
Manual de configuración y compilación de un proyecto con OpenCV en Windows

<http://umh1782.edu.umh.es>

- Requisitos:**
- Windows XP, Windows 7,8,10 - (32-64bits)
 - Microsoft Visual Studio C++ 2010 / 2013
 - Librería OpenCV instalada y configurada

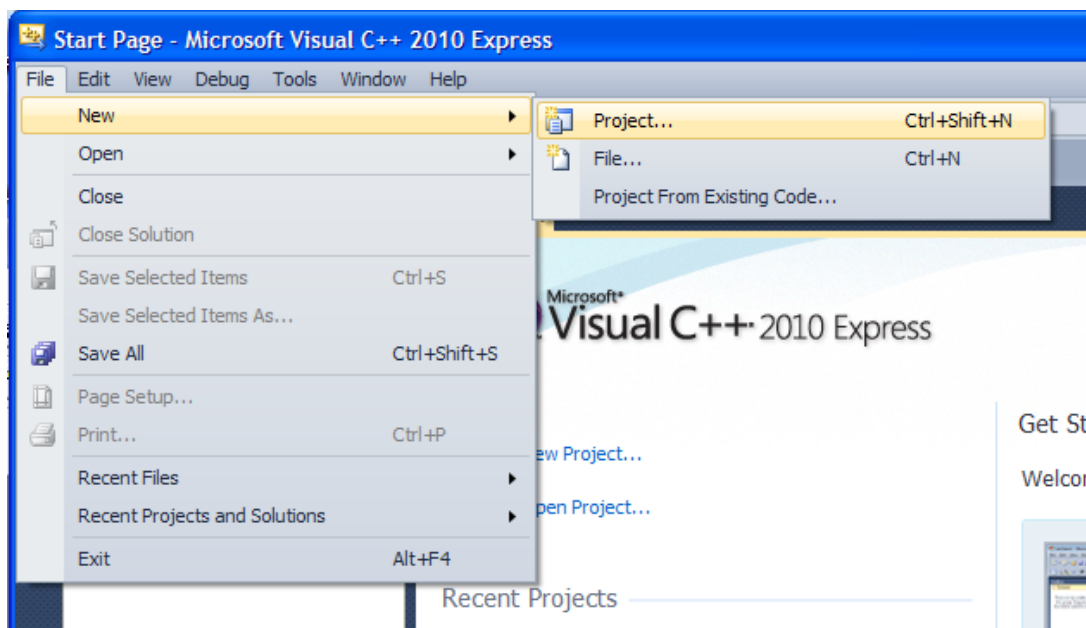
[Tutorial Instalación](#)

Más información en <http://opencv.org>

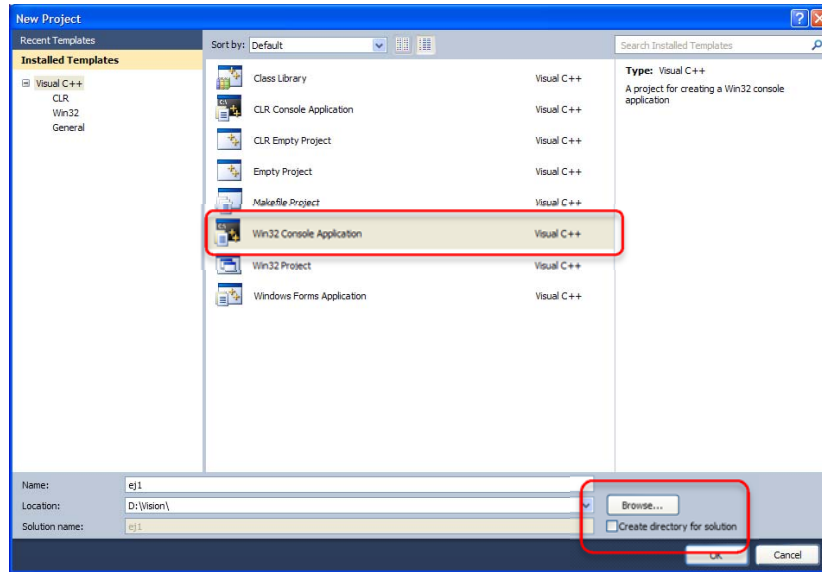
Este tutorial es para la versión en inglés de *Visual Studio C++ Express 2010*.
También es aplicable a la versión Visual Studio 2013

1) Crear el proyecto:

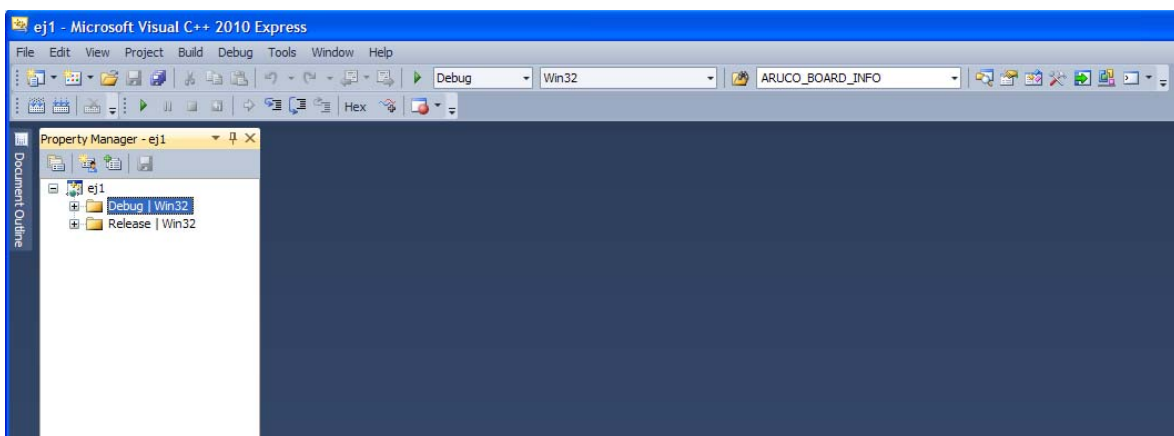
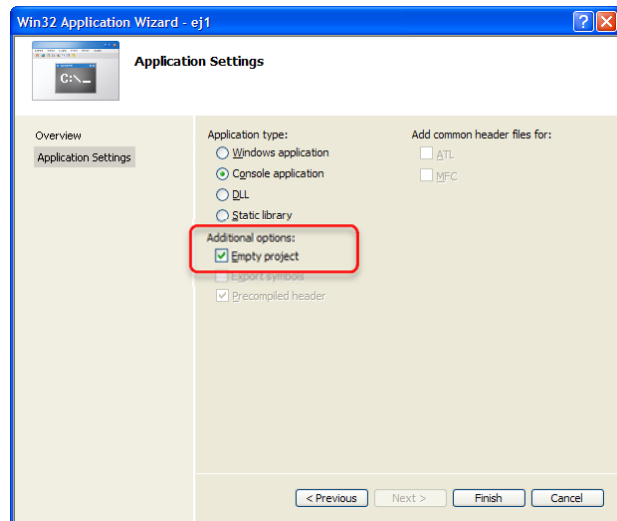
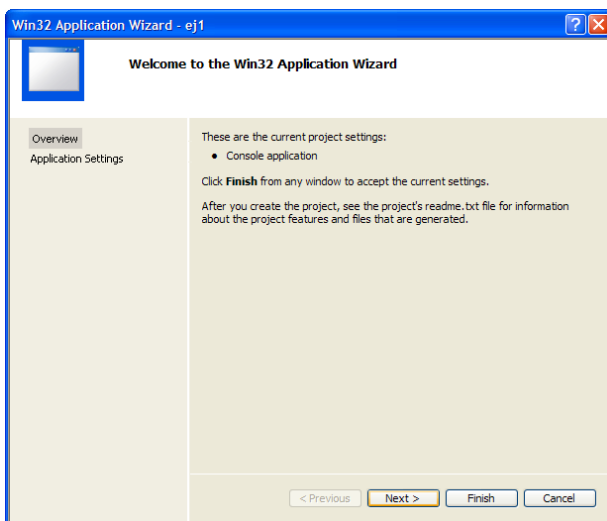
- Ejecutaremos la aplicación **Microsoft Visual C++ 2010** y en el menú **File** seleccionaremos **New/Project**



