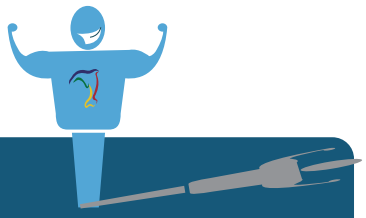
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

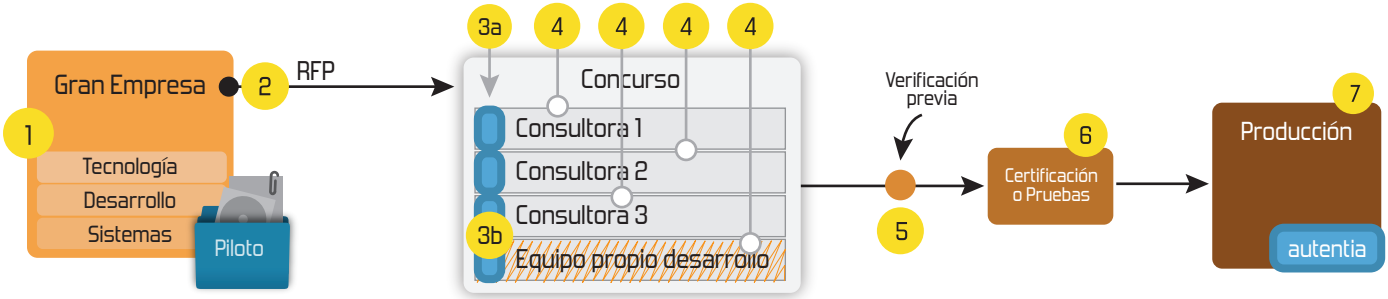
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

adictos altrabajo

Hosting Patrocinado por enREDados.com

Home | Quienes Somos | Empleo | Foros | Tutoriales | Servicios Gratuitos | Contacte



Tutorial desarrollado por:
@ Roberto Canales Mora 2003-2005.

Si crees que te he ayudado con este tutorial tambien **tu me puedes ayudar a mi..**

Es muy fácil, **mándame un correo con los datos de contacto de tu responsable de Formación y/o Recursos Humanos** y es posible que podamos ofrecer un curso interesante a tu organización (y por tanto a tí).
Mas detalles...

Contacta: rcanales@autentia.com.

Descargar este documento en formato PDF dcomyvc.pdf

Empresas diseño web
Especialistas en diseño y promoción Oferta diseño web + buscadores 120€

Novatech Campus Verano
2005 ¿te lo vas a perder? Formate con nosotros 91 395 28 75

Hospedaje Web desde 1,99€
ASP.NET, SQL, dominio, antivirus PHP, tráfico ilimitado, hasta 5 GB

Diseño paginas web
Creacion de paginas web, sencillo comodo y practico. Lo que necesita.

Anuncios Goooooogle

DCOM y Visual C++

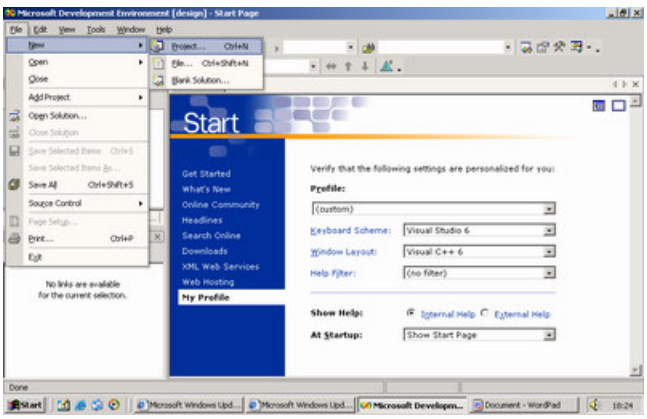
Sin duda, el entorno de trabajo en el que más horas he invertido en el pasado (miles de horas) ha sido Microsoft Visual C++. Simplemente me parece una obra maestra ....

De vez en cuando me toca volver a él para hacer alguna cosilla y voy a aprovechar parra compartir con vosotros como se hacen (y no se hacen) algunas cositas, hoy relacionadas con el desarrollo de clientes y servidores de automatización OLE.

Dando por echo que lo tenéis instalado y lo arrancamos. De una versión a otra cambian algunas cosas de aspecto pero los conceptos son los mismos.

