

# Руководство пользователя

## Dialog Builder for Adobe Creative Suite/Cloud

### Оглавление

Оглавление.....	1
О программе .....	2
Основные возможности .....	2
Планируется в следующих версиях.....	2
Известные проблемы и недостатки .....	3
Установка.....	3
Запуск скрипта .....	3
Вопросы совместимости с различными версиями пакетов Adobe CS/CC .....	4
Использование DialogBuilder с пакетом Adobe Creative Cloud (CC) .....	4
Работа с программой .....	5
Общие принципы.....	6
Клавиатурные сокращения .....	6
Добавление элементов в диалог .....	6
Редактирование свойств элементов .....	7
Перемещение и удаление элементов .....	7
Копирование и вставка элементов .....	7
Редактирование имён переменных.....	8
Открытие файлов.....	8
Сохранение и контроль результата.....	8
Интеграция с библиотекой SimpleUI .....	9
Дополнительные виджеты UIControls .....	9
Separator .....	10
WebLink.....	10
UnitBox .....	10
Работа с настройками программы .....	11
Особенности работы с настройками.....	11
Методика и рекомендации разработки диалога в приложении Dialog Builder .....	12
Дополнительные сведения.....	13
Обновления Dialog Builder в версии 1.90.....	13
Обновления в файле документации .....	14

## О программе

Конструктор диалоговых окон Dialog Builder для Adobe Creative Suite/Cloud реализует простой визуальный способ разработки диалогов для пользовательских сценариев на языке JavaScript в среде Adobe ExtendScript Toolkit, предназначенных для приложений Adobe, поддерживаемых автоматизацию с их помощью: Adobe InDesign, Adobe Photoshop, Adobe Illustrator, Bridge и др...

### Основные возможности:

- Генерация корректного javascript-кода для редактируемого макета диалога с возможностью его сохранения в пользовательский файл документа (файл сценария с расширением `.jsx`), последующего открытия и доработки ранее сохранённых результатов;
- Поддерживается полноценный многодокументный интерфейс — одновременная работа с несколькими документами с возможностью копирования и перемещения элементов диалога между разными окнами документов;
- Работа с макетом диалога реализована в режиме «LiveUpdate», при котором редактирование свойств компонентов сразу отражается на их графическом представлении в макете. Такой подход позволяет на каждом шаге работы с макетом видеть фактический результат своих настроек и получать требуемый результат с минимальными трудозатратами;
- Поддерживаются клавиатурные сокращения (корректная работа с клавиатурными сокращениями возможна только при закрытом редакторе ESTK, в противном случае он производит перехват нажатия “горячих клавиш”);
- Поддерживается возможность управления пользовательскими наборами шрифтов и цветов, которые могут быть сохранены в настройках приложения и использованы повторно;
- Реализована возможность интеграции графической библиотеки SimpleUI с пользовательскими файлами сценариев, а также использование расширенного набора графических компонентов в диалоговых окнах (подробности в справочной документации к библиотеке SimpleUI Guide.pdf, см. <https://github.com/SlavaBuck/Includes/tree/master/SimpleUI>);
- Поддерживается полная локализация интерфейса. Список доступных языков находится в соответствующем пункте настроек приложения, по умолчанию определение языка интерфейса происходит автоматически на основе текущего значения локали целевой системы.

### Планируется в следующих версиях:

- Групповое выделение и выполнение групповых операций с объектами пользовательского интерфейса в документе;
- Поддержка предустановленных наборов из групп элементов управления для быстрого формирования диалогов на основе типовых шаблонов, использования мастеров для их создания;
- Расширение возможностей по настройке объектов списков (ListBox, DropDownList, TreeView) и работы с изображениями в рамках объекта диалога:
  - конвертация и включение изображений в тело скрипта;
  - настройка масштабирования изображений в пользовательских элементах управления;
  - Работа с массивами изображений для элементов, поддерживающих мультисостояния (IconButton/toolbutton и т.п.);
  - реализация работы с многоколоночными списками и включением изображений в списки;
  - привязка списков к источникам данных;
  - реализация более широких возможностей по редактированию элемента управления TreeView;
- Улучшение совместимости с последними версиями пакета Adobe Creative Cloud и реализация совместимости с операционной системой MacOS.

### Известные проблемы и недостатки:

- Целевой платформой для библиотеки SimpleUI является приложение Adobe InDesign (в т.ч. EStoolkit), в этой связи, корректная работа расширенных графических компонентов для остальных приложений Adobe не гарантируется (не гарантируется — не означает “не поддерживается”);
- Работа с изображениями и списками реализована на самом базовом уровне, так, например, не реализована работа с многоколоночными списками, настройкой древовидной структуры для TreeView, массивами изображений для элементов с поддержкой мультисостояний и т.п. (подробнее см. в планах не реализацию);
- В процессе работы с документом могут наблюдаться дефекты пересчёта размеров или автомасштабирование некоторых объектов диалога (Например, при работе с элементом TabbedPanel). Для решения проблемы достаточно в любой момент времени выполнить операцию «обновления/перезагрузки» документа с помощью соответствующей кнопки, расположенной под деревом компонентов — см. Рис.3, рядом с кнопками «Удаление» и «Перемещения» элементов;
- Ограниченная совместимость программы с последней версией пакета Adobe Creative Cloud для Windows (см. “Использование DialogBuilder с пакетом Adobe Creative Cloud (CC)”);
- Программа всё ещё находится в активной стадии разработки, поэтому, наверняка возможны различные внештатные ситуации. В связи с этим рекомендуется чаще сохранять промежуточные результаты своих трудов, особенно при работе с большим количеством одновременно открытых документов.