- Nos aparecerá una ventana donde seleccionaremos el tipo de proyecto (**Win32 Console Application**). Le asignaremos un nombre indicando el directorio donde deseamos que se cree el proyecto y **desactivaremos** la casilla **'Create directory for solution'**. *No te preocupes, ya que aun así se creará el proyecto dentro de una carpeta con el mismo nombre del proyecto.*



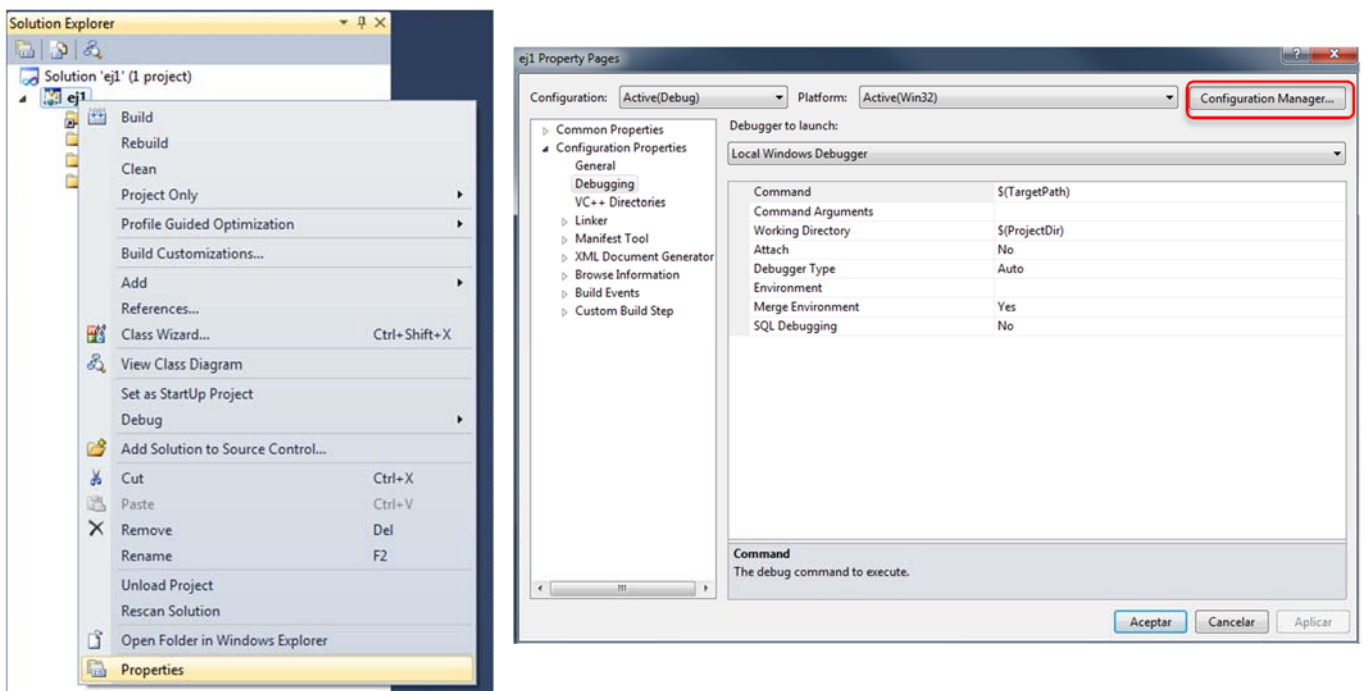
- Lanzará un asistente que nos guiará en los pasos para configurar el proyecto. Al pulsar siguiente, marcaremos la casilla **'Empty project'**. Pulsando el botón **'Finish'** ya tendremos creado nuestro proyecto.



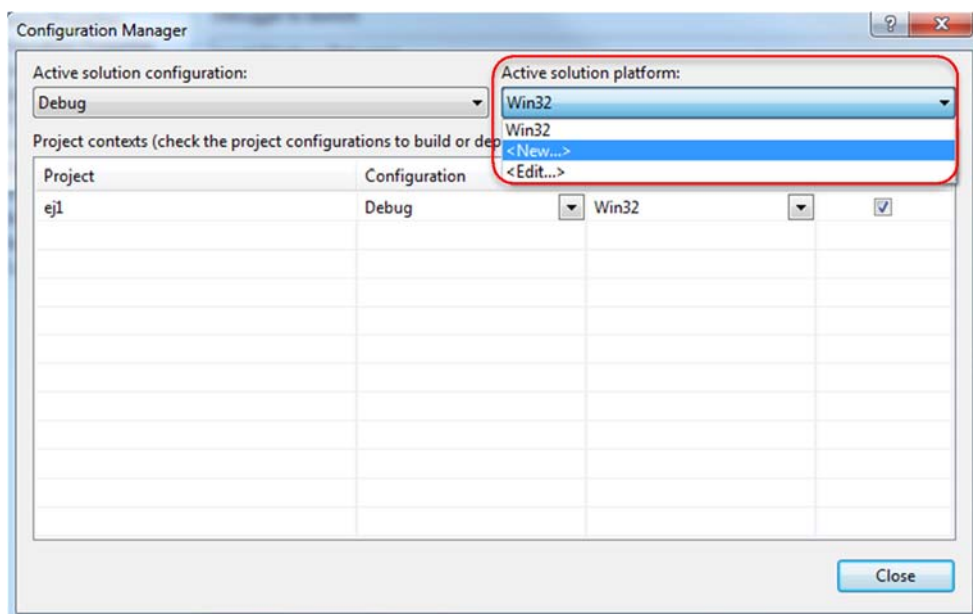
2) Opcional: Configurar el proyecto en la plataforma de 64bits

Por defecto el proyecto que creamos (**Win32 Console Application**) utiliza solo la arquitectura de 32 Bits. Si utilizamos un S.O. de 64 bits y queremos desarrollar nuestra aplicación en 64 bits, deberemos configurarlo manualmente siguiendo las siguientes instrucciones:

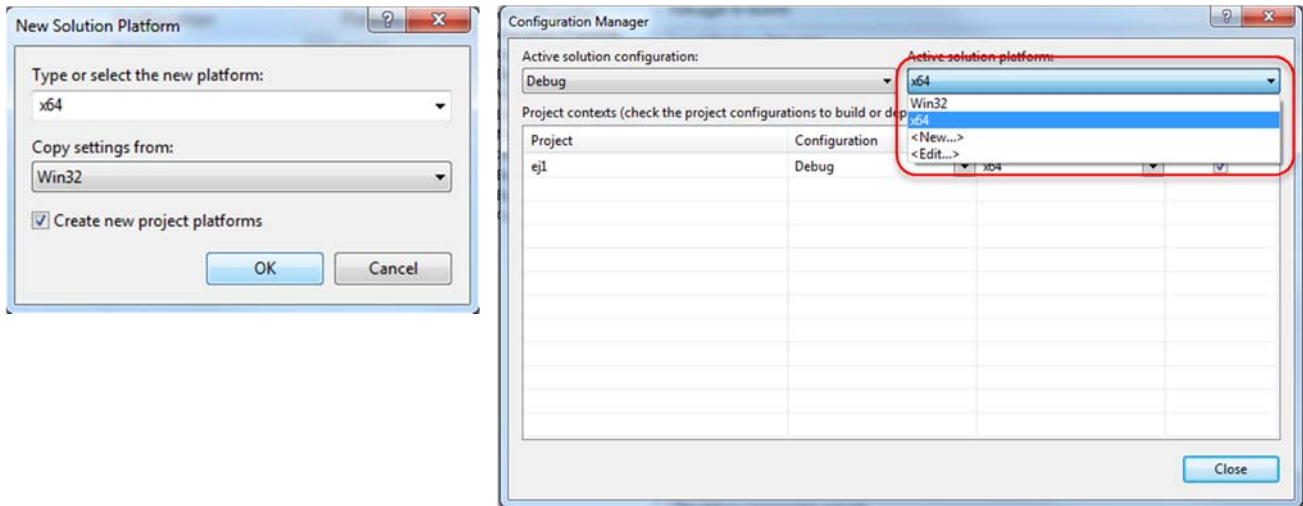
- Pulsaremos el botón derecho del ratón sobre nombre del proyecto '**ej1**', desplegando el menú contextual en el que seleccionaremos '**Properties**'.
- Aparecerá una ventana emergente, pulsaremos en el botón '**Configuration Manager**'.



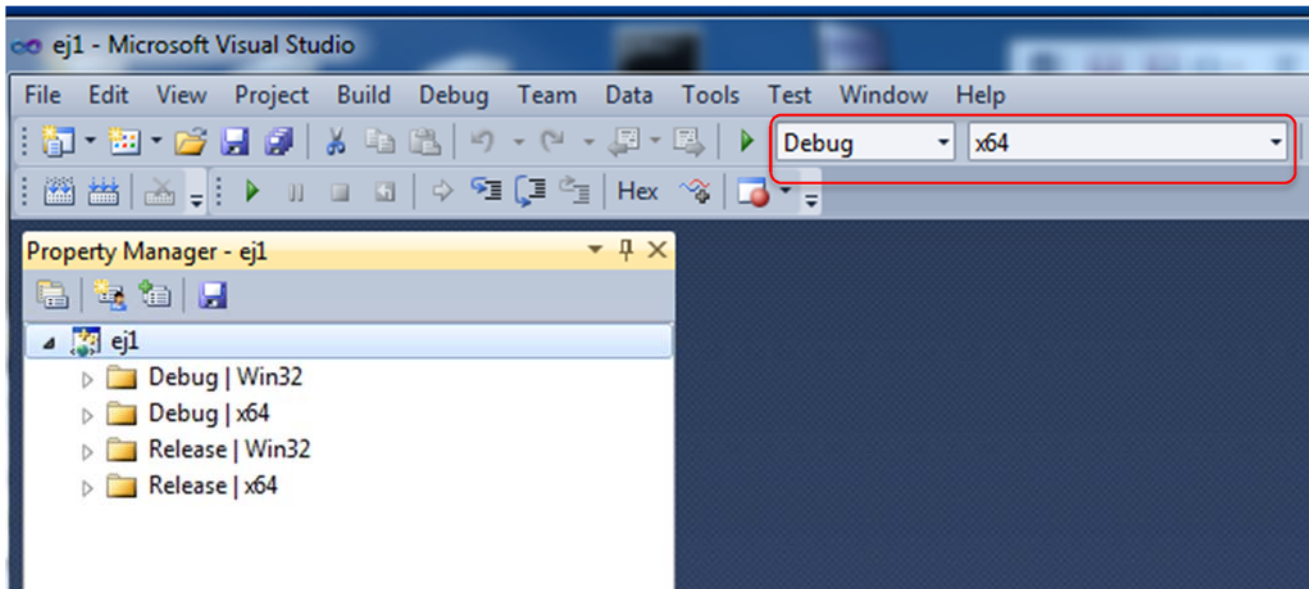
- Aparecerá otra ventana emergente, pulsaremos en el desplegable identificado como '**Active Solution Platform**'. Seleccionando la opción '**New**'.



- En la ventana que nos muestra, dejaremos las opciones por defecto: **'Type of plataform'**-> **'x64'** y **'Copy settings from'**-> **'Win32'**. Pulsaremos el botón **'OK'**. Podemos ver como ya disponemos de las dos plataformas activas (x64, Win32).
- **Nota:** Si lo deseamos podemos borrar la plataforma de 32 bits pulsado en la opción **'Edit'**, pero por compatibilidad con equipos con S.O. de 32 bits no es recomendable.
- Cerraremos las ventanas pulsando **'Close'** y **'Aceptar'**.



Cuando queremos construir nuestro proyecto para una plataforma u otra, lo seleccionaremos en los desplegados disponible en la parte superior del Visual Studio.



A partir de aquí deberemos configurar cada una de las opciones de compilación de forma específica para cada proyecto, tal como se verá en los siguientes apartados.

3) Configurar el proyecto (1ª vez):

Vamos a describir un método para configurar un proyecto que utilice la librería OpenCV, que es reutilizable en varios proyectos o programas. Por tanto distinguiremos dos casos:

- 1) **Primer proyecto:** crearemos unos ficheros especiales con toda la configuración
- 2) **Otros proyectos:** reutilizaremos los ficheros anteriores, por lo que no será necesario repetir los pasos previos

Una vez creado el proyecto, la ventana del programa nos muestra en el panel izquierdo el gestor de configuración del proyecto '**Property Manager**'.

Si no estuviera visible esta ventana, la activaríamos con la opción del menú '**View ▶ Property Manager**' o bien '**View ▶ Other Windows ▶ Property Manager**'.

