command name="cmd2" weight="30"`
`cmdtype="1" capsdisable="0">`

...
`</command>`

`<command name="cmd3" weight="30"`
`cmdtype="1" capsdisable="0">`

...
`</command>`

...
`</package>`

Здесь, как вы уже, наверное, поняли, определяются три команды: `cmd1`, `cmd2` и `cmd3`.

В рамках тегов `<command></command>` можно располагать еще два тега:

`<description></description>` и

`<script></script>`. Опять же с первым, надеюсь, все понятно: это описание команды. Тег `<script></script>` определяет язык, на котором написан скрипт: либо VBScript, либо JScript. Например, вот так:

`<command name="mycommand"`
`weight="30" cmdtype="1" capsdisable="0">`
`<description></description>`

`<script lang="JScript"><![CDATA[`
...здесь расположен код скрипта...

¹ Денис Ожигин. Первые шаги в мир программирования под nanoCAD: <http://habrahabr.ru/company/nanosoft/blog/86970>

```
]]></script>
</command>
```

Обратите внимание на конструкцию `<![CDATA[]]>`: именно в ней прописывается код скрипта.

Теперь, вооружившись этими знаниями, вы без проблем сможете составить свой первый NSF-файл, обертывающий ваш скрипт в команду и регистрирующий его в среде nanoCAD:

```
<?xml version="1.0" encoding="utf-8"?>
<package>
<command name="hello" weight="30"
cmdtype="1" capsdisable="0">
<description>Классическая программа
"Привет, мир!"</description>
<script lang="JScript"><![CDATA[
ThisDrawing.Utility.Prompt("Привет,
мир!")
]]></script>
</command>
</package>
```

Описанная здесь команда *hello* выводит в командную строку классическое приветствие. Сохраним этот XML-файл с именем *userdata.nsf*. Первый шаг мы сделали.

Шаг 2. Загрузка файла команд при запуске nanoCAD

NSF-файл загружается в среду nanoCAD одноименной командой — *nsf*. Введите ее в командной строке запущенного nanoCAD, укажите путь до сформированного на предыдущем шаге файла *userdata.nsf* — и можно запускать описанные в нем команды, например, созданную на предыдущем шаге команду *hello*.



Простейшая команда *hello* и наш первый результат

Конечно, проделывать подобные шаги каждый раз, когда нам нужна своя команда — удовольствие ниже среднего. Автоматизируем...

Создание команды загрузки NSF-файла

Сначала нужно создать свою команду, которая запускает заданный нами NSF-файл (то есть запускает команду *nsf* с параметрами). Для этого создаем текстовый файл *userdata.cfg* (который понадобится нам в дальнейшем) и описываем в нем команду *load_userdata_nsf*:

```
[\\configman\\commands\\load_userdata_nsf]
weight=i30 |cmdtype=i0 |
intername=sload_userdata_nsf
RealCommandName=snsf
Keyword=suserdata.nsf^MCloseDocument
^MNewDocument^M
```

Этим шагом мы создали свой файл конфигураций, где описали новую команду с внутренним именем *load_userdata_nsf*, которая вызывает команду *nsf* (*RealCommandName=snsf*) с опциями, описанными в строке *Keyword*. Обратите внимание на символ *s*, который идет после знака равно (=) — это обязательный символ, необходимый интерпретатору nanoCAD для работы с CFG-файлом.

Все возможности команд я описывать не буду, иначе статья превратится в книгу (интересующиеся могут самостоятельно изучить файл конфигураций для nanoCAD — *nCad.cfg*), но наиболее интересные и часто используемые опции мы все же рассмотрим.

Например, обратите внимание на следующую конструкцию:

```
Keyword=suserdata.nsf^MCloseDocument
^MNewDocument^M
```

Она означает, что после вызова команды (в данном случае — *nsf*) в командную строку подается следующая команда — *userdata.nsf* + ENTER (т.е. загружается файл *userdata.nsf*), затем — *CloseDocument* + ENTER (т.е. закрывается текущий документ) и, наконец, — *NewDocument* + ENTER (т.е. создается новый документ). Думаю, вы уже догадаетесь, что символы `^M` означают ENTER.

Автоматическая загрузка NSF-файла

Теперь нам требуется при запуске nanoCAD автоматически загружать команду *load_userdata_nsf*. Тут все просто: создаем текстовый файл *userdata.ini* со следующими строками:

; Регистрация файла с командами при запуске nanoCAD

```
[\\DefProf\\Startup\\load_userdata_nsf]
```

Шаг 2 завершен, нам остается положить в папку, в которую установлен nanoCAD, три созданных нами файла — *userdata.nsf*, *userdata.ini* и *userdata.cfg* — и запустить программу. Теперь, если вы все сделали правильно, при запуске в среду nanoCAD будет подгружаться файл *userdata.nsf*, а вслед за этим — регистрироваться новая команда *hello*. Если в NSF-файле при запуске будет прописано больше команд, то все они будут доступны из командной строки.