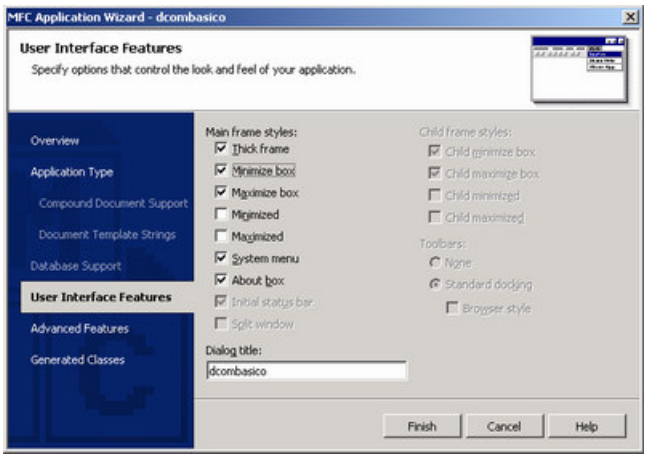
Creación del servidor de Automatización OLE

Creamos un nuevo proyecto

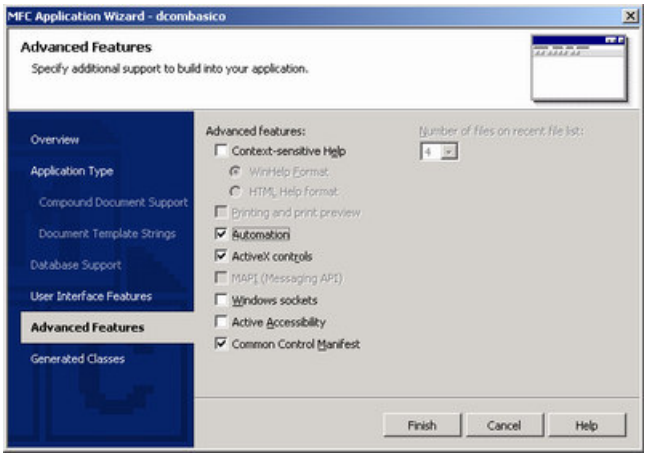


Elegimos una aplicación Visual C++ basada en un diálogo. El echo de que sea un diálogo solo es para que podamos ver que se arranca al ser invocado como un servidor de automatización y para que se registren todos los componentes en otra máquina con solo ejecutarlo (lo suyo sería crearlos como servidores ATL)

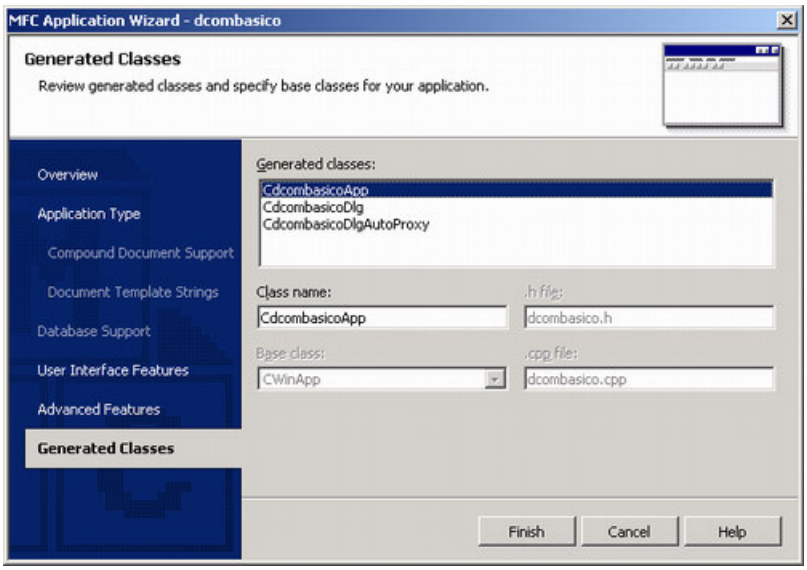
Elegimos las opciones de configuración básica.



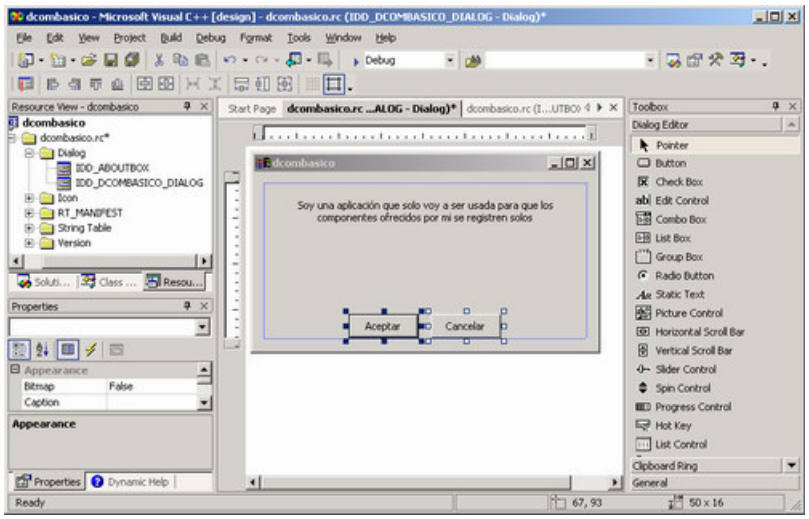
No se os olvide pinchar la opción de automatización



Revisamos los nombres de las clases que se van a generar



Vemos el aspecto de la aplicación y configuramos el diálogo con el editor de recursos

