

Condensed WIX

Manual de supervivència

Bernat Orellana

Contenido

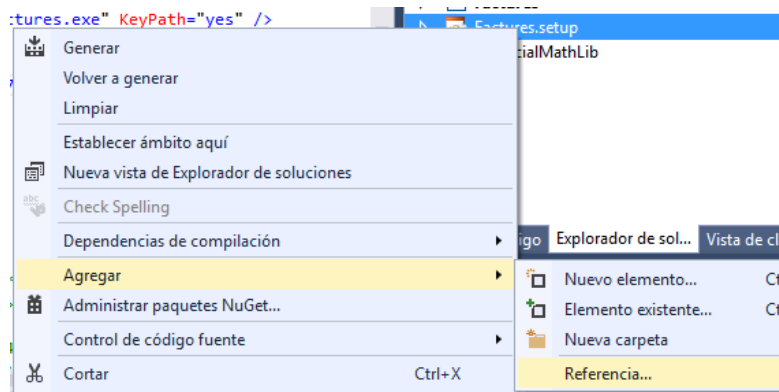
Descarrega i instal·lació de WIX	3
Preparació	3
Creació de la jerarquia de carpetes	3
Components, grups de components i arxius	4
Interfície gràfica bàsica	5
Features	6
Gestió de molts arxius	7
Recopilació d'arxius amb HEAT	7
Shorcut al menú d'inici	8
Shorcut a l'escriptori	9
Afegir icona al shorcut	9
Projecte amb llibreries	9
Opció A) copiar la dll a la mateixa carpeta que l'executable:.....	9
Opció B) Copiar la dll a la GAC (Global assembly cache)	10
Personalitzacions senzilles de la UI	10
Custom actions:	10
Custom actions directes sobre XML: Engegar l'aplicació opcionalment després de l'instal·lador.	10
CustomActions programades en VB.NET	11
Exercicis	14
Exercici 1: programar el reinici de l'ordinador després de la instal·lació (ScheduleReboot)	14
Exercici 2: Custom Action per instal·lar un script de BD com a part final de la instal·lació. Ha d'esborrar les dades en cas de desinstal·lació.....	14

Descarrega i instal·lació de WIX

wixtoolset.org

Preparació

1. Crear un projecte .NET en el que programarem una aplicació
2. Per crear l'instal·lador cal afegir un **nou projecte** WIX al projecte anterior (dins d'una mateixa solució).
3. **IMPORTANT:** Feu que el projecte WIX referencii al projecte .NET (Add Reference / Project i seleccioneu el projecte):



Creació de la jerarquia de carpetes

Creació de les carpetes necessàries a la màquina destí on s'està instal·lant el software.

```
<!-- Definició de l'estructura de carpetes que es crearà en la màquina on s'instal·li el software -->
<Directory Id="TARGETDIR" Name="SourceDir">
  <!-- Archivos de programa -->
  <Directory Id="ProgramFilesFolder">
    <Directory Id="INSTALLDIR" Name="Factures.setup" >
      <Directory Id="DOCFOLDER" Name="doc" />
    </Directory>
  </Directory>

  <!-- Programas -->
  <Directory Id="ProgramMenuFolder">
    <Directory Id="PROGRAM_MENU_FOLDER" Name="Factures" />
  </Directory>

  <!-- Escritori -->
  <Directory Id="DesktopFolder" />

</Directory>
```

Taula de noms per referenciar les carpetes especials :

