



SmartView

**Краткий справочник по
программированию
на языке C#
в среде *ADISRA SmartView***

Область применения документа	3
Информация о C#	3
Почему C#, а не какой-то другой язык?	3
Как я могу изучить C#?	4
Сборка приложения в SmartView	5
Автоматически сгенерированные библиотеки DLL	7
Библиотеки	8
Скрипты SmartView	10
Обзор	10
Расположение скриптов и связи	11
Области применения	12
Скрипты и C#	14
Константы	15
Скрипты и экранные теги	16
Скрипты и теги шаблонов	16
Скрипты и теги	17
Свойства тега	19
Скрипты и экраны	20
Скрипты событий	21
OnOpen	21
OnWhile	21
OnClose	21
Скрипты и экранные объекты	22
Скрипты событий	22
MouseUp	23
MouseDown	24
MouseWhile	24
MouseRightUp	24
MouseRightDown	24
MouseDownDoubleClick	24
SelectionChanged	24

Свойства и функции графического объекта	26
Общие свойства и функции	26
Базовые объекты	27
Изображение (Image)	27
Поле со списком (ComboBox)	27
Текстовое поле (TextBox)	29
Кнопка (Button)	30
Метка (Label)	30
Символ (Symbol)	31
Флажок (CheckBox)	31
Радиокнопка (RadioButton)	32
Геометрические объекты	33
Прямоугольник (Rectangle)	33
Эллипс (Ellipse)	33
Многоугольник (Polygon)	34
Линия (Line)	34
Полилиния (PolyLine)	34
Объекты интерфейса	35
Группа (GroupBox)	35
Вкладка (Tab)	35
Экран (Screen)	36
Меню (Menu)	37
Расширенные объекты	37
.NET Control	37
Тренд (Trend)	37
Тревоги/События (Alarm/Events)	41
Матричная таблица (Matrix Grid)	44
Мультитеговая таблица (MultiTagViewer)	45
Шаблон (Template)	46
Диаграммы (Charts)	47
Диаграмма предок	47

Столбчатая диаграмма (Bar)	48
Линейный график (Line)	48
Круговая диаграмма (Pie)	49
Группировка	49
Группировка (Grouping)	49
Скрипты и библиотека пользовательских функций	51
Скрипты и библиотека системных функций	52
Выпуски документа	53

Область применения документа

Этот документ поможет пользователям SmartView понять С# скрипты, которые широко используются в различных местах приложения. Он также объясняет процесс сборки, выполняемый SmartView и результаты этого процесса. Это является основой для изучения скриптов.

Этот документ обучает только необходимым вещам, которые позволят пользователю писать и понимать скрипты.

Для лучшего понимания были добавлены соответствующие изображения и диаграммы. Рекомендуется первый раз полностью прочитать документ. Также его можно использовать в качестве справочника, используя приведенное выше содержание.

Информация о С#

С# - язык программирования, разработанный Microsoft как часть платформы .NET Framework. Язык высокого уровня, абстрагирует большинство сложных задач. Он известен разработчикам, широко используется и продолжает активно развиваться.

Почему С#, а не какой-то другой язык?

SmartView создавался как гибкий и простой в использовании инструмент, разработанный с помощью С#. Это язык программирования с хорошей производительностью и множеством замечательных функций. Поэтому было естественным использовать С# в качестве языка сценариев (скриптов) внутри SmartView. Пользователи могут реализовать мощный код, используя язык, который хорошо документирован и поддерживается многими другими разработчиками.

Как я могу изучить C#?

Есть много способов выучить это, в зависимости от вашего уровня знаний. Вы можете читать документацию, посещать онлайн-курсы или практические занятия, следовать инструкциям или заниматься самостоятельно. Но на самом деле для написания скриптов для вашего приложения вам не понадобится слишком много знаний, поэтому расслабьтесь, независимо от вашего метода обучения вы сможете создавать очень хорошие скрипты с минимальными усилиями.

Не стесняйтесь использовать ссылки ниже, если вам требуется дополнительная информация и материалы для изучения:

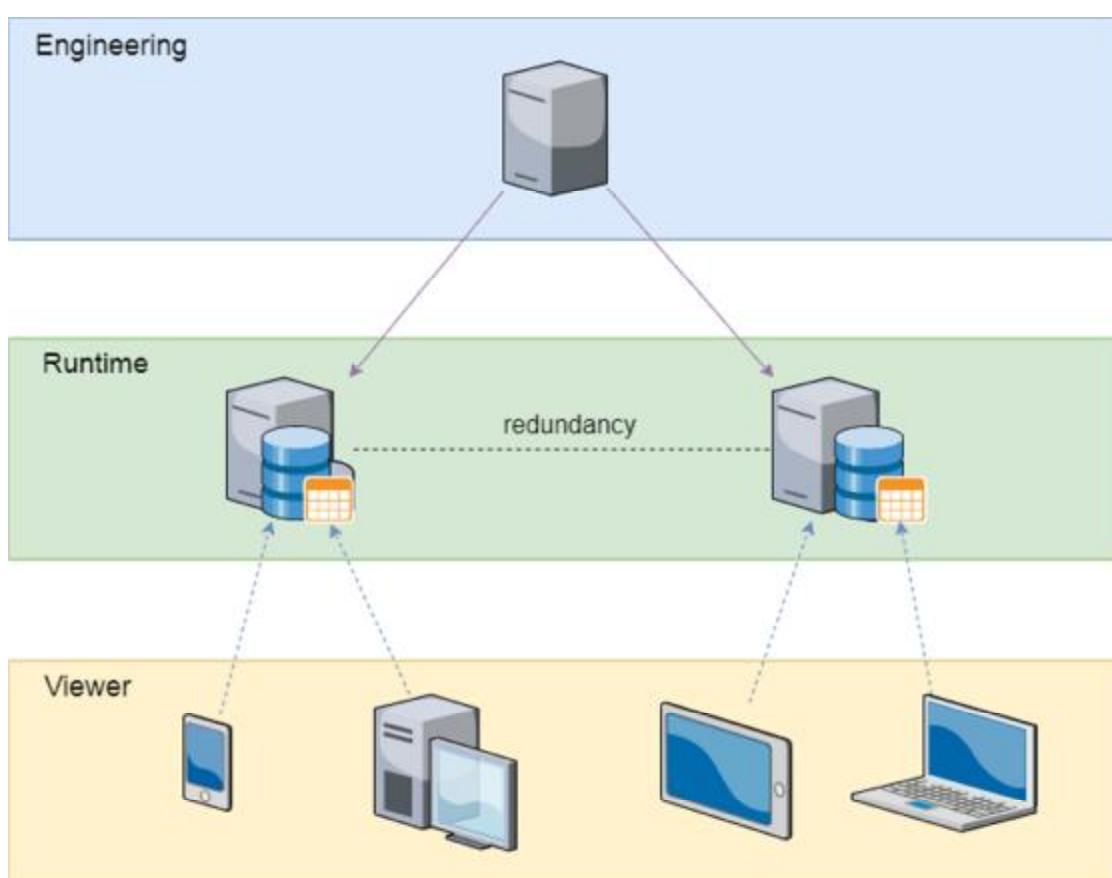
Описание	URL
Microsoft C#. Домашняя страница с учебными пособиями и документацией.	https://docs.microsoft.com/en-us/dotnet/csharp/
Microsoft C#. Руководство по программированию.	https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/
Отличная страница с документацией, руководствами и рабочими примерами.	https://www.w3schools.com/cs/
Еще одна отличная страница, которая включает руководство, советы и подробные объяснения.	https://csharp.net-tutorials.com/
Лидирующая компании в области онлайн-образования.	https://www.codecademy.com/learn/learn-c-sharp

Сборка приложения в SmartView

SmartView можно разделить на 3 основных компонента:

- Среда разработки (Engineering Tool)
- Среда исполнения (Runtime)
- Средство просмотра (Viewer)

Целью данного документа не является подробное объяснение этих компонентов. Но важно упомянуть их для общего представления и целостного усвоения информации.



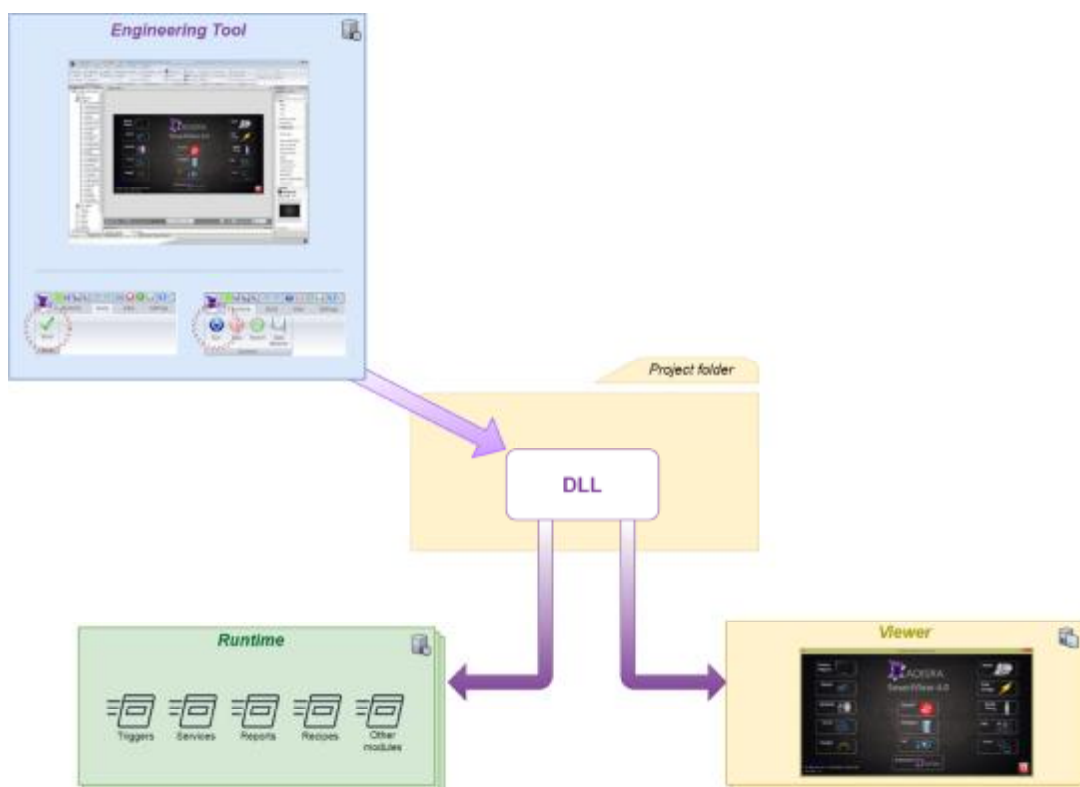
Среда разработки предназначена для создания вашего собственного приложения. Оно может состоять из графики (экранов), служб, триггеров, драйверов связи, клиентов OPC и т.д. В любой момент это приложение может быть собрано или запущено. Оба эти действия требуют полной сборки приложения, поскольку выходные данные этой сборки будут использоваться средой исполнения и средством просмотра.

Но что на самом деле происходит, когда вы собираете приложение?

