



# Triggered Scripts (событийные сценарии) — новое средство автоматизации работы в E<sup>3</sup>.series

Дмитрий Чухран

**Настоящая статья продолжает ряд публикаций, посвященных нововведениям в E<sup>3</sup>.series. В ней рассмотрен новый механизм автоматизации работы пользователя с помощью Triggered Scripts.**

С уверенностью можно сказать, что за последние годы продукт E<sup>3</sup>.series стал полноценным участником рынка программного обеспечения в сегменте САПР для электротехники и АСУТП. На страницах журнала «САПР и графика» регулярно публикуются материалы, в которых описываются принципы работы программы, приводятся обзоры нововведений E<sup>3</sup>.series, рассматриваются вопросы ее внедрения на предприятиях.

Популярность продукт получил благодаря следующим аспектам:

- в основе работы программы лежат фундаментальные законы электротехники;
- развитый и продуманный функционал Редактора базы данных;
- применение скриптов (функциональных макросов), благодаря которым в E<sup>3</sup>.series можно в автоматическом режиме получить выходную документацию практически любого вида;
- поддержка русского языка на уровне ядра программы, а начиная с версии 2008 — полная поддержка Unicode;
- наличие качественной русскоязычной технической поддержки.

Разумеется, это неполный список преимуществ E<sup>3</sup>.series, но именно вышеназванные факторы являются ключевыми.

Особо хотелось бы отметить колоссальные усилия разработчиков, направленные на расширение возможностей программирования скриптов. Проследить это можно хотя бы по числу функций (методов) для программирования, которое увеличивается с каждой версией (табл. 1).

Наряду с автоматизацией получения проектной документации, не менее важным вопросом является автоматизация работы пользователя в E<sup>3</sup>.series. В арсенале программного продукта для этого имеются следующие готовые средства:

- автоматическое наследование атрибутов родительского элемента дочерним элементом;
- автоматическое присвоение кода Места/Устройства изделию, размещенному на листе или в области листа;

- автоматическое присвоение имен (позиционных обозначений) элементам сборки согласно заданному правилу.

Однако, учитывая современные требования к проектированию, этих средств не всегда оказывается достаточно для эффективного решения задач, возникающих перед разработчиками конкретного электротехнического проекта. И вот в программе E<sup>3</sup>.series 2008 появился мощный и, по сути, новый механизм, призванный существенно расширить автоматизацию проектных работ, под названием Triggered Scripts.

Перед тем как дать определение понятия Triggered Scripts и разобраться, в каком контексте разработчики E<sup>3</sup>.series предлагают его использовать, важно рассмотреть определения слов Trigger (триггер) и Script (скрипт, сценарий).

## Триггеры

Возможно, для многих читателей станет открытием, но понятие «триггер» используется во многих областях жизнедеятельности человека: экономике, биологии, схемотехнике, программировании. Несмотря на широту его применения, смысл, который вкладывается в определение этого понятия, во всех случаях сходный. Из всего многообразия наиболее подходящее определение в данном случае следующее: триггер — это средство, обеспечивающее автоматическое выполнение некоторых действий при совершении (возбуждении) определенного события.

## Скрипты

Те из пользователей, кто хоть раз сталкивался с E<sup>3</sup>.series, знают о значимости применения скриптов в программе. По сути, скрипт (от англ. script — скрипт, сценарий, макрос) — это обыкновенный текстовый файл, содержащий набор команд на определенном языке программирования. Скрипты для E<sup>3</sup>.series создаются на VBScript (Visual Basic Script Edition, VBS). Этот язык является урезанной версией языка Microsoft Visual Basic, поэтому для тех, кто программировал на Visual Basic или VBA, язык VBScript окажется хорошо знакомым.

### Дмитрий Чухран

Инженер-проектировщик  
отдела Автоматизации  
Украинского филиала  
«Управляющая компания  
«РусГазИнжиниринг»»,  
г.Северодонецк, Украина.



Таблица 1. Сравнительная динамика числа функций (методов, реализуемых интерфейсами), доступных при программировании скриптов, для разных версий E<sup>3</sup>.series