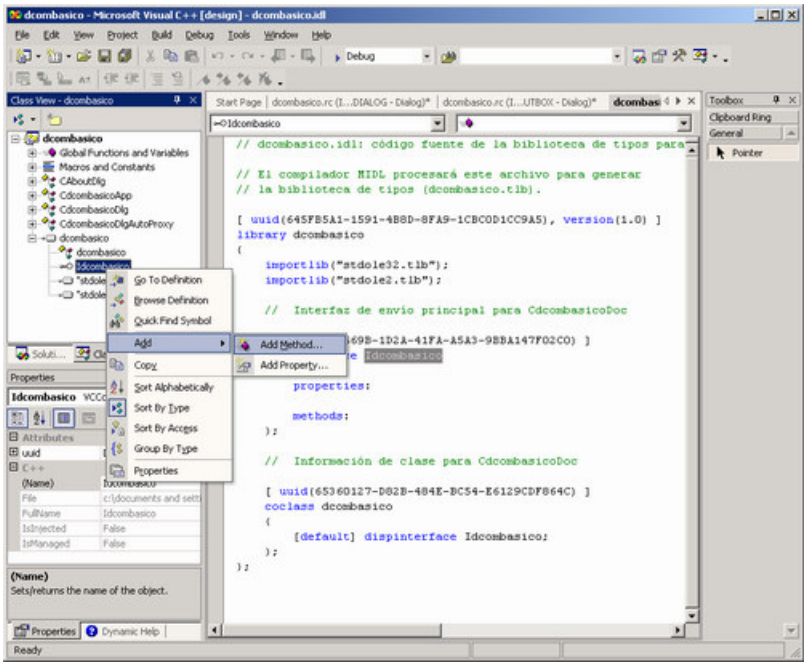


Debemos revisar los componentes creados, sobre todo el señalado (el interfaz) que en nuestro caso se llama **Idombasico**.

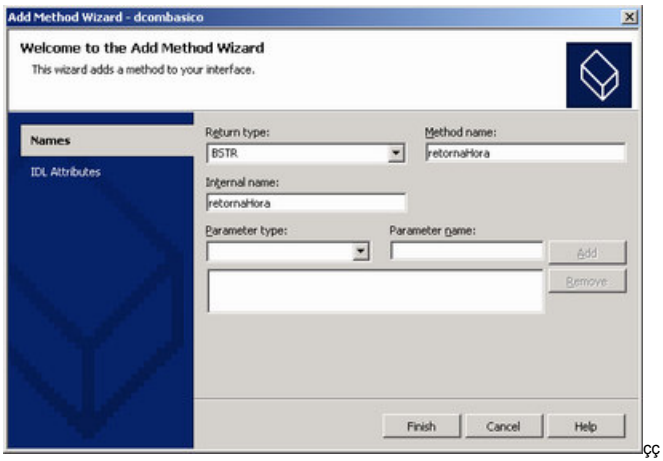
Nota:

- Lo importante de la tecnología de componentes es que existe la definición de un interfaz que define un servicio a ofrecer (en el entorno Microsoft se hace todo tan automático que gente sin los conceptos claros puede trabajar cómodamente y los problemas vienen más tarde).
- Un componente en sí es solamente una implementación de ese interfaz (y algunos otros más).
- Una vez definido un interfaz, este JAMÁS se debe cambiar.
- Es una mala práctica ir añadiendo funciones a un interfaz a medida que lo necesitamos en un componente que lo implementa.
- Una vez que una aplicación está en producción, nunca se debe cambiar el interfaz sino crear uno nuevo (que lo amplíe o modifique) y por técnicas de agregación y/o composición construir un nuevo componente

Añadimos a nuestro interfaz un método



Decimos que retorne una cadena de caracteres



Y vemos como ha quedado el interfaz

```
// dcombasico.idl: código fuente de la biblioteca de tipos para el archivo
dcombasico.exe

// El compilador MIDL procesará este archivo para generar
// la biblioteca de tipos (dcombasico.tlb).

[ uuid(645FB5A1-1591-4B8D-8FA9-1CBC0D1CC9A5), version(1.0) ]
library dcombasico
{
	importlib("stdole32.tlb");
	importlib("stdole2.tlb");

	//  Interfaz de envío principal para CdcombasicoDoc

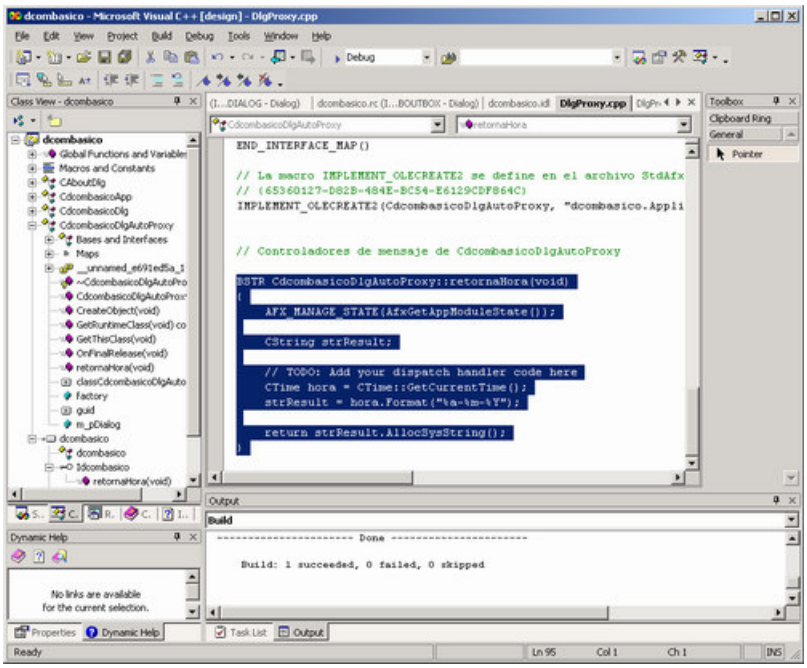
	[ uuid(DE8D669B-1D2A-41FA-A5A3-9BBA147F02C0) ]
	dispinterface Idcombasico
	{
		properties:

		methods:
		[id(1), helpstring("method retornaHora")] BSTR retornaHora(void);
	};

	//  Información de clase para CdcombasicoDoc

	[ uuid(65360127-D82B-484E-BC54-E6129CDF864C) ]
	coclass dcombasico
	{
		[default] dispinterface Idcombasico;
	};
};
```

