

# Chapter 4

## Deploying Windows Application

Mrs. Swati Satpute

# Deployment Overview

---

- What do we mean by Deployment??
  - It's a process of **installing application** on target system.
- Step1: Preparing Installer
  - Packaging Assemblies (dlls / exes), xmls
  - Registry manipulation
  - Database generation
- Step2: Executing installer on required set up

# Deployment Overview

---

- **Deployment** is the process by which you distribute a **finished application** or component to be installed on other computers.
- In Visual Studio you can deploy applications using either
  - **ClickOnce** deployment, or
  - With a traditional Setup using **Windows Installer deployment technology**
- There are many factors to consider when choosing a deployment strategy:
  - the type of application,
  - the type and location of users,
  - the frequency of application updates, and
  - the installation requirements.

# Issue related to Deployment

---

- Installation can be **time consuming** if application needs to be installed on **hundreds of machine**.
  - One can automate this by preparing batch file, but still its time consuming.
- Some deployment expects certain **User level Access Rights**.
- Considering these hassles people have changed their Rich UI – Windows based application to web based.

# ClickOnce Deployment

---

- ClickOnce deployment publishes the **Windows-based applications** to a **centralized location** and the user **installs** or **runs** the application from that location.
- It takes care of three major issues
  - Difficulties in updating applications
  - Impact to the user's computer
  - Security permissions
- ClickOnce technology
  - Enables to create **self-updating** windows-based application.
  - Application can be installed and run with **minimal user interaction**.  
It can be **natural choice** if application **updates** are **frequently expected**.

# What is ClickOnce Deployment?

---

- **ClickOnce application** is any Windows Presentation Foundation, Windows Forms, or console application **published** using ClickOnce technology.
- **Automatic updates**
  - ClickOnce applications are **self updating**, can **check for newer versions** and **update themselves**.
  - Developer can **specify update strategy**
    - Application can be **updated automatically after starting** client application.
    - Or as a **background task**.

# What is ClickOnce Deployment?

---

- ClickOnce applications are **isolated**.
  - installing or running a ClickOnce application **cannot break existing applications**.
- ClickOnce applications are **self-contained**
  - Each ClickOnce application is **installed to and run from a secure per-user, per-application cache**.

# ClickOnce

---

- ClickOnce application can be published in 3 ways
  - from a Web page
  - from a network file share
  - from media such as a CD-ROM
- Two types of installation possible
  - **Offline mode**
    - **Installing** application **on** an end **user's computer** and **run locally**.
  - **Online only**
    - Running application **without permanently installing** anything on the end user's computer.



# Limitations of ClickOnce

---

ClickOnce **can not be** used in following cases:

- If **shared component** is expected to be installed in **GAC** (global assembly cache).
- If application uses COM components which requires **registry settings**.
- If **application** wants to **decide where to get installed**.

# Deployment Architecture

---

- Backbone of ClickOnce deployment are XML based manifest files.
  - **Application manifest (\*.exe.manifest file)**
    - Describes the application assemblies, dependent libraries and lists permissions required by the application
  - **Deployment manifest (\*.application file)**
    - Describes deployment model.
    - Contains information like
      - current version to deploy.
      - update policies
      - publisher identity along with digital signature



