



Deploying Internationalized Applications Using ClickOnce

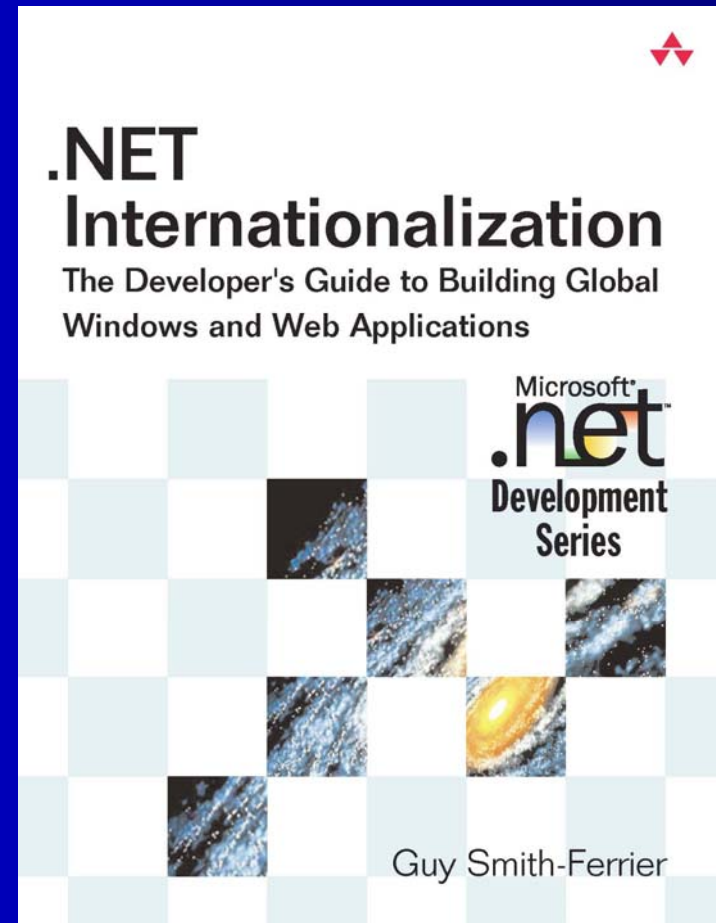


Guy Smith-Ferrier
guy@guysmithferrier.com

Blog: <http://www.guysmithferrier.com>

Author of...

- .NET Internationalization,
Addison Wesley,
ISBN 0321341384
– Due July 2006
- Visit
<http://www.dotneti18n.com>
to download the complete
source code



Agenda

- Publishing Internationalized Applications
 - Using Visual Studio 2005
 - Using msbuild
- Localizing The ClickOnce User Interface
 - publish.htm
 - ClickOnce bootstrapper dialogs
 - ClickOnce dialogs

Publishing Scenarios

- 1. Publish individual cultures
 - fr French
 - fr-FR French (France)
 - fr-CA French (Canada)
- 2. Publish groups of languages
 - fr, fr-FR, fr-CA
 - es, es-ES, es-MX
- 3. Publish all cultures
 - fr, fr-FR, fr-CA, es, es-ES, es-MX

Demos

1. Basic ClickOnce
2. Publishing Individual Cultures
3. Publishing Groups Of Languages
4. Publishing All Cultures

Publishing Using msbuild

- To build an application

```
msbuild /target:build
```

- To publish an application

```
msbuild /target:publish
```

- To publish and override a property

```
msbuild /target:publish /property:ProductName=TimeZones
```

- The publish target does **not** copy files to the deployment location

Publish Individual Cultures Using msbuild

- To publish an individual culture specify the `TargetCulture` and `PublishUrl` properties

```
msbuild /target:publish /property:TargetCulture=fr  
/property:PublishUrl=http://localhost/WindowsApplication1/fr
```

```
msbuild /target:publish /property:TargetCulture=fr-FR  
/property:PublishUrl=http://localhost/WindowsApplication1/fr-FR
```

```
msbuild /target:publish /property:TargetCulture=fr-CA  
/property:PublishUrl=http://localhost/WindowsApplication1/fr-CA
```

```
msbuild /target:publish /property:TargetCulture=es  
/property:PublishUrl=http://localhost/WindowsApplication1/es
```

