

## Лабораторная работа Развертывание приложения

**Цель работы:** Освоить технологию создания инсталляционных программ Windows-приложений для развертывания созданного приложения.

### Задание:

1. Изучить теоретический материал.
2. Разработать инсталлятор для своего приложения из предыдущей лабораторной работы.
3. Протестировать инсталлятор.
4. Продемонстрировать инсталляцию и деинсталляцию приложения преподавателю.

### Общие сведения

После того, как исходный код приложения откомпилирован и завершено тестирование необходимо создать инсталляционный файл для развертывания приложения на компьютере пользователя.

*Visual Studio* позволяет создавать инсталляционные программы для приложений. Для этого предусмотрено шесть типов проектов инсталляторов:

- *Setup Project* – используется для инсталляции клиентских приложений, приложений среднего слоя, а также приложений, запускаемых как службы Windows;
- *Web Setup Project* – используется для инсталляции Web-ориентированных приложений;
- *Merge Module Project* – создает модули слияния (merge modules), которые могут быть использованы с другими приложениями, основанными на технологии Windows Installer;
- *CAB Project* – создает cab-файлы для распространения с применением старых технологий развертывания;
- *Setup wizard* – помощник в создании проекта развертывания;
- *Smart Device CAB Project* – CAB-проект для PocketPC, Smartphone и других CE-приложений.

*Windows Installer* – это служба, управляющая инсталляцией, обновлением, восстановлением и удалением приложений в большинстве операционных систем семейства Windows. Эта служба является частью Windows.

*Windows Installer* ведет базу данных инсталлированных приложений. Когда приложение должно быть деинсталлировано, с помощью этой базы можно легко найти и удалить настройки реестра, которые были добавлены, файлы, которые были скопированы на жесткий диск, а также пиктограммы рабочего стола и меню Start (Пуск). Если определенный файл все еще используется другим приложением, то инсталлятор оставит его на диске, так что использующее его приложение не разрушится. База данных также обеспечивает возможность восстановления. Если настройки реестра или DLL-библиотеки, ассоциированные с приложением, были повреждены или непреднамеренно удалены, инсталляцию можно восстановить. Во время восстановления инсталлятор читает в своей базе информацию о последней инсталляции и повторяет ее.

Проекты развертывания в *Visual Studio* дают возможность создавать пакеты для *Windows Installer*. Проекты развертывания предоставляют доступ к большей части того, что необходимо сделать, чтобы установить определенное приложение.

### Создание инсталлятора

При разработке приложения использовалась отладочная версия проекта (*Debug*), в которой присутствовал дополнительный код для отладки. Инсталлятор приложения должен создаваться для серийного выпуска приложения (*Release*). Для создания серийного выпуска откроем проект приложения и в окне свойств проекта для пункта построителя (*Built*) изменим свойство конфигурации (*Configuration*) на *Active (Release)* (рисунок 1). Далее необходимо провести компиляцию и тестирование приложения.

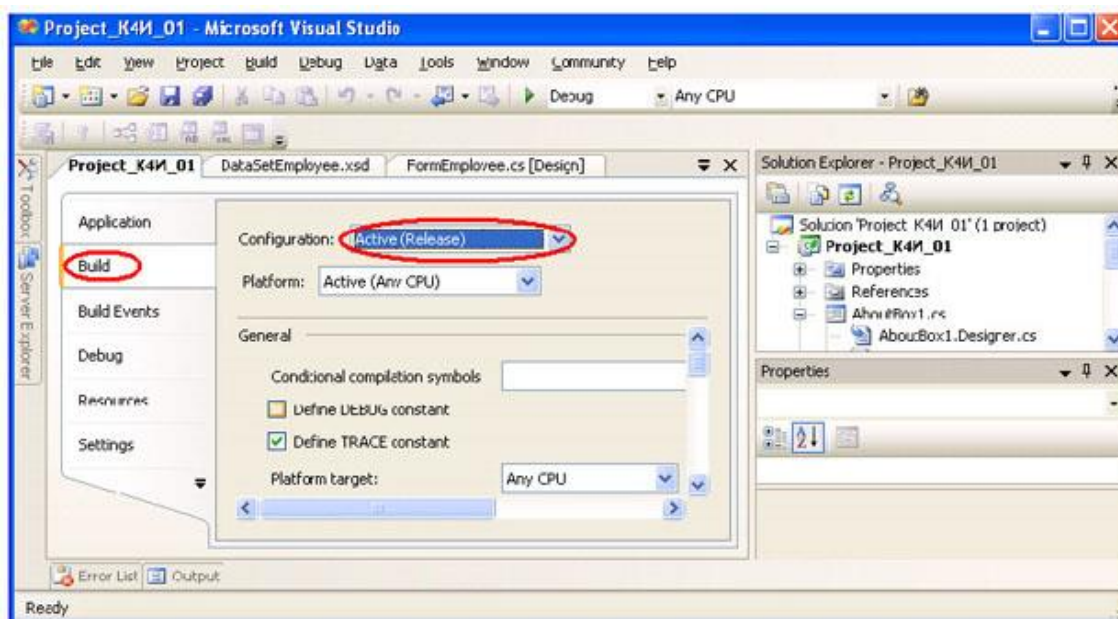


Рис. 1. Установка свойства проекта "Серийный выпуск"

Для развертывания проекта приложения Project\_K4И\_01 создадим отдельный проект развертывания SetupEmployee (рисунок 2).