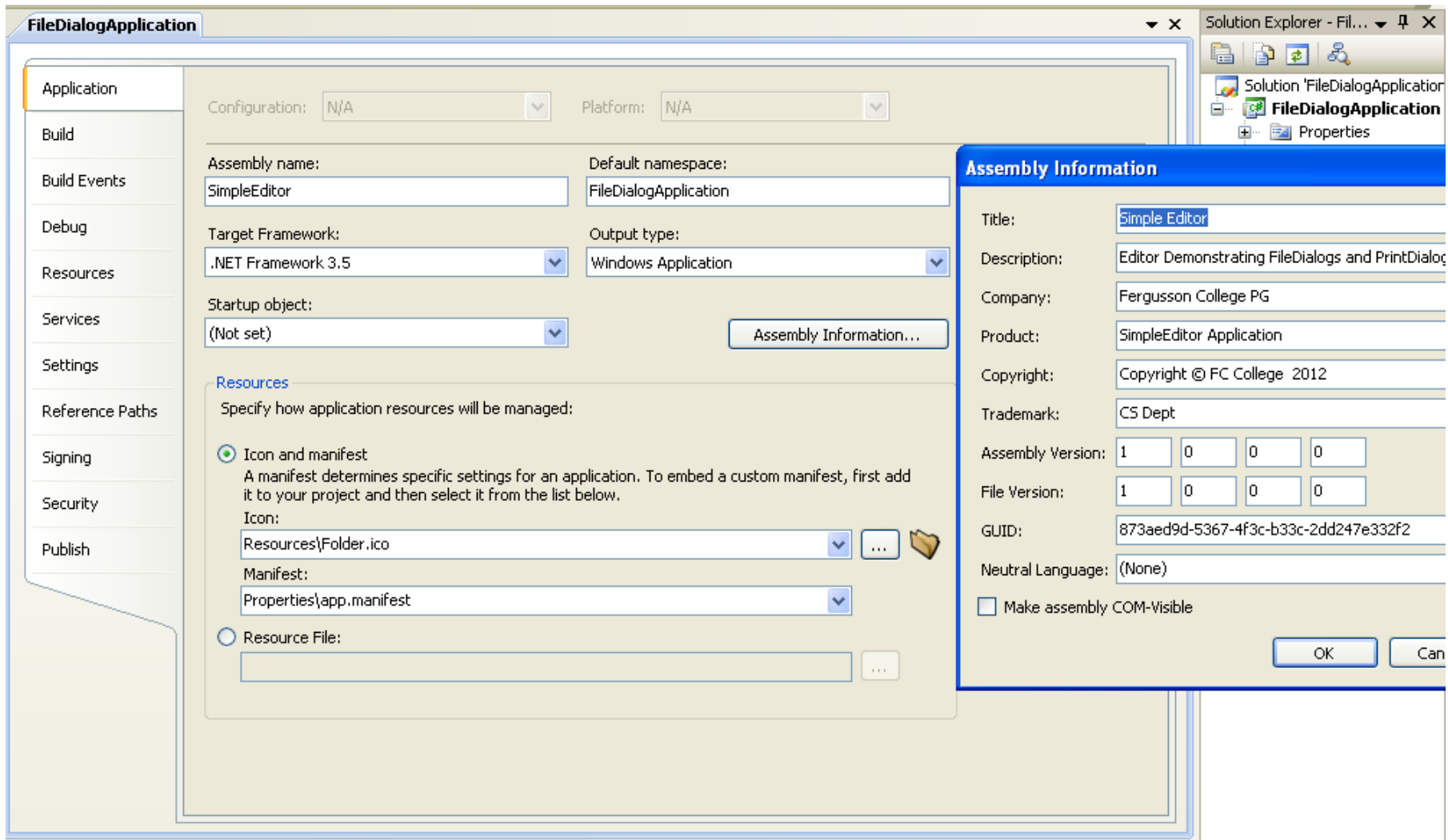
# Creating ClickOnce application demo

---

- Configure ClickOnce deployment
- Publish Application
- Install / launch application
- Update Version

# Configure ClickOnce Application

Launch Project settings – Click on Application tab



# Configure ClickOnce Application

---

- Assembly Name
  - Define the name of the assembly which will get created after building the sources.
- Entered Assembly attributes information will be stored in [AssemblyInfo.cs](#)
- Metadata information will be used by deployment tool.

# Issue Certificate

**FileDialogApplication** [X]

Application

Build

Build Events

Debug

Resources

Services

Settings

Reference Paths

**Signing**

Security

Publish

Configuration: N/A Platform: N/A

☒ Sign the ClickOnce manifests

Certificate:

Issued To	STAFF06\USER
Issued By	STAFF06\USER
Intended Purpose	<All>
Expiration Date	2/26/2013 9:47:33 PM

Select from Store...

Select from File...

Create Test Certificate...

More Details...

Timestamp server URL:

☐ Sign the assembly

Choose a strong name key file:

Change Password...

☐ Delay sign only

When delay signed, the project will not run or be debuggable.

# Issue Certificate

---

- Deploying software expects a **signed manifest**.
- **Certificate** is **must** and **will be shown** to end user while installing software.
- **Certificate** contains the organization details. This **gives confidence** to end user that the software is from trusted source.
- Real certificate should be bought from certificate authority like [VeriSign](#), [Microsoft Certificate Server](#).

# Define Security

FileDialogApplication\*

▼

×

Application

Build

Build Events

Debug

Resources

Services

Settings

Reference Paths

Signing

Security\*

Publish

Configuration: N/A ▼ Platform: N/A ▼

Specify the code access security permissions that your ClickOnce application requires in order to run.  
[Learn more about code access security...](#)

☒ Enable ClickOnce Security Settings

☒ This is a full trust application

☐ This is a partial trust application

ClickOnce Security Permissions

Zone your application will be installed from:  
Local Intranet ▼

Permissions required by the application:

Permission	Setting	Included
EnvironmentPermission	(Zone Default) ▼	✓
FileDialogPermission	(Zone Default) ▼	✓
FileIOPermission	(Zone Default) ▼	
IsolatedStorageFilePermission	(Zone Default) ▼	✓
ReflectionPermission	(Zone Default) ▼	✓
RegistryPermission	(Zone Default) ▼	

Calculate Permissions Properties... Reset

Advanced...