В результате сборки в папке проекта создается несколько библиотечных файлов DLL. Библиотеки DLL должны быть корректными, иначе сборка завершится неудачно, а в инженерном журнале появится сообщение об ошибке. Например, если в объекте "кнопка" есть ошибки, вы сможете запустить приложение на исполнение, а также средство просмотра, но эта кнопка не будет работать. Среда разработки предоставляет несколько механизмов для проверки вашего скрипта. Есть кнопки проверки, генерирующие сведения о корректности скрипта. В случае ошибок границы скрипта окрашиваются, генерируются сообщения в окне инженерных журналов. Если тег не был создан, система предложит его создать.

Как только библиотеки DLL будут созданы, они будут использоваться средой исполнения и средством просмотра. Эти автоматически сгенерированные библиотеки DLL в основном содержат классы С#, которые представляют экранную графику, триггеры, службы. Если вы разработчик, вы знаете, что эти классы имеют разные пространства имен, и для импорта разных классов вам необходимо добавить к ним оператор using. Но давайте не будем углубляться в технические подробности. Нужно только понимать, что каждый из этих классов, сгенерированных внутри этих DLL, будет иметь заранее определенную область видимости, они смогут достигать некоторых классов, и не будут достигать других.

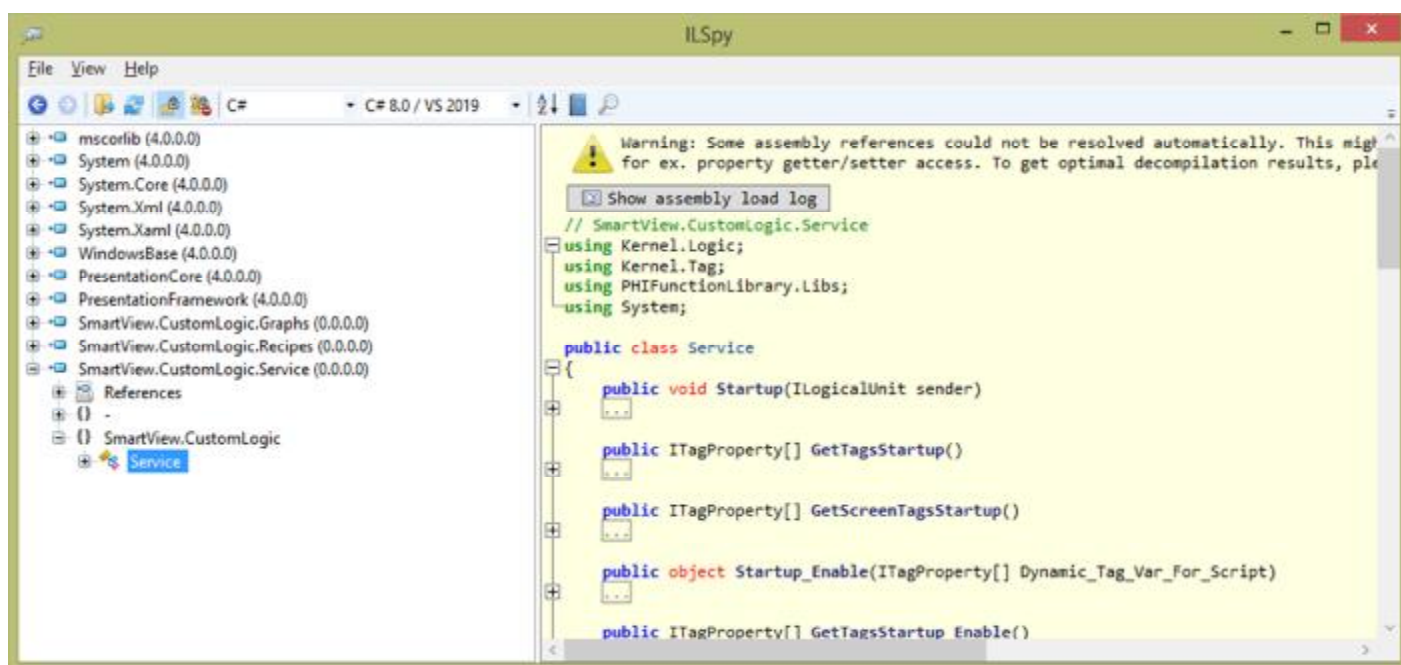
Для более ясного понимания мы рассмотрим сгенерированный класс в следующих разделах. Теперь давайте посмотрим на диаграмму, на которой представлены 3 компонента и библиотеки DLL.



* Среда разработки создает библиотеки DLL, используемые средой исполнения и средством просмотра.

Автоматически сгенерированные библиотеки DLL

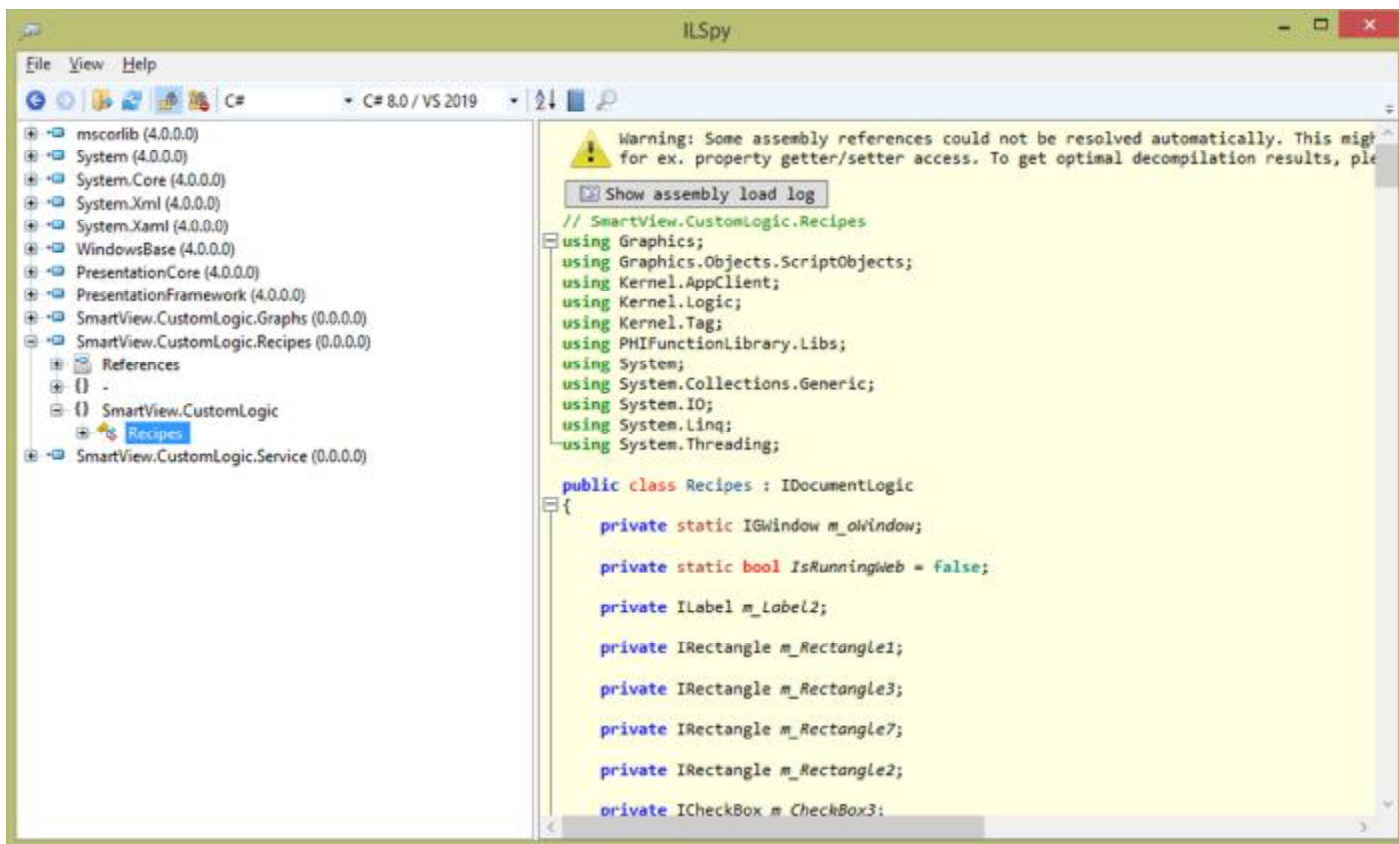
DLL будут созданы внутри папки проекта. Если вы хотите проверить библиотеки Ваших скриптов (служб), то перейдите в папку Services, и вы найдете там соответствующие DLL. Если вы хотите заглянуть в них, есть несколько инструментов, позволяющих увидеть содержимое библиотек DLL. Мы использовали ILSpy при создании документации, вы тоже можете использовать этот инструмент. На первом изображении ниже показан класс служб. Обратите внимание на операторы using, в частности на using Kernel.Tag, который указывает, что в ваших службах могут использоваться теги. Он также добавляет System, представляющее собой пространство имен .NET Framework, которое содержит много часто используемых типов и любые другие подпространства имен. Это важная информация. Вы можете использовать любой другой код С#, как если бы вы разрабатывали в Visual Studio новое программное обеспечение.



* Просмотр SmartView.CustomLogic.Service.dll в ILSpy

На следующем изображении вы увидите графический экран. Для каждого созданного вами экрана будет одна DLL. И нет никакой связи одного экрана с другим (вы не увидите никаких операторов using для других экранных классов). Имя экрана — это имя созданного класса (на изображении - экран рецептов). Тут присутствует больше операторов using: для библиотеки системных функций или System.IO и снова Kernel.Tag.

Все объекты на экране являются приватными свойствами этого класса. Это объясняет, почему скрипт экрана может изменить свойство объекта. Например, щелчок по прямоугольнику 1 (Rectangle1) может изменить ширину прямоугольника 2 (Rectangle2). Все они находятся в одном классе, и знают друг о друге.



* Просмотр SmartView.CustomLogic.Recipes.dll в ILSpy

Библиотеки

Все DLL библиотеки имеют доступ к mscorlib.dll, которая расшифровывается как Microsoft Common Object Runtime Library. Она занимает центральное место.

Когда Microsoft начала работать над .NET Framework, название MSCorLib.dll было аббревиатурой Microsoft Common Object Runtime Library. После того, как ECMA начала стандартизировать CLR и части FCL, MSCorLib.dll официально стала аббревиатурой Multilanguage Standard Common Object Runtime Library.