Directory property	Actual path
AdminToolsFolder	Full path to directory containing administrative tools
AppDataFolder	Full path to roaming folder for current user
CommonAppDataFolder	Full path to application data for all users
CommonFiles64Folder	Full path to the 64-bit Common Files folder
CommonFilesFolder	Full path to the Common Files folder for current user
DesktopFolder	Full path to the Desktop folder
FavoritesFolder	Full path to the Favorites folder for current user
FontsFolder	Full path to the Fonts folder
LocalAppDataFolder	Full path to folder containing local (non-roaming) applications
MyPicturesFolder	Full path to the Pictures folder
NetHoodFolder	Full path to the NetHood folder
PersonalFolder	Full path to the Documents folder for current user
PrintHoodFolder	Full path to the PrintHood folder
ProgramFiles64Folder	Full path to the 64-bit Program Files folder
ProgramFilesFolder	Full path to 32-bit Program Files folder
ProgramMenuFolder	Full path to Program Menu folder
RecentFolder	Full path to Recent folder
SendToFolder	Full path to the SendTo folder for current user
StartMenuFolder	Full path to the Start Menu folder
StartupFolder	Full path to the Startup folder
System16Folder	Full path to the 16-bit system DLLs folder
System64Folder	Full path to the System64 folder
SystemFolder	Full path to the System folder for current user
TempFolder	Full path to the Temp folder
TemplateFolder	Full path to the Template folder for current user
WindowsFolder	Full path to the Windows folder

Components, grups de components i arxius

Cada component conté un arxiu. Un arxiu referencia un arxiu real de la màquina on es compila l'instal·lador, usant l'estructura d'arxius del projecte Visual Studio.

```
<ComponentGroup Id="ProductComponents" Directory="INSTALLDIR">

  <Component Id="CMP_EXECUTABLE" Guid="{0DC546CB-11E4-4BAE-937D-F08C832A5C21}">
    <File Id="FILE_Factures_exe" Source="$(var.Factures.TargetDir)Factures.exe" KeyPath="yes" />
  </Component>

</ComponentGroup>
```

Com es veu a l'exemple, podem usar variables per referenciar a carpetes dels projectes que volem instal·lar, de la forma:

- `$(var.[nom_projecte].TargetDir)`
- `$(var.[nom_projecte].ProjectDir)`

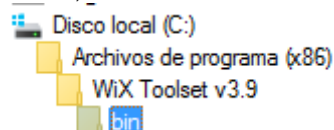
La taula completa de variables disponibles és:

Example usage	Example value
<code>\$(var.MyProject.Configuration)</code>	Debug or Release
<code>\$(var.MyProject.FullConfiguration)</code>	Debug AnyCPU
<code>\$(var.MyProject.Platform)</code>	AnyCPU, Win32, x64 or ia64
<code>\$(var.MyProject.ProjectDir)</code>	C:\users\myusername\Documents\Visual Studio 2010\Projects\MyProject\
<code>\$(var.MyProject.ProjectExt)</code>	.csproj
<code>\$(var.MyProject.ProjectFileName)</code>	MyProject.csproj
<code>\$(var.MyProject.ProjectName)</code>	MyProject
<code>\$(var.MyProject.ProjectPath)</code>	C:\users\myusername\Documents\Visual Studio 2010\Projects\MyProject\MyApp.csproj
<code>\$(var.MyProject.TargetDir)</code>	C:\users\myusername\Documents\Visual Studio 2010\Projects\MyProject\bin\Debug\
<code>\$(var.MyProject.TargetExt)</code>	.exe
<code>\$(var.MyProject.TargetFileName)</code>	MyProject.exe
<code>\$(var.MyProject.TargetName)</code>	MyProject
<code>\$(var.MyProject.TargetPath)</code>	C:\users\myusername\Documents\Visual Studio 2010\Projects\MyProject\bin\Debug\MyProject.exe
<code>\$(var.MyProject.en-US.TargetPath)</code>	C:\users\myusername\Documents\Visual Studio 2010\Projects\MyProject\bin\Debug\en-US\MyProject.msm
<code>\$(var.SolutionDir)</code>	C:\users\myusername\Documents\Visual Studio 2010\Projects\MySolution\
<code>\$(var.SolutionExt)</code>	.sln
<code>\$(var.SolutionFileName)</code>	MySolution.sln
<code>\$(var.SolutionName)</code>	MySolution
<code>\$(var.SolutionPath)</code>	C:\users\myusername\Documents\Visual Studio 2010\Projects\MySolution\MySolution.sln