El método se ha implementado en nuestra clase encargada de su implementación... que simplemente retorna la fecha ...




```
BSTR
CdcombasicoDlgAutoProxy::retornaHora
(void)
{
AFX_MANAGE_STATE
(AfxGetAppModuleState());

CString strResult;

// TODO: Add your dispatch handler
code here
CTime hora = CTime::GetCurrentTime();
strResult = hora.Format("%d/%m/%Y");

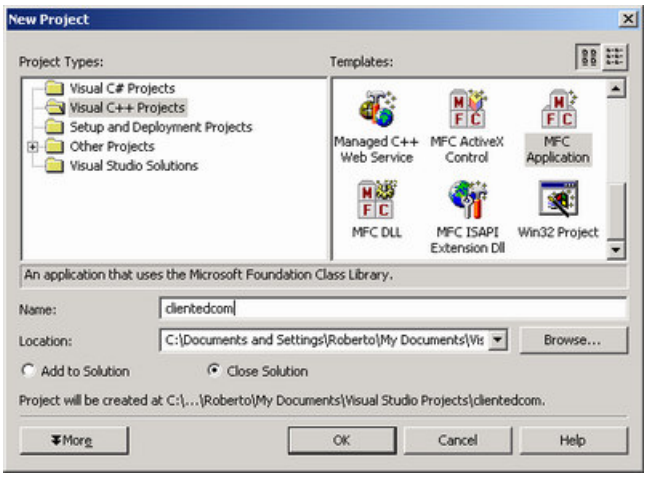
return strResult.AllocSysString();
}
```

Una parte de código que nos interesa ... el nombre necesario para crearlo por el cliente por nombre

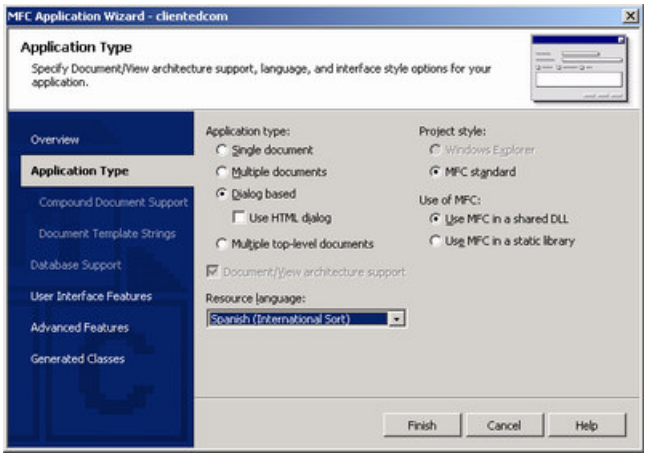
```
// La macro IMPLEMENT_OLECREATE2 se define en el archivo StdAfx.h de este
proyecto
// {65360127-D82B-484E-BC54-E6129CDF864C}
IMPLEMENT_OLECREATE2(CdcombasicoDlgAutoProxy, "dcombasico.Application",
0x65360127, 0xd82b, 0x484e, 0xbc, 0x54, 0xe6, 0x12, 0x9c, 0xdf, 0x86, 0x4c)
```