Ниже приведены некоторые из пространств имен, которые являются частью mscorlib.dll.

```
System
System.Collections
System.Configuration.Assemblies
System.Diagnostics
System.Diagnostics.SymbolStore
System.Globalization
```

```
System.IO System.IO.IsolatedStorage
System.Reflection
System.Reflection.Emit
System.Resources
System.Runtime.CompilerServices
System.Runtime.InteropServices
System.Runtime.InteropServices.Expando
System.Runtime.Remoting
System.Runtime.Remoting.Activation
System.Runtime.Remoting.Channels
System.Runtime.Remoting.Contexts
System.Runtime.Remoting.Lifetime
System.Runtime.Remoting.Messaging
System.Runtime.Remoting.Metadata
System.Runtime.Remoting.Metadata.W3cXsd2001
System.Runtime.Remoting.Proxies
System.Runtime.Remoting.Services
System.Runtime.Serialization
System.Runtime.Serialization.Formatter
System.Runtime.Serialization.Formatter.Binary
System.Security
System.Security.Cryptography
System.Security.Cryptography.X509Certificates
System.Security.Permissions
System.Security.Policy
System.Security.Principal
System.Text
System.Threading
Microsoft.Win32
```

Скрипты SmartView

Обзор

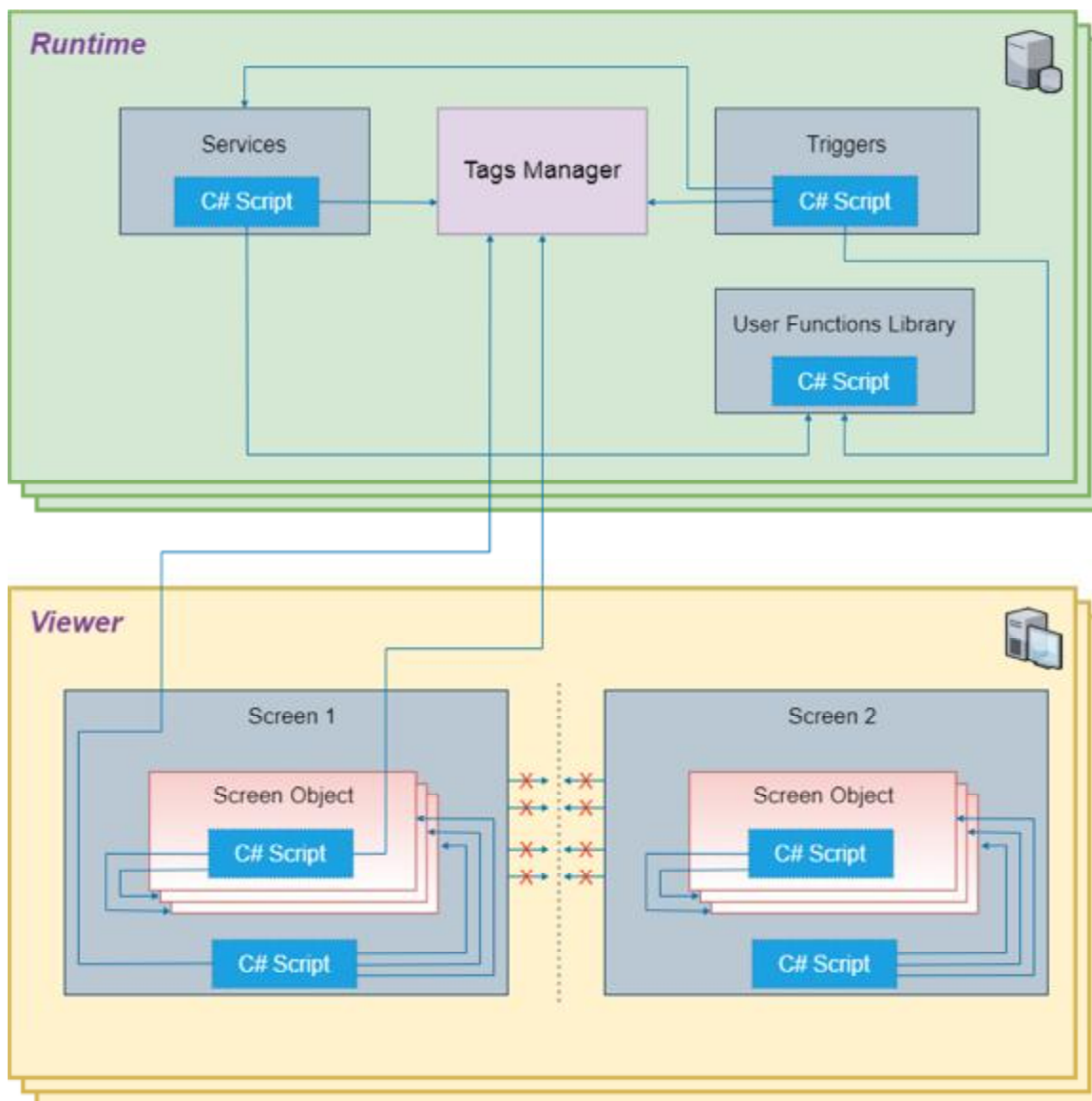
Давайте узнаем побольше о скриптах SmartView. Это не только код C#, это нечто большее, обеспечивающее мощную связь с приложением.

SmartView — это программное обеспечение SCADA (диспетчерский контроль и сбор данных), и важно, чтобы SmartView позволял пользователям манипулировать всеми данными, которые собираются или отправляются в разные места, будь то база данных или ПЛК (программируемый логический контроллер) или любое другое устройство. Все эти данные должны храниться в переменных в SmartView, и эти переменные называются тегами, которые доступны для использования в скриптах.

ПО SmartView было разработано не только для подключения устройств, но и для создания потрясающих интерфейсов для отображения всех этих данных, поэтому оно имеет большую коллекцию экранных объектов от самых простых, таких как метка, текстовое поле, поле со списком, флажок, до самых сложных, таких как матричная таблица (MatrixGrid), мультитеговая таблица (MultiTagViewer), диаграмма, тренд, таблица тревог и т.д. Скрипты также имеют доступ к этим объектам. Это означает, что в одном единственном скрипте вы можете добавить обращение к геометрическому объекту, используя любое значение тегов, добавить новые элементы в поле со списком или добавить новые перья к объекту тренда. Возможности очень широки.

Перед проектированием приложения и написанием скриптов важно понимать архитектуру SmartView и возможности различных модулей. В следующем разделе будут показаны скрипты, которые можно найти в разных модулях, и их области применения, все они представлены на диаграмме.

Расположение скриптов и связи



На приведенной выше диаграмме показаны положения скриптов и связи с другими модулями. Все графические скрипты (экран, объекты и шаблоны) выполняются в средстве просмотра. Службы, триггеры и пользовательские функции выполняются в среде исполнения. Все они имеют доступ к менеджеру тегов. Если тег меняет значения, все модули будут уведомлены и будут использовать одно и то же значение тега. На диаграмме также показано, что между двумя разными экранами нет связи. Таким образом, скрипт экрана 1 нельзя использовать для изменения свойства экранного объекта на экране 2.

Есть отдельные частные случаи, когда скрипт может быть выполнен в среде исполнения или в средстве просмотра, это зависит от местоположения скрипта. Например, функции из библиотеки пользовательских функций. Если они вызываются из графического скрипта, то будут выполняться в средстве просмотра, но, если они вызываются из службы, они будут выполняться в среде исполнения. Также стоит упомянуть библиотеку системных функций. Некоторые ее функции всегда запускаются в среде исполнения (например, SVTags и SVRecipe). Как видите, менеджер тегов находится в среде исполнения, поэтому всякий раз, когда скрипту требуется значение тега или необходимо обновить тег, будет происходить обмен данными между средством просмотра и средой исполнения.

Области применения

Есть много разных мест, где могут быть написаны скрипты в SmartView. Пожалуйста, посмотрите список ниже и обратите внимание на комментарии.

Расположение	Выполнение	Доступ к тегам	Изменение экранных объектов	Комментарий
Скрипты экранных объектов (Screen Object Scripts)	Viewer	Да	Да	Скрипты экранных объектов выполняются в средстве просмотра. Они могут изменять свойства других экранных объектов, но те должны находиться на том же экране. Если они вызывают системную или пользовательскую функцию, она будет выполняться в средстве просмотра, только если не вызывается системная функция, которая изменяет тег (например, SVTags) или рецепт (например, SVRecipe).
Экранные скрипты (Screen Scripts)	Viewer	Да	Да	Может изменять только экранные объекты, расположенные на одном экране.
Службы (Services)	Runtime	Да	Нет	Службы всегда выполняются в среде исполнения.
Триггеры (Triggers)	Runtime	Да	Нет	Триггеры всегда выполняются в среде исполнения. Они могут также запускать службы.
Библиотека пользовательских функций (User Functions Library)	Viewer / Runtime	Нет	Нет	Функция не может изменять значения тегов или использовать их напрямую. Но вы можете передать теги в функцию как параметры и использовать возвращаемые значения для изменения тега. Может выполняться в средстве просмотра или в среде исполнения, в зависимости от того, в каком модуле

				вызывается. Если вызывается в экранном скрипте или экранного объекта, будет выполняться в средстве просмотра. Если вызывается в службе, будет выполнена в среде исполнения.
Объект шаблона (Template Object)	Viewer	Да	Да	Любой объект, добавленный к объекту пользователя, может изменять публичные свойства соседнего объекта в том же шаблоне. Объекты, входящие в состав шаблона, не могут напрямую изменять объекты вне этого шаблона, но они могут изменять теги, которые используют внешние объекты. У них также могут быть внутренние теги, которые можно использовать в скриптах.

В следующих разделах вы более подробно познакомитесь с областями применения и особенностями скриптов.

Скрипты и С#

Редактор скриптов определяет конструкции и ключевые слова языка С# и подсвечивает их определенными цветами.

```
1 // C# Language is identified. Comments can be used and are colored
2
3 for (int i = 0; i < 10; i++)
4 {
5     if (i % 2 == 0)
6     {
7         SVApplications.Output(i.ToString() + " is even");
8     }
9 }
```

При каждом сохранении код компилируется (например, скрипт службы).

The screenshot shows the SmartView interface for a service named 'Services1'. The 'Service Script' section contains the following code:

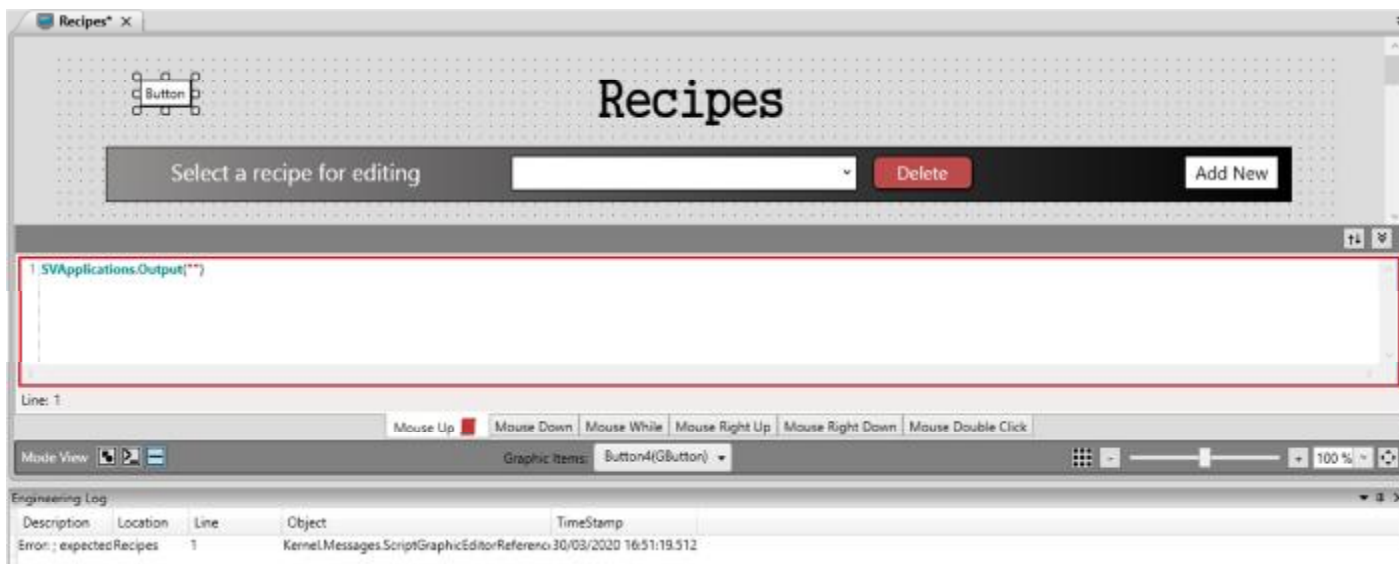
```
1 // C# Language is identified. Comments can be used and are colored
2
3 for (int i = 0; i < 10; i++)
4 {
5     if (i % 2 == 0)
6     {
7         SVApplications.Output(i.ToString() + " is even");
8         SVApplications.Output("This line is missing the semicolon")
9     }
10 }
```

The code is highlighted with a red rectangle, indicating a syntax error. The 'Engineering Log' at the bottom shows the following message:

Description	Location	Line	Object	TimeStamp
; expected	Services1	8	Kernel.Messages.ScriptEditorReference	30/03/2020 16:46:09.531

Красный прямоугольник показывает, что в скрипте есть ошибка. Внизу изображения видно сообщение об ошибке в инженерном журнале.

