

Модуль Конструктор

Краткое описание

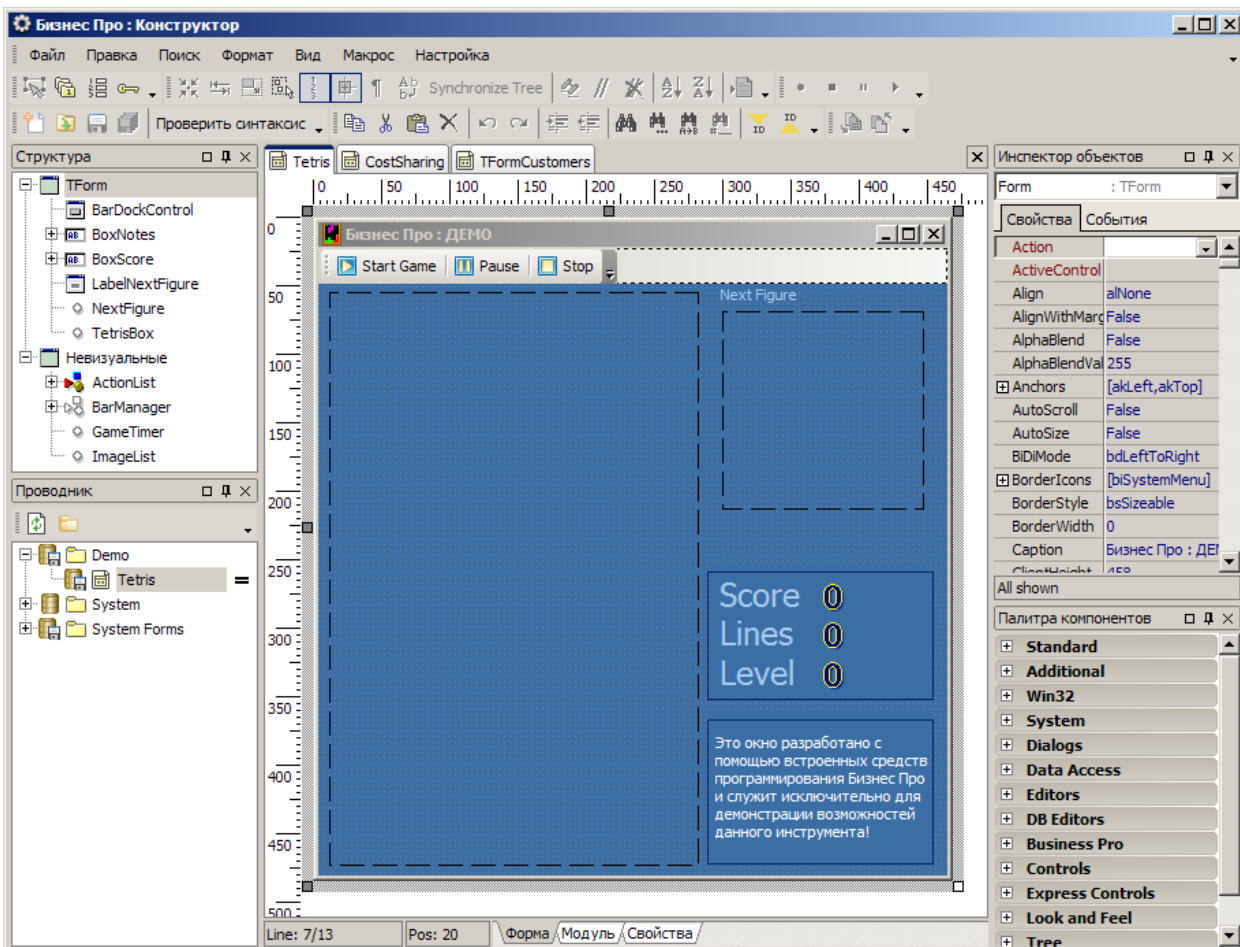
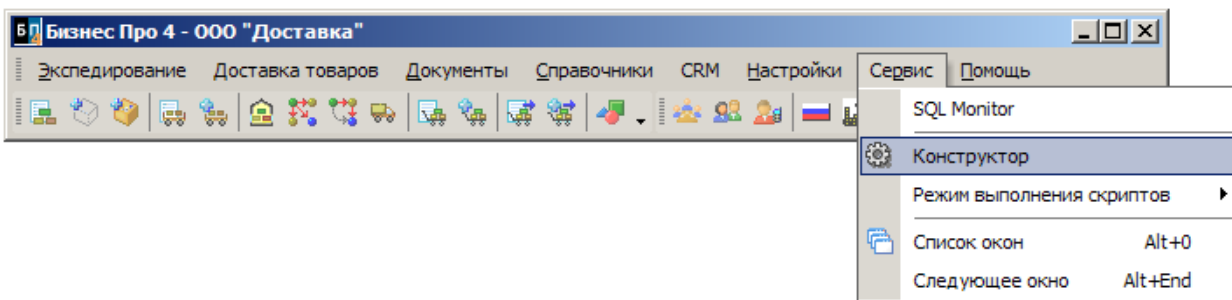
Введение

Конструктор является встроенным инструментом разработки в системе "Бизнес Про" и позволяет расширять функциональность и изменять логику работы системы. Конструктор предоставляет следующие возможности:

- модификация стандартных окон системы
- создание новых окон и связывание их с объектами системы
- написание собственной логики обработки данных (численные расчеты, алгоритмы распределения и т.п.)
- разработка произвольного дополнительного функционала, например процедур импорта/экспорта данных

В состав конструктора входят визуальный дизайнер форм и редактор исходного кода.