# Define Security

---

- Decide if application requires full trust or can do with Partial trust.
- In **full trust**,
  - application has **complete access to the system**.
  - And **can do anything** the **user** running the application **has access for**.
- In **partial trust**,
  - Application is **not allowed to access the file system** other than the isolated storage and registry.

# Publish Application

FileDialogApplication\*

Configuration: N/A Platform: N/A

**Publish Location**

Publishing Folder Location (web site, ftp server, or file path):  
http://localhost/SimpleEditor/

Installation Folder URL (if different than above):

**Install Mode and Settings**

☐ The application is available online only

☒ The application is available offline as well (launchable from Start menu)

Application Files...  
Prerequisites...  
Updates...  
Options...

**Publish Version**

Major: 1 Minor: 0 Build: 0 Revision: 2

☒ Automatically increment revision with each publish

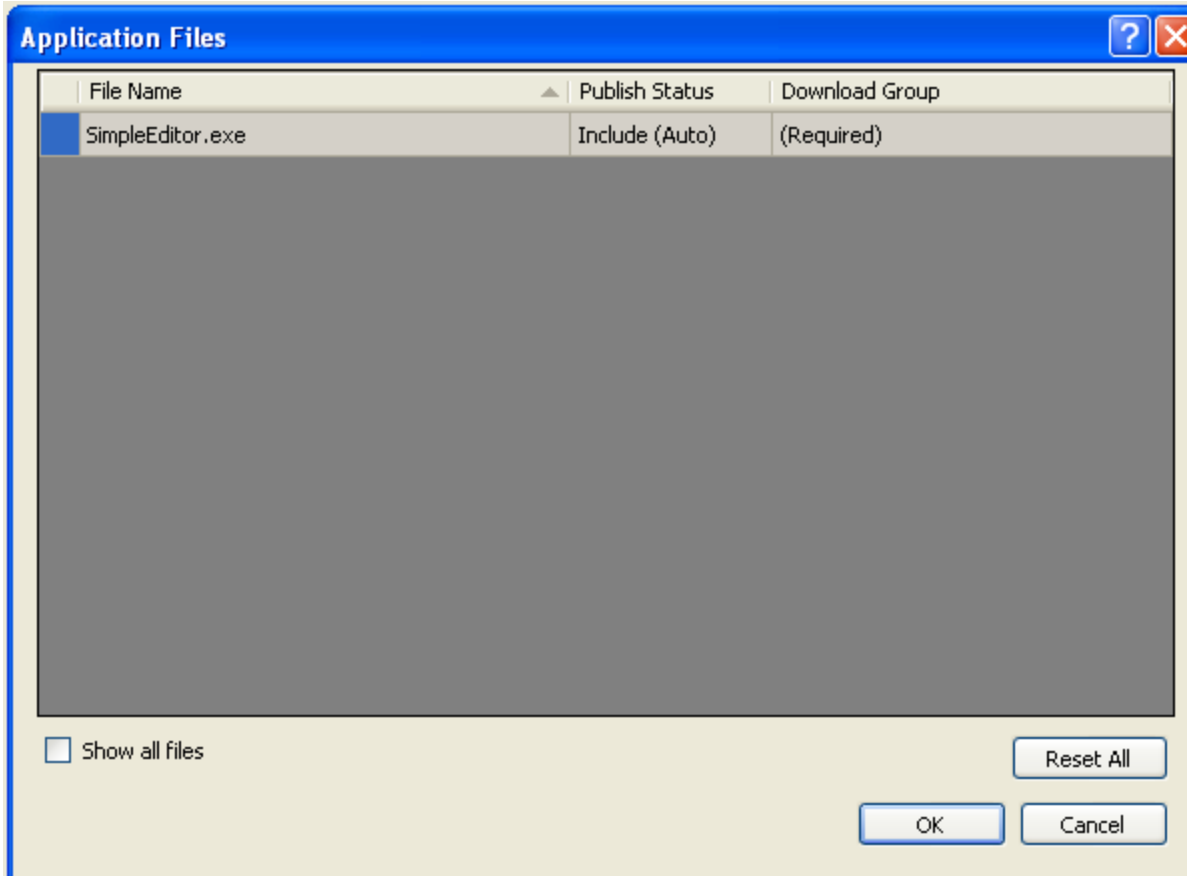
Publish Wizard... Publish Now

Application will be published at this location.

