

Microsoft Visual J++ Versión 1.1

Guía de Evaluación

**Un sistema de programación Java para
la creación de applets y aplicaciones Java para
plataformas cruzadas e integración de Java con
las tecnologías ActiveX™.**

Marzo 1997

©1997 Microsoft Corporation. Todos los derechos reservados.

La información contenida en este documento representa la visión actual de Microsoft Corporation, sobre las cuestiones tratadas, hasta la fecha de la publicación. Debido a que Microsoft debe responder a las condiciones de cambio del mercado, dicha información no debe ser interpretada como un compromiso por parte de Microsoft, y del mismo modo, Microsoft no puede garantizar la exactitud de ninguna información presentada después de la fecha de publicación. Este documento existe sólo con fines informativos.

Microsoft no ofrece garantías, expresa o implícitamente, en este sumario.

Microsoft, ActiveX, DataTips, JScript, VBScript, Visual Basic, Visual C++, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, Windows y Windows NT son todas ellas marcas registradas o marcas de fábrica de Microsoft Corporation en Estados Unidos y/o otros países. Java es una marca de fabricante de Sun Microsystems, Inc. Otros nombres de productos y compañías mencionadas en este documento pueden ser marcas de fabricante de sus respectivos propietarios.

Contenidos

Introducción	1
La Estrategia de Microsoft para Java	2
La Máquina Virtual Microsoft para Java	2
Microsoft Visual J++	3
Requerimientos de Clientes	4
Capacidad para Plataformas Cruzadas	4
Extensibilidad	4
Productividad	4
Mínimo Coste y Riesgo para Conmutar Herramientas	5
Anteproyecto de una Herramienta de Desarrollo para Java	5
Objetivos de Diseño de Microsoft Visual J++	6
Herramienta de Desarrollo en Java para Plataformas Cruzadas	6
Extensibilidad de Java con ActiveX	6
Herramienta de Desarrollo Potente y Profesional	7
IDE Adaptable y Productivo	7
Ayuda Global y Amable con el Desarrollador	7
Herramienta de Desarrollo en Java para Plataformas Cruzadas	8
La Extensibilidad de Java Usando ActiveX	9
Reutilizando Componentes ActiveX	9
<i>Type Library Wizard para Java</i>	9
<i>El Database Wizard</i>	11
Creación de Componentes ActiveX reutilizables con Java	12
<i>El ActiveX Wizard</i>	12
Herramienta de Desarrollo Potente y Profesional	14
Compilador de Alta Velocidad y Sistema de Construcción Adaptable	14
Potentes Herramientas de Depuración	14
Características Avanzadas de Depuración en Visual J++	14
<i>Depuración en el navegador</i>	15
<i>Depuración de múltiples applets simultáneamente</i>	16
<i>Depuración multihilada</i>	16
<i>Depurar Componentes Activos</i>	16
<i>Examinar el Estado de un Programa</i>	16
<i>Ventana Variables</i>	17
<i>DataTips</i>	18
<i>Control del Flujo del Programa</i>	18

<i>Puntos de Ruptura</i>	19
<i>Depuración Paso a Paso</i>	20

IDE Adaptable y Productivo 22

Entorno de Desarrollo Integrado	22
<i>Integración del IDE de Developer Studio</i>	22
Entorno de Desarrollo Integrado Adaptable	23
<i>Escritura de Macros con Visual Basic, Edición para Scripts</i>	23
<i>Ventana Project Workspace</i>	24
<i>ClassView</i>	24
<i>FileView</i>	25
<i>InfoView</i>	25
AppWizards	25
<i>El Asistente para Applets de Java</i>	25
<i>Asistente para el área de trabajo de Java</i>	26
<i>Asistentes para Aplicaciones Add-in</i>	27
Sistema de Proyectos	27
Editor de Código Fuente	27
Editor de Recursos y Asistente de Recursos	28
<i>Editor de Forms y Menús</i>	28
<i>Asistente de Recursos</i>	28
<i>Editor de Imágenes</i>	29

Ayuda Global y Amable con el Desarrollador 30

Documentación y Otras Características Útiles para la Productividad	30
<i>La Tarjeta “Lea Esto Primero”</i>	30
<i>Aprenda Java Ahora</i>	30
<i>Póster con la Jerarquía de Clases</i>	31
<i>Asistente para Applets Java</i>	31
<i>Class View</i>	31
<i>Ayuda en F1</i>	32
<i>Ayuda sobre Errores de Construcción</i>	32
<i>Consejo del Día</i>	32
<i>Qué contiene la Documentación de los Libros en Línea</i>	33
<i>Prueba Controlada</i>	33
<i>Guía para Usuarios</i>	33
<i>Especificaciones del Lenguaje Java de Sun</i>	33
<i>Interface de Programación para Aplicaciones Java de Sun</i>	33
<i>Ejemplos, Ejemplos y más Ejemplos</i>	33
<i>Sitios Web Favoritos</i>	34
<i>Cómo se utiliza la Documentación de los Libros en Línea</i>	34
<i>F1</i>	34
<i>Búsqueda</i>	34
<i>Examinar el Historial del Sistema</i>	34
<i>Revisar los Resultados de la Última Interrogación</i>	34
<i>Definir Subconjuntos</i>	34
<i>Añadir Marcadores</i>	35