Весь процесс разработки происходит непосредственно в программе "Бизнес Про", вызов конструктора осуществляется командой меню [Сервис > Конструктор].



Лицензии

Конструктор является лицензируемым модулем системы и требует лицензии на каждое рабочее место программиста. При отсутствии лицензии ограничиваются возможности визуального дизайнера форм.

Термины

Конструктор - встроенный в систему "Бизнес Про" инструмент разработки

Рабочее место программиста - компьютер, на котором выполняется разработка с использованием конструктора

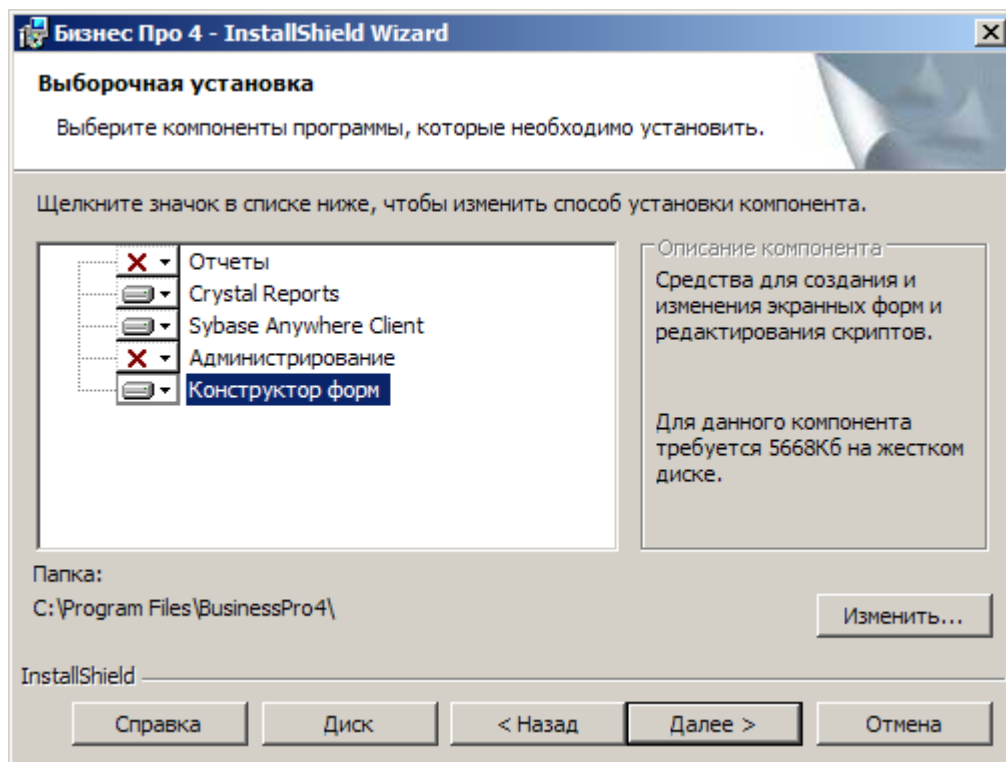
Форма - доступное для редактирования визуальное представление окна в конструкторе

Модуль - программа на встроенном языке программирования

Требования

Рабочее место программиста

На рабочее место программиста необходимо установить клиентскую часть "Бизнес Про", указав тип установки "Выборочная" и включив компонент "Конструктор форм".



Для полноценной работы с визуальным дизайнером форм необходимо, чтобы на рабочем месте программиста была установлена среда разработки Delphi 2006 for Win32 из состава Borland Developer Studio 2006. В отсутствие Delphi 2006 некоторые редакторы компонентов будут недоступны. С web-сайта CodeGear можно загрузить бесплатную версию [Delphi Turbo Explorer 2006](#).

Рабочее место пользователя

К рабочим местам пользователей дополнительных требований, кроме наличия установленной клиентской части "Бизнес Про", не предъявляется.

В случае, если для работы скриптов был произведен импорт новых модулей, отсутствующих в стандартной поставке системы, необходимо обеспечить наличие соответствующих пакетов во вложенной папке "Scripter" рабочей папки программы (по-умолчанию "C:\Program Files\BusinessPro4") на всех рабочих местах пользователей.

Разработка

Описание

Под разработкой скриптов понимаются следующие действия:

- визуальное редактирование форм
- написание программного кода
- тестирование скриптов путем вызова на экран форм и вызова программного кода на выполнение

Разработка скриптов может осуществляться на рабочей базе данных параллельно с работой пользователей или, в целях предосторожности, на копии базы данных.