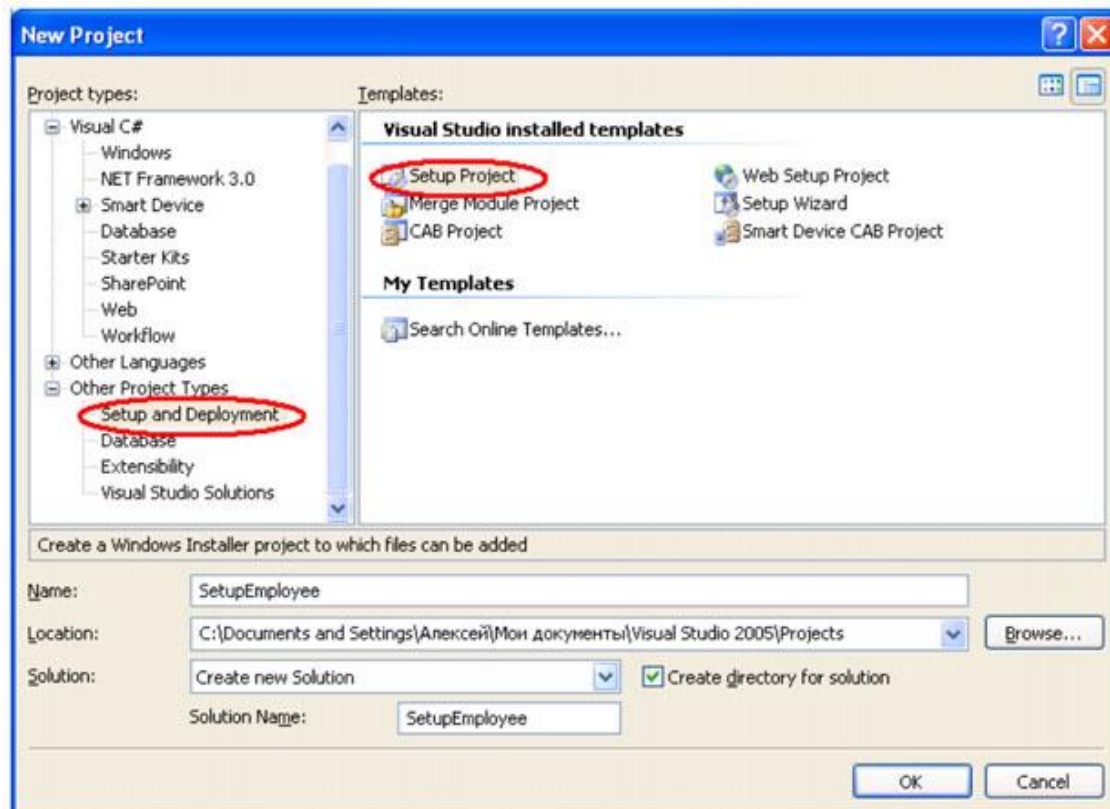


Рис. 2. Создание проекта развертывания

В результате откроется окно создаваемого проекта инсталляции приложения (рисунок 3).



Рис. 3. Окно проекта развертывания

Если в окне *Solution Explorer* щелкнуть на проекте, то в окне *Properties* (Свойства) отобразится список свойств (рисунок 4).



**Рис. 4.** Свойства проекта развертывания

Эти свойства будут отображаться во время инсталляции приложения. Некоторые из них также будут отображаться в приложении панели управления *Add/Remove Programs* (Установка и удаление программ). Поскольку большинство из этих свойств видимы пользователю в процессе инсталляции (или когда он посмотрит на инсталляцию в окне *Add/Remove Programs*), их правильная установка придаст приложению профессиональный вид. Список этих свойств весьма важен, если планируется коммерческая поставка приложения. В таблице 1 приведены свойства и значения, которые необходимо ввести при создании проекта инсталляции.

Таблица 1. Свойства и значения для проекта установки	
Свойства проекта	Описание
AddRemove, ProgramsIcon	Пиктограмма, появляющаяся в диалоговом окне <i>Add/Remove Programs</i> (Установка и удаление программ).
Author	Автор приложения. Обычно это свойство совпадает с производителем ( <i>Manufacturer</i> ). Отображается на странице <i>Summary</i> (Сводка) диалогового окна <i>Properties</i> (Свойства) пакета <i>msi</i> , а также в поле <i>Contact</i> (Контакт) страницы <i>SupportInfo</i> диалогового окна <i>Add/Remove Programs</i> .
Discription	Поле текста свободного стиля, описывающее приложение или инсталлированные компоненты. Эта информация отображается на странице <i>Summary</i> диалогового окна <i>Properties</i> пакета <i>msi</i> , а

	также в поле <i>Contact</i> страницы <i>SupportInfo</i> диалогового окна <i>Add/Remove Programs</i> .
DetectNewer, InstalledVersion	Булевское значение. Если установлено в true, проверяет, не установлена ли уже на машине более новая версия приложения. Если это так, процесс инсталляции останавливается.
InstallAllUsers	Булевское значение. Установка в true означает необходимость инсталляции для всех пользователей компьютера. Установка в false обеспечивает доступ к приложению только текущему пользователю.
Keywords	Ключевые слова, которые могут быть использованы для поиска <i>msi</i> -файла на целевом компьютере. Эта информация отображается на странице <i>Summary</i> диалогового окна <i>Properties</i> пакета <i>msi</i> .
Localization	Локальная настройка, используемая для строковых ресурсов и установок реестра. Влияет на пользовательский интерфейс инсталлятора.
Manufacturer	Наименование компании-производителя приложения или компонента. Обычно это та же информация, что указана и в свойстве <i>Author</i> . Отображается на странице <i>Summary</i> диалогового окна <i>Properties</i> пакета <i>msi</i> , а также в поле <i>Publisher</i> (Издатель) страницы <i>SupportInfo</i> в диалоговом окне <i>Add/Remove Programs</i> . Используется как часть пути инсталляции по умолчанию для данного приложения.
ManufacturerURL	<i>URL Web</i> -сайта, связанного с инсталлируемым приложением или компонентом.
PostBuildEvent	Команда, исполняемая после окончания процесса построения.
PreBuildEvent	Команда, исполняемая перед началом процесса построения.
ProductCode	Строковый идентификатор GUID, уникальный для данного приложения или компонента. <i>Windows Installer</i> использует это свойство, чтобы идентифицировать приложение для последующих обновлений или инсталляций.
ProductName	Имя приложения. Используется для описания приложения в диалоговом окне <i>Add/Remove Programs</i> , а также как часть пути инсталляции по умолчанию данного приложения: C: \Program Files\Manufacturer\ProductName
Remove Previous, Versions	Булевское значение. Если установлено в true, то проверяет наличие на компьютере предыдущей версии приложения. Если она есть, то старая версия деинсталлируется перед тем, как начать инсталляцию новой. Это свойство использует <i>Productcode</i> и <i>UpgradeCode</i> для определения того, должна ли выполняться деинсталляция. <i>UpgradeCode</i> должен быть тем же, <i>ProductCode</i> – должен отличаться.
RunPostBuildEvent	Когда должно запускаться событие <i>PostBuildEvent</i> . Варианты: при успешном построении проекта либо всегда.
SearchPath	Строка, представляющая путь поиска зависимых сборок, файлов или модулей слияния. Используется, когда инсталляционный пакет собран на машине разработчика.
Subject	Дополнительная информация относительно приложения. Эта информация отображается на странице <i>Summary</i> диалогового окна <i>Properties</i> пакета <i>msi</i> .