Publish Groups Of Languages Using msbuild

- msbuild does not have a facility for setting the Publish status of a culture
- Each culture's Publish status is set in the project file:-

```
<PublishFile Include="fr\WindowsApplication1.resources">  
  <InProject>False</InProject>  
  <Group>  
  </Group>  
  <TargetPath>  
  </TargetPath>  
  <PublishState>Exclude</PublishState>  
  <FileType>Satellite</FileType>  
</PublishFile>
```

Publish Groups Of Languages Using msbuild (continued)

- Write a utility to set the publish status of a culture
- Call the utility to set the status before calling msbuild

```
SetPublishState fr:Include fr-FR:Include fr-CA:Include All:Exclude
```

```
msbuild /target:publish /property:TargetCulture=fr  
/property:PublishUrl=http://localhost/WindowsApplication1/fr
```

```
SetPublishState es:Include es-ES:Include es-MX:Include All:Exclude
```

```
msbuild /target:publish /property:TargetCulture=es  
/property:PublishUrl=http://localhost/WindowsApplication1/es
```

Publish All Cultures Using msbuild

- Specify "*" for the TargetCulture to publish all cultures

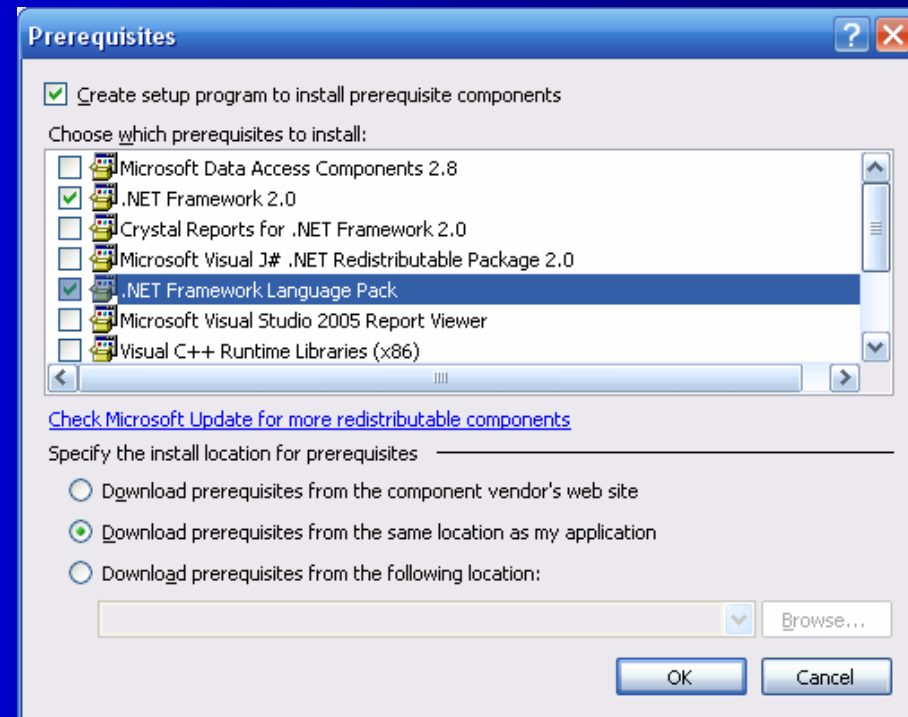
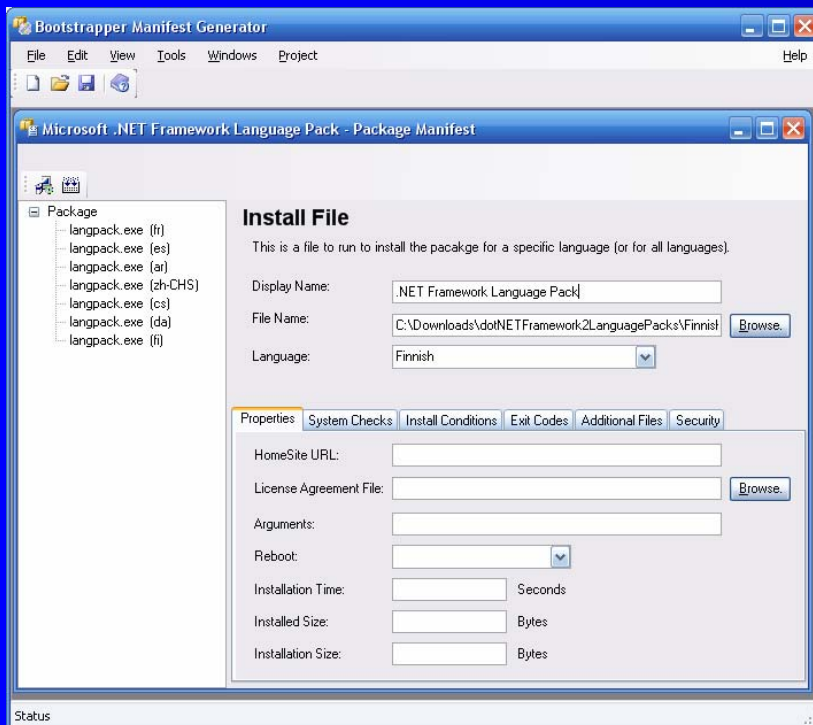
```
msbuild /target:publish /property:TargetCulture=*  
/property:PublishUrl=http://localhost/WindowsApplication1
```