Creación del cliente

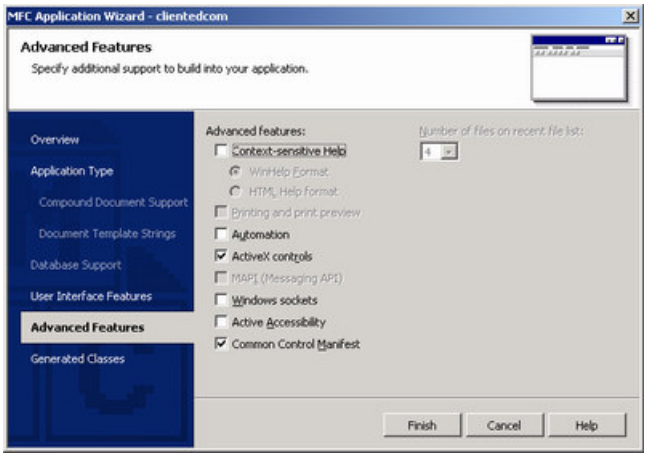
Elegimos una aplicación Visual C++ MFC



Igualmente basada en diálogo



Elegimos las opciones y no marcamos el soporte de automatización



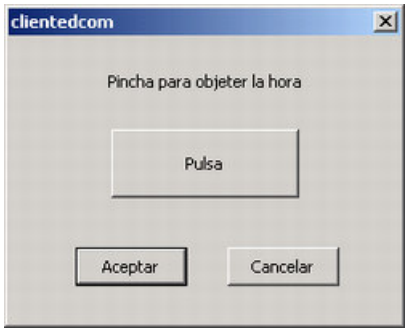
Pero para que funcione el ejemplo necesitamos añadir a mano estas líneas

```
BOOL CclientedcomApp::InitInstance()
{
        .....

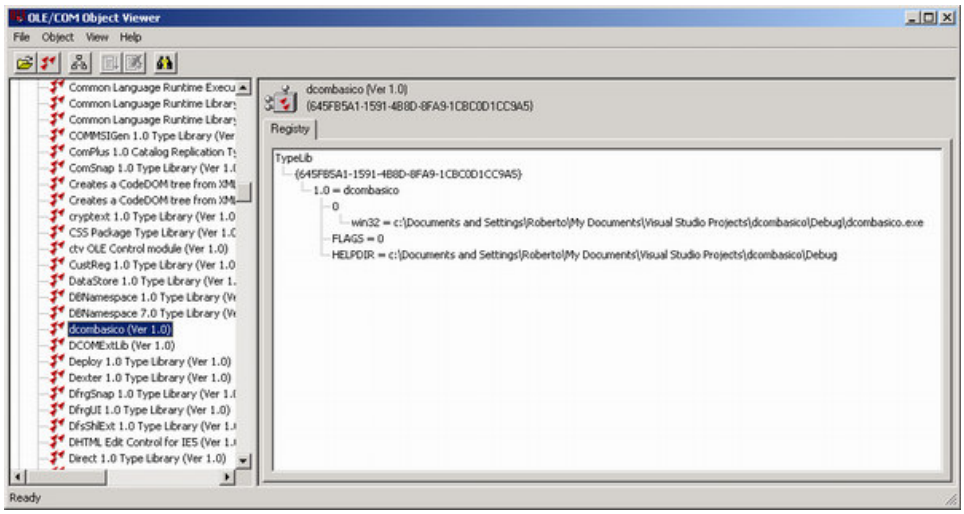
        // Inicializar bibliotecas OLE
        if (!AfxOleInit())
```

```
{
        AfxMessageBox("Error al incializar OLE");
        return FALSE;
}
```