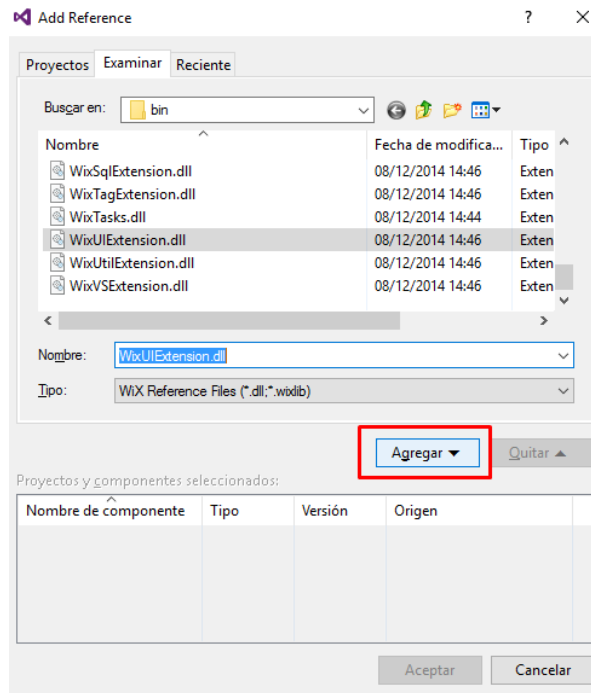
Interfície gràfica bàsica

Abans de poder usar la interfície gràfica de WIX ens cal afegir una referència a una DLL interna de WIX anomenada **WixUIExtensions.dll**.

Feu “Add Reference” sobre el projecte, aneu a la carpeta de binaris de wix (*C:\Archivos de programa(x86)\Wix Toolset vX.X\bin*)



Seleccioneu la pestanya examinar i busqueu la dll i agregueu-la:



Introduir les línies següents abans de tancar el tag `<product>`

```
<!-- Interfície gràfica avançada -->
<UIRef Id="WixUI_Mondo" />
<UIRef Id="WixUI_ErrorProgressText" />

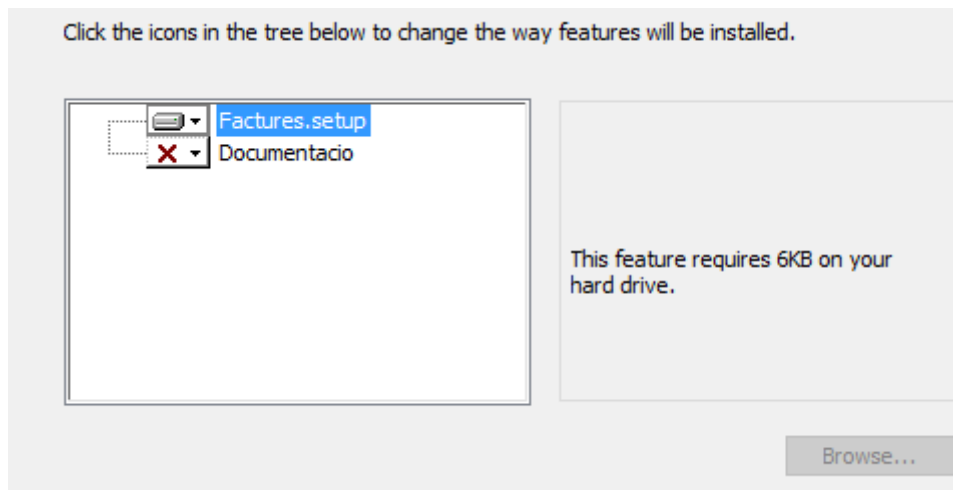
<!-- indica que la interfície d'usuari podrà modificar l'arxiu d'instal·lació. INSTALLDIR és l'identificador
que s'ha de crear dins de ProgramFilesFolder. L'usuari que instal·la pot decidir canviar-la. -->
<Property Id="WIXUI_INSTALLDIR" Value="INSTALLDIR" />
```