Имя интерфейса	Версия E <sup>3</sup> .series / число методов		
	2006	2007	2008
IApplicationInterface	92	100	106
IJobInterface	447	489	535
IBundleInterface	33	33	33
IAttributeInterface	12	12	13
IComponentInterface	13	13	15
IConnectionInterface	34	34	34
IDeviceInterface	108	110	116
IExternalDocumentInterface	12	12	27
IFieldInterface	35	35	35
IGraphInterface	60	61	61
IModuleInterface	22	22	22
IModulePortInterface	11	11	11
INetSegmentInterface	42	42	42
IOptionInterface	12	12	12
IOutlineInterface	12	12	12
ISheetInterface	67	76	76
ISignalInterface	21	21	25
ISupplyInterface	16	16	16
IPinInterface	104	104	114
ISymbolInterface	57	64	84
ITestpointInterface	18	18	18
ITextInterface	41	41	46
IVariantInterface	8	8	8
ISlotInterface	6	6	6
IBoardInterface	7	7	7
ILayerInterface	6	6	6
INetInterface	15	15	15
IUserMenuItemInterface	24	24	28
IBinDataInterface	2	2	2
IStructureNodeInterface	18	23	23
ITreeInterface	11	24	24
IDimensionInterface	-	34	34
IDllInterface	-	-	3
Суммарное число методов	1366	1487	1609

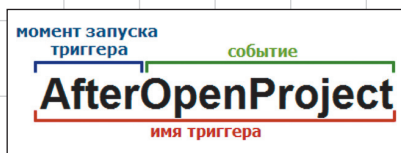


Рис. 1. Принцип именования триггеров

HKEY\_LOCAL\_MACHINE\SOFTWARE\CIM-Team\E3.series\<version>\Triggered-Scripts  
где:  
<version> — число, указывающее версию *E3.series* (на момент написания статьи триггеры поддерживались только *E3.series 2008* — версией 9.0)

Рис. 2. Путь к разделу реестра для регистрации триггеров

Основным назначением скриптов в *E3.series* является автоматическое создание (генерирование) документации на основании данных проекта. Однако благодаря усилиям разработчиков сфера применения скриптов значительно расширилась. Теперь можно не только генерировать документацию, но и автоматизировать работу с базами данных, изделиями, символами, графикой, текстом и настройками программы. Всего сразу и не перечислить, ведь для программирования скриптов в арсенале проектировщика имеется более полутора тысяч функций (см. табл. 1).

## Triggered Scripts

Теперь, зная смысл каждого из слов, составляющих понятие Triggered Scripts, можно дать его полное определение. Triggered Scripts —

это внутренний механизм *E3.series*, обеспечивающий автоматический вызов скрипта при совершении определенного события. При этом вместо скрипта может быть запущена любая внешняя программа. Под событием здесь подразумевается действие, совершаемое пользователем, например открытие проекта,

размещение символа на листе, добавление в проект нового изделия, изменение его атрибутов и т.д.

Далее для упрощения изложения материала Triggered Scripts мы будем называть просто триггерами, а запуск связанного с ним скрипта — запуском триггера.

Тип и количество событий, на которые будут реагировать триггеры, определяются исключительно разработчиком. Для работы с триггерами необходимо определить также моменты их запуска. Предусмотрены следующие моменты:

- Before — триггер запускается до выполнения связанного с ним события;
- After — триггер запускается после выполнения события.

Каждый триггер имеет уникальное имя. Для того чтобы упростить работу с триггерами, их

именование подчинено определенному правилу. Сначала указывается ключевое слово Before или After, то есть момент запуска, а затем — имя события. На рис. 1 показан принцип именования триггеров. Так, по имени AfterOpenProject можно определенно сказать, что триггер будет выполняться всякий раз после события открытия проекта.

Полный список триггеров, реализованных в *E3.series 2008*, приведен в табл. 2.

Информация, связывающая определенный триггер со скриптом, хранится в системном реестре Windows. Для того чтобы зарегистрировать триггер, нужно в разделе реестра (рис. 2) создать строковый параметр (тип параметра REG\_SZ). Его имя должно совпадать с одним из поддерживаемых триггеров (см. табл. 2). А в качестве значения созданного параметра нужно указать абсолютный путь к файлу скрипта (рис. 3).

При запуске система *E3.series* сканирует данный раздел реестра и хранит в оперативной памяти все зарегистрированные в системе триггеры.

Что касается скриптов, то на их программирование не налагается никаких ограничений. В *E3.series*, как известно, вызов скрипта может быть как внутренним, так и внешним. Внутренний вызов — это вызов скрипта, осуществляемый непосредственно из меню программы; внешний вызов — вызов скрипта из операционной системы, например из окна проводника Windows.

Однако при работе с триггерами нужно учитывать несколько особенностей. Во-первых, вызов триггера — это внутренний вызов скрипта, в котором объект *E3.series* создается методом WScript.CreateJobObject(). Во-вторых, разработчики предусмотрели специальные методы, при помощи которых в скрипте можно получить идентификаторы модифицированных элементов. Это методы GetLastAddedItems() и GetLastModifiedItems(). И наконец, в-третьих, триггеры могут возвращать значения (более подробно об этом см. в описании методов GetTriggerReturn() и SetTriggerReturn()).