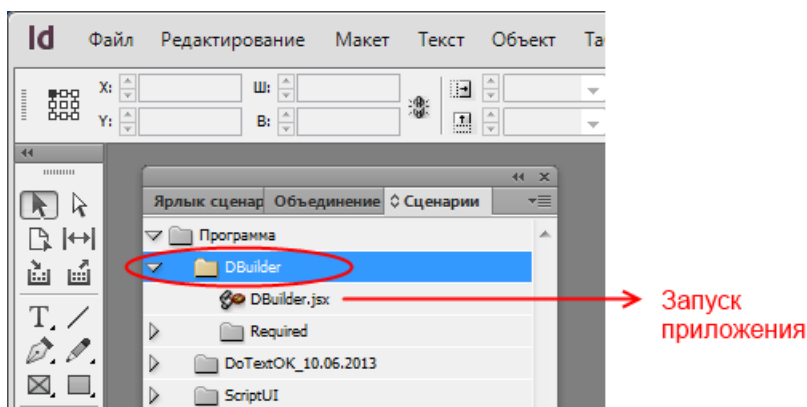
## Установка

1. Скачайте архив, содержащий последнюю версию программы, из репозитория на сайте **GitHub** — <https://github.com/SlavaBuck/DialogBuilder> (прямая ссылка на архив: <https://github.com/SlavaBuck/DialogBuilder/archive/master.zip>);
2. Из скачанного архива (DialogBuilder-master.zip) перенесите папку DBuilder (находится в DialogBuilder-master/DBuilder) в папку со скриптами Adobe InDesign (На своей машине с установленной Windows 7 и Adobe InDesign CS6 я использую путь *C:\Program Files (x86)\Adobe\Adobe InDesign CS6\Scripts\Scripts Panel\*). Строго говоря, вы можете использовать любой желаемый путь для размещения папки с приложением;
3. Распаковав папку по пути, предложенному в п.2, впоследствии вы сможете обнаружить её в палитре скриптов Adobe InDesign так, как это изображено на рис. 1. Чтобы запустить приложение, необходимо выполнить стартовый файл *DBuilder.jsx*.

Программа DialogBuilder распространяется вместе с исходным кодом, который можно найти в папке *src* и в корне архива DialogBuilder-master.zip. Архив также содержит скомпилированную версию программы в папке *DBuilder*. Работа с исходниками в папке *src* дополнительно требует установки библиотек из репозитория <https://github.com/SlavaBuck/Includes>. Описание работы с исходниками можно найти в отдельном файле документации DialogBuilder Programmer Guide.pdf, распространяемом вместе с программой.

## Запуск скрипта

После установки скрипта, придерживаясь рекомендаций из предыдущего раздела, в палитре сценариев Adobe InDesign обнаруживается папка приложения с расположенным в ней стартовым файлом *Dbuilder.jsx* (для открытия палитры используйте “горячую” комбинацию Ctrl+Alt+ F11). На рис. 1 изображён вид палитры сценариев Adobe InDesign CS6 для Windows (подробнее о размещении файлов сценариев см. в документации к вашей версии пакета Adobe CC/CS).



**Рис.1** Стартовый файл программы Dialog Builder

Вы можете запустить программу прямо из палитры, воспользовавшись командой контекстного меню “выполнить сценарий”, или открыть файл *Dbuilder.jsx* в редакторе ExtendScript Toolkit (выполнив “изменить сценарий”), а после — произвести запуск из самого редактора.

### Вопросы совместимости с различными версиями пакетов Adobe CS/CC

Проблема совместимости приложения имеет два аспекта: поддержка целевым приложением написанного вами сценария и поддержка целевой платформой запуска и выполнения самого приложения DialogBuilder.

В отношении пользовательских сценариев и диалогов, разработанных вами с помощью DialogBuilder, никаких особых ограничений не действует. Подход, с использованием ресурсной строки для формирования диалоговых окон, позволяет использовать такие скрипты для любого приложения из пакета Adobe (InDesign, Photoshop, Illustrator, Bridge и т.д.), поддерживающего автоматизацию и сценарии на языке javascript. Это в равной степени относится как к продуктам семейства Adobe Creative Suite (CS), так и для Adobe Creative Cloud (CC). Некоторые ограничения могут возникать только при условии использования в своих сценариях библиотеки SimpleUI. В конечном итоге, работа ваших сценариев и внешний вид диалоговых окон будет определяться степенью поддержки соответствующим целевым приложением собственной реализации подсистемы ScriptUI (к сожалению, эта неприятная закавыка с различными реализациями ScriptUI уже не раз становилась причиной справедливой критики разработчиков в адрес корпорации Adobe Inc.).

В отношении самого конструктора диалоговых окон ситуация несколько иная. Приложение DialogBuilder разрабатывается в среде Adobe ExtendScript Toolkit CS6 с прицелом для работы под Adobe InDesign CS (5, 5.5, 6). В результате, с продуктами семейства CS программа одинаково корректно работает в любых режимах запуска: как с панели скриптов Adobe InDesign (режим запуска Adobe InDesign CS), так и с редактора ExtendScript (в режиме Adobe ExtendScript CS). В то же время, работа в режиме запуска Adobe InDesign CC пока не поддерживается. Использование программы DialogBuilder с пакетом Creative Cloud возможно только в режиме запуска под Adobe ExtendScript CC.

### Использование DialogBuilder с пакетом Adobe Creative Cloud (CC)

Использование DialogBuilder с пакетом Adobe InDesign CC возможно только в режиме запуска Adobe ExtendScript CC:

1. Найдите в палитре скриптов приложения Adobe InDesign CC стартовый файл программы и откройте его в редакторе ESTK;
2. Убедитесь в том, что настройка целевой платформы выставлена в значения для ExtendScript Toolkit;
3. Выполните запуск скрипта из редактора.