Features

Podem definir varis grups de característiques que són instal·lables per separat. Tots els Components han d'estar necessàriament referenciats dins d'un *Feature*, altrament ens donarà un error d'*Orphaned Component*.

```
<Feature Id="ProductFeature" Title="Factures.setup" Level="1">
  <ComponentGroupRef Id="ProductComponents" />
  <ComponentGroupRef Id="ShorcutsProgramMenu" />
  <ComponentGroupRef Id="ShorcutsDesktop" />
</Feature>
<Feature Id="DocFeature" Title="Documentacio" Level="2" >
  <ComponentGroupRef Id="CPG_DOCS" />
</Feature>
```

La configuració anterior provoca que la UI de l'instal·lador ens doni l'opció d'activar o desactivar dues *Features*:

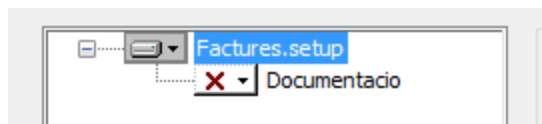


Per defecte, totes les *Features* amb level 1 s'instal·len, i els nivells superiors (level 2 o majors) queden com a opcionals. El nivell de tall , 1 per defecte, es pot configurar si escau usant :

```
<Property Id="INSTALLLEVEL" Value="3" />
```

També podem crear subfeatures dins d'altres *Features*:

```
<Feature Id="ProductFeature" Title="Factures.setup" Level="1">
  <ComponentGroupRef Id="ProductComponents" />
  <ComponentGroupRef Id="ShorcutsProgramMenu" />
  <ComponentGroupRef Id="ShorcutsDesktop" />
  <!-- Subfeature-->
  <Feature Id="DocFeature" Title="Documentacio" Level="2" >
    <ComponentGroupRef Id="CPG_DOCS" />
  </Feature>
</Feature>
```

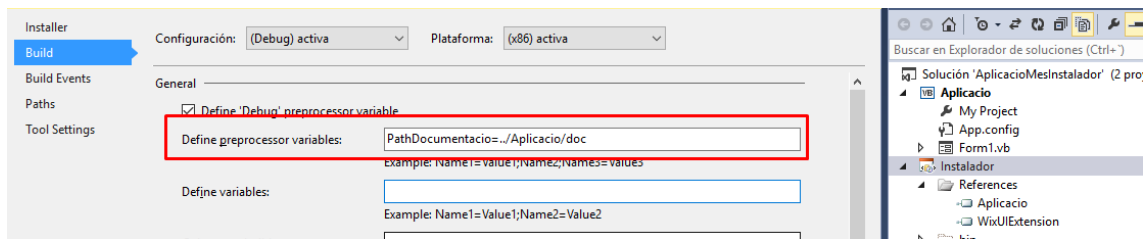


Gestió de molts arxius

Recopilació d'arxius amb HEAT

```
<?xml version="1.0" encoding="utf-8"?>
<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi">
  <Fragment>
    <DirectoryRef Id="DOCFOLDER">
      <Directory Id="dir30D1CB388152588FFE4E338F7C48FFCE" Name="bootstrap" />
    </DirectoryRef>
  </Fragment>
  <Fragment>
    <ComponentGroup Id="CPG_DOCS">
      <Component Id="cmpC7C0CE96F943CC45078812C95858CD4D" Directory="DOCFOLDER" Guid="D265530E-10C0-4B75-88CB-B6743BABA09">
        <File Id="fill118683848C819026AB895602ABF1E689" KeyPath="yes" Source="$(var.PathDocumentacio)\estil.css" />
      </Component>
      <Component Id="cmp5CBA4E06570D803F5B0981E914175470" Directory="DOCFOLDER" Guid="DB637427-CEB4-4AEC-A66A-B7661ACA240F">
        <File Id="fill6D7013AAD63E6DE91D3FB29CAFC8EB7C" KeyPath="yes" Source="$(var.PathDocumentacio)\estil.less" />
      </Component>
      <Component Id="cmpF87DA3734774E89EE3AB6DDAD0A4621D" Directory="DOCFOLDER" Guid="B781C38A-1958-4EBF-B871-C9F0ED34FF77">
        <File Id="fill1A13E9F9972B56451D7025E3853D888" KeyPath="yes" Source="$(var.PathDocumentacio)\exercici.html" />
      </Component>
      <Component Id="cmpA4C98EE64247604DED5745AE1CF33649" Directory="DOCFOLDER" Guid="40924D82-C3D7-4AFC-A716-4FF9FEBEC536E">
        <File Id="fill1E97701E35DE4FC74EAB5CBF967CD76A" KeyPath="yes" Source="$(var.PathDocumentacio)\help.txt" />
      </Component>
    </ComponentGroup>
  </Fragment>
```