NOTA: en la versión **VS2010 Express** es necesario activar esta opción en el Menú:
'**Tools ▶ Settings ▶ Expert**'

NOTA: en la versión **VS2013 Express** es necesario activar esta opción en el Menú:
'**Tools ▶ Customize ▶ Pestaña Commands**'

Seleccionar '**Menu bar**' ▶ **View**, Pulsar botón '**Reset All**'

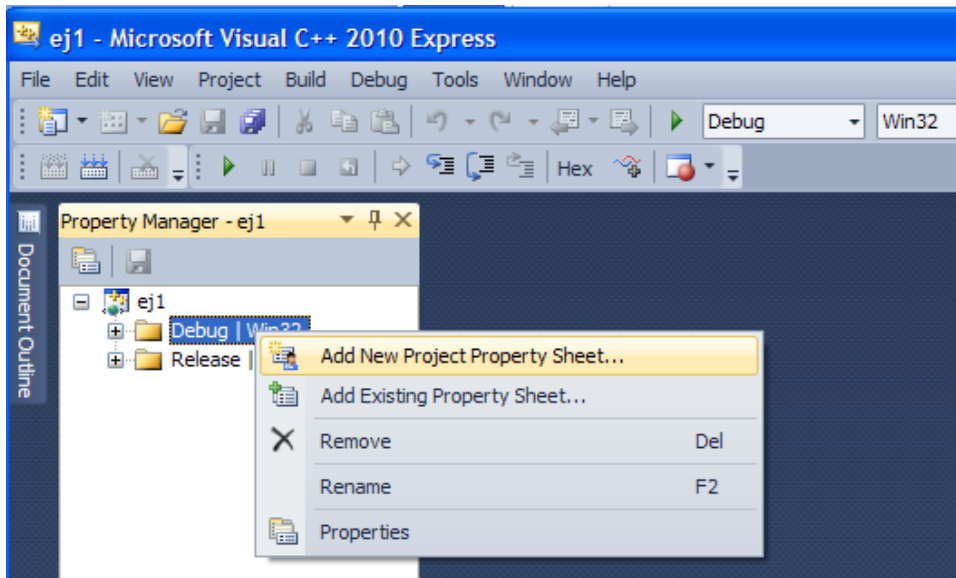
En la ventana del gestor de configuración ('**Property Manager**') nos mostrará la página creada con dos apartados (**Debug/Release**). En Visual C++ tenemos dos modos de compilación, cada uno de ellos con una configuración independiente:

- **Modo Debug:** el compilador añade código extra que nos ayudara a depurar los errores de nuestro programa. Este es el modo que utilizaremos habitualmente
- **Modo Release:** cuando el programa está terminado, la compilación se optimiza eliminando el código de depuración, obteniendo así un ejecutable más pequeño y más rápido.

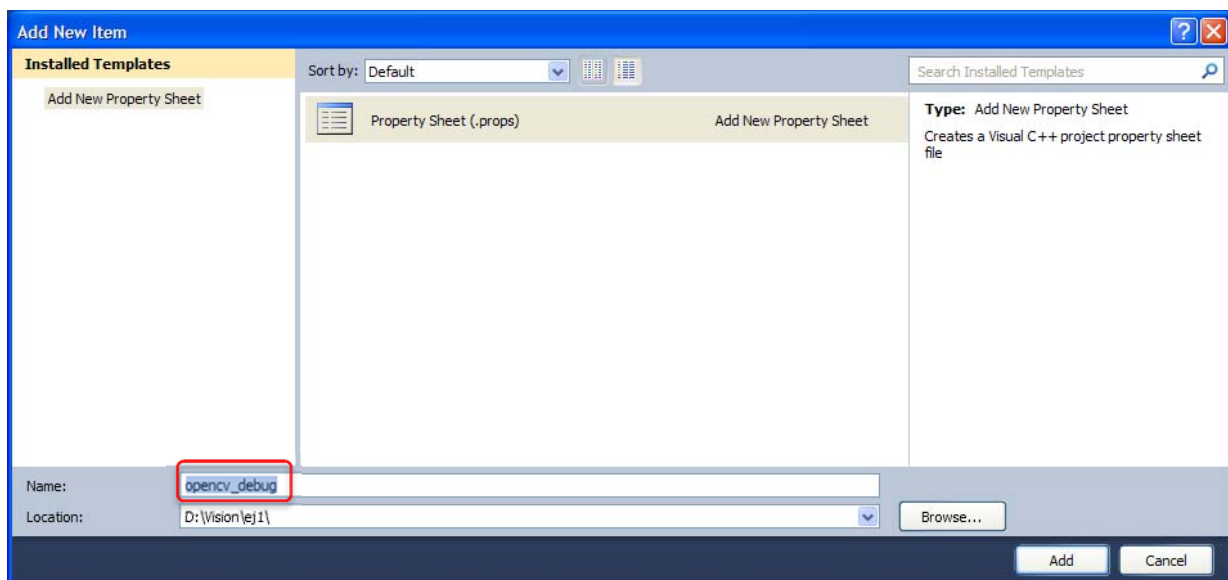
Vamos a describir la configuración de los dos modos:

a) Configuración modo Debug:

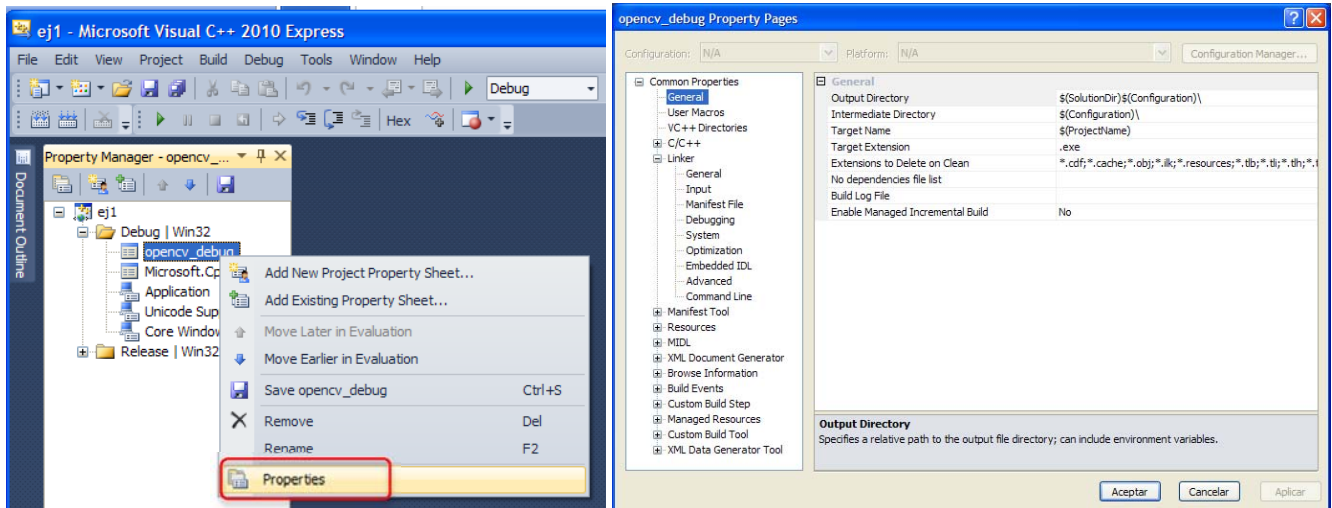
- Pulsaremos el botón derecho del ratón sobre nodo '*Debug*', desplegando el menú contextual en el que seleccionaremos '*Add New Project Property Sheet*'.



- Nos abrirá una nueva ventana donde asignaremos el nombre del fichero de configuración (*opencv_debug*). Pulsando el botón '*Add*' creará el fichero de configuración en la carpeta de nuestro proyecto (*opencv_debug.props*).