Adding .NET Framework Language Packs To The Pre-Requisites

- Bootstrapper Manifest Generator
 - <http://www.gotdotnet.com/workspaces/workspace.aspx?id=ddb4f08c-7d7c-4f44-a009-ea19fc812545>
- Bootstrapper Pre-Requisite Packages can be assigned a culture
 - If they are assigned a culture then the package which matches the publish language is selected

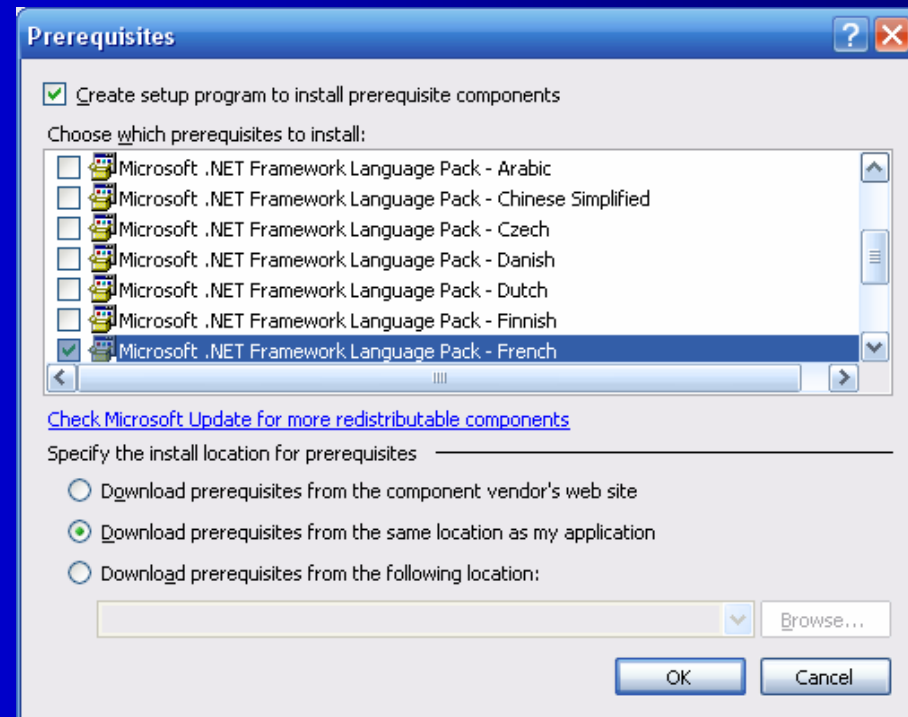
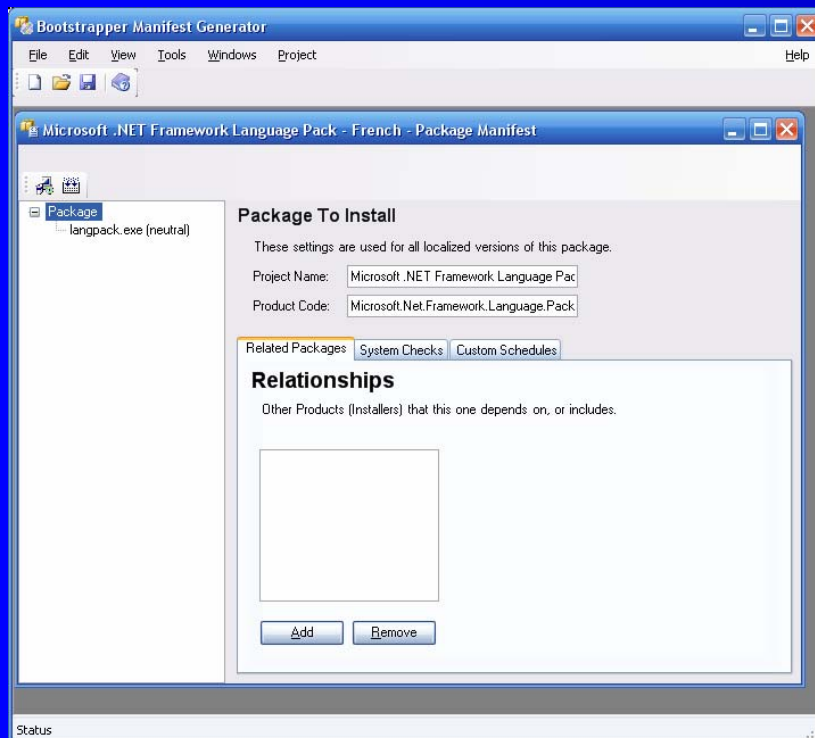
.NET Framework Language Packs Strategy 1

- Create a single .NET Framework Language Pack product and a package for each culture
- The correct package is automatically selected by the choice of Publish language