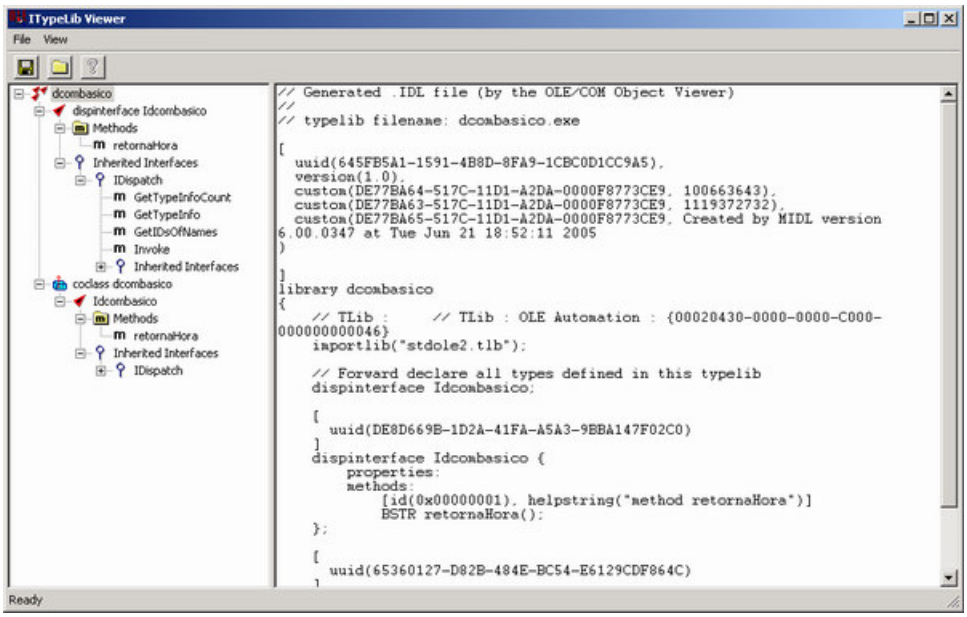
Creamos un diálogo básico



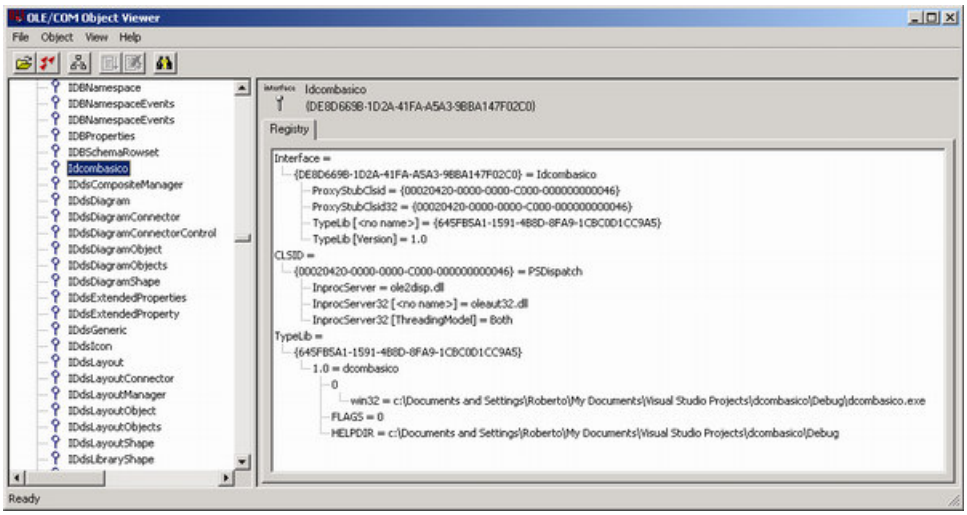
Y revisamos que nuestro interfaz y servidor que lo implementa están registrados en el sistema **ActiveX Control Test Container** (en el menú Tools)



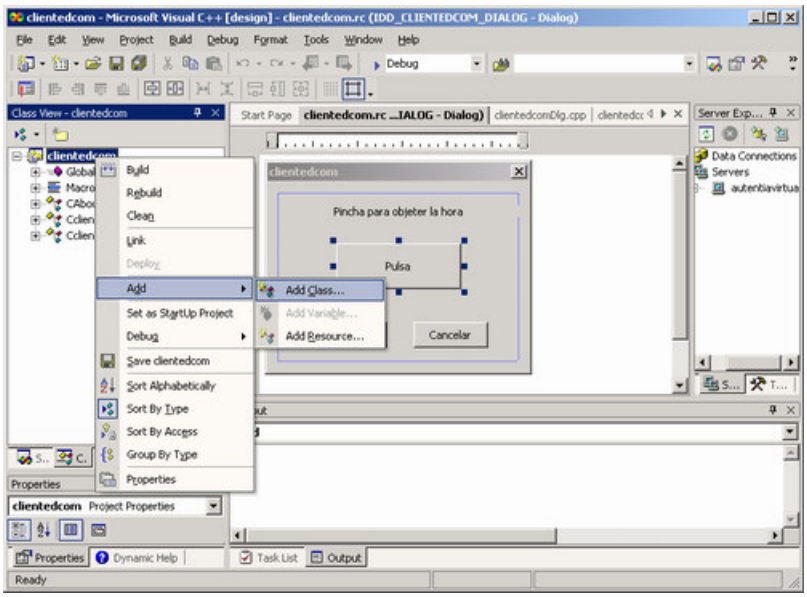
Vemos el detalle



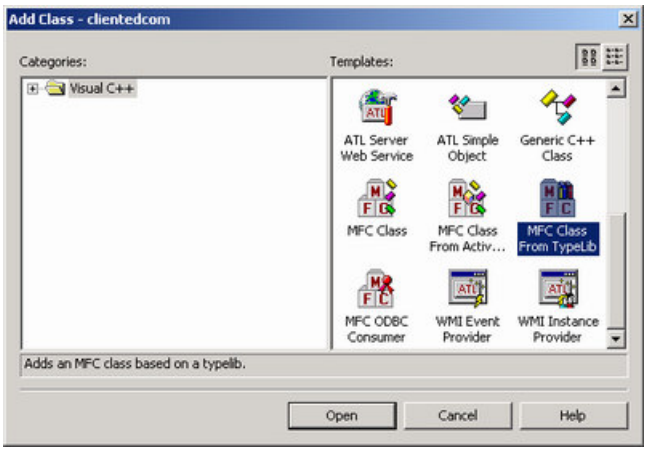
Y el interfaz en el registro



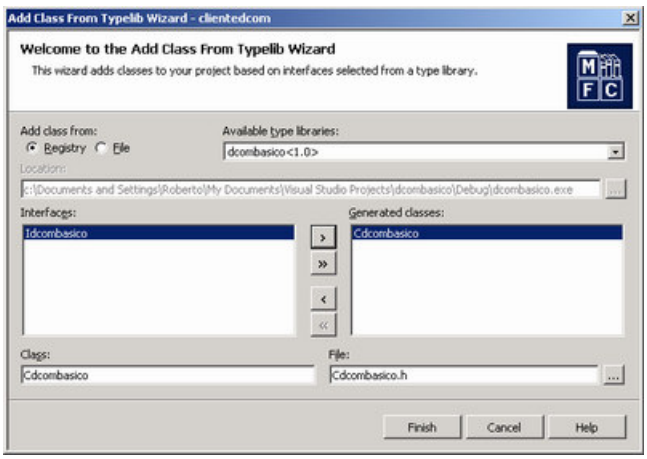
Ahora añadimos una clase nueva, pinchando el botón derecho del editor