Шаг 3. Интеграция скрипт-команд с интерфейсом nanoCAD

За интеграцию команд с интерфейсом nanoCAD отвечает уже знакомый нам файл *userdata.cfg*. Этот файл имеет опре-

деленную структуру описания: фактически все элементы интерфейса nanoCAD прописаны в файлах с таким расширением. Боюсь, что в рамках журнальной публикации я не смогу раскрыть все многочисленные варианты и опции этого файла, но минимально необходимый набор их мы все-таки рассмотрим (отметим, что все команды надо прописывать в файле *userdata.cfg*).

Регистрация меню в nanoCAD

Регистрация меню в nanoCAD осуществляется следующим способом:

```
[\\menu\\mycommans] |name=sМои ко-
манды
[\\menu\\mycommans\\hello] |name=sHo-
вая команда HELLO |intername=shello
```

После этого в nanoCAD появится новый пункт меню *Мои команды* с одним элементом *Новая команда HELLO*, вызывающим команду *hello*. Добавляя новые строки, вы будете расширять свой пункт меню другими командами.

Регистрация панели инструментов в nanoCAD

Регистрация панели инструментов в nanoCAD осуществляется следующим способом:

```
[\\toolbars\\mycommans] |InitialVisible=f1
|name=sМои команды
[\\toolbars\\mycommans\\hello] |inter-
name=shello
```

После этого появится новая панель инструментов *Мои команды* с одной новой командой — *hello*. Опция *InitialVisible* отвечает за видимость панели при запуске (*f1* — видима, *f0* — невидима).

Присвоение команде специализированной иконки

Чтобы у команды появилась своя иконка, вам надо еще раз переопределить команду, указав *dll* с ресурсами изображения. В частности, вы можете использовать файл *newbtns.dll*, устанавливаемый вместе с nanoCAD (просмотрите зарегистрированные в *dll* файлы изображений с помощью любого ресурс-менеджера):

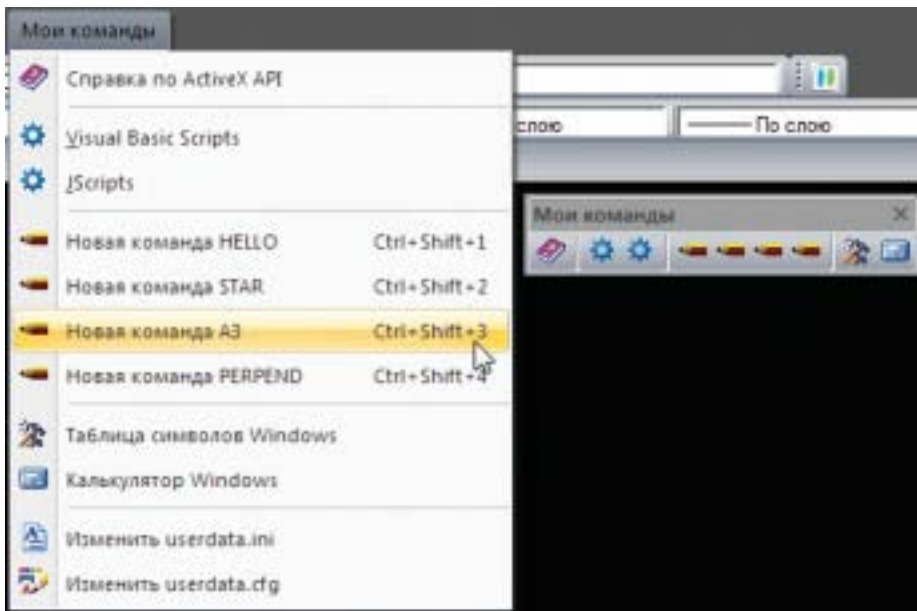
```
[\\configman\\commands\\hello]
weight=i30 |cmdtype=i0 |CapsDisable =
W0xFF
intername=shello
BitmapDll=snewbtns.dll | icon=sPENCIL
```

Горячие клавиши

Команду можно вызывать и с помощью горячих клавиш. Для этого необходимо зарегистрировать их на новую команду следующим образом:

```
[\\Accelerators]
hello=sCtrl+Shift+1
```

Теперь нашу простую команду *hello* можно вызвать сочетанием клавиш CTRL + SHIFT + 1.



