В этом обзоре рассматриваются преимущества интеллектуальных клиентских приложений, основные возможности программирования Windows Forms, а также использование Windows Forms для разработки интеллектуальных клиентов, удовлетворяющих запросам современных предприятий и конечных пользователей.

[Windows Forms и интеллектуальные клиентские приложения](javascript:void(0))

Windows Forms позволяет разрабатывать интеллектуальные клиенты.   *Интеллектуальный клиент* — это приложение с полнофункциональным графическим интерфейсом, простое в развертывании и обновлении, способное работать при наличии или отсутствии подключения к Интернету и использующее более безопасный доступ к ресурсам на локальном компьютере по сравнению с традиционными приложениями Windows.

[Построение многофункциональных интерактивных пользовательских интерфейсов](javascript:void(0))

Windows Forms — это технология интеллектуальных клиентов для .NET Framework. Она представляет собой набор управляемых библиотек, упрощающих выполнение стандартных задач, таких как чтение из файловой системы и запись в нее.  С помощью среды разработки типа Visual Studio можно создавать интеллектуальные клиентские приложения Windows Forms, которые отображают информацию, запрашивают ввод от пользователей и обмениваются данными с удаленными компьютерами по сети.

В Windows Forms *форма* — это видимая поверхность, на которой выводится информация для пользователя.  Обычно приложение Windows Forms строится путем помещения элементов управления на форму и написания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш.   *Элемент управления* — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие.  Приложение реагирует на эти события с помощью кода и обрабатывает события при их возникновении.  Подробнее см. в разделе [Creating Event Handlers in Windows Forms](https://msdn.microsoft.com/ru-ru/library/dacysss4(v=vs.110).aspx).

Windows Forms включает широкий набор элементов управления, которые можно добавлять на формы: текстовые поля, кнопки, раскрывающиеся списки, переключатели и даже веб-страницы.  Список всех элементов управления, которые можно использовать в форме, представлены в разделе [Элементы управления для использования в формах Windows Forms](https://msdn.microsoft.com/ru-ru/library/3xdhey7w(v=vs.110).aspx).  Если существующий элемент управления не удовлетворяет потребностям, в Windows Forms можно создать пользовательские элементы управления с помощью класса [UserControl](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.usercontrol(v=vs.110).aspx).

В состав Windows Forms входят многофункциональные элементы пользовательского интерфейса, позволяющие воссоздавать возможности таких сложных приложений, как Microsoft Office.  Используя элементы управления [ToolStrip](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.toolstrip(v=vs.110).aspx) и[MenuStrip](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.menustrip(v=vs.110).aspx), можно создавать панели инструментов и меню, содержащие текст и рисунки, подменю и другие элементы управления, такие как текстовые поля и поля со списками.

С помощью поддерживающего перетаскивание конструктора Windows Forms в Visual Studio можно легко создавать приложения Windows Forms.  Достаточно выделить элемент управления курсором и поместить его в нужное место на форме.  Для преодоления трудностей, связанных с выравниванием элементов управления, конструктор предоставляет такие средства, как линии сетки и линии привязки.  И при использовании Visual Studio, и при компиляции из командной строки вы можете использовать элементы управления [FlowLayoutPanel](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.flowlayoutpanel(v=vs.110).aspx), [TableLayoutPanel](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.tablelayoutpanel(v=vs.110).aspx) и [SplitContainer](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.splitcontainer(v=vs.110).aspx) для создания сложных макетов форм за меньшее время.