.NET Framework Language Packs Strategy 2

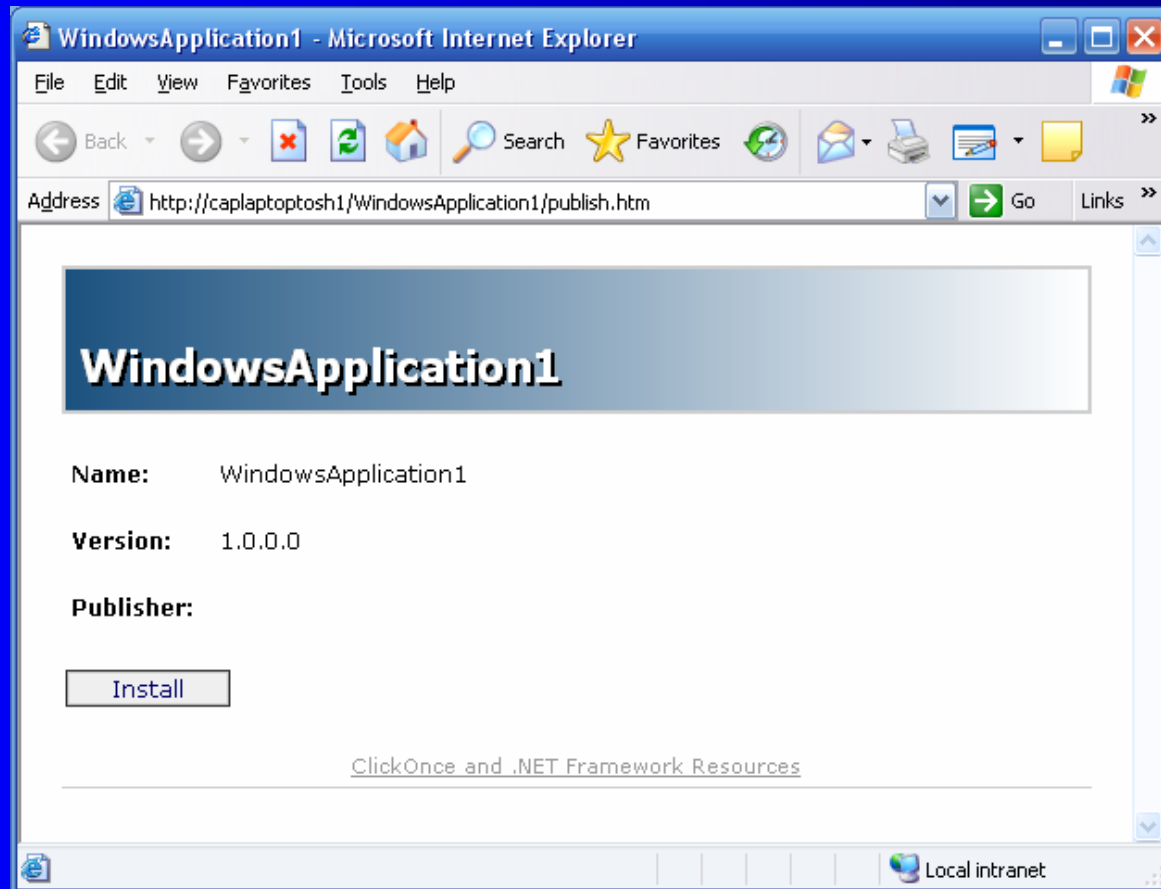
- Create a .NET Framework Language Pack product with a single package for each culture
- The correct product is manually selected and the Publish language is ignored