Cal definir la variable usant Project Properties/Build :



Shorcut al menú d'inici

En primer lloc cal crear una subcarpeta allotjada al menú d'inici:

```
<Fragment>
  <Directory Id="TARGETDIR" Name="SourceDir">
    <Directory Id="ProgramFilesFolder">
      <Directory Id="INSTALLFOLDER" Name="KKApllication" >
        <Directory Id="DOCFOLDER" Name="doc" />
      </Directory>
    </Directory>

    <Directory Id="ProgramMenuFolder">
      <Directory Id="CARPETA_DEL_MENU_INICI" Name="KKApllication" />
    </Directory>

  </Directory>
```

La segona part consisteix a definir un ComponentGroup que s'instal·li a la carpeta anterior:

```
<ComponentGroup Id="ShorcutsProgramMenu" Directory="PROGRAM_MENU_FOLDER">
  <Component Id="CMP_SHORCUT_PROGRAMA" Guid="{60C8998C-4888-4782-8E5F-4B040C121A13}" >
    <Shortcut Id="DocumentationStartMenuShortcut"
      Name="Factures"
      Description="Factures Online"
      Target="[INSTALLDIR]Factures.exe"
      Icon="IconaDePrograma" />

    <!-- accés directe al desinstal·lador -->
    <Shortcut Id="UninstallShortcut"
      Name="Uninstall Factures"
      Description="Uninstalls Factures and all of its components"
      Target="[System64Folder]msiexec.exe"
      Arguments="/x [ProductCode]" />

    <!-- RemoveMyShortcutsDir = ID escombraries ( garbage ) -->
    <RemoveFolder Id="RemoveMyShortcutsDir" On="uninstall" />

    <!-- deixem marca al registre -->
    <RegistryValue Root="HKCU"
      Key="Software\Microsoft\Factures"
      Name="installed"
      Type="integer"
      Value="1"
      KeyPath="yes" />
  </Component>
</ComponentGroup>
```


Shortcut a l'escriptori

```
<ComponentGroup Id="ShortcutsDesktop" Directory="DesktopFolder">
  <Component Id="CMP_SHORCUT_PROGRAMA_DESKTOP" Guid="{43326C47-A4CA-43BB-95FB-B8C5DE70D1D0}" >
    <Shortcut Id="DocumentationDesktopShortcut"
      Name="Factures"
      Description="Factures Online"
      Target="[INSTALLDIR]Factures.exe"
      Icon="IconaDePrograma" />

    <RegistryValue Root="HKCU"
      Key="Software\Microsoft\Factures"
      Name="installedDesktop"
      Type="integer"
      Value="1"
      KeyPath="yes" />
  </Component>
</ComponentGroup>
```

Afegir icona al shortcut

- 1) Al projecte "normal" assignar una icona a l'aplicació (propietats del projecte, pestanya *Aplicación*, desplegable *Icono*) Podeu usar <http://imagen.online-convert.com/es/convertir-a-ico> per convertir les vostres icones a format *.ico.
- 2) Al projecte wix, dins de l'arxiu *.wxs. cal afegir el següent fragment abans de tancar <product>:

```
<!-- Definim la icona a utilitzar en els accessos directes . La referenciarem més endavant. -->
<Icon Id="IconaDePrograma" SourceFile="$(var.Factures.TargetDir)Factures.exe" />
```