Интерфейс nanoCAD, настроенный на новые команды

Финальные шаги: особенности регистрации интерфейса

На данный момент любые изменения интерфейса в nanoCAD применяются только после однократного сброса настроек программы и очистки реестра. Поэтому необходимо добавить в *userdata.ini* следующие строчки:

```
[\Configuration]
ClearRegistry=fl
```

Теперь, запустив программу посредством *ncad.exe* (важно!), мы получим в среде новую команду *hello*, новый пункт меню, новую панель инструментов и горячие клавиши. После однократного запуска останавливаем сброс настроек, закомментировав строку "ClearRegistry=fl" в файле *userdata.ini*:

```
[\Configuration]
;ClearRegistry=fl
```

Почему важно запускать программу напрямую через *ncad.exe*? Дело в том, что штатный ярлык программы при запуске проверяет целостность установки программы, и если мы внесли в интерфейс свои изменения, то ярлык будет стараться восстановить файлы и реестр nanoCAD, запуская установщик Windows. Это не очень удобно, когда вы тренируетесь со скриптами.

Но если ваше приложение оттестировано и зарегистрировано в системе, то в дальнейшем программу можно по-прежнему запускать через ярлык — программа, один раз проверив целостность ключевых файлов, будет запускаться корректно.

Заключение

Итак, мы добились того, что хотели: с помощью трех настроечных файлов (*userdata.nsf*, *userdata.cfg* и *userdata.ini*) получили

возможность расширять функционал бесплатной платформы nanoCAD. При этом новые команды добавляются в NSF-файл, а настройки интерфейса — в CFG-файл. Все вместе это увязывает INI-файл.

Команды, которые вы можете использовать в nanoCAD, описаны в справочном руководстве по ActiveX API, установленном в папку с программой по следующему пути: *%product_dir%\help\api\ncX_devguide.chm*.

Нельзя не отметить, что ActiveX API предоставляет пользователю огромные возможности: небольшие инструменты автоматизации позволяют обходиться без платных приложений, избавляют от рутинных операций и ускоряют работу. На мой взгляд, это must have для студентов и настоятельно рекомендуется для опытных САПР-пользователей.

На нашем форуме вы можете скачать демонстрационные файлы *userdata.nsf*, *userdata.cfg* и *userdata.ini*², что избавит вас от необходимости создавать их самостоятельно.

Кстати, на форуме уже начали появляться полезные команды, которые расширяют функционал nanoCAD: включение/отключение рамки вокруг раstra, построение касательных к двум окружностям и т.д. Там же вы можете выложить свои наработки. Приходите, обсуждайте, делитесь и получайте удовольствие от проектирования!

Денис Ожигин
ЗАО "Нанософт"
E-mail: denis@nanocad.ru

НОВОСТЬ



nanoCAD Механика 3.0 – переход на новую платформу nanoCAD!

Компания "Нанософт" объявила о выходе новой версии популярного продукта — nanoCAD Механика 3.0. Программа предназначена для оформления чертежей в соответствии с ЕСКД, проектирования систем гидropневмоэлементов, зубчатых зацеплений, валов, выполнения инженерного анализа, расчета размерных цепей. Благодаря применению технологии интеллектуального чертежа обеспечены высокая скорость работы и автоматизация операций оформления.

"nanoCAD Механика — это еще один аргумент в пользу платформы nanoCAD. И уже четвертый программный продукт, который перешел на новейшую версию САПР-платформы nanoCAD 3.0, вышедшую в июне этого года, — говорит директор по стратегическому развитию Денис Ожигин. — В новой версии nanoCAD Механики вы получите все преимущества новой платформы: усовершенствованные алгоритмы работы с файлами чертежа AutoCAD, переработанную систему печати, новые возможности работы с видовыми экранами, оптимизированную работу большого числа команд, а также многое другое. Получили развитие и специализированные инструменты nanoCAD Механики для работы с таблицами и табличными отчетами, элементы оформления, редактор формул. Все это дает ощутимые преимущества пользователям наших программ".

nanoCAD Механика 3.0 распространяется по стандартным схемам продаж: абонементной и коробочной. Стоимость абонемента — 8000 руб., стоимость коробочной версии — 22 000 руб.

Владельцы действующих абонементов на nanoCAD Механика могут бесплатно получить серийные номера (в Личном кабинете сайта www.nanocad.ru).

Владельцы коробочной версии с приобретенной подпиской также переходят на новую версию бесплатно. Для владельцев коробочных версий без подписки стоимость перехода составит 6600 руб.

Скачать оценочную версию nanoCAD Механика 3.0 можно с сайта www.nanocad.ru, с официального ftp ЗАО "Нанософт" и через torrent-сеть www.rutracker.org.

Оформить годовой абонемент на право коммерческого использования вы можете на сайте или обратившись к авторизованному партнеру компании "Нанософт".

² <http://forum.nanocad.ru/index.php?showtopic=4026>