На следующем изображении пример скрипта экранного объекта. Ошибка в скрипте кнопки. На этот раз ошибка возникает, как только вы переключаете фокус со скрипта (даже до его сохранения).



Константы

Скрипты SmartView содержат некоторые константы, которые также могут использоваться разработчиками. Взгляните на приведенную ниже таблицу с примерами и описанием.

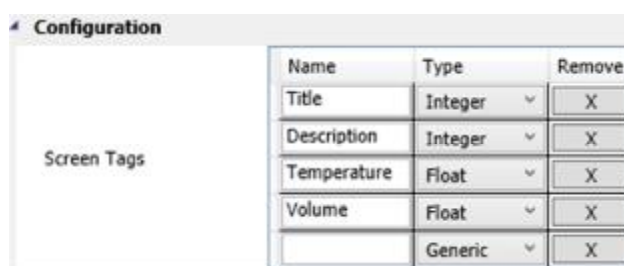
Константа	Пример	Описание
IsRunningWeb	<pre>if (IsWebRunning) { // Выполняется что-нибудь для Web }</pre>	Проверяет, находится ли текущее средство просмотра в Web или нет. Эта константа имеет логический тип, возвращается true в случае, если она находится в Web средстве просмотра.

Скрипты и экранные теги

SmartView позволяет создавать экранные теги. Область действия этих тегов - локальная, ограничена экраном, их можно использовать в объектах экрана или скриптах. Поскольку темой данного документа являются скрипты, давайте взглянем на них.

Экранные теги позволяют создать общий экран, который можно использовать с разными входными параметрами, не ограничивая вас в использовании одних и тех же глобальных тегов.

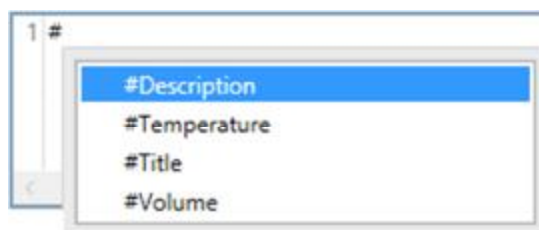
Их можно создать в таблице свойств экрана, как показано на изображении ниже.



The screenshot shows a 'Configuration' dialog box with a 'Screen Tags' section. It contains a table with the following data:

Name	Type	Remove
Title	Integer	X
Description	Integer	X
Temperature	Float	X
Volume	Float	X
	Generic	X

Экранные теги можно использовать с помощью символа # в скриптах. Как только вы введете знак #, в помощь вам в всплывающем окне будут перечислены все экранные теги.

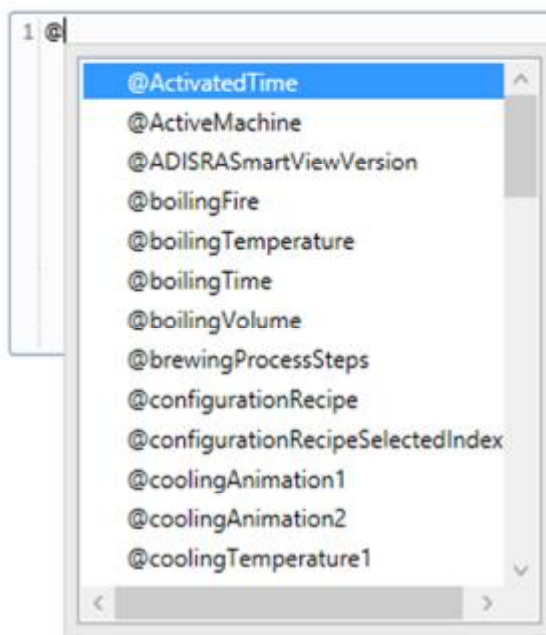


Скрипты и теги шаблонов

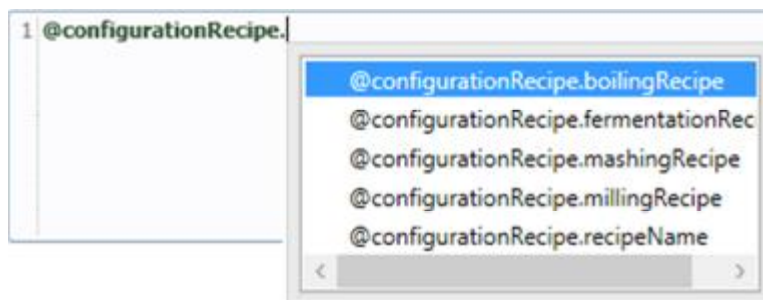
В шаблонных объектах применяется такой же принцип. Как только вы создадите новый шаблонный объект, у вас появится возможность создавать локальные теги объекта в его таблице свойств, к которым также можно будет получить доступ с помощью знака #. Это делает шаблонные объекты невероятно гибкими и универсальными, что позволяет использовать их с разными входными параметрами.

Скрипты и теги

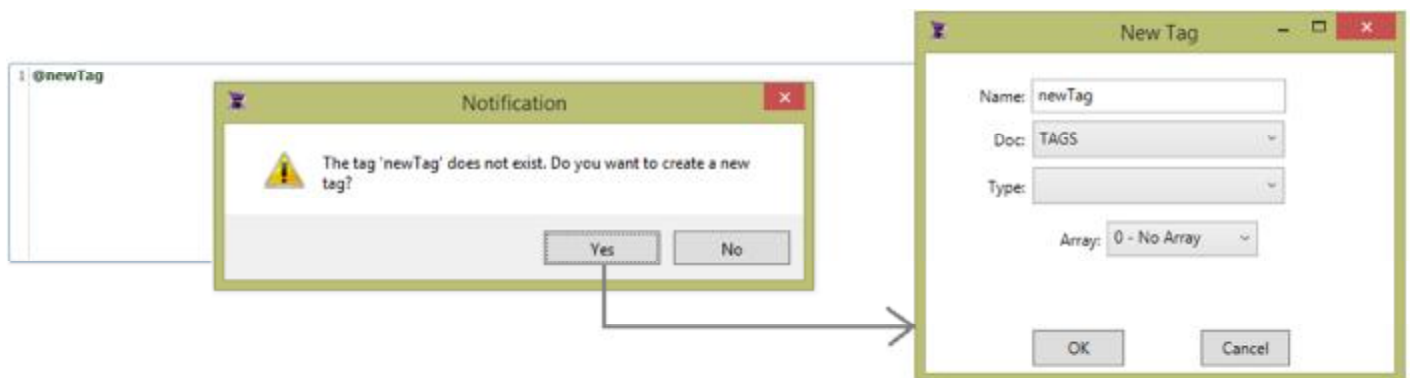
В SmartView все теги, созданные в приложении, можно использовать в скриптах. Вам нужно только ввести символ @, и в скрипте отобразятся все доступные теги.



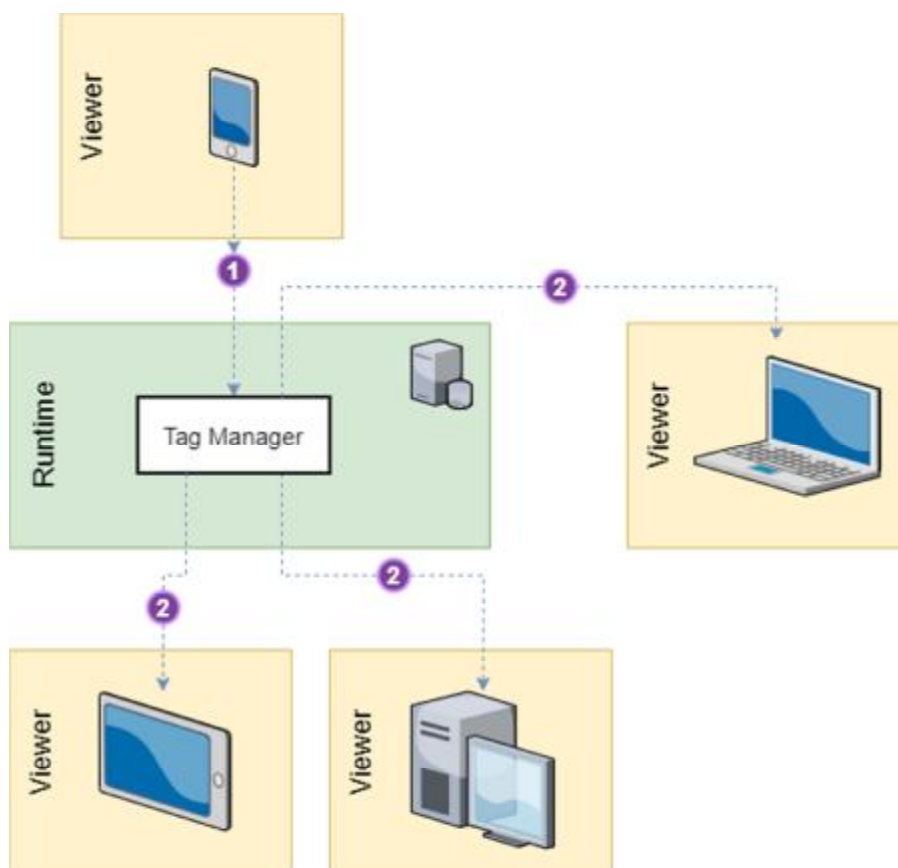
Если тег является типом данных, автодополнение поможет вам найти вложенный тег, как в примере ниже.



Если тег не существует, SmartView предложит его создать. Вы сможете выбрать тип данных и документ-местоположение.