SupportPhone	Телефонный номер службы поддержки приложения или компонента Эта информация отображается в поле <i>Support Information</i> (Информация о поддержке) страницы <i>SupportInfo</i> диалогового окна <i>Add/Remove Programs</i> .
SupportURL	URL службы поддержки приложения или компонента. Эта информация отображается в поле <i>Support Information</i> страницы <i>SupportInfo</i> диалогового окна <i>Add/Remove Programs</i> .
TargetPlatform	Поддержка 32-разрядной или 64-разрядной версий <i>Windows</i> .
Title	Заголовок инсталлятора. Отображается на странице <i>Summary</i> диалогового окна <i>Properties</i> пакета <i>msi</i> .
UpgradeCode	Строковый GUID, представляющий идентификатор различных версий одного и того же приложения. UpgradeCode не должен меняться для разных версий или разных языковых версий приложения. Используется <i>DetectNewerInstalledVersion</i> и <i>RemovePreviousVersion</i> .
Version	Номер версии инсталлятора, <i>cab</i> -файла или модуля слияния.

Для добавления сборок (в данном примере единственная сборка, которую необходимо добавить – основная исполняемая программа (*Project\_K4И\_01.exe*)) необходимо щелкнуть правой кнопкой мышки на проекте в *Solution Explorer*, либо выбрать команду меню *Project/Add* (Проект/Добавить), которая имеет четыре опции:

- *Project Output* – выходные данные проекта;
- *File* (Файл) – используется для добавления текстового файла *readme* или любого другого файла, не являющегося частью процесса сборки;
- *Merge Module* (Модуль слияния) – модуль слияния, созданный отдельно;
- *Assembly* (Сборка) – эту опцию следует использовать для выбора сборки, являющейся частью инсталляции.

В нашем случае выберем *Assembly* (рисунок 5).

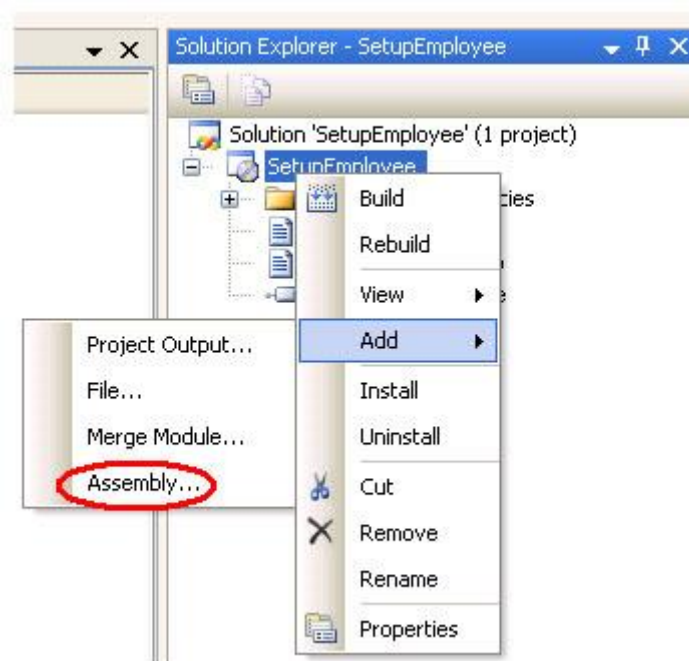
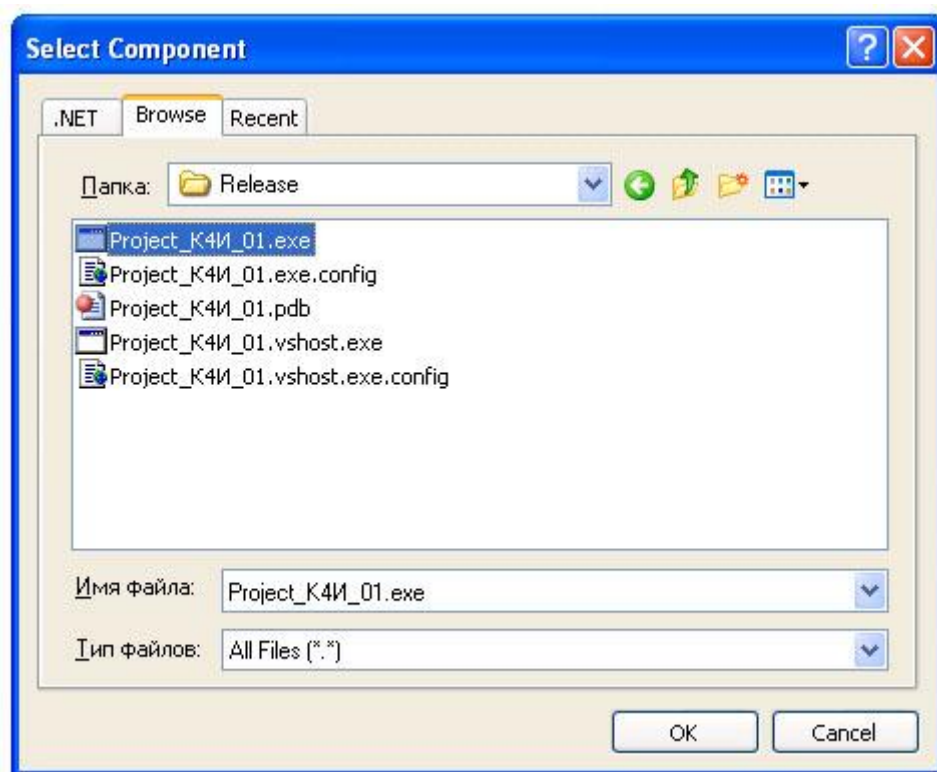