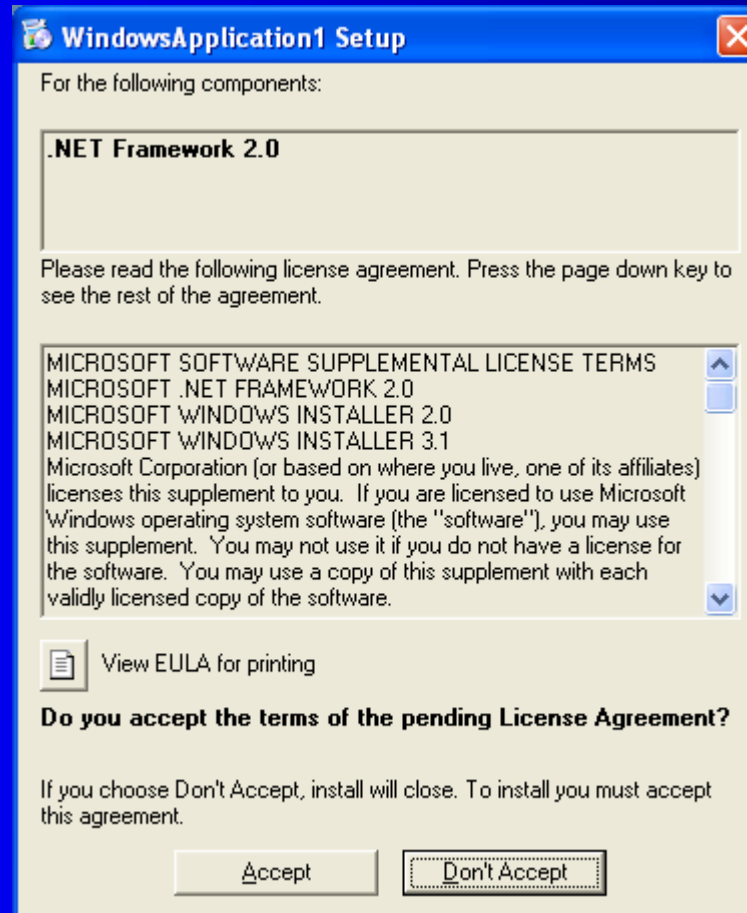
Publishing Conclusions

Tool	Set Assembly Include/Exclude	Set Publish Language	Bandwidth Requirement	Verdict
Publish individual cultures				
VS2005	All files to Auto	Yes	Low	Laborious
msbuild	All files to Auto	Yes	Low	Simple
Publish groups of languages				
VS2005	Group files to Include Others to Exclude	Yes	Low	Laborious
msbuild	Group files to Include Others to Exclude	Yes	Low	Complex
Publish all cultures				
VS2005	All files to Include	Yes or No	High	Simple
msbuild	All files to Include	Yes or No	High	Simple