Наконец, если нужно создать свои собственные элементы пользовательского интерфейса, пространство имен[System.Drawing](https://msdn.microsoft.com/ru-ru/library/system.drawing(v=vs.110).aspx) содержит широкий набор классов, необходимых для отрисовки линий, кругов и других фигур непосредственно на форме.

|  |
| --- |
| **System_CAPS_noteПримечание** |
| Элементы управления Windows Forms не предназначены для маршалинга между доменами приложений.  По этой причине технологии Майкрософт не поддерживают передачу элементов управления Windows Forms через границы[AppDomain](https://msdn.microsoft.com/ru-ru/library/system.appdomain(v=vs.110).aspx), хотя на первый взгляд базовый тип [Control](https://msdn.microsoft.com/ru-ru/library/system.windows.controls.control(v=vs.110).aspx) класса [MarshalByRefObject](https://msdn.microsoft.com/ru-ru/library/system.marshalbyrefobject(v=vs.110).aspx) подразумевает такую возможность.  Приложения Windows Forms с несколькими доменами приложений поддерживаются только при условии, что элементы управления Windows Forms не передаются через границы доменов приложения. |

[Справка по созданию форм и элементов управления](javascript:void(0))

Пошаговые инструкции по использованию этих возможностей можно найти в приведенных ниже разделах справки.

|  |  |
| --- | --- |
| **Описание** | **Раздел справки** |
| Использование элементов управления в формах | [Практическое руководство. Добавление элементов управления в формы Windows Forms.](https://msdn.microsoft.com/ru-ru/library/0h5y8567(v=vs.110).aspx) |
| Использование элемента управления [ToolStrip](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.toolstrip(v=vs.110).aspx) | [Практическое руководство. Создание в форме Windows Forms базового элемента управления ToolStrip со стандартными элементами с помощью конструктора](https://msdn.microsoft.com/ru-ru/library/571c1z99(v=vs.110).aspx) |
| Создание графических элементов с помощью[System.Drawing](https://msdn.microsoft.com/ru-ru/library/system.drawing(v=vs.110).aspx) | [Приступая к программированию графики](https://msdn.microsoft.com/ru-ru/library/da0f23z7(v=vs.110).aspx) |
| Создание пользовательских элементов управления | [Практическое руководство. Наследование класса UserControl.](https://msdn.microsoft.com/ru-ru/library/00ctb4z0(v=vs.110).aspx) |

[Отображение и обработка данных](javascript:void(0))

Во многих приложениях нужно отображать данные из базы данных, XML-файла, веб-службы XML или другого источника данных.  Windows Forms предоставляет гибкий элемент управления с именем [DataGridView](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview(v=vs.110).aspx) для отображения таких табличных данных в традиционном формате строк и столбцов так, что каждый фрагмент данных занимает свою собственную ячейку.  С помощью [DataGridView](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview(v=vs.110).aspx) можно, помимо прочего, настроить внешний вид отдельных ячеек, зафиксировать строки и столбцы на своем месте, а также обеспечить отображение сложных элементов управления внутри ячеек.

При использовании интеллектуальных клиентов Windows Forms можно легко подключаться к источникам данных по сети.  Компонент Windows Forms [BindingSource](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.bindingsource(v=vs.110).aspx), появившийся в Visual Studio 2005 и .NET Framework 2.0, представляет подключение к источнику данных и содержит методы для привязки данных к элементам управления, перехода к предыдущей или следующей записи, редактирования записей и сохранения изменений в исходном источнике.  Элемент управления [BindingNavigator](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.bindingnavigator(v=vs.110).aspx) предоставляет простой интерфейс на основе компонента [BindingSource](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.bindingsource(v=vs.110).aspx) для перехода между записями.

Вы можете легко создавать элементы управления с привязкой к данным с помощью окна "Источники данных".  В нем приводятся имеющиеся в проекте источники данных, такие как базы данных, веб-службы и объекты.  Создавать элементы управления с привязкой к данным можно путем перетаскивания объектов из этого окна в формы проекта.  Также можно связывать существующие элементы управления с данными, перетаскивая объекты из окна "Источники данных" в существующие элементы управления.

Другой тип привязки к данным в формах Windows Forms — это *параметры*.  Большинство интеллектуальных клиентских приложений должны сохранять некоторые сведения о своем состоянии во время выполнения, такие как последние известные размеры форм, а также сохранять пользовательские предпочтения, например место сохранения файлов по умолчанию.  Параметры приложения отвечает этим требованиям, предоставляя простой способ хранения обоих типов сведений на клиентском компьютере.  Параметры, определенные один раз с помощью Visual Studio или редактора кода, сохраняются в XML-файле и автоматически считываются обратно в память во время выполнения.