Рис. 5. Добавление сборки в проект развертывания

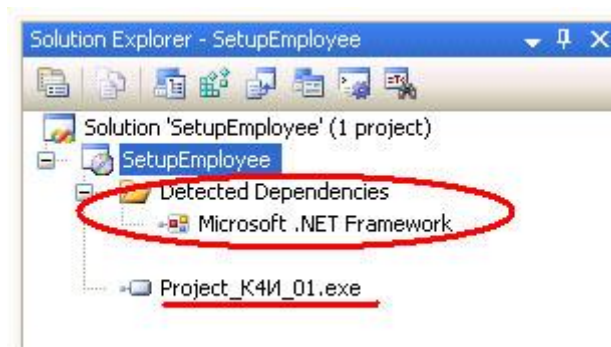


Появится диалоговое окно *Component Selector* (Выбор компонентов). Пройдем до папки `\bin\release` созданного ранее приложения и выберем – `Project_K4И_01.exe`, затем щелкнем на кнопке *OK* (рисунок 6).



**Рис. 6.** Выбор приложения проекта развертывания

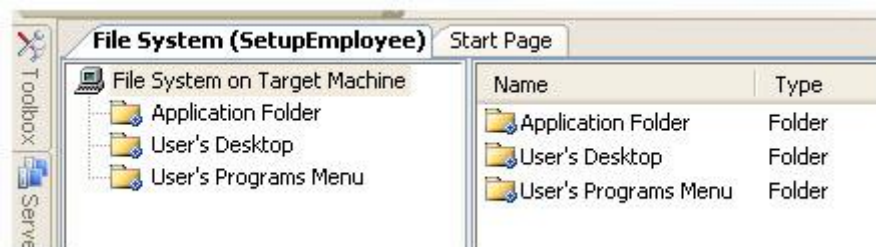
После этого увидим `Project_K4И_01.exe` в *Solution Explorer* проекта развертывания. В разделе *Detected Dependencies* (Обнаруженные зависимости) увидим, что *Visual Studio* опросил `Project_K4И_01.exe` на предмет сборок, которых он зависит – в данном случае *Microsoft .NET Framework* включен автоматически (рисунок 7).



**Рис. 7.** Окно проекта развертывания с добавленной сборкой

В общем случае, если приложение состоит из нескольких сборок необходимо продолжать этот процесс до тех пор, пока все сборки приложения не будут зарегистрированы в *Solution Explorer* для проекта развертывания.

Далее необходимо определить, где должны быть развернуты сборки. По умолчанию *Visual Studio* отображается редактор *File System* (Редактор файловой системы), вид которого приведен на рисунке 8.

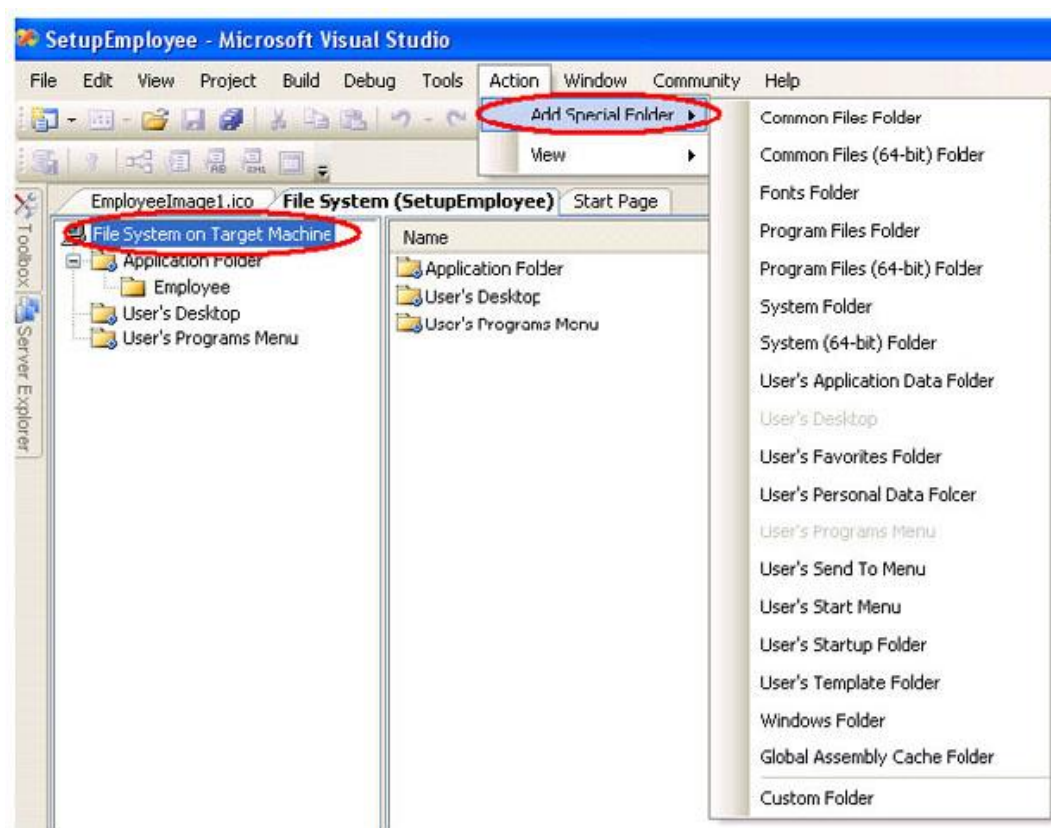


**Рис. 8.** Редактор файловой системы

Этот редактор разделен на две панели. Левая показывает иерархическую структуру файловой системы целевой машины, а правая панель отображает детальное представление выбранной папки. Имена папок могут отличаться от того, что можно было ожидать, но следует помнить, что это – целевая машина; например, папка с меткой *User's Program Menu* (Меню Программы пользователя) отображается на *C:\Documents and Settings\User Name\Start Menu\Programs* целевого клиента.