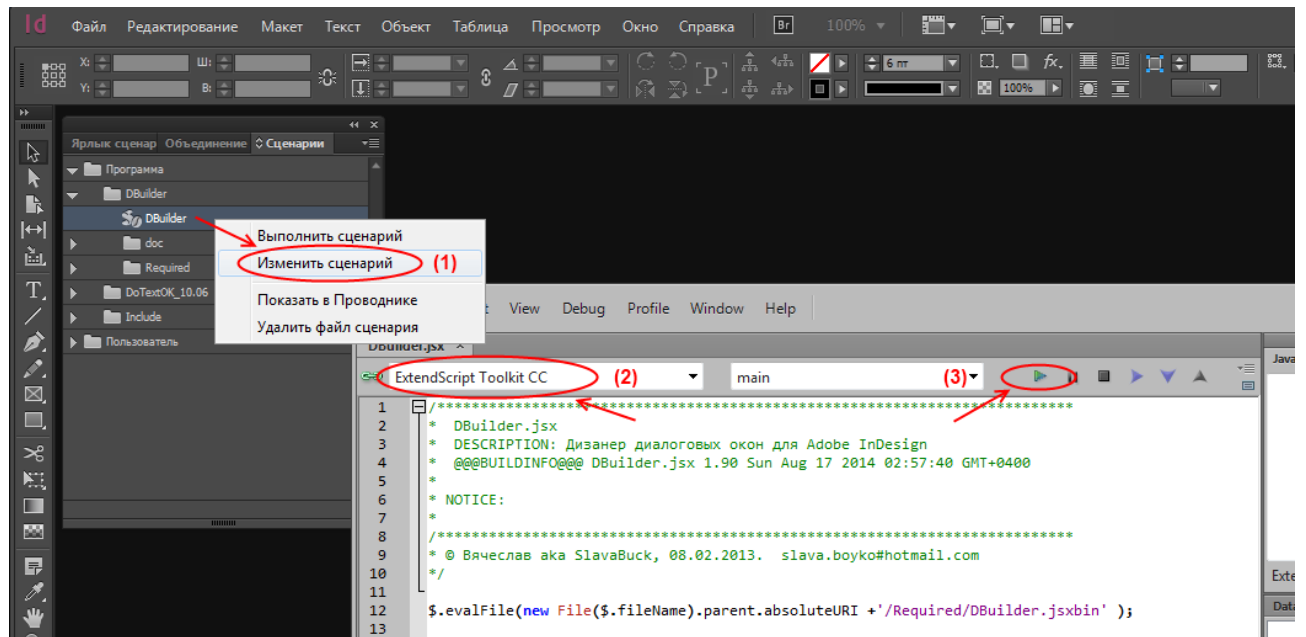


Рис.2 Запуск Dialog Builder в режиме Adobe ExtendScript CC

## Работа с программой

Идея написания программы заключается в предоставлении возможности максимально простой и быстрой разработки диалоговых окон для сценариев, работающих в среде Adobe ExtendScript (ESTK) и InDesign. На рис. 3 изображен основной интерфейс конструктора Dialog Builder:

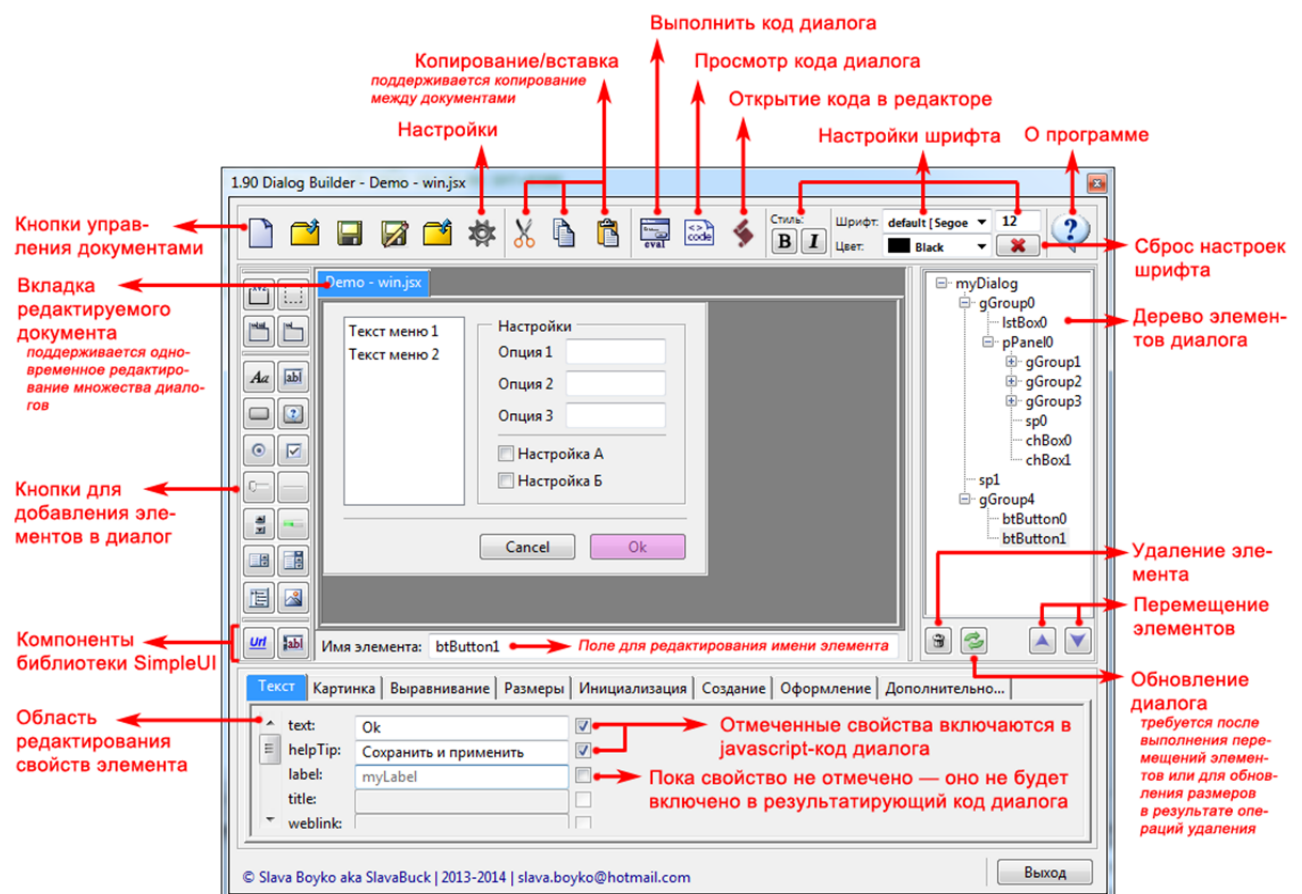


Рис.3 Основной интерфейс конструктора диалоговых окон Dialog Builder

## Общие принципы

Разработка диалога опирается на стандартные возможности компоновщика ScriptUI (AutoLayout manager) при которой размеры и положение элементов диалога вычисляется автоматически в зависимости от их настроек и содержимого. Таким образом, для формирования нужного окна последовательно добавляйте в него компоненты. Добавляйте группы и панели, наполняйте их нужными элементами. Используя выравнивание компонентов относительно друг друга, добивайтесь получения требуемого вида диалогового окна. Для удобства работы с макетом диалога поддерживается работа с горячими клавишами.