- Pulsaremos el botón derecho del ratón sobre la etiqueta '*opencv_debug*' de la hoja de propiedades del modo '*Debug*' que hemos creado. En el menú contextual seleccionaremos la opción '*Properties*'. Esto abrirá la ventana donde realizaremos la configuración del proyecto.



A continuación modificaremos tres apartados del fichero de configuración:

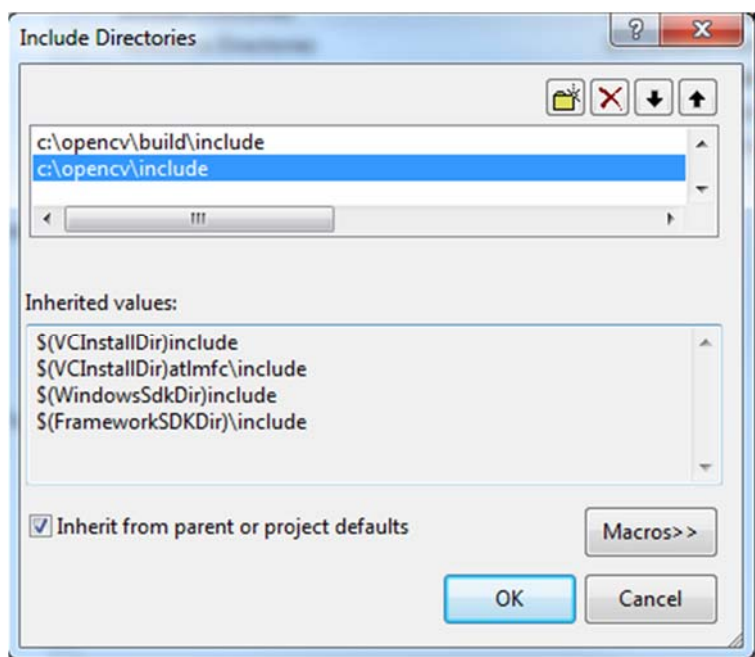
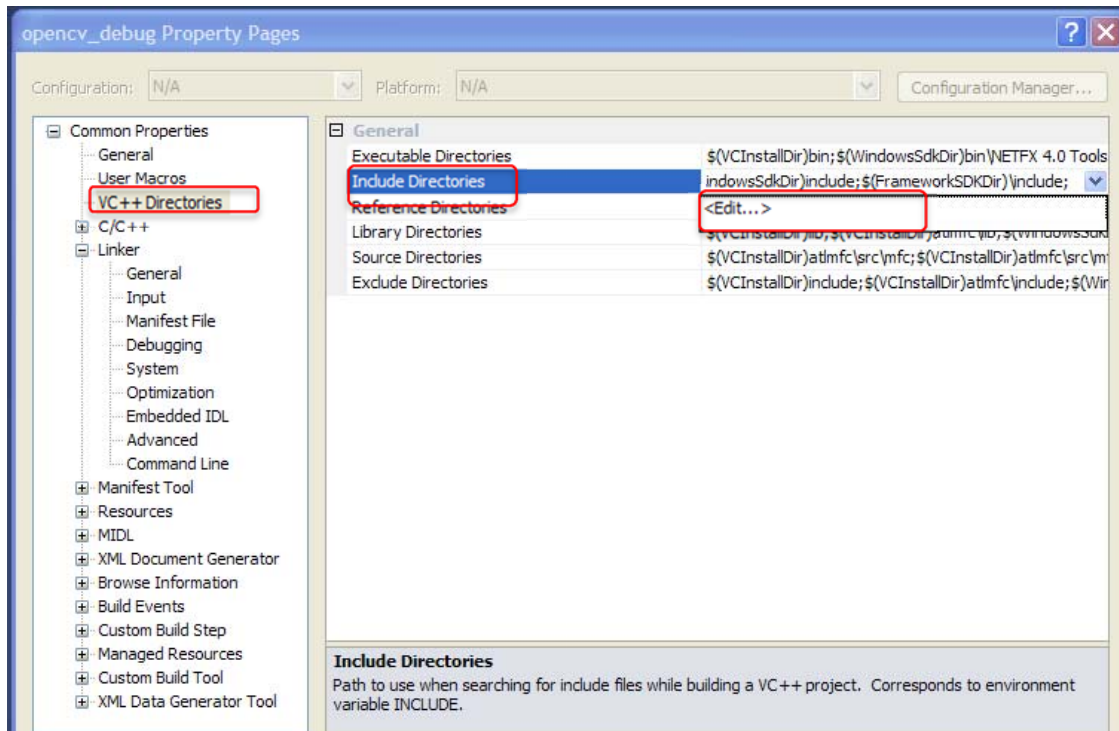
- *VC++ Directories* -> ***Include Directories***
- *VC++ Directories* -> ***Library Directories***
- *Linker* -> *Input* -> ***Additional Dependencies***

Vamos por partes:

a.1) Directorio de cabeceras:

- Editaremos la opción **VC++ Directories -> Include Directories**, pulsando sobre la flecha azul del lateral derecho nos muestra un menú con la opción **<Edit>** que pulsaremos y abrirá una ventana de edición. Añadiremos el directorio: **c:\opencv\build\include**

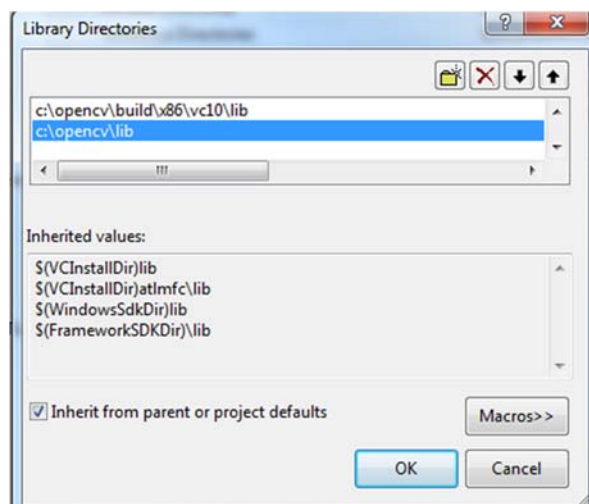
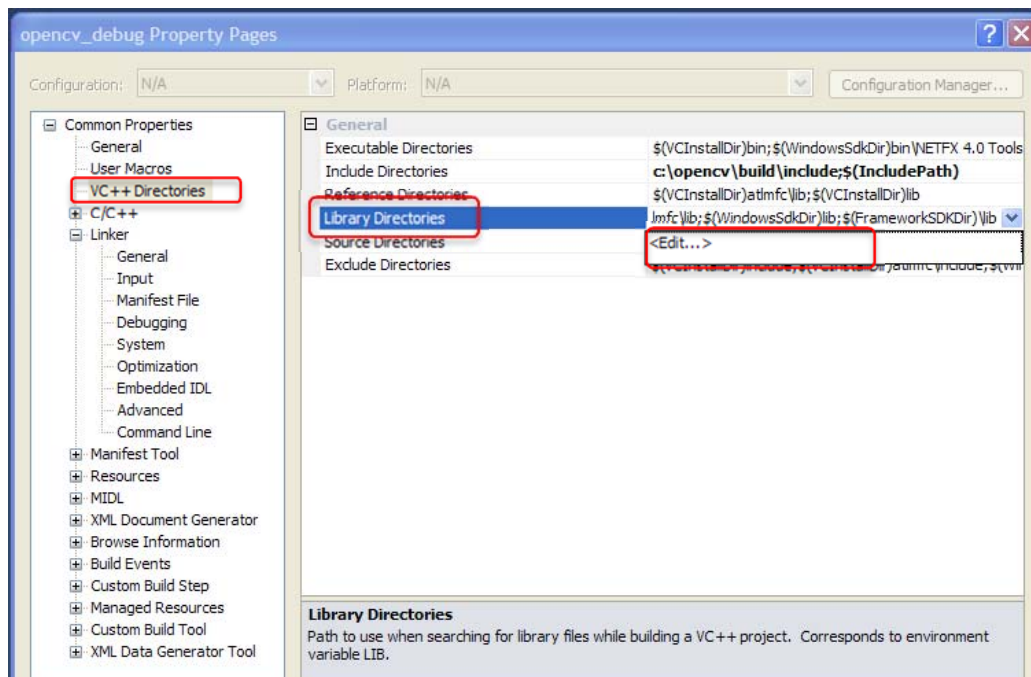
Opcionalmente añadiremos la carpeta: **c:\opencv\include**