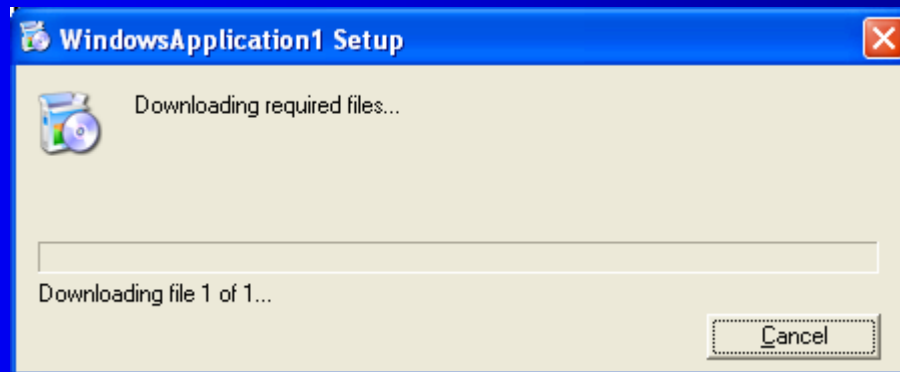
ClickOnce UI StoryBoard publish.htm



ClickOnce UI StoryBoard Bootstrapper Dialogs

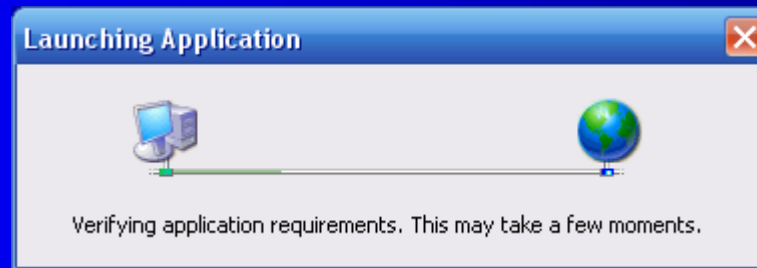


ClickOnce UI StoryBoard Bootstrapper Dialogs



ClickOnce UI StoryBoard

ClickOnce Dialogs



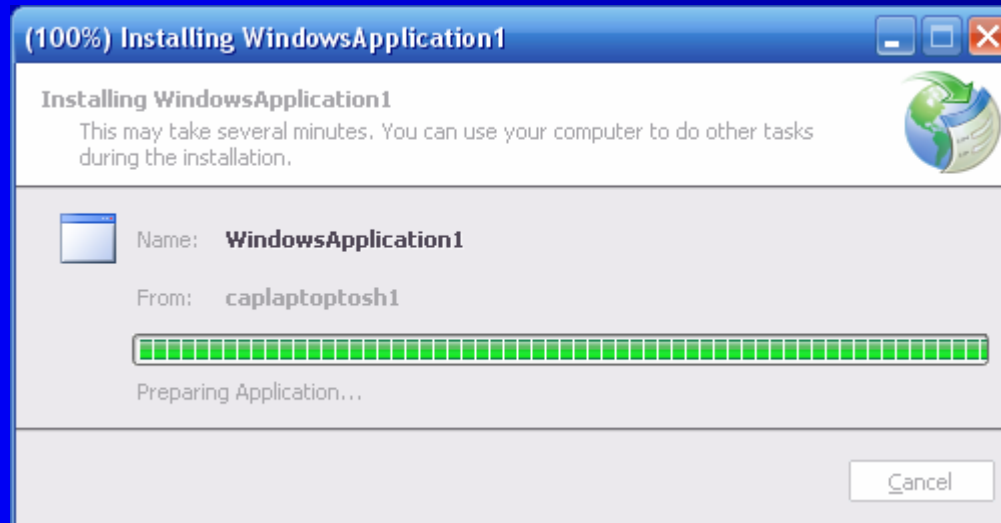
ClickOnce UI StoryBoard

ClickOnce Dialogs



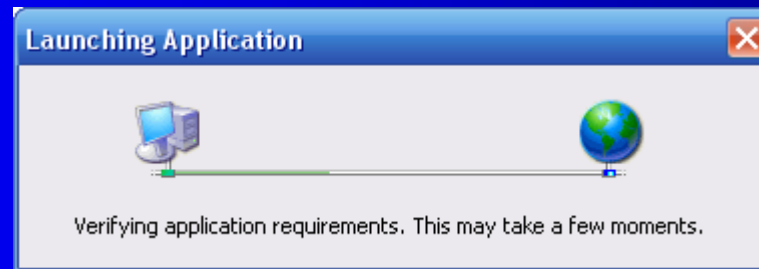
ClickOnce UI StoryBoard

ClickOnce Dialogs



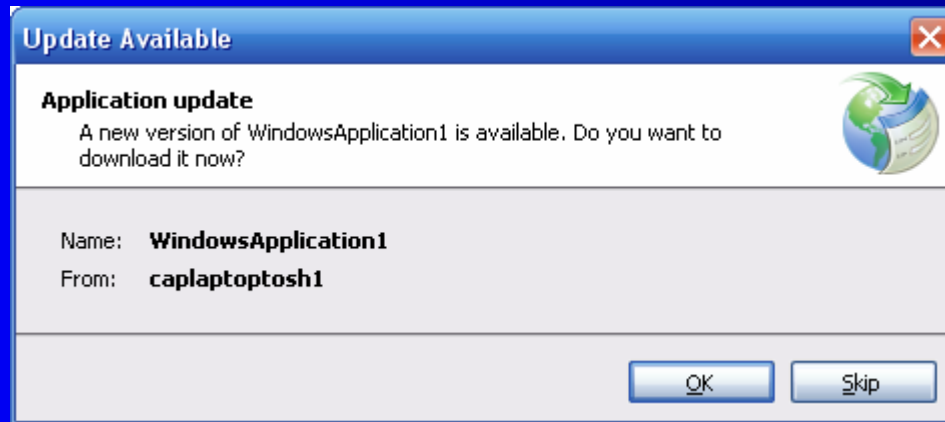
ClickOnce UI StoryBoard

ClickOnce Dialogs



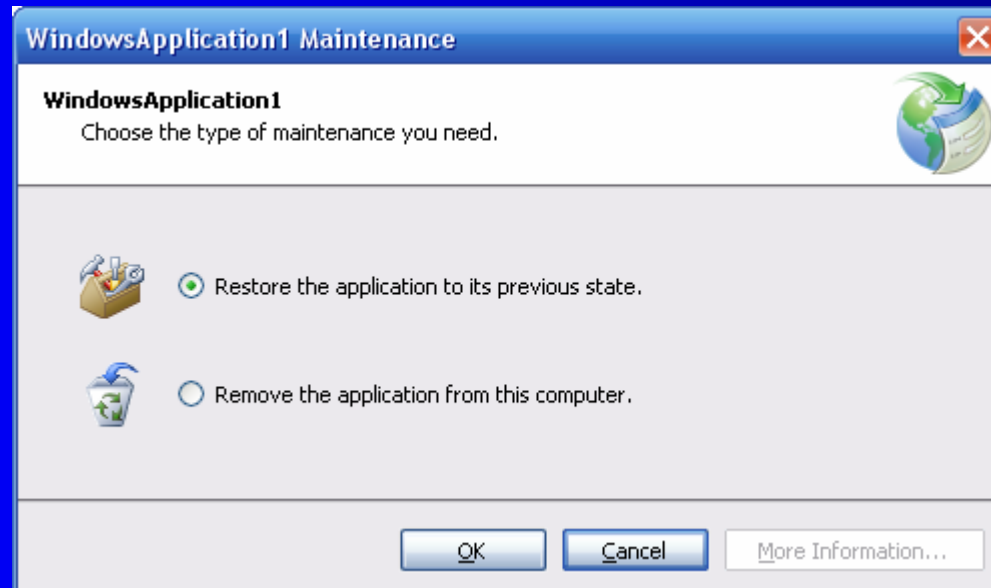
ClickOnce UI StoryBoard

ClickOnce Dialogs



ClickOnce UI StoryBoard

ClickOnce Dialogs



Origins Of The ClickOnce User Interface

Dialog Or Page

Origin

publish.htm

Visual Studio 2005

Bootstrapper (setup.exe) dialogs

ClickOnce bootstrapper resources

“Launching Application” dialog

.NET Framework Language Pack

“Security Warning” dialog

.NET Framework Language Pack

“Installing” dialog

.NET Framework Language Pack

“Update Available” dialog

.NET Framework Language Pack

“Updating” dialog

.NET Framework Language Pack