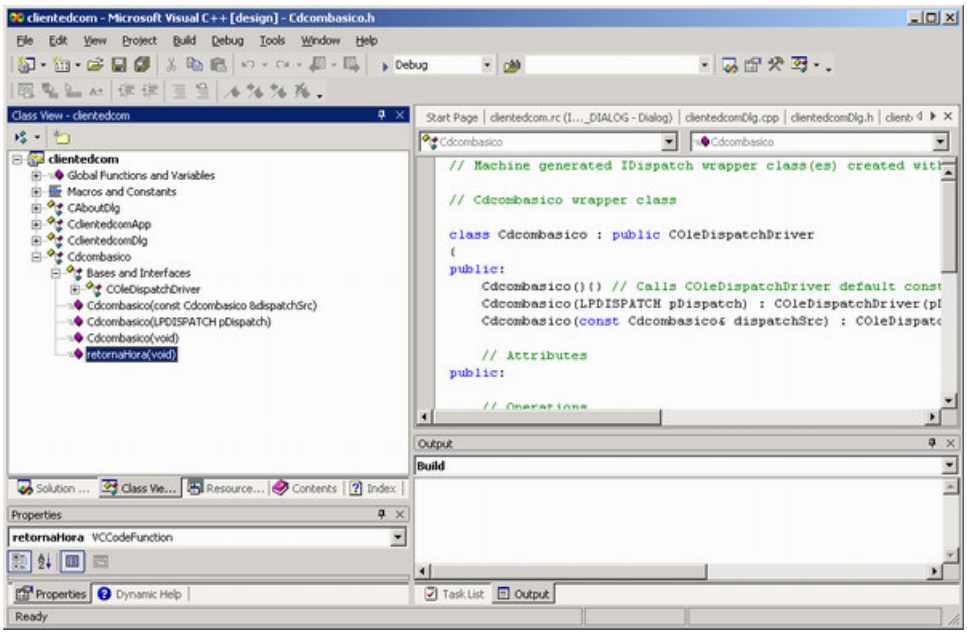
Creamos la clase desde un fichero descriptor



Elegimos el componente servidor



Y revisamos el código generado



Y añadimos el código al invocar el botón (no se os olvide incluir el .h):

#include "Cdcombasico.h"

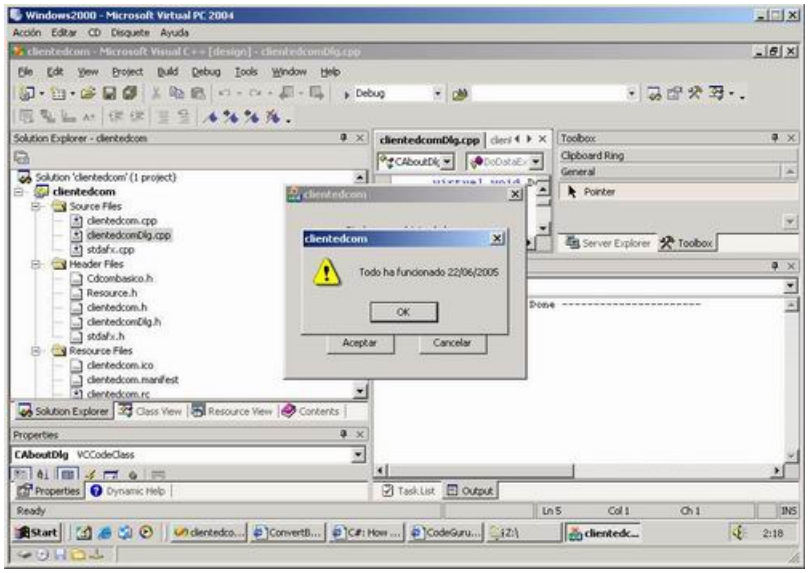
```
void CclienteligeroDlg::OnBnClickedButton1()
{
	// TODO: Add your control notification handler code here
	COleDispatchDriver disp;
	COleException *e = new COleException;

	try {
		// Create instance of Microsoft System Information Control
		// by using ProgID.
		if (disp.CreateDispatch("dcombasico.Application",
e))
		{
			Cdcombasico a = Cdcombasico(disp);
			CString x = a.retornaHora();
			AfxMessageBox("Todo ha funcionado " + x);

		}

		disp.DetachDispatch();
	}
	catch (COleDispatchException * e)
	{
		AfxMessageBox(e->m_strDescription);
		e->Delete();
	}
}
```

Y ejecutamos.