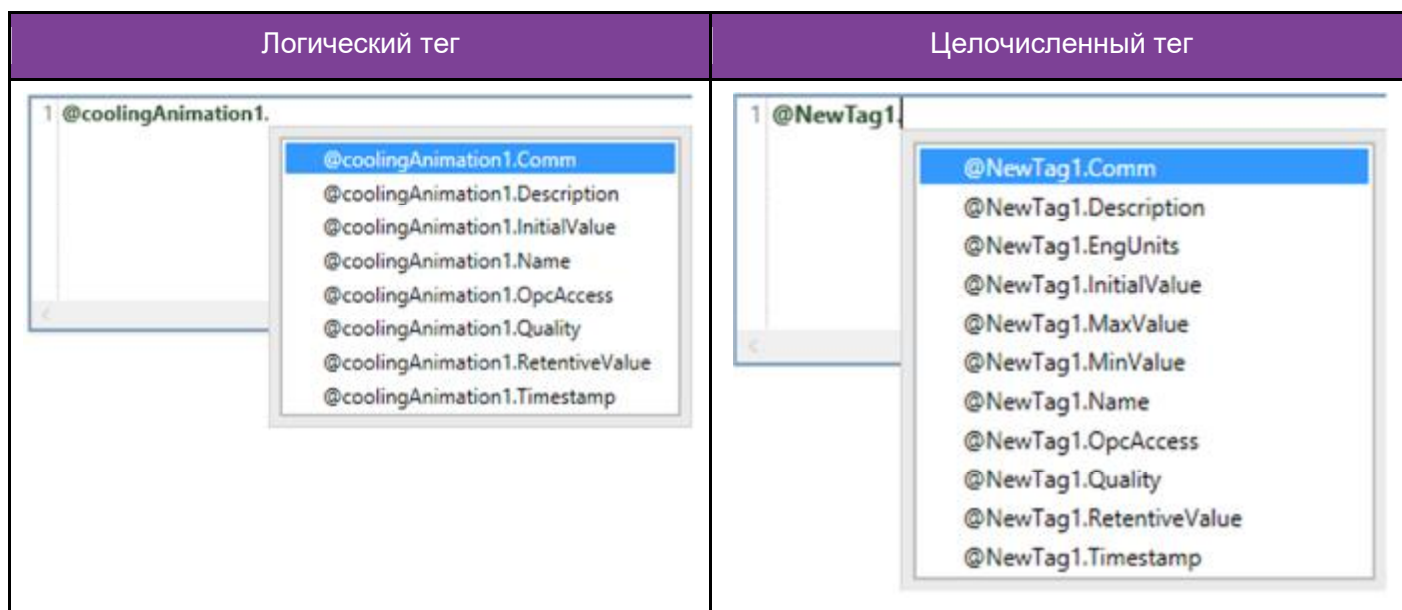


Всякий раз, когда значение тега изменяется в скрипте, это изменение отправляется в менеджер тегов среды исполнения, тот в свою очередь отвечает за уведомление всех средств просмотра.



Свойства тега

В скриптах можно не только изменить значение тега, но и прочитать некоторые его свойства. Свойства, доступные для использования в скриптах, зависят от типа тега. В качестве примера вы можете увидеть свойства логического и целочисленного тегов на изображениях ниже.



В следующей таблице вы увидите все доступные свойства, сгруппированные по типам тегов, варианты доступа (чтение или запись) и описание.

Свойство	Тип	Чтение / Запись	Описание
Comm	Все	Чтение	Указывает, задействован ли тег в какой-нибудь связи. Это может быть "нет", симуляция, драйвер, клиент OPC и т.д. Тег может быть связан только с одним типом связи.
Description	Все	Чтение	Получить описание тега.

EngUnits	Integer, Float	Чтение	
InitialValue	Все	Чтение	Получить начальное значение тега.
MaxValue	Integer, Float	Чтение	
MinValue	Integer, Float	Чтение	
Name	Все	Чтение	Получить имя тега.
OpAccess	Все	Чтение	Получить значения доступа OPC к тегу (чтение, запись или чтение/запись).
Quality	Все	Чтение	Получить качество тега. 192 - хорошее качество. Иное – ошибка.
RetentiveValue	Все	Чтение	Проверить, установлена ли для тега сохраняемость (значение true/false). Если да, значение тега будет сохранено и восстановлено при перезапуске приложения.
TimeStamp	Все	Чтение	Последнее время изменения значения тега. Формат - дата и время.

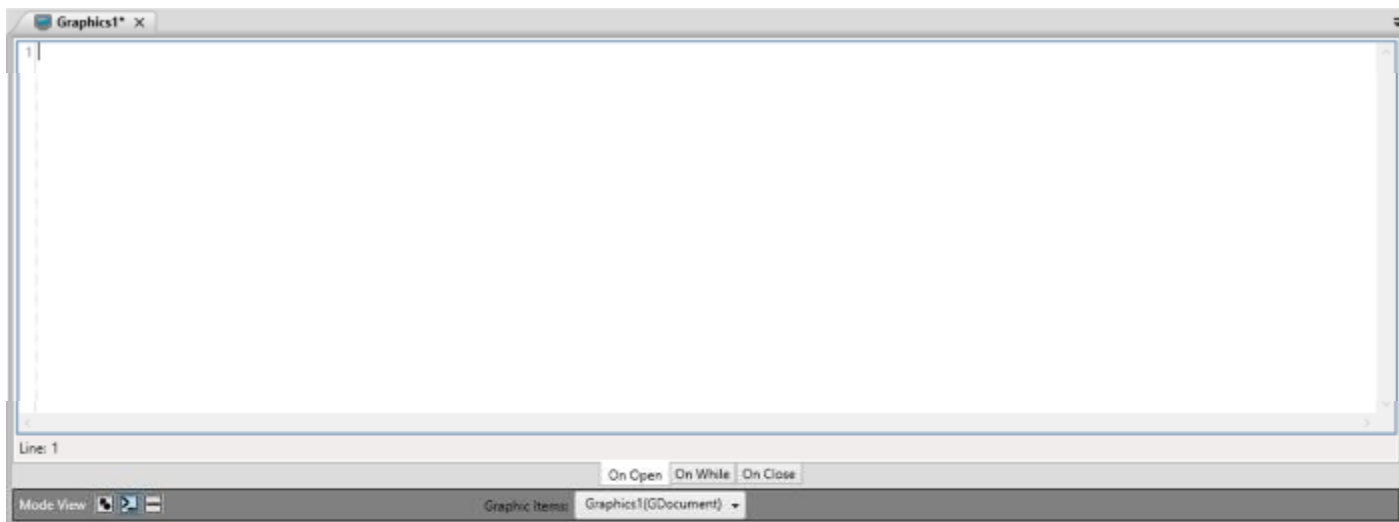
Скрипты и экраны

Для каждого экрана существует 3 события, к каждому из которых привязан соответствующий скрипт. Чтобы получить доступ к окну скриптов, вы можете выбрать либо режим просмотра скрипта, либо разделенный просмотр, при этом должен быть открыт экран. Найдите компонент режима просмотра (Mode View) и выберите желаемый режим отображения.



* Выбор режима просмотра

На следующем изображении показано окно скрипта в режиме просмотра скрипта.



Как ранее упоминалось, тут можно использовать любой код C#, а также теги с символом @, пользовательские функции, системные функции, управлять свойствами экранных объектов. Эти скрипты могут состоять из множества строк, но вы не можете объявлять тут функции или классы. Представляйте данный скрипт как функцию, уже объявленную без параметров и возвращаемого значения.

Скрипты событий

В скриптах экрана можно увидеть 3 типа событий. OnOpen, OnWhile и OnClose

OnOpen

Выполняется один раз при каждом открытии экрана. Не имеет значения каким образом открывается экран: через объект экрана (интерфейс объекта), системной функцией SVGraphics или любым другим способом.

OnWhile

Выполняется постоянно циклически пока открыт экран. Период исполнения может быть небольшим (около 300 мс) и зависит от того, насколько нагружено средство просмотра. Обычно используется для анимации на экране или проверок состояний тегов (хотя вы можете использовать триггер, если ожидаете действие при определенных значениях тега).

OnClose

Выполняется один раз при каждом закрытии текущего экрана.

Скрипты и экранные объекты

В SmartView доступно множество объектов для размещения на экранах. Они включают разные категории, такие как геометрические объекты, базовые объекты, объекты интерфейса, расширенные объекты и диаграммы. Ниже описаны свойства и функции, которые можно использовать для управления этими объектами.

Важно понимать, что свойства и функции экранных объектов могут использоваться только экранными скриптами, причем того экрана, на котором размещен объект.

Для доступа к свойствам и функциям вам понадобится имя объекта. По умолчанию SmartView уже присваивает имя объекту при размещении, но вы можете изменить имя на свое.



* Панель графических объектов

Скрипты событий

В скриптах экранных объектов можно увидеть 6 общих типов событий. MouseUp, MouseDown, MouseWhile, MouseRightUp, MouseRightDown и MouseDoubleClick. Для объекта ComboBox есть еще одно событие SelectionChanged. Рассмотрим каждое событие более подробно.

Есть два способа, как получить доступ к скрипту события объекта.

- 1) Вы можете выбрать объект и изменить режим просмотра на просмотр скрипта (вы также можете дважды щелкнуть объект, включится этот режим просмотра) или выбрать разделенный вид.

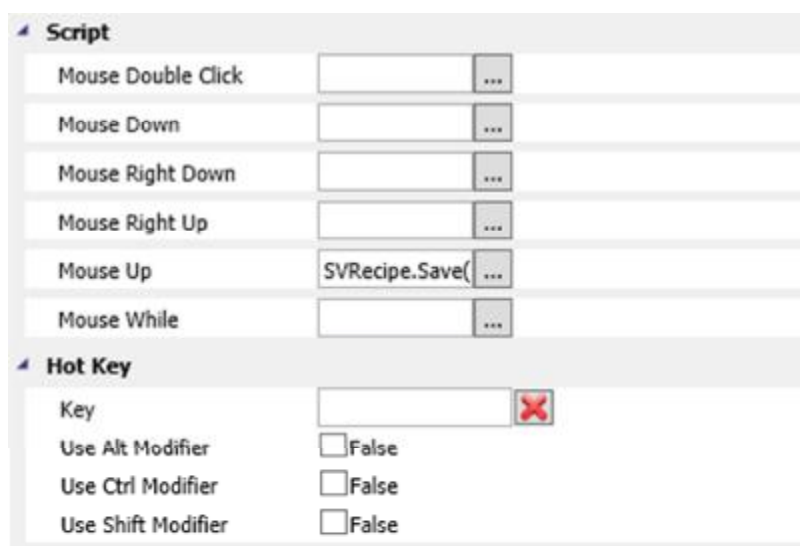


* Выбор режима просмотра

На следующем изображении показаны события объекта прямоугольник (Rectangle) в режиме просмотра скрипта. Те же события характерны для большинства объектов.



- 2) Доступ к скриптам событий можно получить с помощью таблицы свойств (Property Grid) при выборе графического объекта, как изображено ниже.



** Доступ к скриптам объекта с помощью таблицы свойств*

MouseUp

Выполняется каждый раз, когда пользователь отпускает левую кнопку мыши над выбранным объектом. Это событие обычно используется для действия, которое может быть наиболее важным, при этом существует некоторая возможность отменить его. Если пользователь, например, нажимает кнопку, но пока она нажата, он решает не завершать действие, он может увести курсор мыши от кнопки, а затем отпустить ее. Таким образом, событие не будет выполнено. Событие также может запускаться настроенной горячей клавишей (HotKey).