## Клавиатурные сокращения:

**Ctrl + N** - Новый документ;

**Ctrl + O** - Открыть документ;

**Ctrl + W** - Закрыть документ;

**Ctrl + S** - Сохранить документ;

**Ctrl + K** - Открыть настройки;

**Ctrl + X** - Вырезать;

**Ctrl + C** - Скопировать;

**Ctrl + V** - Вставить;

**Ctrl + R** - Выполнить;

**Ctrl + D** - Показать код диалога;

**Ctrl + H** - О программе;

**Ctrl + Shift + S** - Сохранить как;

**Ctrl + Shift + E** - Открыть jsx-файл в редакторе (предварительно файл сохраняется);

Корректная работа с клавиатурными сокращениями возможна только при закрытом редакторе ESTK, в противном случае он производит перехват нажатия горячих клавиш.

## Добавление элементов в диалог

Новые элементы в диалог добавляются с помощью соответствующих кнопок в левой части окна приложения (см. Рис 5. Кнопки элементов). Новые элементы всегда добавляются “в хвост” активного контейнера (под контейнерами подразумеваются такие элементы как: группы, панели и вкладки / group, panel, tabbedpanel, tab). После добавления элемента, на него автоматически переключается фокус редактирования свойств, при этом в нижней области главного окна приложения отображаются значения всех доступных для него свойств, сгруппированные согласно категориям в соответствующих вкладках (см. Рис. 3.). Для добавления элементов в контейнер необходимо предварительно переключить на него “фокус добавления” — выделить его мышкой (кликнуть либо по самому элементу, либо по соответствующему узлу в дереве компонентов). Можно настроить автоматическую переустановку фокуса на элементы контейнерного типа после их добавления в диалог, для этого необходимо установить соответствующую настройку «Автофокус» на вкладке «Основные» в панели настроек приложения (Рис. 4.). По умолчанию автофокус отключён (галочка с настройки снята).

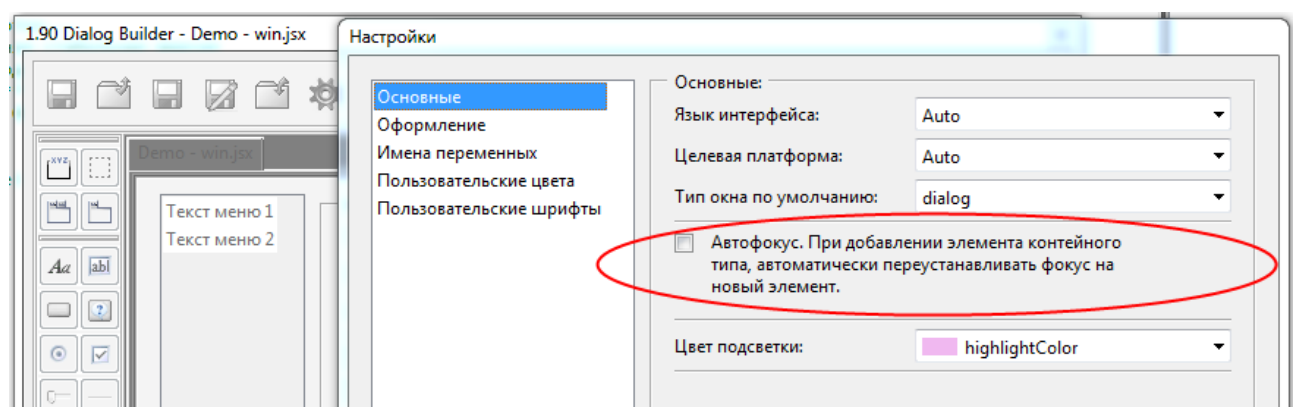


Рис.4 Настройка автофокуса

### Редактирование свойств элементов

После добавление элемента в диалог, на него автоматически переустанавливается фокус редактирования, о чём свидетельствует соответствующее цветовое выделение (цвет выделения может выбираться в настройках: окно “Настройки” → “Основные” → “Цвет подсветки” — см. Рис. 4) и переключаются поля свойств в нижней части экрана. При этом, свойства недоступные для данного элемента блокируются. Справа от поля редактирования свойства расположено поле для установки флажка, который указывает на необходимость включать данное свойство в результирующий код диалога. Свойство не включается в код диалога, пока оно не будет отмечено флажком, даже несмотря на наличие в нём какого-либо значения. Для визуального выделения неотмеченных свойств дополнительно используется серый цвет для текста в соответствующих полях редактирования. Редактирование свойств организовано по принципу “LiveUpdate” — при котором любое изменение свойства сразу отображается на внешнем виде элемента в макете.

### Перемещение и удаление элементов

Любой выделенный элемент можно удалить или переместить в рамках своего контейнера. Эти операции допускаются над целыми группами элементов. Для их выполнения служат кнопки в нижней области дерева элементов, расположенного в правой части окна приложения. После начала процесса перемещения элемента все действия с документом блокируются до тех пор, пока не будет использована кнопка обновления документа. После начала процесса перемещения не допускается переключаться между вкладками документа, пока не будет выполнена операция «обновления/перезагрузки» документа, в то же время перед обновлением документа можно выполнить любое кол-во перемещений для одного или нескольких элементов диалога. О необходимости перезагрузки документа дополнительно свидетельствует блокирование всех функциональных областей интерфейса главного окна приложения и красный цвет иконки на кнопке «обновить».