На данном этапе можно добавить другие папки – как специальные, так и произвольные. Чтобы добавить специальную папку, нужно убедиться, что на левой панели выделена опция *File System on Target Machine* (Файловая система на целевой машине), и подменю *Action* (Действие) главного меню (рисунок 9).

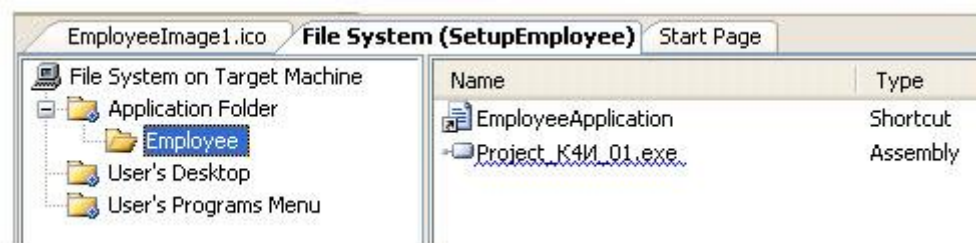
Для случая, рассматриваемого в лабораторной работе, добавим папку *Employee* в папку приложения *Application Folder* (внутреннее вложение).



**Рис. 9.** Окно добавления специальной папки

Для этого выберем папку *Application Folder* (Папка приложения) в панели редактора и затем выберем меню *Action*. На этот раз обратимся к пункту *Add* (Добавить), который позволит создать новую папку. Переименуем ее, присвоив имя *Employee*. Перетащим файлы из папки *Application Folder* в папку *Employee* (рисунок 10).





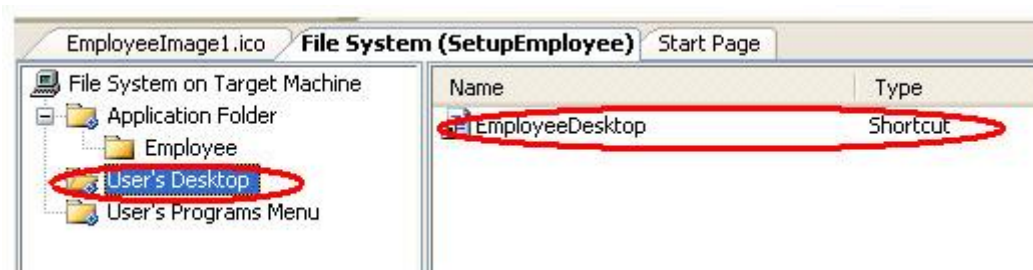
**Рис. 10.** Окно проекта развертывания

Добавим ярлык приложения на рабочий стол пользователя. Для создания ярлыка на рабочем столе, необходимо обратиться к папке *Application Folder*. На правой панели редактора выберите приложение, перейдите к меню *Action* и выберите пункт *Create Shortcut* (Новый ярлык), чтобы создать ярлык приложения (рисунок 11).



**Рис. 11.** Окно создания ярлыка приложения

После создания ярлыка переименуйте его, задав имя *EmployeeDesktop*, и перетащите его в папку *User's Desktop* (рисунок 12).



**Рис. 12.** Окно создания ярлыка приложения на рабочем столе

Созданному ярлыку необходимо придать индивидуальный вид с помощью иконки. Для этого добавим файл любой иконки в проект (рисунок 13).

В результате в проект добавится файл с расширением *.ico*, который можно переименовать, например *EmployeeImage.ico* (рисунок 14), и открыв в редакторе изменить графическое представление (рисунок 15).