Скрипт

Скрипт является именованной единицей функциональности. Для удобства скрипты организуются в папки по аналогии с обычными файловыми системами, таким образом любой скрипт может быть однозначно идентифицирован с помощью пути вида "Folder1\Folder2\MyScript". Как видно из примера, разделителем имен в пути является символ обратной косой черты "\". Скрипт может состоять из модуля и необязательной формы:

- Модуль содержит описание логики работы системы в виде программного кода на встроенном языке программирования [DelphiScript](#).
- Форма представляет собой пользовательский интерфейс, отображаемый на экране. На форме размещаются визуальные (поля ввода, кнопки, таблицы, меню, панели инструментов и т.д.) и невидимые компоненты (наборы данных, таймеры и т.д.). В модуле, связанном с формой, содержатся обработчики событий компонентов, размещенных на форме и вспомогательные процедуры/функции. В одном скрипте может содержаться описание не более чем одной формы.

Расположение

Рабочие версии скриптов хранятся в базе данных системы и загружаются на рабочие места пользователей для выполнения. При разработке программист может изменять существующие скрипты или создавать новые. Чтобы работа программиста не мешала работе пользователей, отладочные версии скриптов хранятся локально на рабочем месте программиста в виде файлов (по-умолчанию во вложенной папке "BusinessPro4 Scripts" папки "Мои документы"). Структура папок на рабочем месте программиста повторяет структуру папок скриптов в базе данных, каждому скрипту соответствует один файл на диске с именем, совпадающим с именем скрипта, и расширением ".xml". После отладки и тестирования программист "загружает" скрипт со своего диска в базу данных, тем самым делая его доступным для всех пользователей системы.

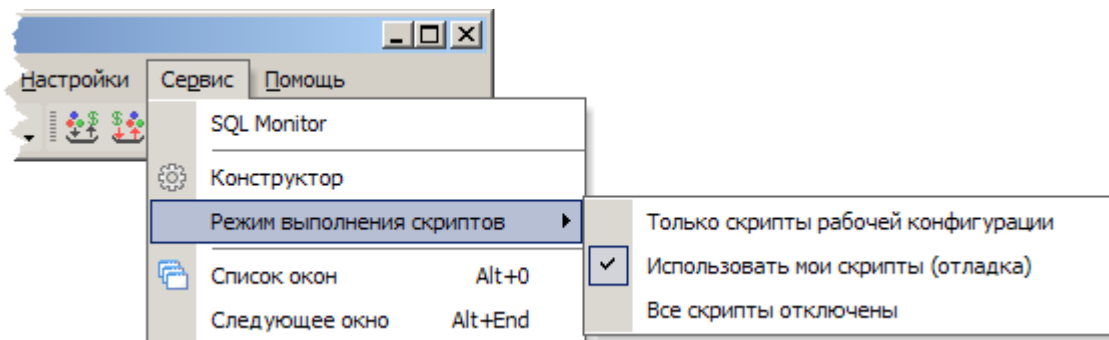
Режим выполнения

В процессе разработки существуют несколько наборов скриптов:

- рабочий набор скриптов, которые загружаются для обычных пользователей системы - хранится в базе данных
- собственный отладочный набор скриптов у каждого программиста - хранится локально на рабочем месте программиста

Настройка режима выполнения [Сервис > Режим выполнения] предназначена для программиста и определяет каким образом скрипты загружаются системой на выполнение на его рабочем месте:

1. **Только скрипты рабочей конфигурации** - загружаются только общедоступные скрипты из базы данных, отладочные скрипты на рабочем месте программиста игнорируются
2. **Использовать мои скрипты (отладка)** - загружаются отладочные скрипты с рабочего места программиста, если необходимый скрипт отсутствует локально, он загружается из базы данных
3. **Все скрипты отключены** - не загружаются никакие скрипты