a.3) Directorio de librerías:

- Editaremos la opción **VC++ Directories** -> **Library Directories**, pulsando sobre la flecha azul del lateral derecho nos muestra un menú con la opción **<Edit>** que pulsaremos y abrirá una ventana de edición.
 - Añadiremos el directorio específico, aunque en nuestro caso siempre vamos a trabajar con aplicaciones en consola **Win32** independientemente de la arquitectura de nuestro S.O.:
- Windows con arquitectura de 32 bits: **c:\opencv\build\x86\vc10\lib**
 - Windows con arquitectura de 64 bits: **c:\opencv\build\x64\vc10\lib**
 - Opcionalmente añadiremos la carpeta: **c:\opencv\lib**

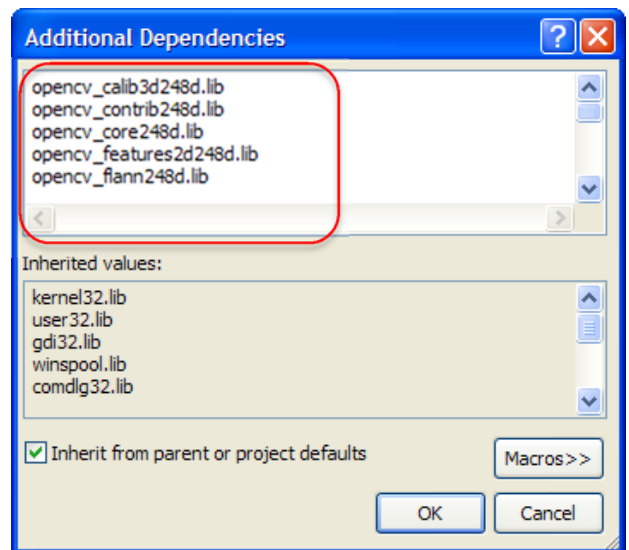
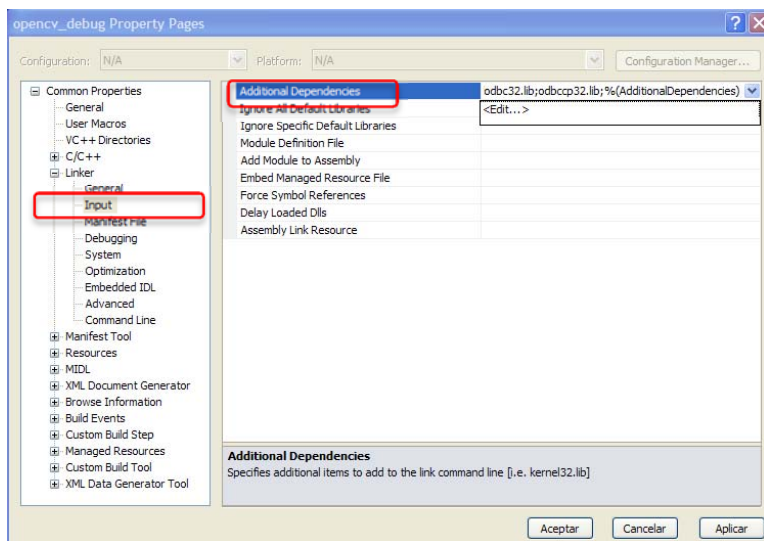
Nota: Optaremos por esta segunda opción sólo en caso de que compilemos una aplicación en un S.O Win64.



a.3) Librerías:

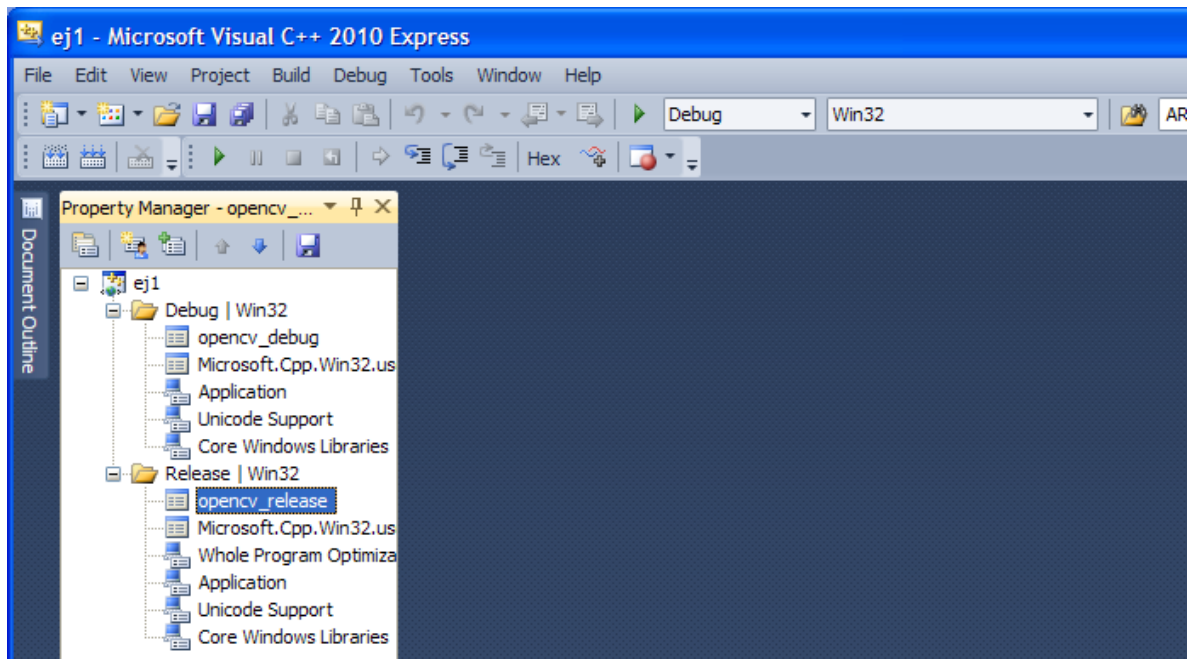
- Editaremos la opción **Linker -> Input -> Additional Dependencies**, pulsando sobre la flecha azul del lateral derecho nos muestra un menú con la opción **<Edit>** que pulsaremos y abrirá una ventana de edición. Añadiremos el listado de librerías de la **opencv**
- **Nota:** En la página web tienes disponible el listado para varias versiones de la biblioteca

```
opencv_calib3d2411d.lib  
opencv_contrib2411d.lib  
opencv_core2411d.lib  
opencv_features2d2411d.lib  
opencv_flann2411d.lib  
opencv_highgui2411d.lib  
opencv_imgproc2411d.lib  
opencv_legacy2411d.lib  
opencv_ml2411d.lib  
opencv_nonfree2411d.lib  
opencv_objdetect2411d.lib  
opencv_photo2411d.lib  
opencv_stitching2411d.lib  
opencv_superres2411d.lib  
opencv_ts2411d.lib  
opencv_video2411d.lib  
opencv_videostab2411d.lib
```