В случае возникновения ошибок в скрипте *E3.series* перехватывает все сообщения и отображает их в своем стандартном окне вывода информации (рис. 4).

## Управление триггерами

Предложив в версии *E3.series 2008* механизм триггеров, разработчики, к сожалению, не предоставили никаких средств управления ими. То есть, когда необходимо зарегистрировать новый триггер, приходится открывать редактор реестра, находить нужный раздел и вручную создавать параметр, что весьма утомительно. В случае использования большого числа триггеров не исключена ситуация, когда неудобства управления ими могут запросто перечеркнуть все их преимущества.

Таблица 2. Список триггеров, реализованных в *E3.series 2008*

Имя триггера	Момент запуска триггера
AfterChangeComponentCode	После изменения, удаления типа изделия или провода
AfterCreateComponent	После создания нового изделия в Редакторе баз данных
AfterCreateModel	После создания новой модели в Редакторе баз данных
AfterCreateSymbol	После создания нового символа в Редакторе баз данных
AfterDeleteComponent	После удаления изделия в Редакторе баз данных
AfterModifyAttribute	После добавления, изменения или удаления атрибута
AfterModifyDeviceName	После изменения позиционного обозначения (имени) изделия
AfterModifySymbol	После размещения и изменения символа
AfterOpenProject	После открытия файла проекта
AfterPlacePart	После размещения шаблона схемы из базы данных или меню
AfterSaveComponent	После сохранения изделия в Редакторе баз данных
AfterSaveModel	После сохранения модели в Редакторе баз данных
AfterSaveSymbol	После сохранения символа в Редакторе баз данных
AfterUpdateBlock	После добавления и обновления блока
AfterWireConnected	После вставки (назначения) провода
AfterWireDisconnected	После отсоединения провода
BeforeCheckInSheet	Перед проверкой работы в текущем листе (многопользовательский режим)
BeforeCheckOutSheet	Перед проверкой работы вне текущего листа (многопользовательский режим)
BeforeCloseProject	Перед закрытием файла проекта
BeforeDeleteComponent	Перед удалением изделия
BeforeDeleteModel	Перед удалением модели
BeforeDeleteSymbol	Перед удалением символа
BeforeEditComponent	Перед редактированием изделия
BeforeEditModel	Перед редактированием модели
BeforeEditSymbol	Перед редактированием символа
BeforeMoveSymbol	Перед смещением символа
BeforeSaveProject	Перед сохранением файла проекта

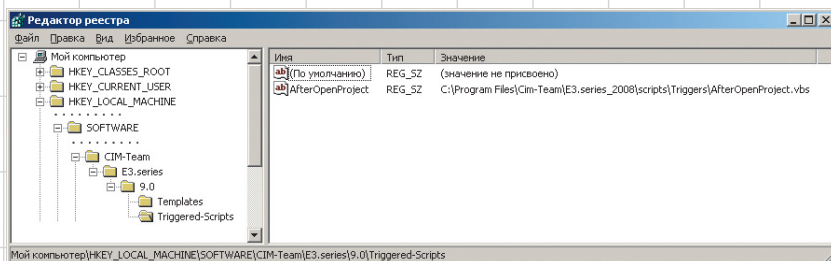


Рис. 3. Регистрация триггера

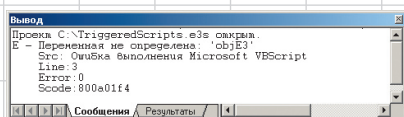


Рис. 4. Перехват сообщения об ошибке в скрипте

Чтобы упростить работу с триггерами, предлагается применять приложение (скрипт) Triggered Scripts Manager (рис. 5). В его основном окне можно выделить такие элементы:

1. **Версия E3.series (Version E3)** — в системе может быть установлено несколько версий программы. Данный список позволяет управлять триггерами для различных версий. Напомним, что на момент написания статьи триггеры поддерживались только 9-й версией программы.
2. **Имя триггера (Trigger Name)** — список содержит имена всех поддерживаемых триггеров для выбранной версии программы. Выберите нужный триггер, чтобы зарегистрировать, изменить его параметры или удалить.
3. **Показать только используемые триггеры (Show Only Used Triggers)** — если выбрана эта опция, то в списке имен триггеров (2) будут присутствовать имена только тех триггеров, которые уже зарегистрированы в системе.
4. **Путь к файлу скрипта (File Path)** — в этом поле указывается абсолютный путь к файлу скрипта или внешней программы.
5. **Редактор скрипта (Script Editor)** — позволяет производить ввод кода скрипта и/или его редактирование. Если в поле (4) указан путь к уже существующему файлу, то в поле редактора отобразится содержимое скрипта.