Операция «обновления/перезагрузки» документа также оказывается полезной для нормализации размеров диалога в результате удаления или редактирования отдельных его элементов. В результате выполнения этой процедуры все элементы диалога переинициализируются таким образом, как если бы они создавались в рамках окна средствами самой подсистемы ScriptUI. Перезагрузка документа оказывается крайне полезной для нормализации размеров объектов TabbedPanel, а также выравнивания размеров элементов и контейнеров в результате удаления смежных элементов управления или смены настроек `spacing`/межэлементных расстояний (в случаях, когда программа не смогла выполнить это автоматически). Операция «обновления/перезагрузки» документа не подразумевает операцию «сохранение» и не отражается на текущем содержимом файла документа.

### Копирование и вставка элементов

Копирование и вставка (в т.ч. и вырезание) осуществляется с сохранением настроек для всех свойств копируемых элементов. Элементы можно копировать как в рамках одного документа, так и между разными открытыми документами. Копироваться могут как элементы, так и группы элементов с неограниченной глубиной вложенности. Операции копирования/вставки могут выполняться как с помощью горячих клавиш (см. «Работа с программой») так и с помощью соответствующих кнопок в заголовочной панели приложения. При этом, для клавиши «Вставка» предусмотрена всплывающая подсказка, содержащая код элемента, находящегося в данный момент времени в буфере обмена.

Вставка элементов всегда происходит в «хвост» текущего активного контейнера (группы или панели) в документе-приёмнике. После вставки элемента его, в случае необходимости, можно перемещать в произвольном порядке. После перемещения следует «перезагрузить документ», если перемещение не выполняется — перезагрузка не требуется.



## Редактирование имён переменных

При добавлении элементов в диалог, для них происходит автоматическое генерирование имён переменных. Переменные генерируются на основании шаблона, заданного в настройках на вкладке «Имена переменных». На выбор доступен один из трёх шаблонов: “small” – короткие двух- и трёх-буквенные имена; “full” – длинные имена (установлено по умолчанию); “user” – пользовательская таблица имён (для незадаанных “пользовательских” значений используется значение по умолчанию, отображаемое серым шрифтом в соответствующем поле элемента на странице настроек коротких имён). При редактировании имени переменной для элемента в составе диалога следует не допускать использование одинакового имени для нескольких элементов в пределах одного контейнера, в то же время элементы из разных контейнеров могут иметь одинаковые имена.

## Открытие файлов

Программа допускает открывать любой javascript файл, содержащий код создания и отображения ScriptU-диалога. Предварительно файл сценария может быть создан с помощью любого инструмента (редактора или IDE), главное, что бы он соответствовало определённым критериям:

- ресурсная строка в объявлении “var <win> = new Window({...})”, где <win> — имя переменной диалога, разбита на строки, причём один графический компонент занимает строго одну строку;
- следом за ресурсной строкой следуют блоки с объявлениями локальных ссылок на компоненты диалога и инициализирующим кодом. Блоки между собой могут быть разделены пустой строкой. Началом блока выступает объявление вида “var <control> = <win>.<...>;”, где <control> — имя элемента, за которым могут следовать объявления вида “var gfx = <control>.graphics;” и т.п.
- последней значащей строкой считается строка “<win>.show();”, всё остальное — игнорируется.
- все строки до объявления диалогового окна “var <win> = new Window({...})” и после инструкции его открытия “<win>.show();” — игнорируются, всё что между ними — отработывается при открытии файла.

## Сохранение и контроль результата

При окончании работы над макетом диалога его можно сохранить в файл скрипта. В ходе работы можно контролировать фактический внешний вид диалога с помощью кнопки «Eval» в заголовочной области приложения (см. Рис. 3.). Если в результате вид диалога существенно отличается от редактируемого макета — это, скорее всего, может свидетельствовать о том, что вы забыли поставить галочку напротив одного из установленных вами свойств одного из элементов (например, установили, но не отметили одно из свойств выравнивания “alignment” / “alignChildren” или размера “preferredSize” / “size” и т.п.).

Просмотреть текущий код без предварительного сохранения можно с помощью кнопки «Code». Также, при помощи кнопки “Открыть в ESTK” можно напрямую передать код диалога в редактор ExtendScript Toolkit (при этом происходит автоматическое сохранение сценария и, в случае необходимости, зарос имени файла).

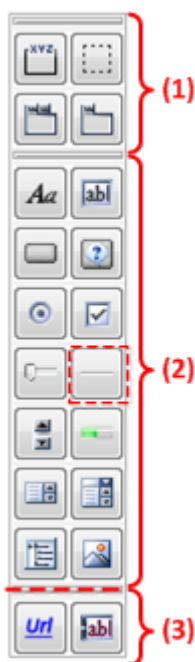
Следует знать, что при сохранении текущего документа его содержимое перезаписывается. Это важно, если файл предварительно редактировался и содержал дополнительный пользовательский код или комментарии — всё содержимое будет перезаписано автоматически сформированным кодом, относящимся исключительно к формированию диалогового окна. В связи с этим, в процессе разработки рекомендуется макет диалога и целевой сценарий держать в разных файлах.



## Интеграция с библиотекой SimpleUI

Библиотека SimpleUI состоит из нескольких, относительно независимых модулей, в числе которых модуль UIControls, содержащий набор дополнительных графических элементов управления (виджетов) таких, как **Separator**, **WebLink**, **UnitBox** и др... . В следующем разделе приводится краткое описание того, как Dialog Builder предлагает интегрировать данную библиотеку с пользовательскими сценариями и об использовании дополнительных элементов управления, которые в неё входят. Подробнее о структуре и составе дополнительных модулей, а также о компонентах самой библиотеки можно узнать в сопроводительной к ней документации (файл SimpleUI Guide.pdf, распространяемый вместе с приложением Dialog Builder) и на сайте GitHub с её репозиторием <https://github.com/SlavaBuck/Includes/tree/master/SimpleUI>.