Veremos que todo funciona correctamente

Otros temas avanzados

Ojito con estos ejemplos.... la tecnología de componentes es algo serio y requiere formación y experiencia.

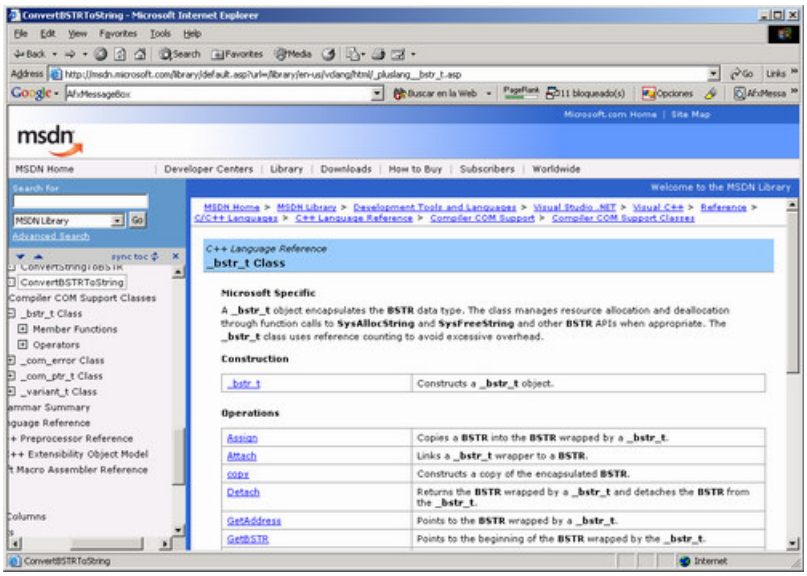
Si miráis un poquito el código, en el servidor de automatización OLE se crea una cadena (que se sube a la memoria compartida) que se consume en el cliente ¿y quién la limpia? Este código generado desde el asistente, ¿dejará lagunas de memoria .... no detectable por el depurador?

Os aconsejo que miréis este artículo

http://www.codeguru.com/forum/showthread.php?t=231156

y esta porción de código .....

```
BSTR bstr;
   SomeOLEFunction(bstr);
   _bstr_t tmp(bstr, FALSE); //wrap the BSTR
   CString cs(static_cast<const char *>(tmp)); //convert it
   AfxMessageBox(cs, MB_OK, 0); //test
   // when tmp goes out of scope it will free the BSTRs memory
```



Conclusiones

Los entornos de desarrollo son cada día más potentes e impresionantes pero esto no quita para que realicemos un estudio profundo de las tecnologías base en las que se apoya.

En la mayoría de las empresas, la gente utiliza los entornos de desarrollo de Microsoft por lo rápido que podemos ponernos en marcha.... los problemas vienen en ejecución y cuando queremos ampliar la funcionalidad haciendo cosas que ya no nos hacen los asistentes (Wizards) automáticamente (por fallas graves de conceptos esenciales).... pero bueno .... de algo tenemos que vivir los demás .....

Si desea contratar formación, consultoria o desarrollo de piezas a medida puede contactar con



Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	Descripción
Ejecución de entorno Unix en Windows	Os enseñamos como ejecutar simulteneamente un entorno Windows y Unix en el mismo equipo gracias a CygWin (herramienta gratuita)
Primeros programas OpenGL con Visual C++	Con este tutorial, como insertar código OpenGL en un programa Visual C++, creando un sistema planetario básico.
Novedades en Java 1.5	Ya está disponible la versión Beta del J2SDK 1.5. Os mostramos algunas de las nuevas características introducidas en el lenguaje Java: Clases genéricas, enumeraciones, bucles simplificados, etc.
Repositorio CVS en Windows	Os mostramos como montar un servidor para el control de versiones CVS en Windows asi como acceder a él a través de WinCVS
Como crear un Cron en Unix	Cuando desarrollamos aplicaciones, numerosas veces defemos programar tareas que se ejecuten periodicamente. Os ensañamos a hacerlo en Unix
Nombrar dispositivos USB en Debian	En este tutorial vamos a ver como podemos asignar nombres a los dispositivos USB en Debian, independientemente del nombre que le de el kernel.
Decompilar Java	Os mostramos como recuperar el fuente de vuestro código a partir de los ficheros compilados .class
Almacenamiento en Windows Pocket 2003	Cesar Crespo nos enseña como utilizar ObjectStore en nuestros programas Visual C++. El "Object Store" (ObS) en Windows Pocket 2003 cumple en muchos casos la misma función que el disco duro en un equipo de escritorio.
Framework desarrollo eclipse	Aquí os mostramos algunas de las características de Eclipse
Instalación de Oracle8i	Cesar Crespo nos enseña las particularidades de la instalación de Oracle8i

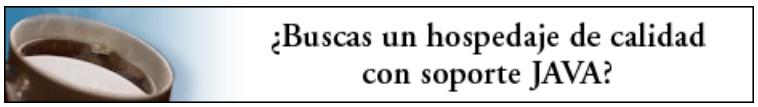
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE



www.AdictosAlTrabajo.com Opimizado 800X600