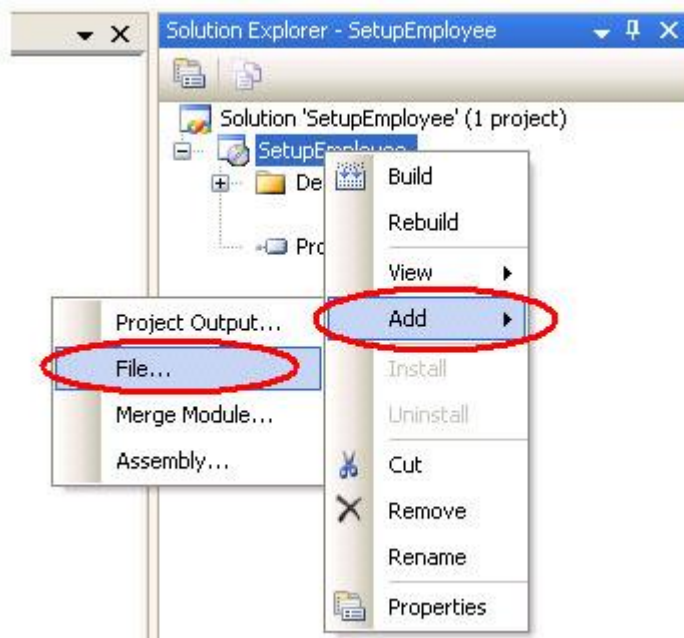


Рис. 13. Добавление файла иконки в проект

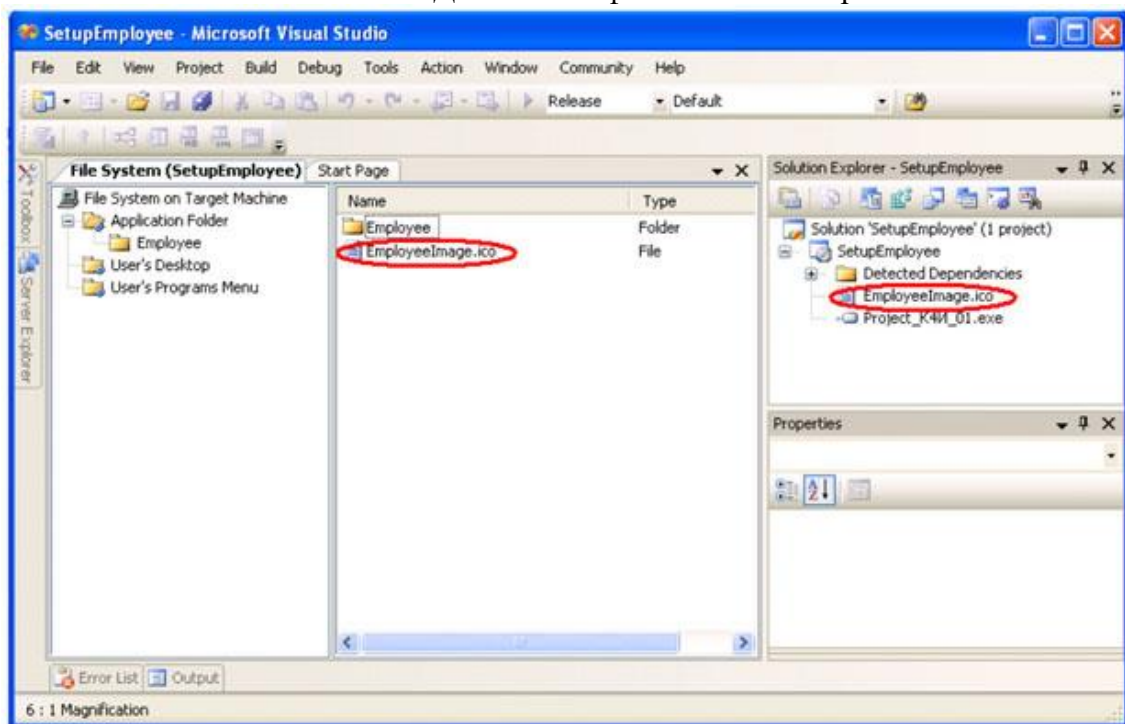


Рис. 14. Проект с файлом иконки

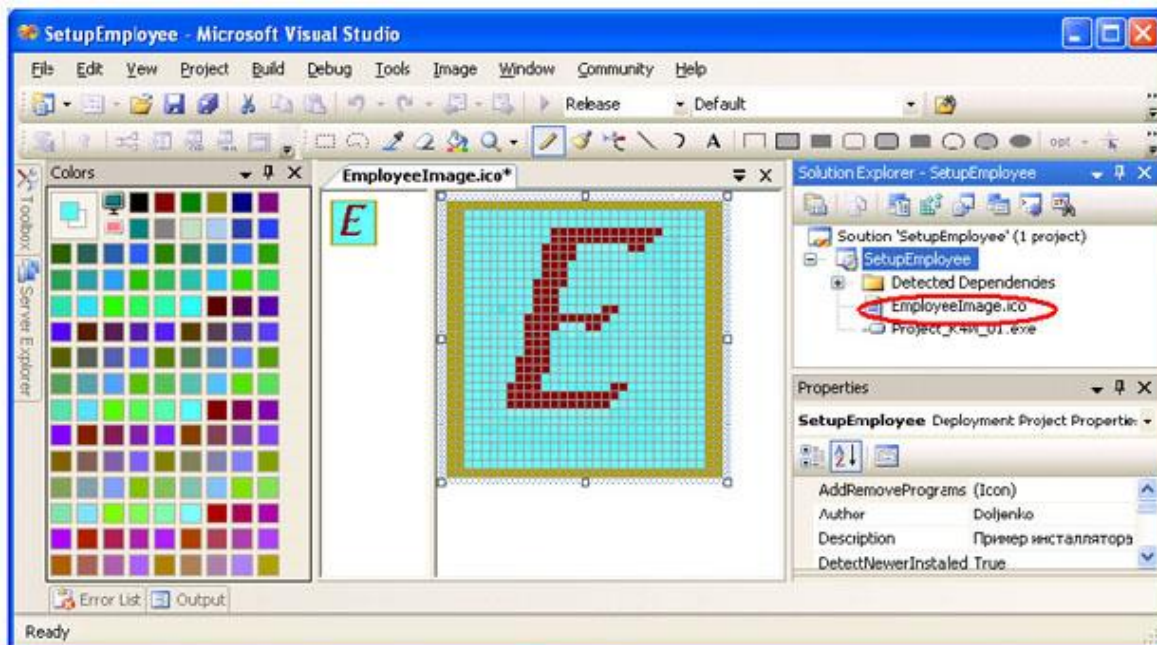


Рис. Редактирование иконки

Теперь необходимо связать с ярлыком на рабочем столе созданную нами иконку. Для этого необходимо задать свойство Icon для ярлыка рабочего стола (рисунок 16).