b) **Configuración modo Release:**

- Repetiremos los mismos pasos del apartado anterior pero sobre la etiqueta '**Release**'. El fichero de propiedades lo llamaremos '**opencv_release**' y en lugar de las versiones de depuración de las librerías de la *OpenCV*, deberemos incluir las librerías sin información de depuración:



```
opencv_calib3d2411.lib
opencv_contrib2411.lib
opencv_core2411.lib
opencv_features2d2411.lib
opencv_flann2411.lib
opencv_highgui2411.lib
opencv_imgproc2411.lib
opencv_legacy2411.lib
opencv_ml2411.lib
opencv_nonfree2411.lib
opencv_objdetect2411.lib
opencv_photo2411.lib
opencv_stitching2411.lib
opencv_superres2411.lib
opencv_ts2411.lib
opencv_video2411.lib
opencv_videostab2411.lib
```

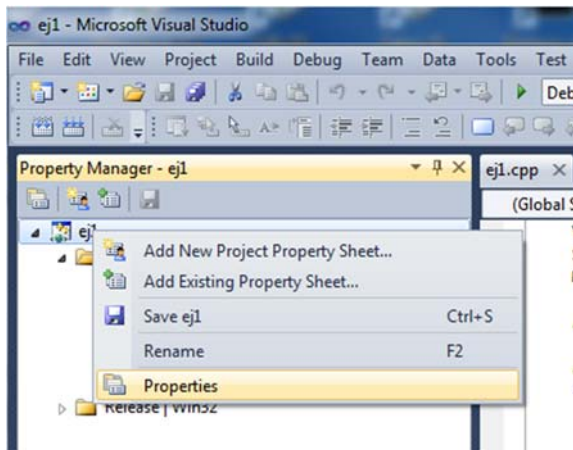
Al terminar este apartado tendremos en la carpeta de nuestro proyecto dos ficheros con toda la configuración que guardaremos para uso futuro:

opencv_debug.props
opencv_release.props

c) Configuración opcional

En el tutorial de instalación de la librería OpenCV vimos como configurar la variable de entorno (**PATH**) con la ubicación de las **DLLs** de la librería. Si estamos trabajando con diferentes versiones de la librería o diferentes compiladores, podemos realizar esta configuración de forma específica para nuestro proyecto.

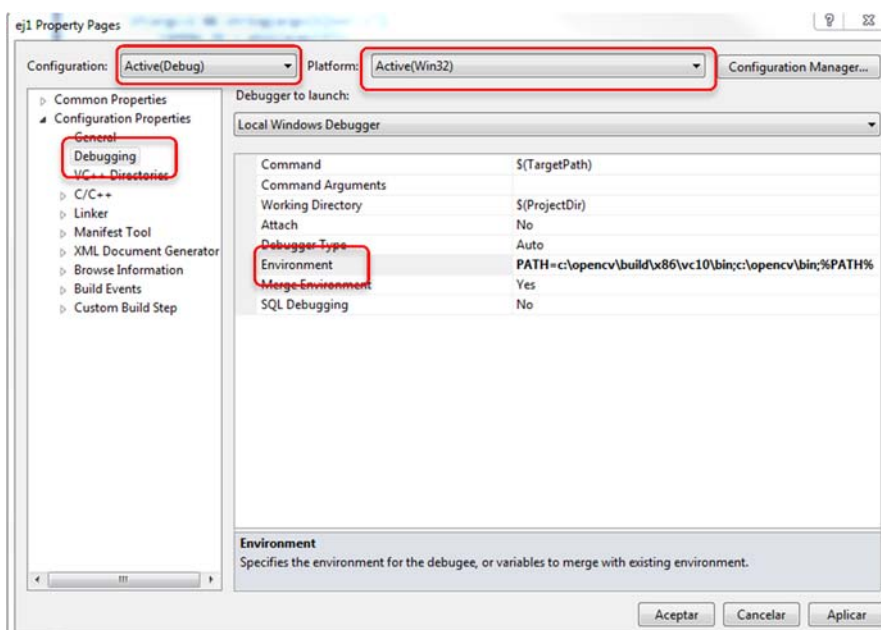
- Pulsaremos el botón derecho del ratón sobre nodo principal con el nombre del proyecto '**ej1**', desplegando el menú contextual en el que seleccionaremos '**Properties**'.



- Elegiremos la **Configuración /Plataforma (Debug/Win32)** que deseamos configurar. Marcaremos la opción '**Debugging**' y editaremos la opción '**Enviroment**'.

Para el compilador VC2010 y arquitectura **Win32** pondríamos:
PATH=c:\opencv\build\x86\vc10\bin;c:\opencv\bin;%PATH%

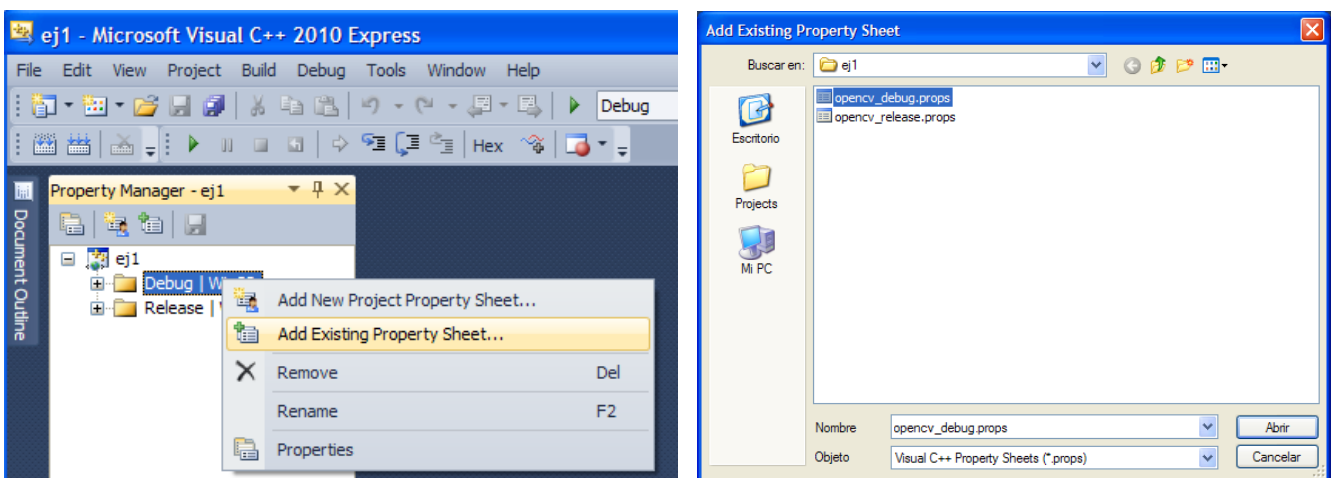
Para el compilador VC2010 y arquitectura **Win64** pondríamos:
PATH=c:\opencv\build\x64\vc10\bin;c:\opencv\bin\x64;%PATH%



4) Configurar el proyecto a partir de una página de propiedades previa:

En este caso reutilizaremos los ficheros de configuración creados en el proyecto anterior.