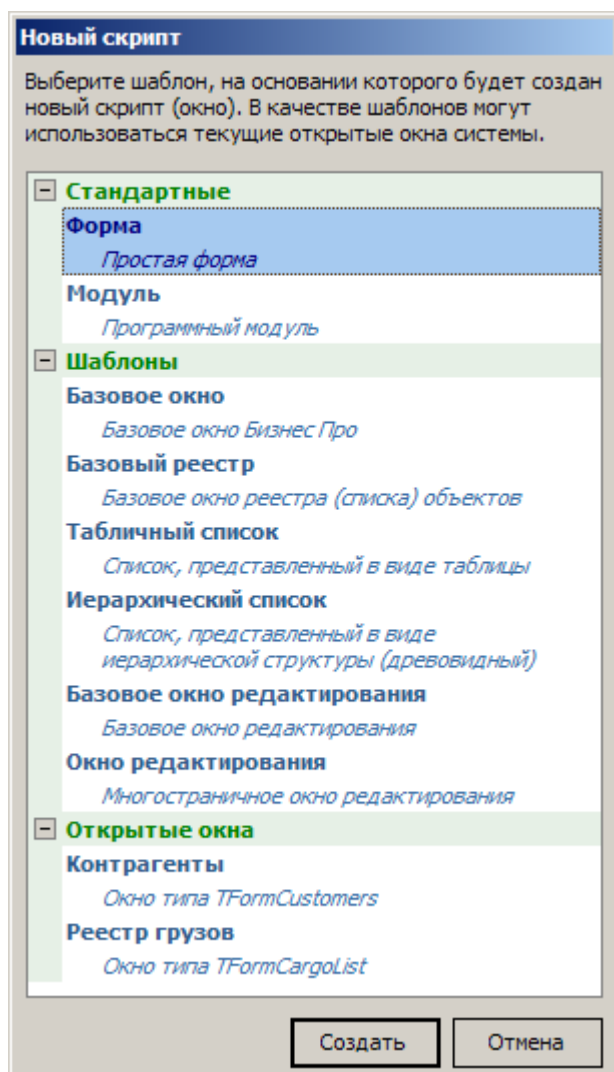
Вызов скриптов

Скрипт вызывается системой для выполнения в следующих случаях:

- Пользователь открывает стандартное окно системы, которое было модифицировано с помощью конструктора. В этом случае загружается модифицированная версия окна.
- В логике работы системы предусмотрен вызов функции из скрипта с определенным именем, например для выполнения расчета или проверки введенных данных.

Создание скриптов

Для создания нового скрипта необходимо выбрать в меню конструктора команду [Файл > Новый]. На экране отобразится окно выбора шаблона, предоставляющее возможность создать пустую форму, модуль (программный код без формы) или форму на основании шаблона.



Использование шаблона для формы означает, что вновь создаваемая форма уже будет включать в себя некоторый базовый пользовательский интерфейс и логику работу, предназначенную для типовых случаев использования, например реестра или окна редактирования. Шаблоны позволяют добиться того, что вновь разрабатываемые окна имеют стандартный для системы "Бизнес Про" пользовательский интерфейс и органично вписываются в работу системы.

Использование в качестве шаблона открытого окна системы означает, что изначально (т.е. до внесения каких-либо изменений с помощью конструктора) форма из нового скрипта будет вести себя в точности также как и выбранное окно системы. У программиста имеется возможность скорректировать работу окна, добавить или скрыть элементы пользовательского интерфейса, назначить дополнительные обработчики событий и т.д. Созданные таким образом формы являются модифицированными *копиями* стандартных окон системы и сами по себе ниоткуда вызываться не будут. Если необходимо изменить стандартное окно системы, смотрите раздел [Стандартные окна](#).

После выбора шаблона и нажатия кнопки "Создать" в конструкторе будет открыт на редактирование новый скрипт.

По окончании редактирования скрипта необходимо сохранить его с помощью команды [Файл > Сохранить]. На экран будет выведен диалог сохранения файла на диск, изначально указывающий на локальную папку со скриптами. Чтобы иметь возможность использовать скрипт, необходимо сохранить его в корневой папке скриптов или вложенных в нее папках, т.к. при необходимости загрузки скрипта система будет производить поиск на диске, начиная отсчет от корневой папки со скриптами. Таким образом, для загрузки скрипта "Folder1\Folder2\MyScript", система обратится к файлу "%UserProfile%\My Documents\BusinessPro4 Scripts\Folder1\Folder2\MyScript.xml".

Использование новых скриптов

Для создания объекта формы предназначена специальная глобальная функция CreateForm, принимающая в качестве параметра путь к скрипту, в котором описана форма. Пример:

```
procedure Button1Click(Sender: TObject);
var
  f: TForm;
begin
  f := CreateForm('MyForms\MyForm1');
  try
    f.ShowModal;
  finally
    f.Release;
  end;
end;
```

Для подключения скрипта используется конструкция uses, с помощью которой определенные в подключаемом скрипте объекты (процедуры, функции, константы) становятся доступными в текущем скрипте. Пример:

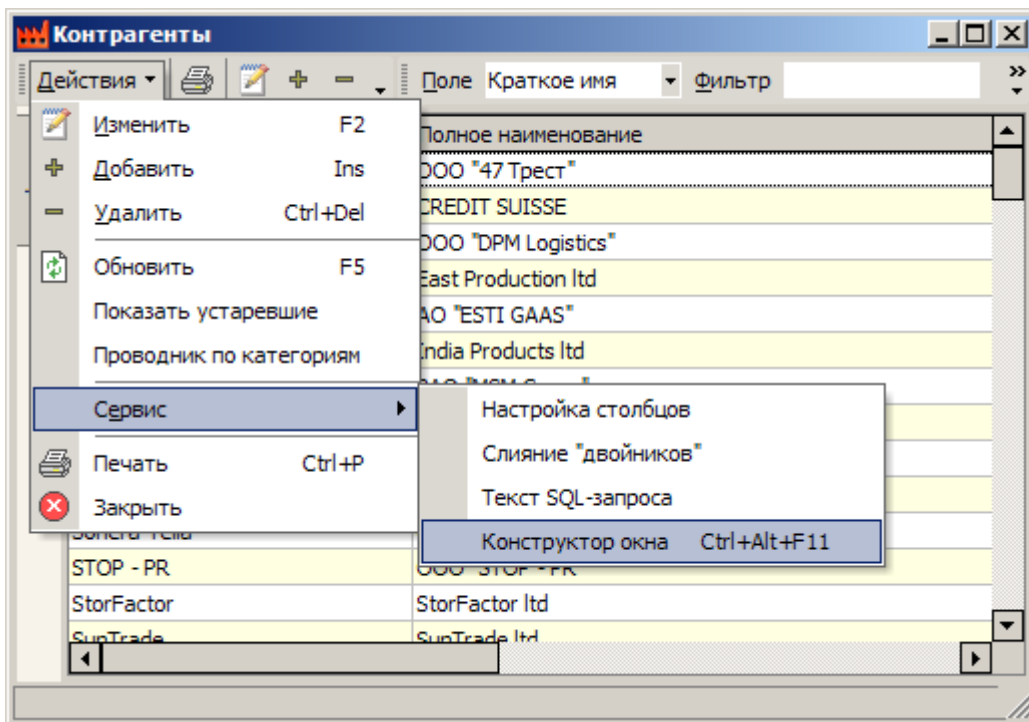
```
uses 'MyUnits\Unit1'; // обязательно заключайте имя скрипта в одиночные кавычки!
```

```
procedure Button1Click(Sender: TObject);
begin
  ProcFromUnit1; // определена в Unit1
end;
```