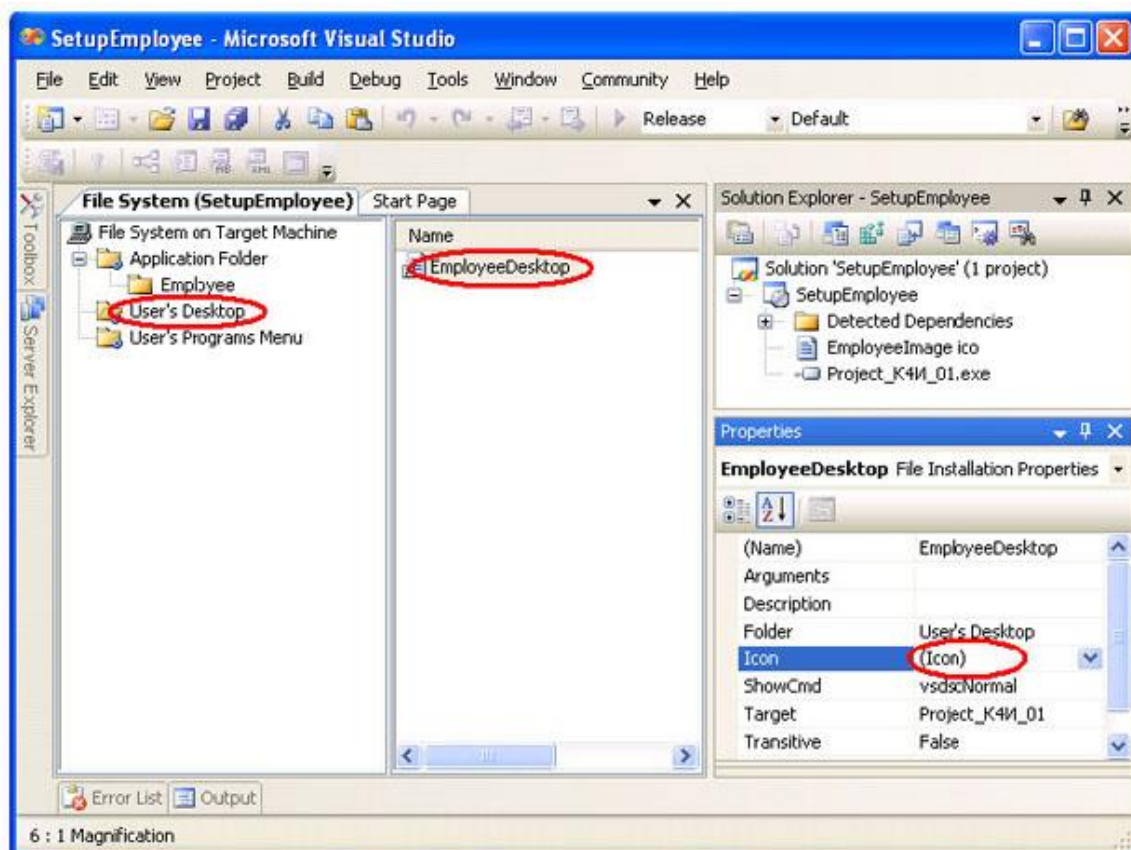
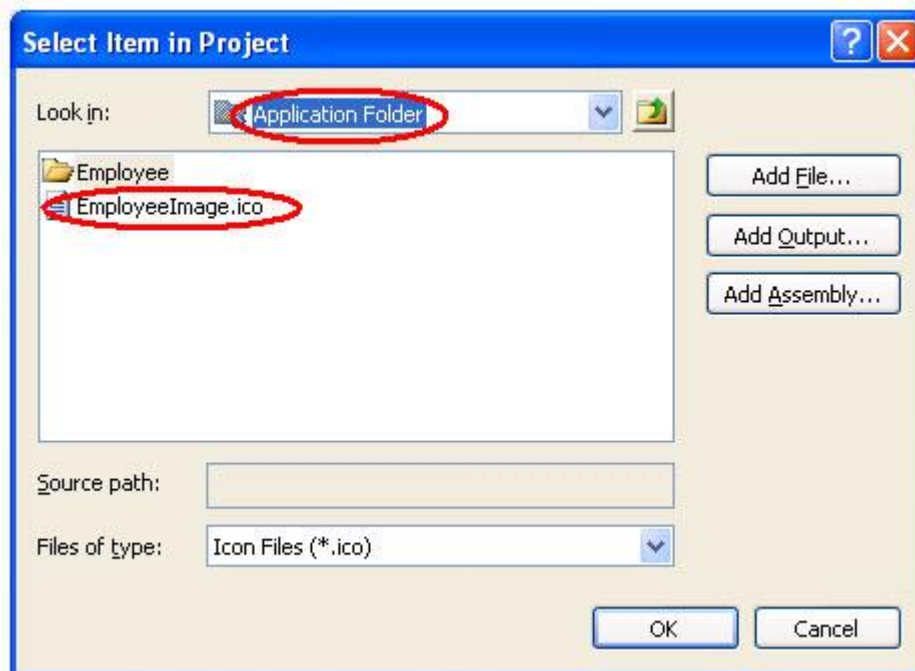
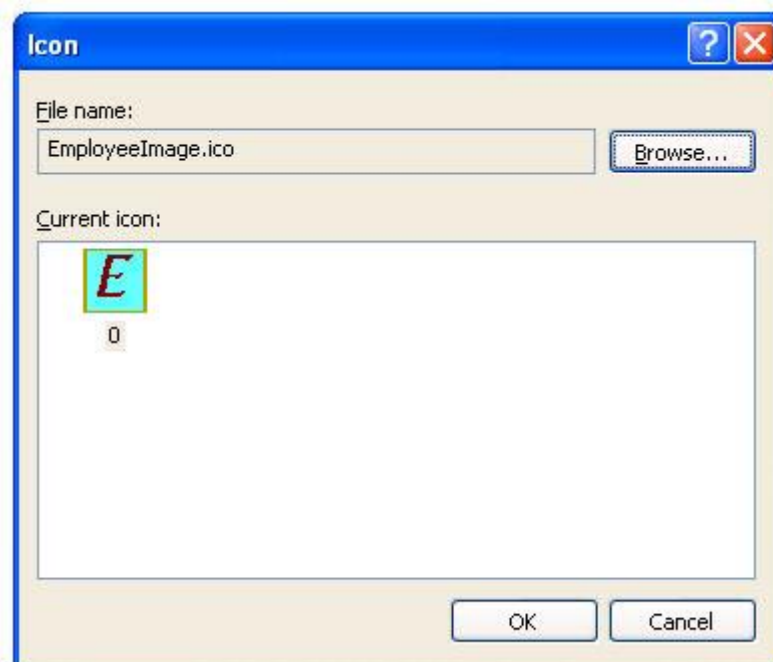


Рис. 16. Окно задания иконки ярлыка приложения

Процесс задания иконки сопровождается выбором файла (рисунок 17) и подтверждением выбранной иконки (рисунок 18).