<i>Incluir Anotaciones</i>	35
<i>Ver el Código y la Ayuda Juntos</i>	35
<i>Incluye la Biblioteca Microsoft Developer Network en CD</i>	35
Servicios de Soporte	35

Introducción

Durante 1996, la tecnología Java™ ha comenzado a evolucionar desde ser tan sólo una forma fácil de añadir efectos de animaciones y sonido sencillos a las páginas Web, hasta convertirse en un lenguaje de programación más robusto. Java tiene el potencial de simplificar el desarrollo y la distribución de software debido a sus capacidades de multihilo, su recolector de basura y su ejecución sobre diversas plataformas.

Para aprovechar completamente las posibilidades de Java, los desarrolladores necesitan una herramienta y una Máquina Virtual Java (Java VM) que no sólo maximice la potencia de Java como lenguaje, sino que también obtenga beneficios de su software y experiencia, con el fin de minimizar la curva de aprendizaje. Las herramientas de desarrollo en Java deben ser más que un entorno gráfico. Necesitan ofrecer un entorno global para construir software permita Web e intranet, así como sistemas de misión crítica (sistemas que dependen de operaciones y decisiones tomadas día a día).

La Estrategia de Microsoft para Java

Microsoft está completamente comprometido con Java. La estrategia de Microsoft para Java se centra en satisfacer al desarrollador mediante:

- La oferta del mejor modo de desarrollar y ejecutar aplicaciones Java a través de la mejor herramienta de desarrollo, la mejor máquina virtual Java y las mejores bibliotecas de clases.
- La proposición a los desarrolladores de una opción que soporta completamente aplicaciones de plataformas cruzadas y proporciona las herramientas necesarias para crear aplicaciones maduras que puedan ejecutarse en la familia de sistemas operativos de Microsoft® Windows®.

De este modo, Microsoft equipa al desarrollador con un amplio espectro de opciones, desde el más extenso soporte para múltiples plataformas hasta la más dilatada experiencia posible del usuario sobre Windows. La solución global de Microsoft para Java consiste en la Máquina Virtual Microsoft (VM) para Java y Microsoft Visual J++.

La Máquina Virtual Microsoft para Java

Microsoft ha creado una de las mejores máquinas virtuales para Java existentes hasta la fecha. Además de ser 100% compatible con Java, es una de las máquinas virtuales para Java más rápidas, fiables y funcionales. Esta VM líder está licenciada de forma gratuita a Sun Microsystems para su distribución a otros vendedores. Microsoft pretende mantener la VM Microsoft en vanguardia de la tecnología Java.

Parte de los motivos por los que la VM Microsoft obtiene un alto rendimiento, es el compilador instantáneo (JIT). Además de proporcionar un compilador de alto rendimiento, la VM Microsoft permite la incorporación de cualquier compilador JIT. Utilizando esta característica, los desarrolladores pueden concebir compiladores JIT para mejorar el rendimiento de la aplicación. La VM también proporciona la capacidad de ampliar Java con tecnologías ActiveX. Esto simplifica la integración de componentes de software ya existentes en Java, así como crear componentes reutilizables usando el lenguaje Java. Es importante advertir que la extensibilidad ActiveX se ofrece sin la necesidad de alterar el lenguaje Java.

La VM Microsoft se incluye con Microsoft Internet Explorer 3.0 para Windows 3.1, Windows 95, Windows NT® y Apple Macintosh. Con el lanzamiento de la versión de Microsoft Internet Explorer 4.0, la VM pasa a formar parte del sistema operativo. Para Apple Macintosh, Microsoft está colaborando con Metrowerks para ofrecer la mayor calidad posible en la VM, que también incluirá soporte ActiveX. Microsoft también está trabajando para ofrecer una de las mejores VMs para UNIX con Internet Explorer 3.0.

Los principales vendedores de herramientas, tales como Borland, Symantec, Powersoft, y Metrowerks, han anunciado que se orientarán hacia la VM Microsoft para Windows. Microsoft está trabajando estrechamente con ellos, además de con Sun Microsystems para conseguir los objetivos de esta fructífera colaboración.

Microsoft Visual J++

Microsoft introdujo Visual J++ en el verano de 1996. Usando Visual J++, los desarrolladores pueden construir aplicaciones y applets para plataformas cruzadas con una elevada calidad y un alto rendimiento, de forma 100% compatible con Java. Con las herramientas de desarrollo de Microsoft Visual Studio™ 97, Visual J++ ofrece un entorno de desarrollo mejorado, para desarrollar con Java de forma más fácil y rápida. También permite la reutilización de la inversión en tecnologías ya existentes. Esto significa que un desarrollador puede crear aplicaciones con Java que pueden integrar controles y aplicaciones ya existentes, y crear componentes de software reutilizables, que puedan ser usados por otros lenguajes y herramientas, tales como Microsoft Visual Basic®, Microsoft Visual C++®, Delphi, Powerbuilder, Microsoft Excel y otros.