MouseDown

Выполняется каждый раз, когда пользователь нажимает левую кнопку мыши над выбранным объектом. Событие также может запускаться настроенной горячей клавишей.

MouseWhile

Выполняется, пока над объектом нажата левая кнопка мыши, между событиями MouseDown и MouseUp.

MouseRightUp

Выполняется каждый раз, когда пользователь отпускает правую кнопку мыши над объектом.

MouseRightDown

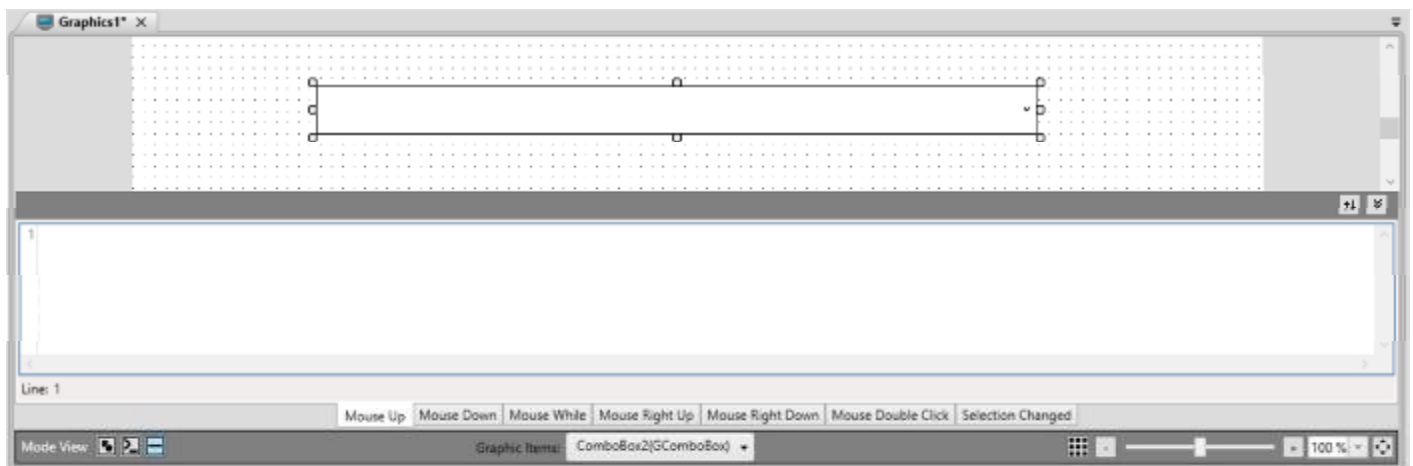
Выполняется каждый раз, когда пользователь нажимает правую кнопку мыши над объектом.

MouseDoubleClick

Выполняется при двойном щелчке левой кнопкой мыши над объектом.

SelectionChanged

Выполняется при изменении выбора элемента ComboBox. На рисунке ниже список событий ComboBox.



Посмотрите на список объектов, их категорию и наличие у них событий клика.

Имя	Категория	Скрипты событий
Изображение (Image)	Базовый объект	Да
Поле со списком (ComboBox)	Базовый объект	Да
Текстовое поле (TextBox)	Базовый объект	Да
Кнопка (Button)	Базовый объект	Да
Метка (Label)	Базовый объект	Да
Символ (Symbol)	Базовый объект	Да
Флажок (CheckBox)	Базовый объект	Да
Радиокнопка (RadioButton)	Базовый объект	Да
Прямоугольник (Rectangle)	Геометрический объект	Да
Эллипс (Ellipse)	Геометрический объект	Да
Многоугольник (Polygon)	Геометрический объект	Да
Линия (Line)	Геометрический объект	Да
Полилиния (PolyLine)	Геометрический объект	Да
Группа (GroupBox)	Объект интерфейса	Да
Вкладка (Tab)	Объект интерфейса	Нет
Экран (Screen)	Объект интерфейса	Нет
Меню (Menu)	Объект интерфейса	Да
.NET Control	Расширенный объект	Нет
Тренд (Trend)	Расширенный объект	Нет
Тревоги/События (Alarm/Events)	Расширенный объект	Нет
Матричная таблица (Matrix Grid)	Расширенный объект	Да
Мультитеговая таблица (MultiTagViewer)	Расширенный объект	Да
Шаблон (Template)	Расширенный объект	Да

Столбчатая диаграмма (Bar)	Диаграммы	Нет
Линейный график (Line)	Диаграммы	Нет
Круговая диаграмма (Pie)	Диаграммы	Нет
Группировка (Grouping)	Группировка	Да

Свойства и функции графического объекта

В этом разделе вы узнаете, какие объекты содержат скрипты, а также обо всех свойствах и функциях, доступных для каждого объекта.

Как упоминалось ранее, свойства и функции объекта могут использоваться только внутри графического экрана, которому принадлежит объект. Их могут использовать скрипты экрана или любой другой объект, находящийся в том же скрипте.

Некоторые свойства наследуются и являются общими для всех объектов (или, по крайней мере, для большинства объектов). Эти свойства будут перечислены здесь, чтобы избежать повторов.

Общие свойства и функции

1. Общие свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
isEnabled	bool	Чтение / Запись	Включает или отключает объект.
isVisible	bool	Чтение / Запись	Показывает или скрывает объект.
Width	double	Чтение / Запись	Получает или задает ширину объекта.
Height	double	Чтение / Запись	Получает или задает высоту объекта.
Left	double	Чтение / Запись	Получает или задает левую сторону объекта.

Top	double	Чтение / Запись	Получает или задает верх объекта.
-----	--------	-----------------	-----------------------------------

2. Свойства вращения

Имя	Тип	Доступ (Чтение / Запись)	Описание
Angle	double	Чтение / Запись	Получает или задает угол объекта.

Базовые объекты

Изображение (Image)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (чтение / запись)	Описание
Source	string	Чтение / Запись	Получает или задает источник изображения. Исходное изображение должно находиться в папке Images проекта. Нужно передать имя файла изображения. (Например, Image1.Source = "ADISRA.png");

2. Функции

Нет доступных функций.

Поле со списком (ComboBox)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
Items	List<string>	Чтение / Запись	Получает или задает список элементов ComboBox.
SelectedIndex	int	Чтение / Запись	Получает или задает выбранный элемент ComboBox с помощью индекса.
SelectedText	string	Чтение / Запись	Получает или задает выбранный элемент ComboBox, используя текст.
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта элементов ComboBox.

2. Функции

Имя	Возвращает	Описание
InsertItem(string item)	void	Вставляет новый элемент в ComboBox. Это должна быть строка, переданная как параметр.
InsertItemsFromTag(string tag)	void	Вставляет элементы из тега. Это может быть простой тег или массив тегов. Значения будут преобразованы в строки.
InsertItemsFromTag(string tag, bool isScreenTag)	void	Вставляет элементы из тега аналогично предыдущей функции. Но дополнительно указывается, является ли тег экранным или нет. Для экранного тега необходимо использовать эту функцию.
InsertItemsFromTag(string tag, string member)	void	Вставляет элементы из тега сложного типа данных. Дополнительно нужно указать имя члена типа данных.
InsertItemsFromTag(string tag, string member, bool isScreenTag)	void	Вставляет элементы из тега сложного типа данных

		аналогично предыдущей функции. Но дополнительно указывается, является ли тег экранным или нет. Для экранного тега необходимо использовать эту функцию.
RemoveSelectedItem()	void	Удаляет выбранный элемент из ComboBox.
RemoveItem(int index)	void	Удаляет элемент из ComboBox для заданного индекса. Индекс начинается с 0-го элемента.
RemoveAllItems()	void	Удаляет все элементы ComboBox.

Текстовое поле (TextBox)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
Text	string	Чтение / Запись	Получает или задает текст в TextBox.
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта TextBox. (Например, TextBox1.FontColor = "red";)

2. Функции

Имя	Возвращает	Описание
ResetPasswordBox()	void	Очищает поле, если TextBox имеет свойство Password, установленное в значение true в таблице свойств.

GetFocus()	void	Устанавливает фокус на TextBox.
SelectAll()	void	Выделяет весь текст TextBox.

Кнопка (Button)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
Text	string	Чтение / Запись	Получает или задает текст Button. (Например, Button1.Text = "Welcome";)
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта Button. (Например, Button1.FontColor = "red";)
IsChecked	bool	Чтение / Запись	Получает или устанавливает для Button включенное или выключенное положение. Работает, только если установлено значение true для свойства переключающая кнопка в таблице свойств. (Например, btnlabel.Text = btn.IsChecked.ToString();)

2. Функции

Имя	Возвращает	Описание
changeBorderThickness(int left, int top, int right, int bottom)	void	Изменяет толщину границы Button.
changeBorderThicknessPressed(int left, int top, int right, int bottom)	void	Изменяет толщину границы кнопки в нажатом положении.

Метка (Label)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
Text	string	Чтение / Запись	Получает или задает текст Label. (Например, Label1.Text = "Welcome");
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта Label. (Например, Label1.FontColor = "red");

2. Функции

Нет доступных функций.

Символ (Symbol)

(Общие свойства, Свойства вращения)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций.

Флажок (CheckBox)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
-----	-----	--------------------------	----------

Text	string	Чтение / Запись	Получает или задает текст Checkbox. (Например, CheckBox1.Text = "Checkbox 1";)
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта Checkbox. (Например, CheckBox1.FontColor = "red";)
IsChecked	bool	Чтение / Запись	Получает или устанавливает Checkbox (установлен или снят). Если Checkbox настроен на три состояния, true будет назначено только для установленного состояния (не для снятого флажка или третьего неожиданного состояния).
IsCheckedInt	int	Чтение / Запись	Получает или устанавливает Checkbox с использованием целого числа (0 - не отмечено флажком, 1 - отмечено). Если Checkbox настроен на три состояния, значение -1 будут соответствовать третьему неожиданному состоянию.

2. Функции

Нет доступных функций.

Радиокнопка (RadioButton)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
Text	string	Чтение / Запись	Получает или задает текст RadioButton. (Например, RadioButton1.Text = "Option 1";)
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта RadioButton. (Например, RadioButton1.FontColor = "red";)

IsChecked	bool	Чтение / Запись	Получает или устанавливает выбранную RadioButton.
-----------	------	-----------------	---

2. Функции

Нет доступных функций.

Геометрические объекты

Прямоугольник (Rectangle)

(Общие свойства, Свойства вращения)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций.

Эллипс (Ellipse)

(Общие свойства, Свойства вращения)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций.

Многоугольник (Polygon)

(Общие свойства, Свойства вращения)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций.

Линия (Line)

(Общие свойства, Свойства вращения)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
X1	double	Чтение / Запись	Получает или задает первую точку по оси X.
Y1	double	Чтение / Запись	Получает или задает первую точку по оси Y.
X2	double	Чтение / Запись	Получает или задает вторую точку по оси X.
Y2	double	Чтение / Запись	Получает или задает вторую точку по оси Y.

2. Функции

Нет доступных функций.

Полилиния (PolyLine)

(Общие свойства, Свойства вращения)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций.