Стандартные окна

Изменение стандартного окна подразумевает, что во всех местах системы, где по логике работы происходит создание данного окна будет создаваться его модифицированная версия.

Для вызова стандартного окна на редактирование необходимо вызвать его на экран и выбрать в меню команду [Действия > Сервис > Конструктор окна] или нажать комбинацию клавиш Ctrl+Alt+F11.



В результате в конструкторе будет открыт скрипт, соответствующий окну "Контрагенты". После сохранения внесенные изменения будут видны во всех вновь открытых окнах данного класса.

Скрипты, отвечающие за стандартные окна системы, хранятся в специальной системной папке скриптов "System Forms". Имя скрипта соответствует классу формы (в случае реестра контрагентов это "TFormCustomers"). Чтобы отказаться от внесенных в стандартное окно изменений и вернуть исходное стандартное окно необходимо с помощью [проводника](#) из папки "System Forms" удалить скрипт с именем класса формы окна, в случае реестра контрагентов это будет скрипт "System Forms\TFormCustomers".

DelphiScript

Написание программного кода в системе выполняется на встроенном языке DelphiScript, который представляет собой подмножество языка Delphi.

Ограничения

Возможности Delphi, отсутствующие в DelphiScript:

- интерфейсы
- процедурные указатели
- метаклассы

Работа с множествами

В DelphiScript не поддерживается конструктор множеств "[]" и оператор проверки принадлежности множеству "in". Их заменяют следующие функции:

MkSet - создание множества

```
Font.Style := MkSet(fsBold, fsItalic); // вместо Font.Style := [fsBold, fsItalic]
```

InSet - проверка на вхождение в множество

```
if InSet(fsBold, Font.Style) then ShowMessage('Font is bold!'); // вместо if (fsBold in Font.Style) then ...
```