### Дополнительные виджеты UIContros



Группа кнопок с элементами разделена на несколько групп:

- 1) Группа стандартных элементов ScriptUI контейнерного типа (Group, Panel и т.д.);
- 2) Группа стандартных элементов ScriptUI (StaticText, EditText и т.д.);
- 3) Группа дополнительных элементов из набора UIControls библиотеки SimpleUI.

Дополнительные элементы представляют собой либо расширенные версии стандартных элементов управления, либо являются комбинацией сразу нескольких стандартных элементов, представленных как отдельный расширенный элемент с собственным набором специальных свойств.

Использование дополнительных элементов (группа 3) требует включения в пользовательский файл сценария библиотеки SimpleUI. Необходимость включения библиотеки определяется автоматически в процессе анализа содержимого диалога на этапе его сохранения в файл.

Отдельного внимания заслуживает элемент **Separator** (обозначен красным прямоугольником в группе 2). В простейшем случае он представляет собой обычную разделительную линию между соседними ScripUI элементами и выступает в таком качестве как статический элемент Panel шириной в два пикселя, не требующий включения и поддержки со стороны дополнительных библиотек.

**Рис. 5** Кнопки элементов

Однако, библиотекой SimpleUI предусматривается расширенная версия этого элемента и возможность создания сепараторов, которые “умеют” реагировать на перетаскивание мышью и позволяют реализовать в рамках диалога области с возможностью их масштабирования в процессе работы. За тип сепаратора отвечает его свойство `dragged` (доступно на вкладке «Инициализация»).

Приложением Dialog Builder, на основании анализа содержимого диалогового окна, принимается решение о включении библиотеки SimpleUI в пользовательский файл сценария. Включение производится только в случае необходимости и происходит при условии наличия в файле сценария одного из элементов группы 3, или наличия хотя бы одного сепаратора с установленным для него свойством `dragged` в значение **true**. По умолчанию свойство имеет значение **false**.

Подробнее с дополнительными элементами управления можно познакомиться в справочной документации к библиотеке SimpleUI. Ниже коротко описаны дополнительные элементы из расширенного набора и приводится перечень свойств, разбитый по категориям, доступных для данных элементов в соответствующих вкладках нижней панели редактирования Dialog Builder:

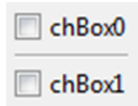


Рис 6. Простой сепаратор.

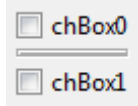


Рис 7. Перемещаемый сепаратор.

## Separator

Разделительная линия между двумя элементами диалога

Инициализация	
<b>dragged:</b> {bool}	Если <b>true</b> — линия динамическая (реагирует на перетаскивание мышкой, требуется поддержка библиотеки SimpleUI). По умолчанию — <b>false</b> , поддержка библиотеки не требуется.
Размеры	
<b>width:</b> {number}	Ширина линии сепаратора. По умолчанию: 2 для статической линии, 5 для динамической.

## WebLink

Гиперссылка. Элемент построен на базе стандартного ScriptUI-компонента StaticText. Реагирует на клик — открывает заданную ссылку в браузере. В дополнение к стандартным свойствам StaticText обладает следующими свойствами:

Текст	
<b>text:</b> {string}	Стандартное свойство для StaticText, отображаемое в качестве ссылки (значение присваивается произвольно, по умолчанию формируется на основе свойства weblink).
<b>weblink:</b> {string}	url-ссылка в корректном формате для передачи браузеру, например "http://domain.com/". По умолчанию, если не задано свойство text, ему будет присвоена часть строки weblink без "http://" и завершающей косой "/".

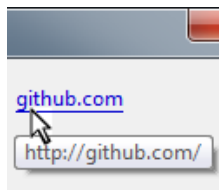


Рис 8. Элемент WebLink

## UnitBox

Настраиваемое редактируемое поле ввода, на базе стандартного элемента EditText для определённого типа числовых данных, которое определяется свойством unittype и представляет название одного из типов данных UnitValue (см. справку для UnitValue). Элемент обладает счётчиком для увеличения/уменьшения значений в заданном свойствами minvalue и maxvalue диапазоне с шагом, заданным значением stepdelta (аналогично как для ScriptUI-элемента Scroll). Кроме этого, элемент реагирует на клавиши "Up", "Down" и комбинации "Ctrl+Up", "Ctrl+Down" (Удержание клавиши Ctrl заставляет значение изменяться с шагом, заданным свойством jumpdelta, по умолчанию 20% от диапазона). Поле также позволяет на лету вычислять значения математических выражений. В случае наличия в поле редактирования некорректных данных (не числового типа или значений вне диапазона) — поле подсвечивается красным шрифтом.

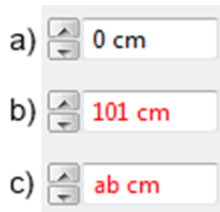


Рис 9. Элемент UnitBox  
(a — норм. состояние, b — вне диапазона, c — не число)