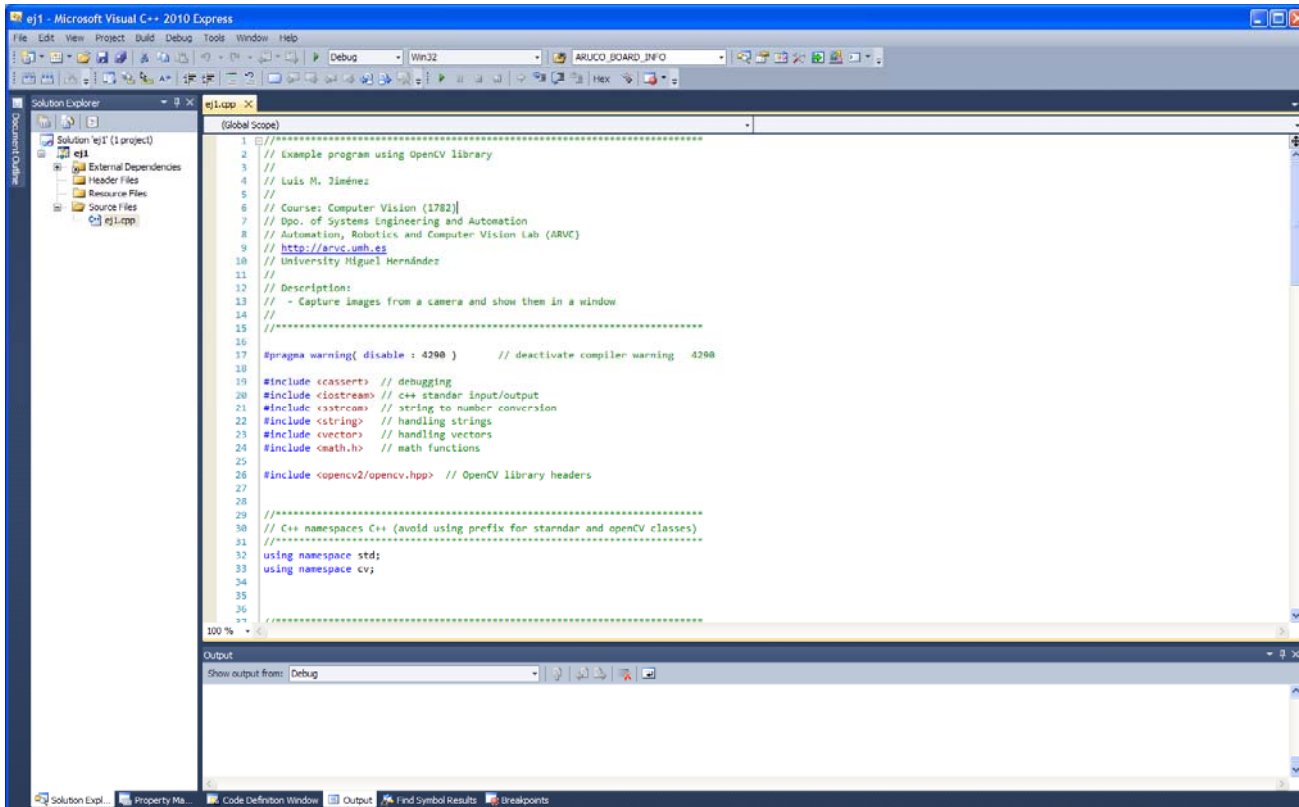
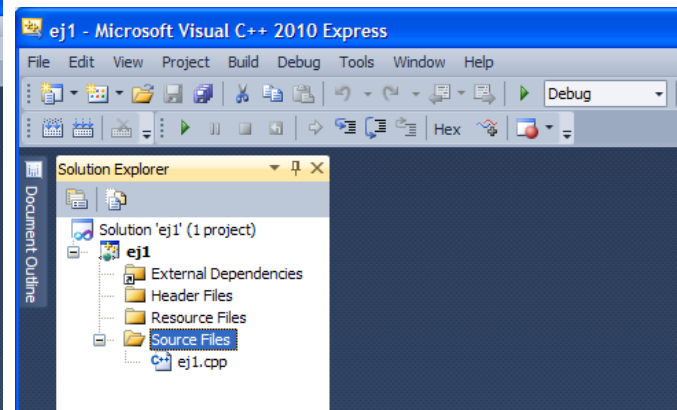
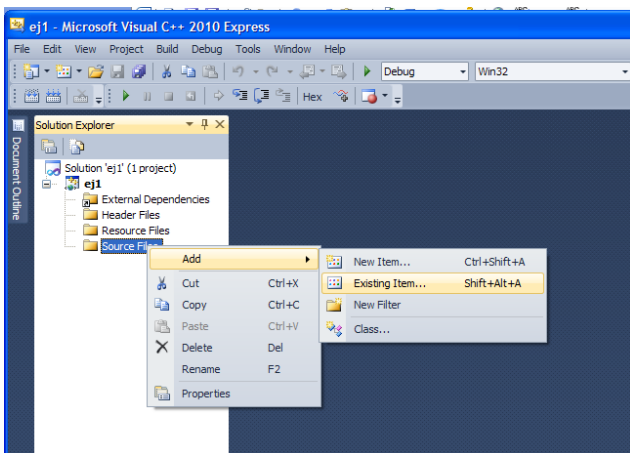
- Copiaremos los ficheros '*opencv_debug.props*' y '*opencv_release.props*', en la carpeta de nuestro proyecto.
- En la ventana '*Property Manager*' de nuestro proyecto, pulsando con el botón derecho del ratón sobre las pestañas '*Debug*' y '*Release*' añadiremos las hojas de propiedades correspondientes con la opción del menú '*Add Existing Property Sheet*'



- Realizaremos, si lo deseamos, la configuración opcional de las variables de entorno para depuración.

5) Añadir el código fuente de nuestro programa:

- Descargaremos el fichero de ejemplo '**ej1.cpp**' del blog de la asignatura. Lo copiaremos en la carpeta creada para nuestro proyecto.
- Activaremos la vista del '**Solution Explorer**'. Menú '**View-> Solution Explorer**'
- Sobre el nodo de la aplicación (**ej1**) desplegaremos las subcarpetas y pulsaremos el botón derecho del ratón sobre la carpeta '**Source Files**'. Seleccionaremos la opción '**Add/Existing Files**' y cargaremos el fichero de código **ej1.cpp**.



- A continuación compilaremos el programa (Menú '**Build/Build Solution**' o **F7**) y lo ejecutaremos (Menú **Debug/Start Debugging**' o **F5**).