Интерфейс

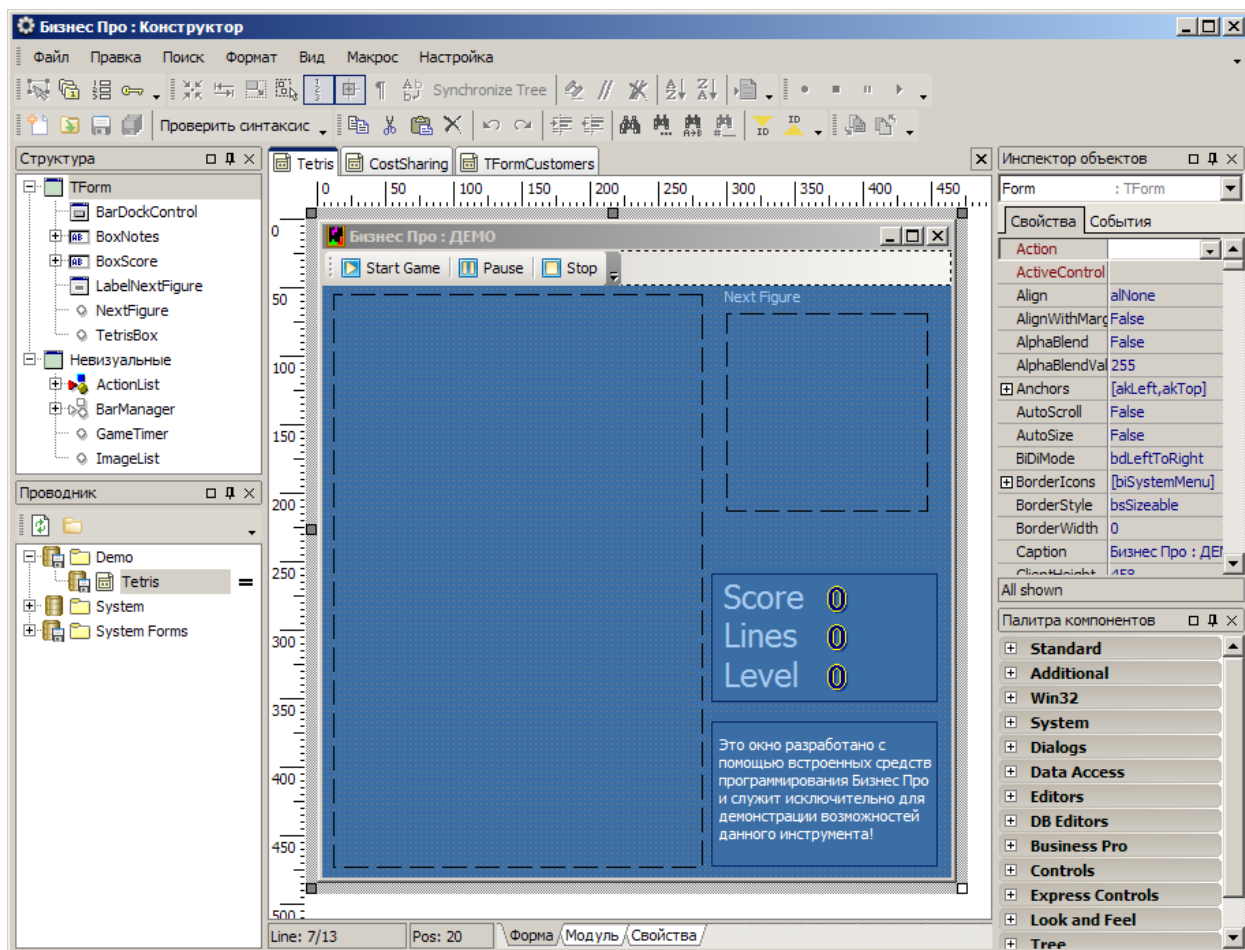
Конструктор

Главное окно конструктора состоит из рабочей области в центре и вспомогательных окон: [структура](#), [проводник](#), [инспектор объектов](#) и [палитра компонентов](#).

В рабочей области могут быть открыты на редактирование несколько скриптов, переключение между ними осуществляется с помощью закладок в верхней части.

Для редактирования скрипта в нижней части имеются вкладки:

- "Форма" - визуальный дизайнер формы. Отсутствует в случае, если скрипт состоит только из модуля.
- "Модуль" - редактор исходного кода.
- "Свойства" - примечание.



Горячие клавиши

- **F12** - переключение между формой и модулем
- **Alt+F12** - переключение между графическим и текстовым представлением формы

Структура

Структура отображает в виде дерева иерархические отношения между размещенными на форме компонентами.

Некоторые визуальные компоненты являются контейнерами для других визуальных компонентов, т.е. на их поверхность могут быть помещены любые визуальные компоненты. В таком случае, в структуре узел

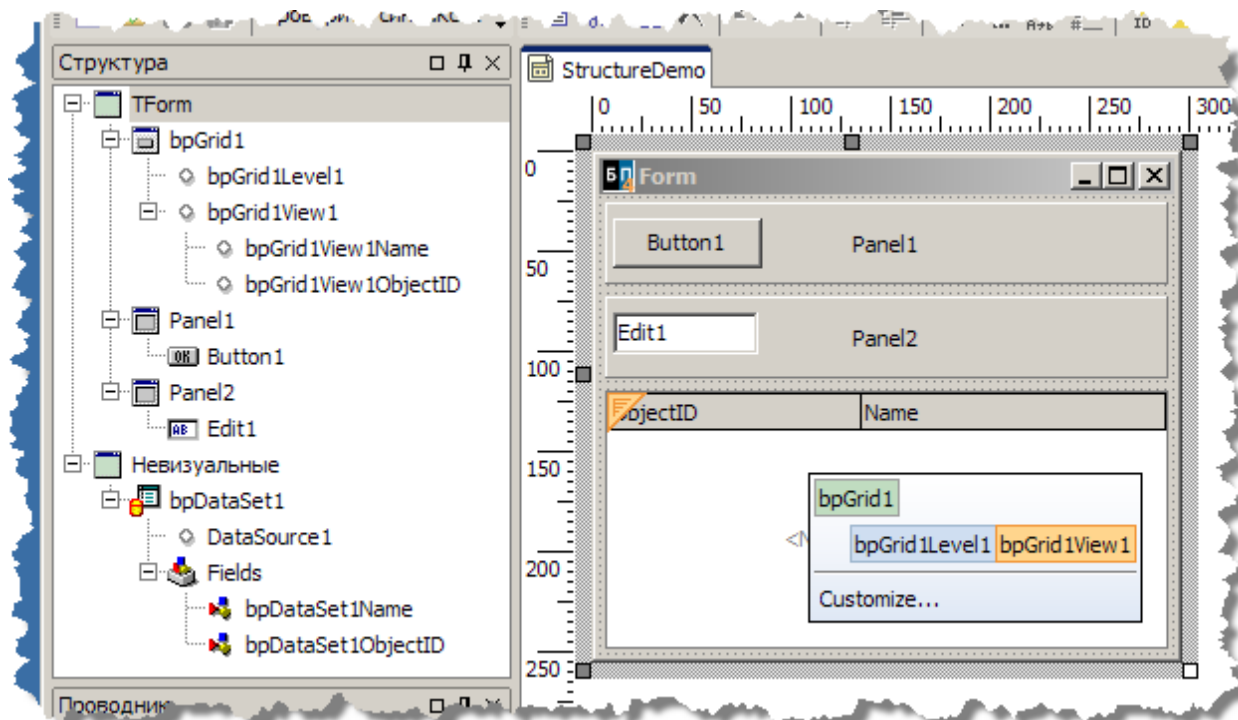
контейнера будет содержать в качестве дочерних узлов все размещенные на контейнере компоненты. Примерами контейнеров являются компоненты TPanel и TGroupBox.