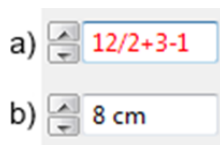


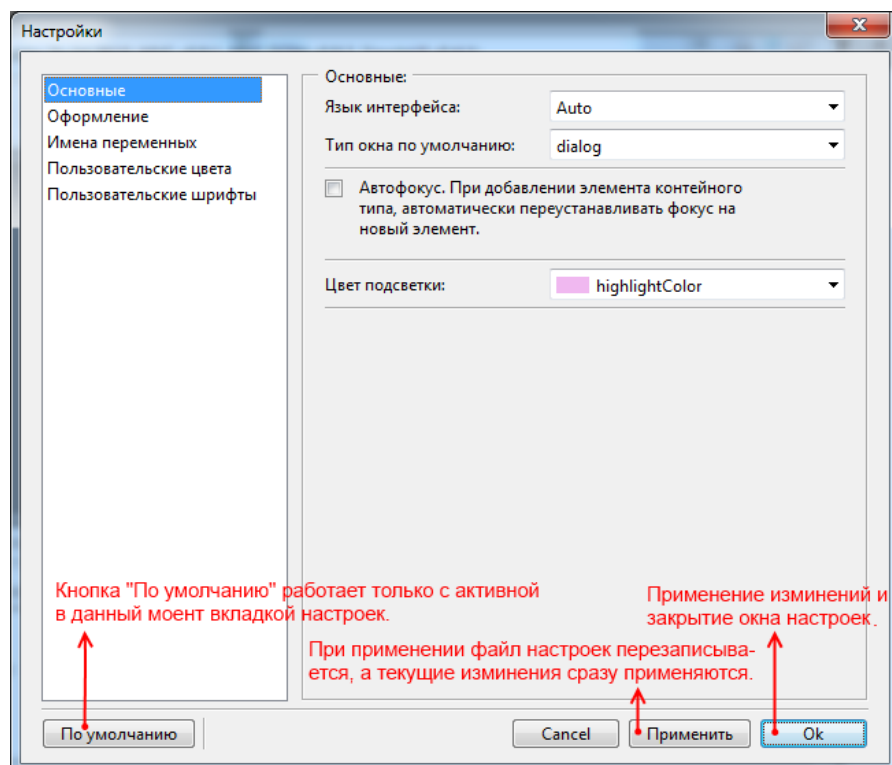
Рис 10. Вычисления в эл. UnitBox на лету  
(a — после ввода мат. выражения и нажатия Enter, b — поле получает результат вычисления мат. выражения)

Текст	
<b>unittype:</b> {string}	Тип объекта UnitValue, представленного в данном UnitBox-е свойством value и отображаемое в поле редактирования (см. раздел UnitValue object в справочном документе Adobe JavaScript Tools Guide)
Размеры	
<b>characters:</b> {number}	Размер поля редактирования (По умолчанию 8 симв.). Не исключается возможность установки размеров с помощью свойств preferredSize и size.
Инициализация	
<b>minvalue:</b> {number}	По умолчанию: 0 — нижняя граница значений value;
<b>maxvalue:</b> {number}	По умолчанию: 100 — верхняя граница значений value;
<b>stepdelta:</b> {number}	По умолчанию: 1 — шаг приращения значений value;
<b>jumpdelta:</b> {number}	По умолчанию: 20 — шаг приращения значений value при нажатии "Ctrl+Up/Down";
<b>value:</b> { UnitValue}	Отображаемое и редактируемое значение заданного типа

## Работа с настройками программы

Окно настроек доступно по нажатию кнопки «Настройки» в заголовочной области приложения. Первые три вкладки «Основные», «Оформление» и «Имена переменных» содержат все доступные опции для настройки приложения. Вкладки «Пользовательские цвета» и «Пользовательские шрифты» содержат соответствующие таблицы. Все добавленные в них значения становятся доступными в соответствующих элементах управления в рамках всего приложения. Все добавленные значения сохраняются в файле настроек и остаются доступными после перезапуска приложения.

Нажатие кнопок «Ок» и «Применить» приводит к перезаписи файла настроек и применению внесённых изменений. Нажатие кнопки «По умолчанию» сбрасывает значения настроек открытой в данный момент вкладки в значения по умолчанию.



### Особенности работы с настройками:

При запуске программы происходит попытка обнаружения файла *options.jsxinc* в папке Required, расположенной в папке с программой. В случае обнаружения файла (и наличия в нём сигнатуры <DBuilder options> ) происходит применение настроек, объявленных в файле. Если файл с настройками не обнаруживается – используются настройки по умолчанию. Файл автоматически создаётся и перезаписывается при применении или сохранении настроек из приложения.

Файл допустимо редактировать как с помощью графического интерфейса, предоставляемого приложением, так и в ручном режиме. Все допустимые настройки и их описание содержатся в файле *options.sample.jsxinc*, расположенном рядом с файлом настроек в той же папке Required.

При ручном редактировании файла настроек следует знать, что настройки хранятся в JSON формате, принятом для объектов JavaScript. Таким образом, в файле они должны синтаксически представлять корректный JavaScript-объект, в случае нарушения правил синтаксиса или наличия других ошибок, содержимое файла настроек будет проигнорировано и перезаписано значениями по умолчанию (наиболее частая ошибка – пропуск запятой после определения свойства и другие подобные синтаксические ошибки, специфичные для JavaScript...).

## Методика и рекомендации разработки диалога в приложении Dialog Builder

Данный раздел предполагает наличие определённых базовых знаний и некоторого опыта создания диалогов в среде ExtendScript. При разработке и создании макета для диалога следует понимать общие принципы работы системного компоновщика элементов (AutoLayout Manager) для графической подсистемы ScriptUI. Получить базовые навыки работы со ScriptUI можно с помощью замечательно работы **ScriptUI for dummies**, автором которой является Peter Kahrel. Последнюю версию документа можно скачать по адресу — <http://www.kahrel.plus.com/indesign/scriptui.html>, также на сайте <http://adobeindesign.ru/> на странице <http://adobeindesign.ru/2010/12/26/scriptsui/> доступен перевод (более ранней версии) этого руководства (прямая ссылка на архив с переводом: <http://adobeindesign.ru/wp-content/uploads/Написание-скриптов.-Пользовательский-интерфейс26.12.2010.zip>).

Приступая к работе над макетом диалога, следует знать, что в пределах контейнера (элемента контейнерного типа, которым выступает один из элементов group, panel, tabbedpanel, tab и сам объект диалога: dialog или palette) элементы располагаются один за другим только по вертикали или по горизонтали. Свойство orientation в контейнере отвечает за способ размещения в нём элементов. Оно имеет текстовый формат и может обладать одним из трёх значений: 'row' (элементы располагаются горизонтально в ряд, по умолчанию для контейнеров group); 'column' (элементы располагаются вертикально, по умолчанию для всех, кроме group) и 'stack' (элементы располагаются один над другим).