Объекты интерфейса

Группа (GroupBox)

(Общие свойства)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
Text	string	Чтение / Запись	Получает или задает текст GroupBox. (Например, GroupBox1.Text = "GroupBox 1";)
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта GroupBox. (Например, GroupBox1.FontColor = "red";)

2. Функции

Нет доступных функций.

Вкладка (Tab)

(Общие свойства)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
-----	-----	--------------------------	----------

FontColor	string	Чтение / Запись	Получает или задает цвет шрифта Tab. (Например, Button1.FontColor = "red");
-----------	--------	-----------------	---

2. Функции

Имя	Возвращает	Описание
SetVisibility(int index, bool isVisible)	void	Изменяет видимость вкладки для данного индекса, начиная с 0.
SetIsEnabled(int index, bool isEnabled)	void	Включает или отключить вкладку для данного индекса, начиная с 0.
ChangeSelectedIndex(int index)	void	Изменяет выбор вкладки для данного индекса, начиная с 0.
IsTabVisible(int index)	bool	Проверяет видимость вкладки для данного индекса, начиная с 0

Экран (Screen)

(Общие свойства)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Имя	Возвращает	Описание
GetCurrentScreen()	string	Получает текущий открытый графический экран, открытый в экранном объекте.

Меню (Menu)

(Общие свойства)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
FontColor	string	Чтение / Запись	Получает или задает цвет шрифта Menu. (Например, Menu1.FontColor = "red");

2. Функции

Нет доступных функций.

Расширенные объекты

.NET Control

(Общие свойства)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций.

Тренд (Trend)

(Общие свойства)

Объект тренда имеет множество функций, которые можно использовать в скриптах. Некоторые из них связаны с SPC (статистический контроль процесса), который является стандартной методологией для измерения и контроля качества в процессе производства.

1. Свойства

Нет доступных свойств, кроме унаследованных.
--

2. Функции

Имя	Возвращает	Описание
Play()	void	Включение обновления отображения объекта тренда.
Pause()	void	Пауза обновления объекта тренда.
AddPen(string tagName, string description)	void	Добавляет новое перо к объекту тренда. Используется существующий тег и описание пера.
AddPen(string tagName, string description, bool showSpc)	void	Добавляет новое перо к объекту тренда, как и в предыдущей функции и включает видимость всей информации SPC.
AddPen(string tagName, string description, bool showSpcMin, bool showSpcMax, bool showSpcAvg, bool showSpcDev, int spcDevSigma)	void	Добавляет новое перо к объекту тренда и позволяет пользователю указать, какая информация SPC должна быть видна.
AddPen(string tagName, string description, string penColor)	void	Добавляет новое перо к объекту тренда и позволяет пользователю указать цвет пера.
AddPen(string tagName, string description, string penColor, bool showSpc)	void	Добавляет новое перо к объекту тренда и позволяет пользователю указать цвет пера и видимость всей информации SPC.
AddPen(string tagName, string description, string penColor, bool showSpcMin, bool showSpcMax, bool showSpcAvg, bool showSpcDev, int spcDevSigma)	void	Добавляет новое перо к объекту тренда и позволяет пользователю указать цвет пера, и какая информация SPC должна быть видна.
AddPen(string tagName, string description, int lineStyle, double thickness, string	void	Добавляет новое перо к объекту тренда и позволяет пользователю указать

penColor, bool interpol, int marker, bool showLegend)		стиль линии (начиная с 0, используется тот же стиль линии, как в таблице свойств тренда), толщину пера, цвет пера, интерполяцию, тип маркера (начиная с 0 - нет, 1 - круг, 2 - треугольник) и видимость легенды.
AddPen(string tagName, string description, int lineStyle, double thickness, string penColor, bool interpol, int marker, bool showLegend, bool showSpc)	void	Добавляет новое перо к объекту тренда, как и в предыдущей функции и включает видимость всей информации SPC.
AddPen(string tagName, string description, int lineStyle, double thickness, string penColor, bool interpol, int marker, bool showLegend, bool showSpcMin, bool showSpcMax, bool showSpcAvg, bool showSpcDev, int spcDevSigma)	void	Добавляет новое перо к объекту тренда и позволяет пользователю указать, какая информация SPC должна быть видна.
SetPenDescription(string tagName, string description)	void	Устанавливает описание пера по имени тега.
SetPenLineStyle(string tagName, int lineStyle)	void	Устанавливает стиль линии пера по имени тега (начиная с 0, используется тот же стиль линии, как в таблице свойств тренда).
SetPenThickness(string tagName, double thickness)	void	Устанавливает толщину пера по имени тега.
SetPenColor(string tagName, string penColor)	void	Устанавливает цвет пера по имени тега.
SetPenInterpolated(string tagName, bool interpol)	void	Устанавливает интерполяцию пера по имени тега.
SetPenMin(string tagName, double min)	void	Устанавливает минимальное значение для пера по имени тега.
SetPenMax(string tagName, double max)	void	Устанавливает максимальное значение для пера по имени тега.
SetPenMarker(string tagName, int marker)	void	Устанавливает маркер пера по имени тега (начиная с 0 - нет, 1 - круг, 2 - треугольник).
SetPenShowLegend(string tagName, bool showLegend)	void	Устанавливает видимость легенды пера.

HasPenSPCMin(string tagName)	bool	Проверяет, отображается ли SPCMin.
HasPenSPCMax(string tagName)	bool	Проверяет, отображается ли SPCMax.
HasPenSPCAvg(string tagName)	bool	Проверяет, отображается ли SPCAvg.
HasPenSPCDev(string tagName)	bool	Проверяет, отображается ли SPCDev.
GetPenSPCDevSigma(string tagName)	int	Получает SPCDevSigma, используемое при вычислении отклонения пера.
GetPenSPCMin(string tagName)	double	Получает текущее значение PenSPCMin видимых точек.
GetPenSPCMax(string tagName)	double	Получает текущее значение PenSPCMax видимых точек.
GetPenSPCAvg(string tagName)	double	Получает текущее значение PenSPCAvg видимых точек.
GetPenSPCDev(string tagName)	double	Получает текущее значение PenSPCDev видимых точек.
GetPenSPCDevPositive(string tagName)	double	Получает значение положительного отклонения (верхняя линия отклонения).
GetPenSPCDevNegative(string tagName)	double	Получает значение отрицательного отклонения (нижняя линия отклонения).
SetSPC(bool showSPC)	void	Устанавливает видимость всей информации SPC для всех перьев в тренде.
SetSPC(bool showSpcMin, bool showSpcMax, bool showSpcAvg, bool showSpcDev, int spcDevSigma)	void	Устанавливает выборочную видимость информации SPC для всех перьев в тренде.
SetPenSPC(string tagName, bool showSPC)	void	Устанавливает видимость всей информации SPC по имени тега.
SetPenSPC(string tagName, bool showSpcMin, bool showSpcMax, bool showSpcAvg, bool showSpcDev, int spcDevSigma)	void	Устанавливает выборочную видимость информации SPC по имени тега.
HidePen(string tagName, bool isHide)	void	Скрывает или показывает желаемое перо в объекте тренда.

RemovePen(string tagName)	void	Удаляет перо из объекта тренда.
UpdateStartDate(DateTime date)	void	Обновляет начальную дату объекта тренда.
UpdateEndDate(DateTime date)	void	Обновляет конечную дату объекта тренда.
UpdateDuration(int duration)	void	Обновляет интервал отображения объекта тренда. Параметр в секундах.
UpdateDuration(TimeSpan duration)	void	Обновляет интервал отображения объекта тренда. Интервал времени в качестве параметра.
UpdateDurationEnd(int duration, DateTime date)	void	Обновляет EndDuration объекта тренда. StartDuration рассчитывается как EndDuration - Duration. Интервал отображения (duration) в секундах.
UpdateDurationEnd(TimeSpan duration, DateTime date)	void	Обновляет EndDuration объекта тренда. Интервал отображения (duration) типа TimeSpan.

Тревоги/События (Alarm/Events)

(Общие свойства)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
SelectedLine	int	Чтение / Запись	Получает или задает выбранную строку в объекте тревог.

2. Функции

Имя	Возвращает	Описание
Ack()	void	Подтвердить выбранную тревогу.

AckAll()	void	Подтвердить все тревоги.
DeselectLines()	void	Снять выделение со всех строк.
Filter(string type, object filter)	bool	Настраивает фильтр для объекта тревог. (Например, Alarm1.Filter("Name.Value", NameTextBox.Text);)
SetAbsolutStartDate(DateTime date)	void	Устанавливает абсолютную начальную дату объекта тревог.
SetAbsolutStartDate(string date)	void	Устанавливает абсолютную начальную дату объекта тревог.
SetAbsolutEndDate(DateTime date)	void	Устанавливает абсолютную конечную дату объекта тревог.
SetAbsolutEndDate(string date)	void	Устанавливает абсолютную конечную дату объекта тревог.
SetAbsolutStartEndDate(DateTime startDate, DateTime endDate)	void	Устанавливает абсолютную начальную и конечную дату объекта тревог.
SetAbsolutStartEndDate(string startDate, string endDate)	void	Устанавливает абсолютную начальную и конечную дату объекта тревог.
SetAbsolutStartEndDate(DateTime startDate, string endDate)	void	Устанавливает абсолютную начальную и конечную дату объекта тревог.
SetAbsolutStartEndDate(string startDate, DateTime endDate)	void	Устанавливает абсолютную начальную и конечную дату объекта тревог.
SetRelativeDate(DateTime date)	void	Устанавливает относительную дату объекта тревог.
SetRelativeDate(string date)	void	Устанавливает относительную дату объекта тревог.
SetRelativePeriod(TimeSpan period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativePeriod(string period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativePeriod(double period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться

		в объекте тревог.
SetRelativePeriod(TimeSpan period, bool isBackward, bool isForward)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativePeriod(string period, bool isBackward, bool isForward)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativePeriod(double period, bool isBackward, bool isForward)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(DateTime date, TimeSpan period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(DateTime date, string period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(DateTime date, double period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(string date, TimeSpan period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(string date, string period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(string date, double period)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(DateTime date, TimeSpan period, bool isBackward, bool isForward)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(DateTime date, string period, bool isBackward, bool isForward)	void	Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(DateTime date, double period, bool isBackward, bool isForward)		Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.

SetRelativeDatePeriod(string date, TimeSpan period, bool isBackward, bool isForward)		Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(string date, string period, bool isBackward, bool isForward)		Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
SetRelativeDatePeriod(string date, double period, bool isBackward, bool isForward)		Устанавливает период, в течение которого тревоги будут отслеживаться в объекте тревог.
AddFilter(string newFilter)		Добавляет новый фильтр для объекта тревог.
RemoveFilter(string filterToRemove)		Удаляет фильтр из объекта тревог.
CleanFilter()		Удаляет все фильтры из объекта тревог.
ClearEvents()		Удаляет все события из объекта тревог.

Матричная таблица (Matrix Grid)

(Общие свойства)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
ColumnIndex	int	Чтение / Запись	Возвращает индекс столбца выбранной ячейки (индекс увеличивается, начиная с 0).
RowIndex	int	Чтение / Запись	Возвращает индекс строки выбранной ячейки (индекс увеличивается, начиная с 0).
DepthIndex	int	Чтение / Запись	Возвращает индекс глубины выбранной ячейки (индекс увеличивается, начиная с 0 для 1-го измерения).