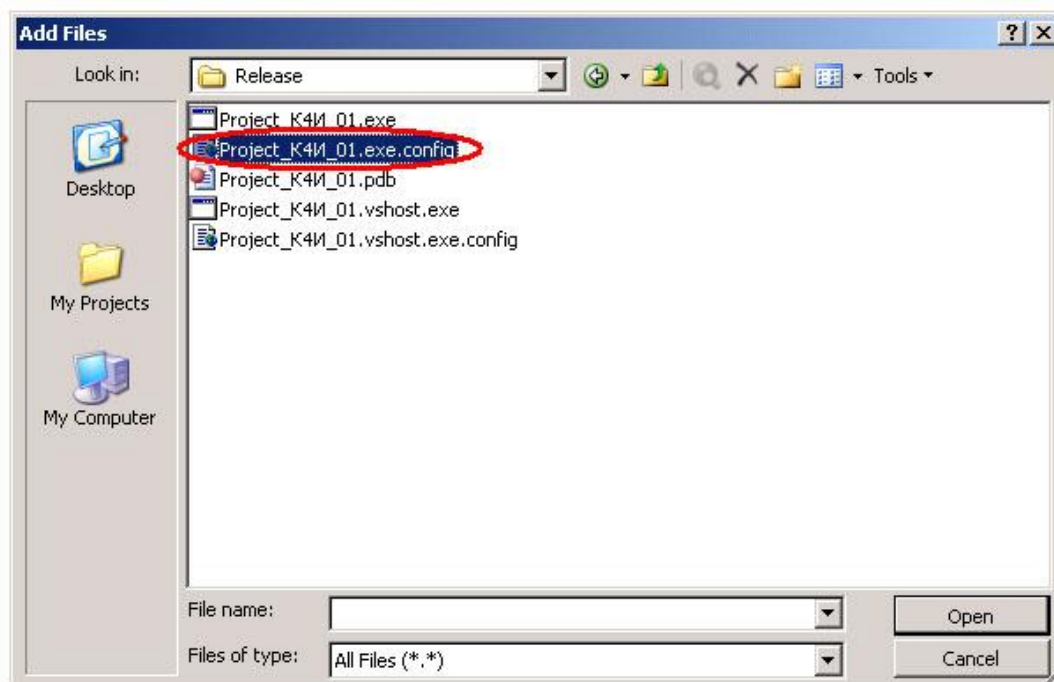
**Рис. 17.** Окно выбора файла иконки



**Рис. 18.** Окно задания иконки

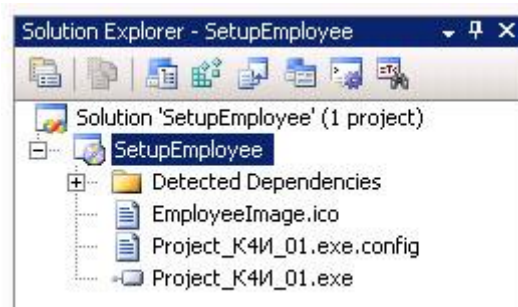
Теперь при инсталляции приложения его ярлык появится на рабочем столе. Точно так же выглядит процесс создания ярлыка в Start.

В инсталляционный проект необходимо добавить конфигурационный файл Project\_K4И\_01.exe.config для обеспечения возможности настройки подключения к базе данных пользователя без перекомпиляции проекта (рисунок 19).



**Рис. 19.** Подключение к проекту конфигурационного файла

После подключения конфигурационного файла окно *Solution Explorer* будет иметь вид, аналогичный, приведенному на рисунке 20.



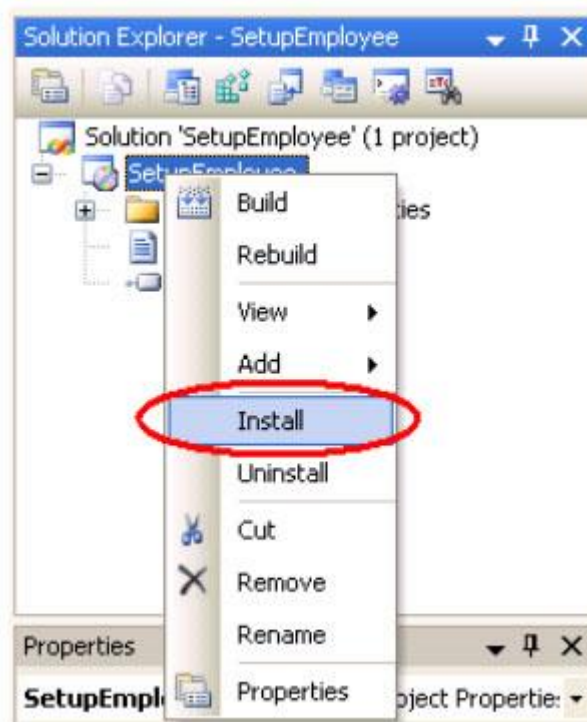
**Рис. 20.** Окончательное окно Solution Explorer инсталляционного проекта

Прежде чем собирать проект разворачивания, может понадобиться проверить некоторые свойства проекта. Если выбрать команду меню *Project/Properties* (Проект/Свойства) для проекта *SetupEmployee*, то отобразится диалоговое окно *Property Pages* (Страницы свойств) (рисунок 21).





	<ul style="list-style-type: none"> <li>• <i>Microsoft Visual J# .NET Redistributable Package 2.0;</i></li> <li>• <i>SQL Server Express Edition;</i></li> <li>• <i>Microsoft Data Access Components 2.8.</i></li> </ul> <p>Имеется также возможность указать необходимое программное обеспечение, которое можно загрузить с указанных <i>URL</i>-адресов либо из того места, откуда выполняется основная инсталляция.</p>
Compression	<p>Специфицирует тип сжатия включенных файлов. Выбор такой:</p> <ul style="list-style-type: none"> <li>• Optimized for speed (Оптимизация по скорости). Файлы больше по размеру, но быстрее инсталляция (установлено по умолчанию);</li> <li>• Optimized for size (Оптимизация по размеру). Маленькие файлы, но медленная инсталляция;</li> <li>• None (Без сжатия). Без сжатия.</li> </ul>
CAS size	<p>Это свойство доступно, когда установка Package files установлена в In cabinet files (В кабинетных файлах). Неограниченный (Unlimited) размер создает единственный кабинетный файл; настраиваемый (Custom) позволяет установить максимальный размер каждого кабинетного файла.</p>
Authenticode signature	<p>Когда включено, то выход проекта развертывания подписывается с использованием Authenticode; по умолчанию отключено.</p>
Certificate file	<p>Сертификат, используемый для подписи.</p>
Private key file	<p>Секретный ключ, содержащий цифровой ключ шифрования подписанных файлов.</p>
Timestamp server URL	<p><i>URL</i>-адрес сервера временных меток. Используется также для подписи кодом Authenticode.</p>



**Рис. 22.** Тестирование инсталляционного проекта

При инсталляции приложения Project\_K4И\_01.exe на рабочем столе должен появиться ярлык EmployeeDesktop, с помощью которого можно запустить приложение. После запуска приложения ярлык появляется и в главном меню.

Для инсталляции приложения Project\_K4И\_01.exe на клиентском компьютере необходимы два созданных в проекте файла setup.exe и SetupEmployee.msi.