- 3) als <shortcut afegir l'atribut:

```
Icon="IconaDePrograma"
<Shortcut Id="DocumentationStartMenuShortcut"
  Name="Factures"
  Description="Factures Online"
  Target="[INSTALLDIR]Factures.exe"
  Icon="IconaDePrograma" />
```

Projecte amb llibries

Opció A) copiar la dll a la mateixa carpeta que l'executable:

```
<ComponentGroup Id="ProductComponents" Directory="INSTALLDIR">

  <Component Id="CMP_EXECUTABLE" Guid="{0DC546CB-11E4-4BAE-937D-F08C832A5C21}">
    <File Id="FILE_Factures_exe" Source="$(var.Factures.TargetDir)Factures.exe" KeyPath="yes" />
  </Component>

  <Component Id="CMP_FINANCIAL_LIB_DLL" Guid="{6E5950E9-AF46-4848-9475-A23711D9A47C}">
    <File Id="FILE_FINANCIAL_LIB_DLL"
      Source="$(var.Factures.TargetDir)FinancialMathLib.dll"
      KeyPath="yes" />
  </Component>

</ComponentGroup>
```

Opció B) Copiar la dll a la GAC (Global assembly cache)

Tot i que ho referenciem a una carpeta, si li posem l'atribut Assembly=".net" l'assembly es copiarà a la GAC.

IMPORTANT: Això requereix que la dll estigui **firmada**.

```
<ComponentGroup Id="ProductComponents" Directory="INSTALLDIR">

  <Component Id="CMP_EXECUTABLE" Guid="{0DC546CB-11E4-4BAE-937D-F08C832A5C21}">
    <File Id="FILE_Factures_exe" Source="$(var.Factures.TargetDir)Factures.exe" KeyPath="yes" />
  </Component>

  <Component Id="CMP_FINANCIAL_LIB_DLL" Guid="{6E5950E9-AF46-4848-9475-A23711D9A47C}">
    <File Id="FILE_FINANCIAL_LIB_DLL"
      Source="$(var.Factures.TargetDir)FinancialMathLib.dll"
      Assembly=".net"
      KeyPath="yes" />
  </Component>

</ComponentGroup>
```

Personalitzacions senzilles de la UI

Abans de tancar el tag <product>

```
<!-- Customitzacions simples de la UI: llicència i l'aspecte de l'instal·lador. -->
<WixVariable Id="WixUILicenseRtf" Value="llicencia.rtf" />
<WixVariable Id="WixUIBannerBmp" Value="kraken.bmp" />
<WixVariable Id="WixUIDialogBmp" Value="dialog.bmp" />
```

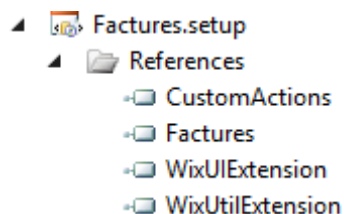
Més informació de configuració/customització de la UI a

<http://wix.tramontana.co.hu/tutorial/user-interface/ui-wizardry>

Custom actions:

Custom actions directes sobre XML: Engegar l'aplicació opcionalment després de l'instal·lador.

Abans de començar, cal agregar al projecte WIX les referències **WixUIExtension.dll** i **WixUtilExtension.dll**



Després indiquen que a la UI cal afegir un checkbox a la finestra de diàleg de finalització de la instal·lació (dins de <Product>):

```
<!-- Afegir un checkbox a la finestra de diàleg final per triar si es vol llançar l'aplicació -->
<Property Id="WIXUI_EXITDIALOGOPTIONALCHECKBOXTEXT" Value="Engegar l'aplicació ara" />
```

Seguidament (just a continuació, també dins de **<Product>**) declarem una “*Custom Action*”, o acció personalitzada consistent a engegar un shell i executar-hi la nostra aplicació:

```
<!-- Definició d'una custom action que engega en el shell de wix l'aplicació -->
<Property Id="WixShellExecTarget" Value="[INSTALLDIR]Factures.exe" />
<CustomAction Id="LaunchApplication" BinaryKey="WixCA" DllEntry="WixShellExec" Impersonate="yes" />
```

Important: INSTALLDIR és l'Id del <Directory> on s'instal·la l'aplicació, i heu de reemplaçar “Factures.exe” pel nom del vostre executable.