2. Функции

Имя	Возвращает	Описание
ChangeDataSource(string tagName)	void	Изменяет источник данных MatrixGrid с помощью тега, имя которого передается в качестве параметра.

Мультитеговая таблица (MultiTagViewer)

(Общие свойства)

1. Свойства

Имя	Тип	Доступ (Чтение / Запись)	Описание
SelectedRowIndex	int	Чтение / Запись	Получает или задает выбранную по индексу строку таблицы, начиная с 0.
SelectedArrayIndex	int	Чтение / Запись	Получает или задает индекс массива тегов, начиная с 0. Это свойство полезно, когда пользователь сортирует строки таблицы, и индекс строки таблицы не соответствует индексу массива.
Count	int	Read	Получает количество строк в таблице.

2. Функции

Имя	Возвращает	Описание
ChangeImageSource(int line, string key, string newSource)	void	Настраивает изображение, которое будет отображаться в определенном свойстве (ключе) в MultiTagViewer.
ChangeBackground(int line, string key, string color)	void	Настраивает цвет фона, который будет использоваться в определенном свойстве (ключе) в MultiTagViewer.

Remove()	void	Удаляет выделенную строку в объекте MultiTagViewer.
Add(string property, object value)	void	
Add(Dictionary<string, object> properties)	void	
Filter(string property, object value)	void	Настраивает фильтр для объекта MultiTagViewer. (Например, MTV1.Filter("Name.Value", NameTextBox.Text);)
Filter(Dictionary<string, object> properties)	void	Настраивает несколько фильтров для объекта MultiTagViewer.
RefreshIndex()	void	Обновляет индекс объекта MultiTagViewer.
HideColumn(int index)	void	Скрывает столбец в объекте MultiTagViewer.
ShowColumn(int index)	void	Показывает столбец в объекте MultiTagViewer.
IsColumnVisible(int index)	bool	Проверяет, видимость столбца по индексу.

Шаблон (Template)

(Общие свойства)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций.

Диаграммы (Charts)

Для диаграмм добавим еще несколько общих свойств и функций, поскольку они имеют одного и того же предка.

Диаграмма предок

1. Свойства

Имя	Тип	Чтение / Запись	Описание
Title	string	Чтение / Запись	Получает или задает заголовок диаграммы
TitleHorizontalAlignment	string	Чтение / Запись	Получает или задает выравнивание заголовка графического объекта. По центру (Center), слева (Left), справа (Right).
BackgroundColor	string	Чтение / Запись	Получает или задает цвет фона диаграммы.
PlotAreaColor	string	Чтение / Запись	Получает или задает цвет области построения графического объекта. (Например, Chart1.PlotAreaColor = "blue")
ShowGridBorder	bool	Чтение / Запись	Показывает или скрывает границу сетки.
ShowVerticalLines	bool	Чтение / Запись	Показывает или скрывает вертикальные линии.
ShowHorizontalLines	bool	Чтение / Запись	Показывает или скрывает горизонтальные линии.

2. Функции

Имя	Возвращает	Описание
UpdateTitleFormat(string family, int size, bool isBold, bool isItalic)	void	Обновляет шрифт заголовка диаграммы. (Например, Chart1.UpdateTitleFormat("Arial", 10, true, true);)

UpdateLegendFormat(string family, int size, bool isBold, bool isItalic)	void	Обновляет шрифт легенды диаграммы. (Например, Chart1.UpdateLegendFormat("Arial",10,true,false);
UpdateLegendSize(double width, double height)	void	Обновляет размер легенды диаграммы.

Столбчатая диаграмма (Bar)

(Общие свойства, Диаграмма предок)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций, кроме унаследованных.

Линейный график (Line)

(Общие свойства, Диаграмма предок)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Имя	Возвращает	Описание
InsertItem(int serieNumber, object x, object y)	void	Вставляет элемент в объект точечной диаграммы. (Например, LineChart.InsertItem(serieNumber, x, y);)

		Тип объекта может быть строковым или числовым.
InsertItems(int serieNumber, List<object> x, List<object> y)	void	Вставляет список элементов в объект линейного графика (Например, LineChart.InsertItems(serieNumber, list1,list2);) Тип объекта может быть строковым или числовым.
ClearItems(int serieNumber)	void	Удаляет всю серию из объекта линейного графика.
ChangeLineType(int serieNumber, int type)	void	Изменяет тип линии серии в объекте линейного графика.
ChangeSeriesThickness(int serieNumber, double thickness)	void	Изменяет толщину серии линейного графика.

Круговая диаграмма (Pie)

(Общие свойства, Диаграмма предок)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

Нет доступных функций, кроме унаследованных.

Группировка

Группировка (Grouping)

(Общие свойства)

1. Свойства

Нет доступных свойств, кроме унаследованных.

2. Функции

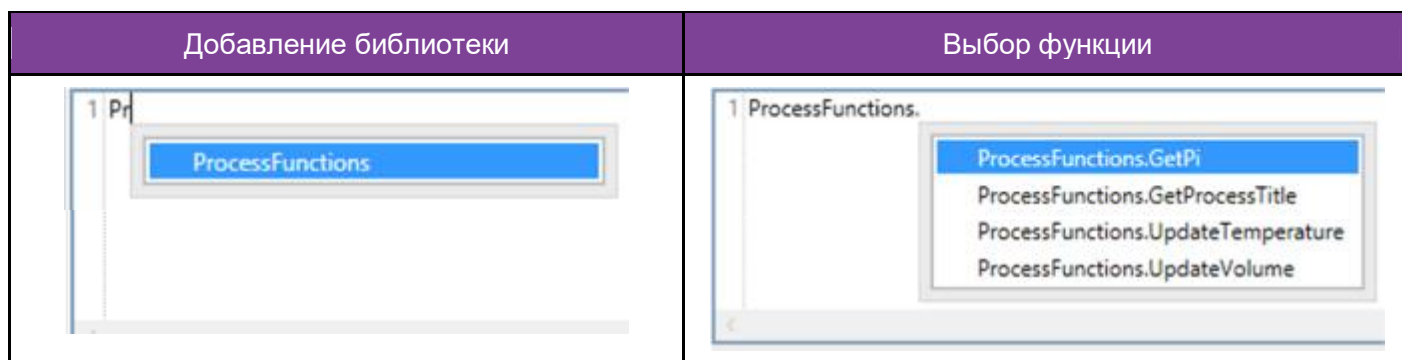
Нет доступных функций, кроме унаследованных.

Скрипты и библиотека пользовательских функций

Также вы можете создавать свои собственные пользовательские функции в SmartView, вам просто нужно добавить функции в библиотеку (документ), и эти функции будут доступны для использования в разных местах вашего приложения. В приведенном ниже примере библиотека была названа ProcessFunctions, и в нее были добавлены четыре функции.

```
ProcessFunctions X
1 void UpdateTemperature(int index, float temperature)
2 {
3     //Code here
4 }
5
6 void UpdateVolume(int index, float volume)
7 {
8     //Code here
9 }
10
11 float GetPi()
12 {
13     return 3.14159265359f;
14 }
15
16 string GetProcessTitle(string title)
17 {
18     return "Title is " + title;
19 }
```

Функция автодополнения поможет вам добавить библиотеку ProcessFunction и выбрать одну из четырех функций, как показано ниже.

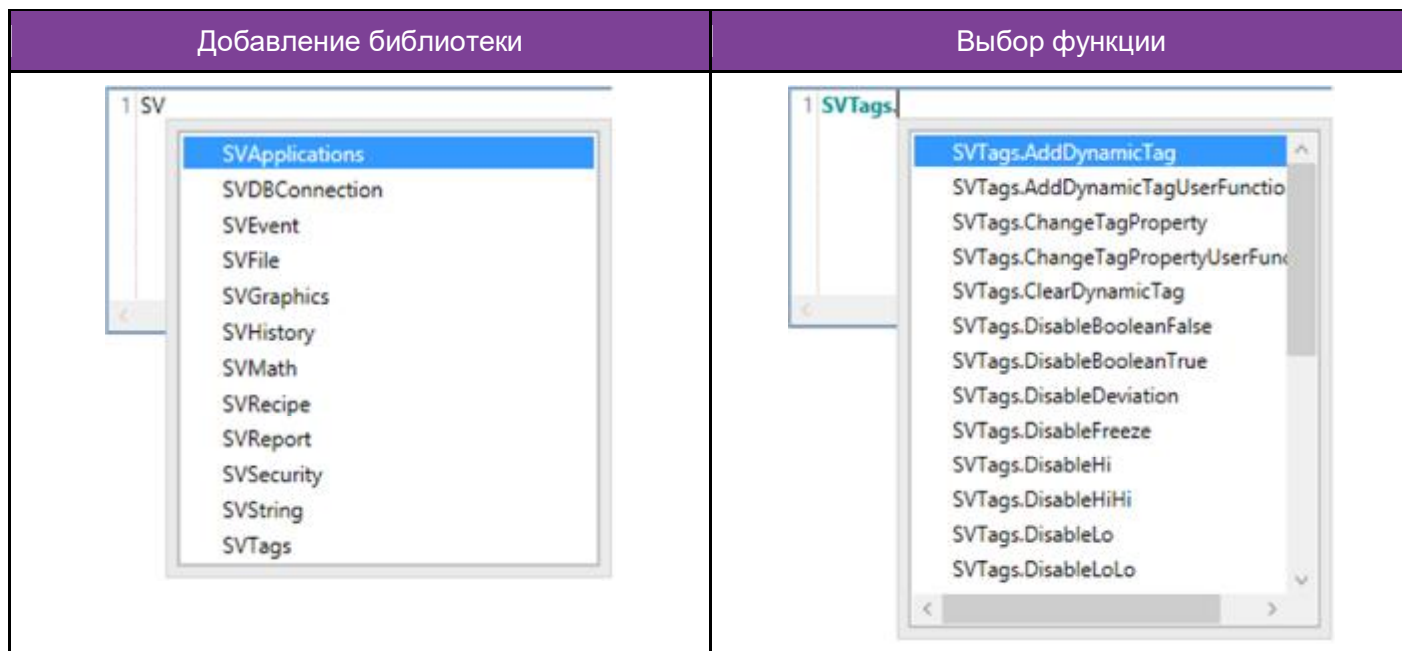


Скрипты и библиотека системных функций

Библиотека системных функций также доступна для использования в скриптах. Она содержит полезные библиотеки, которые сделают вашу жизнь проще. Вы можете найти их в дереве навигации.



Все они имеют одинаковый префикс SV. Как только вы наберете SV в скрипте, они все будут перечислены. После выбора библиотеки добавьте точку, и получите список доступных функции библиотеки.



Выпуски документа

Версия	Комментарии	Дата	Автор
1.0	Первоначальный вариант	02-April-2020	Fabio Carvalho
1.1	В новый подраздел добавлена константа IsWebRunning	06-April-2020	Fabio Carvalho
1.2	Изменено имя функции ChangeSerieThickness на ChangeSeriesThickness	09-April-2020	Fabio Carvalho