Specify Installation options:

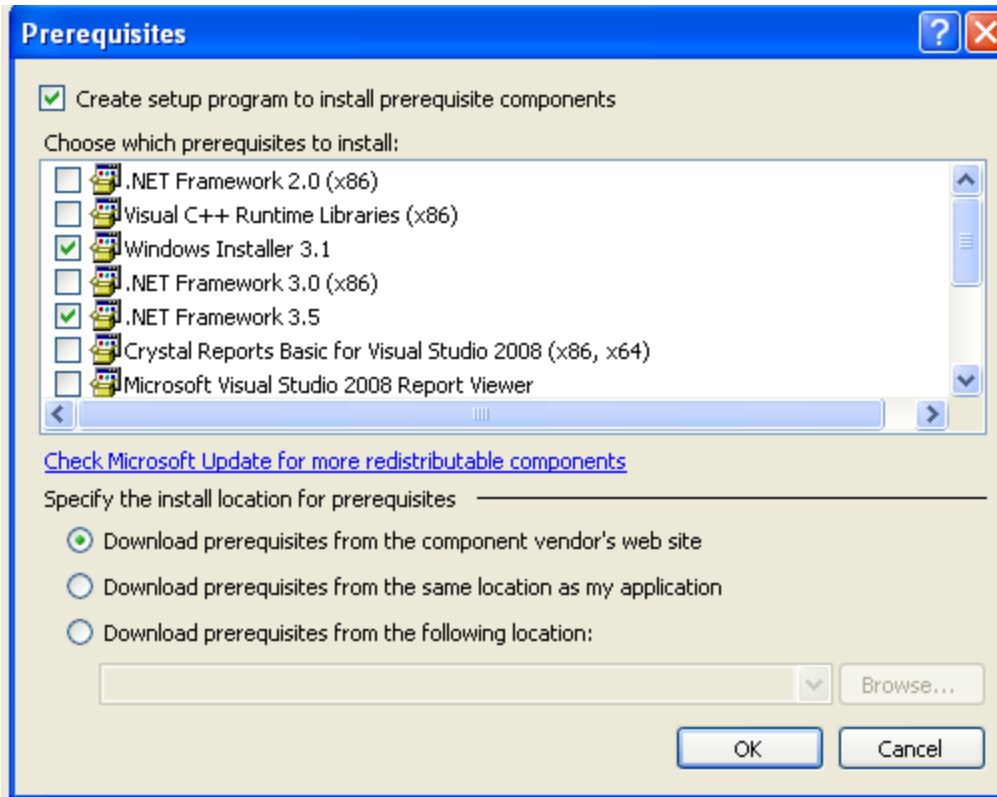
- Online only
- Offline

# Publish Application – Application Files



Shows the list of files, which gets generated as project output.

# Publish Application - Prerequisites



One can specify list of required software which are expected for using this software.

# Publish Application - Updates

**Application Updates**

☒ The application should check for updates

Choose when the application should check for updates:

☐ After the application starts  
Choose this option to speed up application start time. Updates will not be installed until the next time the application is run.

☒ Before the application starts  
Choose this option to ensure that users who are connected to the network always run with the latest updates.

Specify how frequently the application should check for updates:

☒ Check every time the application runs

☐ Check every: 7 day(s)

☐ Specify a minimum required version for this application

Major: Minor: Build: Revision:

Update location (if different than publish location):