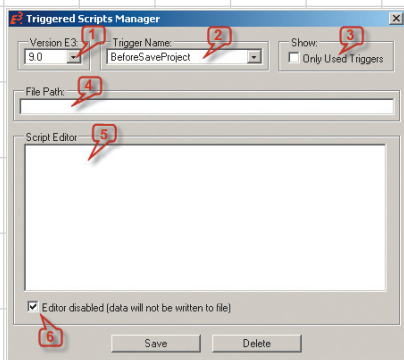


Рис. 5. Triggered Scripts Manager — приложение для управления триггерами

6. **Выключить редактор (Editor Disabled)** — если выбрана эта опция, то поле (5) редактора будет неактивным и внести изменения в код скрипта будет невозможно.

Кнопка **Сохранить (Save)** сохранит данные о триггере, то есть зарегистрирует его в системе, а кнопка **Удалить (Delete)** удалит всю информацию о выбранном триггере.

Более подробная информация о том, как скачать Triggered Scripts Manager и его описание, приводится в разделе «Ссылки».

## Примеры использования

Жестких правил, устанавливающих, где именно применять триггеры, нет. Здесь все зависит от пользователей, ведь разработчики предоставили им достаточно средств. Примеров использования триггеров множество. Например, можно организовать автоматическое резервирование проектов, то есть при каждом открытии проекта будет создаваться отдельная копия файла (с последующей архивацией), которая будет размещаться в определенном разделе на сервере резервирования.

Еще пример. В зависимости от некоторых предустановленных условий (значение атрибута проекта, имя учетной записи пользователя, IP-адрес компьютера и т.д.) возможно изменить параметры текущей сессии программы, например применить нужный файл настроек или создать дополнительное меню.

Чтобы увидеть всю мощь триггеров в E3.series, далее мы рассмотрим более сложный вариант применения Triggered Scripts.

Как правило, системы АСУТП разбиты на отдельные контуры. Всем средствам автоматизации присваиваются позиционные обозначения (позиции), сохраняющиеся во всех документах проекта.

Зачастую позиции средства автоматизации присваиваются на основании позиции контура, в состав которого они входят. Буквенно-цифровые обозначения как контуров АСУТП, так и всех их элементов регламентируются ГОСТ 21.404-85. Порядок расположения буквенных обозначений должен быть следующим: обозначение основной измеряемой величины; обозначение, уточняющее (если необходимо) основную измеряемую величину; обозначение функционального признака. Далее следует нумерация контура. Это может быть как по-

рядковый номер контура, так и любая другая информация (номер объекта, код технологической стадии и т.д.). Например, в состав контура с позицией P1CAHL2311 могут входить устройства: PT2311, PVS2311 и т.д.

Присвоение имени цепи для вывода изделия также можно подчинить определенному правилу. Довольно распространенной практикой является именование цепи по позиции прибора и имени его вывода (контакта). Например, имя цепи, подключенной к выводу 1 устройства PT2311, будет PT2311-1.

Как не сложно заметить, описанные выше правила сводятся к единому алгоритму именования, который без особых сложностей может быть программно реализован в скрипте. Назначив данный скрипт определенному триггеру, можно полностью автоматизировать процесс именования изделий в проекте. Это можно представить следующим образом. При добавлении изделия в проект его имя будет переопределяться в зависимости от позиции контура, в который оно добавлено. Имена цепей выводов этого изделия также будут автоматически формироваться в зависимости от текущего имени изделия. При изменении позиции контура должны будут измениться имена всех изделий, входящих в его состав.

Более подробно о том, как скачать демонстрационное видео работы с триггерами для описанного примера, см. в разделе «Ссылки».

## Заключение

Система E3.series непрерывно развивается и совершенствуется. Но при этом, несмотря на популярную ныне тенденцию создания программного обеспечения по типу «все в одном», в E3.series новые функциональные возможности вводятся из соображений актуальности и необходимости.

При разумном подходе и детальном изучении механизма Triggered Scripts можно существенно автоматизировать работу в программе. Алгоритм автоматизации реализуется программно в скрипте, что является очень гибким подходом, способным удовлетворить самые высокие запросы пользователей. Поэтому здесь, как и в любом деле, необходимо чувство меры: чрезмерная автоматизация может только усложнить процесс проектирования.

Подводя итог, можно сказать, что Triggered Scripts — это новый инструмент, которым необходимо научиться пользоваться. Или если не научиться, то хотя бы ознакомиться с ним, ведь, как известно, «осведомлен — значит вооружен!».

## Ссылки

Интернет-ссылки для скачивания всего предлагаемого к настоящей статье материала находятся на форуме поддержки русскоязычных пользователей E3.series по адресу: <http://forum.e3series.ru/viewtopic.php?f=13&t=68>. ■