Add/Remove Programs dialog

.NET Framework Language Pack

publish.htm

- The publish.htm language is determined by:-
 - The language version of Visual Studio 2005
 - The presence of a corresponding .NET Framework Language Pack
 - The Windows O/S Language

Visual Studio 2005	.NET Framework Language Pack	Windows Culture	publish.htm Language
English	English	English	English
English	English	French	English
English	French	English	English
English	French	French	English
French	French	English	English
French	French	French	French

ClickOnce Bootstrapper Dialogs

- The language used in the bootstrapper dialogs is determined by the Publish language
- Bootstrapper files are in
`%FrameworkSDK%\Bootstrapper`
- The bootstrapper (setup.exe) is created by combining:-
 - setup.bin
 - `<language>\setup.xml`
- Create new languages by creating new setup.xml files
- The language of the bootstrapper is determined at publish time – not at runtime

ClickOnce Dialogs

- The ClickOnce Engine (dfsvc.exe) is a .NET Framework application
 - It uses ResourceManager to load resources
 - ResourceManager defaults to CurrentUICulture
 - CurrentUICulture defaults to the Windows O/S language
- ClickOnce dialogs are determined by:-
 - The Windows O/S language
 - The presence of a .NET Framework Language Pack
- The language of the ClickOnce dialogs on a single machine is the same for **all** applications

ClickOnce UI Conclusions

- Can Control:-
 - Publish.htm
 - ClickOnce Bootstrapper
 - Installation of .NET Framework Language Packs
- Cannot Control:-
 - User's O/S language
 - Creation of new .NET Framework Language Packs

The Importance Of The Publish Language

- The Publish language determines:-
 - The inclusion/exclusion of satellite assemblies (when Publish Status is Auto)
 - The selection of the corresponding bootstrapper pre-requisite package
 - The language of the bootstrapper dialogs
- A single setting is used for all of these issues

Summary

- Choose your publishing strategy wisely considering:-
 - Footprint and bandwidth
 - Updates and bandwidth
 - Complexity of build process
- Consider the language of the ClickOnce UI
 - Avoid changing languages during installation
- Consider what impact setting the Publish Language has