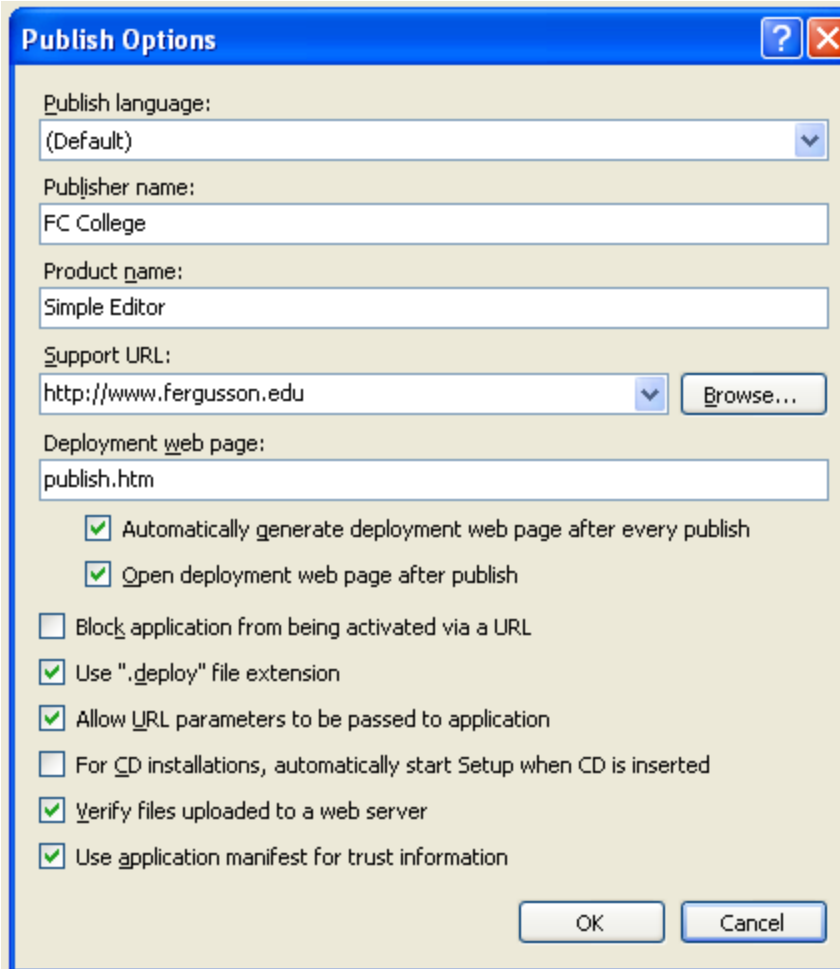
Browse...

OK Cancel

One can specify how the application should be upgraded to latest version.

- a) Before starting application
- b) After starting application

# Publish Application – Publish Options



**Publish Options**

Publish language:  
(Default)

Publisher name:  
FC College

Product name:  
Simple Editor

Support URL:  
http://www.fergusson.edu Browse...

Deployment web page:  
publish.htm

☒ Automatically generate deployment web page after every publish

☒ Open deployment web page after publish

☐ Block application from being activated via a URL

☒ Use ".deploy" file extension

☒ Allow URL parameters to be passed to application

☐ For CD installations, automatically start Setup when CD is inserted

☒ Verify files uploaded to a web server

☒ Use application manifest for trust information

OK Cancel

These are publishing related details like

- a) Product Name
- b) Name of the publisher

# Publishing Application

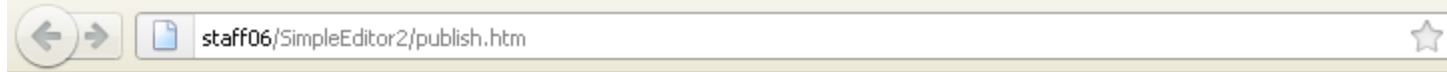
---

- To publish application, either follow [publish wizard](#) or [Publish Now](#) option.

## Working of Publish Wizard

- Website gets created on local IIS.
- Following files will be copied to the web server
  - Application (assemblies of application)
  - Application and deployment manifest files
  - Setup.exe
  - Sample web page (publish.htm)
- Based on Installation mode, entry in the start up menu will be done.

# Published Application



## FC College Simple Editor

**Name:** Simple Editor

**Version:** 1.0.0.2

**Publisher:** FC College

The following prerequisites are required:

- Windows Installer 3.1
- .NET Framework 3.5

If these components are already installed, you can [launch](#) the application now. Otherwise, click the button below to install the prerequisites and run the application.

[Install](#)



# Summary

---

- ClickOnce deployment can be a better choice if
  - application is self contained
  - doesn't require administrative privilege
  - No NTFS access expected apart from Users work area.

# References

---

- Book referred “Beginning Visual C# 2010” by Wrox publication.
- <http://msdn.microsoft.com/en-us/library/t71a733d%28v=vs.80%29.aspx>
- <http://msdn.microsoft.com/en-us/library/142dbbz4%28v=vs.90%29.aspx>

# Question Bank

---

1. What is ClickOnce deployment? When we can use this type of deployment.
2. What are the limitations of ClickOnce deployment?
3. How Automatic updates are handled in case of ClickOnce deployment?
4. Name 2 manifest files which are used by ClickOnce deployment.
  1. Application manifest (\*.exe.manifest file)
  2. Deployment manifest (\*.application file)