Finalment cal lligar el checkbox amb l'execució de la *CustomAction*:

```
<!-- Interfície gràfica avançada -->
<UI>
  <UIRef Id="WixUI_Mondo" />
  <UIRef Id="WixUI_ErrorProgressText" />
  <Publish Dialog="ExitDialog"
    Control="Finish"
    Event="DoAction"
    Value="LaunchApplication">WIXUI_EXITDIALOGOPTIONALCHECKBOX = 1 and NOT Installed</Publish>
</UI>
```

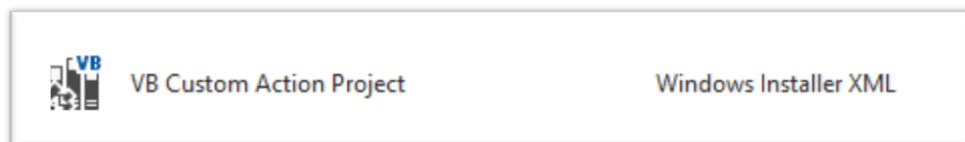
Amb això indiquem que en el diàleg de *ExitDialog*, quan cliquem sobre finish es llança una acció “LaunchApplication” (la que hem definit a abans) quan és certa la condició expresada (checkbox actiu i aplicació encara no instal·lada 100%)

CustomActions programades en VB.NET

Les *CustomActions* també ens permeten executar un petit programa en VB.NET des de WIX, de forma que tinguem plena llibertat per fer operacions complexes sobre la màquina d'instal·lació. A més, podem crear una interfície gràfica a mida usant Windows Forms per exemple.

Els passos a seguir són els següents:

Afegir a la nostra solució (on ja tenim un projecte .NET i un projecte WIX Setup) un nou projecte de tipus WIX VB Custom Action Project:

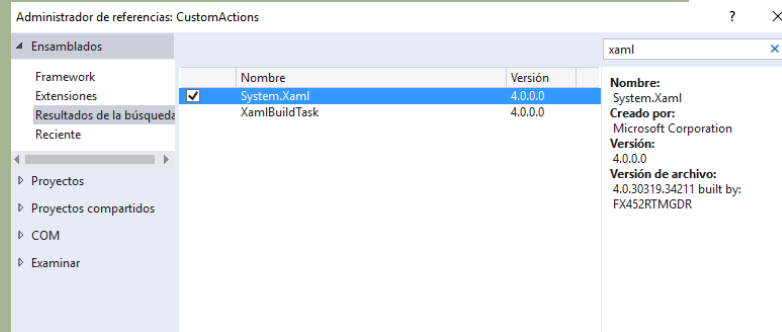


Editar l'arxiu *CustomActions.vb* i programar aquí una *Custom Action*. El nom del mètode no és rellevant, el que importa és que estigui prefixat per l'anotació **<CustomAction()>**. Dins d'aquest mètode crearem un petit programa que faci quelcom. A l'exemple més a sota es mostra una *Custom Action* que obriria un *Form* i en funció de quin botó es premi, es retorna un èxit o fracàs de la *Custom Action*.

IMPORTANT: Per poder usar interfície gràfica des d'una *CustomAction* cal afegir una referència a :

- UI Windows Forms: **System.Windows.Forms.dll**

- *UI WFP: System.Xaml*



Versió Windows Forms/ VB

Imports System.Windows.Forms

0 referencias

```
]Public Class CustomActions
```

```
]    <CustomAction()> _
```

0 referencias

```
Public Shared Function CASaluda(ByVal session As Session) As ActionResult
    session.Log("Begin CustomAction1")
```

```
    Console.WriteLine("Estem passant pel custom action")
```

```
    Dim form As New SetupForm()
```

```
    Dim d As DialogResult = form.ShowDialog()
```

```
    If d = DialogResult.OK Then
```

```
        Return ActionResult.Success
```

```
    Else
```

```
        Return ActionResult.Failure
```

```
    End If
```

```
End Function
```