[Справка по отображению и обработке данных](javascript:void(0))

Пошаговые инструкции по использованию этих возможностей можно найти в приведенных ниже разделах справки.

|  |  |
| --- | --- |
| **Описание** | **Раздел справки** |
| Использование компонента [BindingSource](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.bindingsource(v=vs.110).aspx) | [Практическое руководство. Связывание элементов управления Windows Forms с компонентом BindingSource с помощью конструктора](https://msdn.microsoft.com/ru-ru/library/801dxw2t(v=vs.110).aspx) |
| Работа с источниками данных ADO.NET | [Практическое руководство. Сортировка и фильтрация данных ADO.NET с помощью компонента BindingSource в Windows Forms](https://msdn.microsoft.com/ru-ru/library/ya3sah92(v=vs.110).aspx) |
| Использование окна "Источники данных" | [Пошаговое руководство. Отображение данных на форме в приложении Windows](https://msdn.microsoft.com/ru-ru/library/wwh8ka92(v=vs.110).aspx) |
| Использование параметров приложения | [How to: Create Application Settings](https://msdn.microsoft.com/ru-ru/library/ms171565(v=vs.110).aspx) |

[Развертывание приложений на клиентских компьютерах](javascript:void(0))

После создания приложения необходимо отправить его пользователям, чтобы они могли установить и запустить его на своих клиентских компьютерах.  С помощью технологии ClickOnce можно развертывать приложения из Visual Studio всего в несколько щелчков и предоставлять пользователям URL-адрес приложения в Интернете.  ClickOnce управляет всеми элементами и зависимостями в приложении и обеспечивает его правильную установку на клиентском компьютере.

Приложения ClickOnce можно настроить так, чтобы они работали только при подключении к сети или как в сетевом, так и в автономном режиме.  Если вы указываете, что приложение должно поддерживать работу в автономном режиме, то технология ClickOnce добавляет ссылку на приложение в меню **Пуск** пользователя.  Пользователь может открыть приложение без использования URL-адреса.

Когда вы обновляете приложение, на веб-сервере публикуется новый манифест развертывания и новая копия приложения.  ClickOnce обнаруживает доступное обновление и обновляет пользовательскую установку. Для обновления старых сборок написание специального кода не требуется.

[Справка по развертыванию приложений ClickOnce](javascript:void(0))

Общие сведения о ClickOnce можно найти в разделе [Развертывание и безопасность технологии ClickOnce](https://msdn.microsoft.com/ru-ru/library/t71a733d(v=vs.110).aspx).  Пошаговые инструкции по использованию этих возможностей можно найти в приведенных ниже разделах справки.

|  |  |
| --- | --- |
| **Описание** | **Раздел справки** |
| Развертывание приложения с помощью ClickOnce | [Практическое руководство. Публикация приложения ClickOnce с помощью мастера публикации](https://msdn.microsoft.com/ru-ru/library/31kztyey(v=vs.110).aspx)  [Разбор примера: развертывание вручную приложения ClickOnce](https://msdn.microsoft.com/ru-ru/library/xc3tc5xx(v=vs.110).aspx) |
| Обновление развертывания ClickOnce | [Практическое руководство. Управление обновлениями для ClickOnce-приложения](https://msdn.microsoft.com/ru-ru/library/1zyc39fb(v=vs.110).aspx) |
| Управление безопасностью с помощью ClickOnce | [Практическое руководство. Включение параметров безопасности ClickOnce-приложений.](https://msdn.microsoft.com/ru-ru/library/1sfbfyk0(v=vs.110).aspx) |

[Другие элементы управления и функции](javascript:void(0))

В Windows Forms имеется множество других возможностей, которые упрощают и ускоряют реализацию общих задач, таких как создание диалоговых окон, печать, добавление справки и документации, а также локализация приложений на различных языках.  Кроме того, в Windows Forms применяется эффективная система безопасности .NET Framework.  Благодаря ей можно создавать более надежные приложения.