Для расположения нескольких различных элементов с требуемой ориентацией относительно друг друга существует два основных способа: 1) Вычислять и задавать требуемые значения свойства location: [x, y] для каждого элемента управления; 2) Объединять элементы с помощью контейнеров в группы и размещать такие группы с помощью родительских контейнеров в нужной ориентации. Второй способ хоть и звучит сложно, на деле является гораздо более простым и наглядным, так как не требует дополнительных вычислений и легко реализуем на практике. Хотя приложение Dialog Builder позволяет использовать оба подхода, именно второй способ является основным в силу своей простоты и наглядности.

Таким образом, общий подход в разработке макета диалога с помощью конструктора диалогов Dialog Builder заключается в создании необходимой структуры вложенных, размещением в них требуемых элементов управления и, в случае необходимости, настройки свойств выравнивания для них. Свойство "alignment" отвечает за настройку выравнивания самого элемента, для контейнеров дополнительно доступно свойство "alignChildren", отвечающее за выравнивание для всех элементов в контейнере.

Проще всего понять вышеизложенный подход позволяет короткое видео на сайте YouTube <http://youtu.be/i6P0OuBvmql>, в котором продемонстрировано создание небольшого диалога, приведённого на рис. 3 в данном руководстве.

Вторым, исключительно практическим аспектом, является контроль над размерами определённых блоков диалога или всего диалога в целом, с целью придания ему и его составляющим элементам нужных пропорций, выравнивания и размещения. Тут также возможны варианты. Например, можно задавать нужное свойство preferredSize: [x, y] для контейнеров и использовать свойство alignment в элементах управления или alignChildren – в самих контейнерах.

На практике, практически никогда не нужно использовать заранее предопределённые свойства размеров (`preferredSize` или `size`) для элементов управления контейнерного типа. Вместо этого, следует задавать требуемые размеры самим элементам управления, а расчёт и масштабирование контейнеров оставить на попечение у системного компоновщика — `AutoLayout Manager`-а. При этом, если элемент управления позволяет задавать собственный размер с помощью свойства `characters` (`StaticText`, `EditText`, ...) — лучше использовать именно его, такой подход избавит вас от неприятностей неправильного размера и неудачного обрезания текста в элементах, при использовании для них различных размеров шрифтов и начертаний.

На практике могут возникать ситуации, когда в результате удаления элементов или редактирования некоторых их свойств (например, это характерно для элемента «`TabbedPanel`») программа не может корректно обновить размеры `ui`-элементов в редактируемом диалоге. В таком случае следует воспользоваться клавишей «Обновление/Перезагрузка». В результате выполнения данной процедуры будут переинициализированы все элементы диалога так же, как это происходит в рамках обработки элементов нативными средствами `ScriptUI`-компоновщика. Таким образом, элементы примут ту же форму и размеры, какие они будут иметь в рамках создания диалога в результате реального вызова метода `new Window("...")`.

## Дополнительные сведения

- Домашняя страница проекта: <https://github.com/SlavaBuck/DialogBuilder>;
- Ветка обсуждения приложения на форумеRuDTP : <http://forum.rudtp.ru/threads/dialog-builder-v1-18-texnicheskaja.58536/>;
- Открытый репозиторий библиотеки SimpleUI: <https://github.com/SlavaBuck/Includes/tree/master/SimpleUI>;
- Связаться с автором можно со страницы проекта либо по эл. почте [slava.boyko@hotmail.com](mailto:slava.boyko@hotmail.com);

## Обновления Dialog Builder в версии 1.90:

- Добавлена интеграция с Библиотекой SimpleUI:
  - Добавлены элементы `WebLink`, `UnitBox`, `Separator` (расширенная версия);
  - Реализован механизм внедрения кода библиотеки в пользовательский файл сценария.
- Исправлен ряд ошибок, среди наиболее существенных:
  - Открытие файла сценария в редакторе ESTK в случае неудачи приводило к зависанию;
  - Копирование одиночного элемента `Separator` происходило некорректно;
  - Устранён фриз и неадекватная реакция на обнуление полей в процессе редактирования свойств на вкладка «Размеры».
- Улучшен механизм формирования моделей при парсинге содержимого диалогового окна, как следствие, ускорено формирование дерева элементов, открытие и обновление документов, операций копирования и вставки сложных объектов диалога контейнерного типа;
- Полностью обновлена документация:
  - `DialogBuilder readme.pdf` кардинально обновлён и переименован в `DialogBuilder User Guide.pdf`;
  - Добавлен `DialogBuilder Programmer Guide.pdf` с краткой инструкцией для разработчиков;
  - Добавлен `SimpleUI Guide.pdf` с описанием библиотеки SimpleUI.

## Обновления в файле документации:

Версия	Описание	Дата	Автор(ы)
1.90	<ul style="list-style-type: none"><li>• DialogBuilder readme.pdf (версия 1.82) был существенно обновлён и переименован в DialogBuilder User Guide.pdf (номера версий продолжены);</li><li>• Добавился новый раздел «Интеграция с библиотекой SimpleUI»;</li><li>• Добавились иллюстрации и ссылки на дополнительные файлы с документацией;</li></ul>	20.08.2014	Вячеслав Бойко
1.82	<ul style="list-style-type: none"><li>• В соответствии с новой функциональностью обновлены разделы: «Предназначение и возможности», «Работа с программой», «Контроль и сохранение результатов»</li><li>• В разделе «Работа с программой» обновлена основная иллюстрация рабочей области приложения и добавлен список горячих клавиш;</li><li>• Добавлены подразделы: «Перемещение и удаление элементов», «Копирование и вставка элементов», «Обновления в версии X.XX»</li><li>• Изменён формат таблицы версий файла документации — полная история изменений удалена, теперь в ней содержаться только предыдущие и текущие изменения.</li></ul>	03.08.2014	Вячеслав Бойко

© Вячеслав Бойко, 2014

Украина, г. Киев

slava.boyko@hotmail.com