End Class

Versió WPF / C#

```

public class CustomActions
{
    static Thread _t;
    static ManualResetEvent _done = new ManualResetEvent(false);
    [CustomAction]
    0 referencias | 0 cambios | 0 autores, 0 cambios
    public static ActionResult MostrarPantalla(Session session)
    {
        session.Log("Begin CustomAction1");

        _t = new Thread(WorkerThread);
        _t.SetApartmentState(ApartmentState.STA);
        _t.Start();
        _done.WaitOne();

        return ActionResult.Success;
    }
}

```

```

[STAThread] |
1 referencia | 0 cambios | 0 autores, 0 cambios
static void WorkerThread()
{
    try
    {
        Dialeleg d = new Dialeleg();
        d.ShowDialog();
    }
    finally
    {
        _done.Set();
    }
}
}

```

Un cop programada la *Custom Action*, cal definir-ne la seva utilització des del XML de WIX.

El primer pas és definir un objecte `<Binary>` que apunti a la DLL resultant del projecte de *CustomActions* anteriorment creat. Important, cal referencieu l'arxiu {nom_projecte_custom_actions}.CA.dll (fixeu-vos en el sufix .CA !!)
 Com a *DllEntry* especifiqueu el nom de la funció que cal cridar, p.ex. *CASaluda*,
Return="check" obliga a WIX a validar el retorn de la *Custom Action*. Si és *ActionResult.Failure* es cancel·la la instal·lació.

```

<Binary Id="CustomActionsDLL" SourceFile= "${(var.CustomActions.TargetDir)CustomActions.CA.dll" />

<CustomAction Id="CA_CustomAction"
    BinaryKey="CustomActionsDLL"
    DllEntry="CASaluda" Execute="deferred"
    Return="check" />

```

Finalment, dins de `<Product>` definim en quin moment de la seqüència d'instal·lació llancem la nostra *CustomAction*:

```

<InstallExecuteSequence>
    <Custom Action="CA_CustomAction" After="InstallInitialize" />
</InstallExecuteSequence>

```

A l'atribut *After* indiquem després de quina fase es crida la *Custom Action*:

- AppSearch
- LaunchConditions
- ValidateProductID
- CostInitialize
- FileCost
- CostFinalize
- InstallValidate
- InstallInitialize
- ProcessComponents
- UnpublishFeatures
- RemoveShortcuts
- RemoveFiles
- InstallFiles
- CreateShortcuts
- RegisterUser
- RegisterProduct
- PublishFeatures
- PublishProduct
- InstallFinalize
- RemoveExistingProducts

Condicionant l'execució d'una *CustomAction* per a que només s'executi en desinstal·lacions:

```
<Custom Action="CA_CustomAction" After='PublishProduct'>
    Installed AND NOT UPGRADINGPRODUCTCODE
</Custom>
```

	Install	Uninstall	Change	Repair	Upgrade
Installed	False	True	True	True	True
REINSTALL	False	False	False	True	False
UPGRADINGPRODUCTCODE	False	False	False	False	True
REMOVE	False	True	False	False	True

Also assuming a full repair & uninstall the actual values of properties could be:

	Install	Uninstall	Change	Repair	Upgrade
Installed	" ""	"00:00:00"	"00:00:00"	"00:00:00"	"00:00:00"
REINSTALL	" ""	" ""	" ""	"ALL"	" ""
UPGRADINGPRODUCTCODE	" ""	" ""	" ""	" ""	"{New Product/@Id GUID}"
REMOVE	" ""	"ALL"	" ""	" ""	"ALL"

Exercicis

Exercici 1: programar el reinici de l'ordinador després de la instal·lació (ScheduleReboot)

Exercici 2: Custom Action per instal·lar un script de BD com a part final de la instal·lació. Ha d'esborrar les dades en cas de desinstal·lació.