Некоторые компоненты (как визуальные, так и невидимые) являются неотъемлемой частью своего родительского компонента и не могут существовать вне его, например поля набора данных TbpDataSet или столбцы таблицы TbpGrid. В структуре узлы таких компонентов являются дочерними для узла родительского компонента.

С помощью окна структуры можно совершать с компонентами следующие действия:

- с помощью перетаскивания (drag-and-drop) изменять принадлежность визуальных компонентов родительским контейнерам
- двойным щелчком мыши по узлу компонента вызывать на экран редактор компонента, если таковой назначен

Для удобства визуальные и невидимые компоненты разделены в структуре на два поддерева.

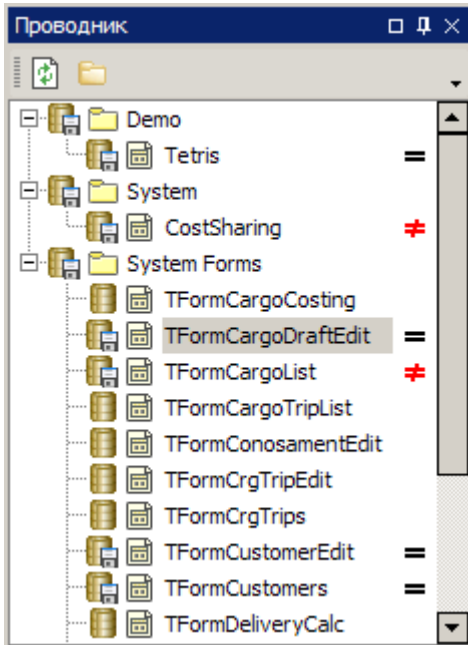


В примере на рисунке на форме размещены:

- два компонента-контейнера TPanel, на первом из которых расположена кнопка Button1, на втором - поле ввода Edit1
- набор данных bpDataSet1 с двумя полями bpDataSet1ObjectID и bpDataSet1Name и источник DataSource1 для него
- таблица bpGrid1View1, отображающая набор данных bpDataSet1 с двумя столбцами bpGrid1View1ObjectID и bpGrid1View1Name

Проводник








Проводник отображает файловую систему скриптов в виде дерева, содержащего скрипты и папки со скриптами, и позволяет управлять ими.



Проводник позволяет производить следующие действия при помощи контекстного меню:

- открывать скрипты на редактирование
- переименовывать скрипты и папки (клавиша F2)
- удалять скрипты и папки
- создавать новые папки и вложенные папки
- выгружать скрипты на диск
- загружать скрипты в базу данных

Значения пиктограмм:

-  скрипт
-  папка
-  скрипт или папка присутствует только в базе данных
-  скрипт или папка присутствует только на диске
-  скрипт или папка присутствует и в базе данных и на диске
-  скрипт на диске и скрипт в базе данных различаются
-  скрипт на диске и скрипт в базе данных одинаковы

Инспектор объектов

Предназначен для задания свойств и обработчиков событий выделенного на форме компонента.

Для установки свойства компонента необходимо:

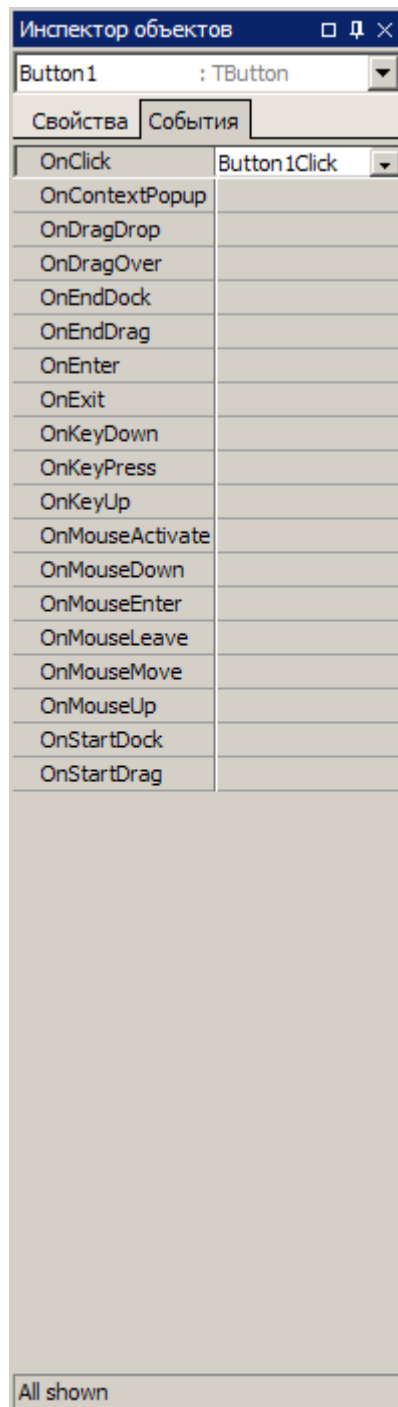
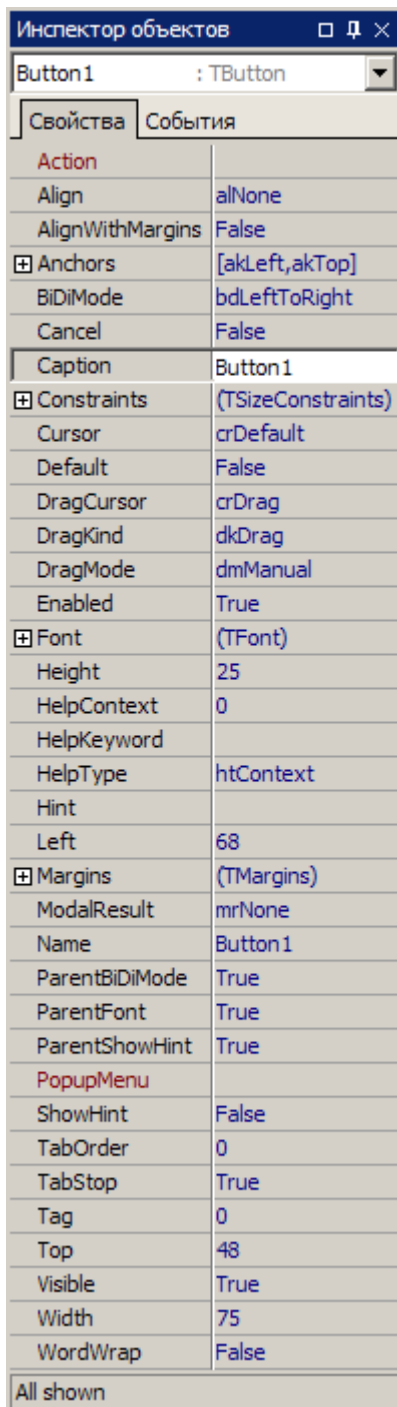
1. На форме или в структуре выполнить щелчок по компоненту, чтобы выделить его.
2. Переключить инспектор объектов на закладку "Свойства".
3. Выбрать свойство, которое требуется изменить и, в зависимости от типа свойства, ввести текст в поле ввода, выбрать значение из выпадающего списка или щелкнуть по многоточию [...] справа от поля ввода и воспользоваться ассоциированным редактором свойства.

Для установки обработчика события компонента необходимо:

1. На форме или в структуре выполнить щелчок по компоненту, чтобы выделить его.
2. Переключить инспектор объектов на закладку "События".

3. Если обработчик для события уже существует, то выбрать его в выпадающем списке. Иначе, для его создания, выполнить двойной щелчок мышью по полю ввода или ввести имя обработчика и нажать клавишу Enter. Произойдет переключение на редактор исходного кода.
4. Ввести код обработчика.

На рисунках показаны закладки инспектора объектов для компонента TButton:



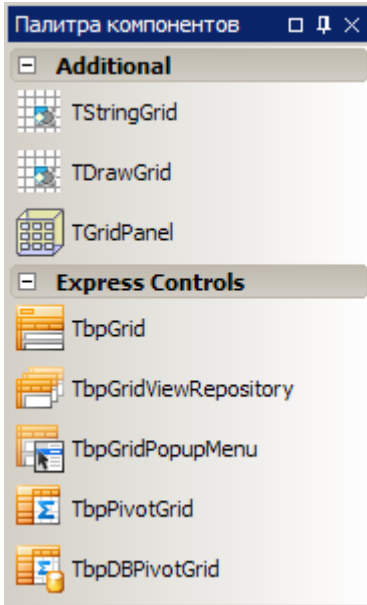
Палитра компонентов

На палитре отображаются все типы компонентов, которые могут быть помещены на форму, с разбиением на категории.

Чтобы поместить компонент на форму необходимо:

1. Выбрать его на палитре, щелкнув по строке компонента, после чего компонент на палитре окажется подсвеченным.
2. Выполнить щелчок левой кнопкой мыши в требуемом месте формы для создания компонента с размерами по-умолчанию или, удерживая нажатой левую клавишу мыши, начертить на форме прямоугольную область, тем самым задав размеры нового компонента.

Если необходимо создать сразу несколько копий компонента можно при выборе компонента на палитре удерживать клавишу Shift.



Палитра поддерживает фильтрацию по имени компонента. Чтобы найти компонент, необходимо установить фокус на палитру компонентов и набрать на клавиатуре часть имени компонента. После этого на палитре останутся лишь те компоненты, имя которых включает введенную строку. Для отмены режима фильтрации служит клавиша Escape.

На рисунке показана палитра компонентов, отфильтрованная по строке "grid".