Cuando Visual J++ 1.0 fue lanzado, fue ampliamente pregonado como "...el estándar de hecho para los desarrolladores de Java sobre Windows" (*JavaWorld*, Agosto de 1996). Durante dos meses de publicación beta, se descargaron más de 150.000 copias de 14MB de archivos de programas de Visual J++.

Requerimientos de Clientes

Como uno de los principales vendedores de herramientas de desarrollo en el mundo, Microsoft está vivamente interesado en las necesidades de la gente que construye software. A través de la interacción con miles de desarrolladores profesionales que utilizan Java y compañías interesadas en Java, siguiendo las conversaciones de desarrolladores en foros tales como comp.lang.java, y a través de discusiones concretas con un grupo consistente en 50 desarrolladores líderes y escritores vinculados con el lenguaje Java, han surgido los siguientes requerimientos como respuesta a las necesidades recurrentes de los desarrolladores que usan Java.

Capacidad para Plataformas Cruzadas

“Escribir una vez y ejecutar donde se quiera” ha sido durante mucho tiempo el sueño de los desarrolladores. Un beneficio clave de Java es su potencial para proporcionar capacidad para plataformas cruzadas destinada a una amplia gama de soluciones de software. Por consiguiente, los desarrolladores interesados en utilizar Java insisten en que cualquier herramienta de desarrollo en Java cumpla al 100% con las especificaciones para Java de Sun Microsystems.

Extensibilidad

La habilidad para integrar aplicaciones **cliente/servidor** con las nuevas aplicaciones **Internet** y Java es fundamental. Ni los desarrolladores independientes ni las organizaciones desean abandonar el código ya existente y empezar el desarrollo desde cero. Por consiguiente, una herramienta Java necesita proporcionar la posibilidad de integrar, de forma sencilla, la nueva aplicación Java con aplicaciones existentes, componentes y otros lenguajes.

Productividad

Los desarrolladores tienen grandes esperanzas en el entorno de desarrollo que utilizan. Ellos quieren y esperan ser altamente productivos y además quieren ser capaces de adaptar sus entornos de desarrollo para que cada pulsación de tecla cuente. Últimamente, su objetivo es llegar al ciclo editar-construir-depurar tan rápidamente como sea posible. De este modo, el editor necesita adaptarse a las necesidades del desarrollador, el compilador necesita ser extremadamente rápido y

el entorno de depuración necesita estar completamente caracterizado y estrechamente integrado con el resto de la herramienta.

Mínimo Coste y Riesgo para Conmutar Herramientas

Las herramientas de desarrollo para Java necesitan minimizar el riesgo y los obstáculos asociados con la adaptación a un nuevo lenguaje y tecnología. Los desarrolladores quieren proteger las inversiones que han hecho en el código existente y formación, al trasladarlos a la era **Internet**. De igual modo, los desarrolladores quieren que sus nuevas inversiones en herramientas de desarrollo para Java, les proporcione valor de futuro al ofrecerles las nuevas destrezas y código de un lenguaje orientado a objeto que será fácilmente ampliable para reunir las necesidades futuras.

Anteproyecto de una Herramienta de Desarrollo para Java

En resumen, para ayudar a los desarrolladores a satisfacer los requerimientos de los clientes mientras compiten en el mercado de desarrollo de software, las herramientas de desarrollo en Java tienen la necesidad de ofrecer:

- 100% de compatibilidad con Java.
- La capacidad de realizar potentes aplicaciones que sean equivalentes a las que los desarrolladores profesionales y sus clientes están acostumbrados, para Microsoft Windows y Apple Macintosh, utilizando otros lenguajes de programación.
- Un entorno de desarrollo extremadamente productivo que ayuda a los programadores a conseguir pasar por el ciclo de edición-construcción-depuración tan rápido como sea posible.
- Una solución completa que permita a los programadores obtener una mayor velocidad en la programación con Java, incluyendo no sólo la propia herramienta, sino también extensas Ayudas, documentación impresa y en línea, tutoriales y programas de ejemplo documentados.

Objetivos de Diseño de Microsoft Visual J++

El sistema de desarrollo Microsoft Visual J++ está diseñado con las necesidades diarias de los desarrolladores que tienen en mente el uso de Java. Ofrece juntas las capacidades de información, productividad y programación que tanto los desarrolladores profesionales como los noveles buscan en una herramienta de desarrollo en Java. Para cumplir los requerimientos de los clientes esbozados en la sección anterior, Microsoft Visual J++ ha sido implementado para alcanzar estos objetivos de diseño.

Herramienta de Desarrollo en Java para Plataformas Cruzadas

Visual J++ está diseñado para ser la mejor herramienta de desarrollo disponible en Java destinada a la producción para plataformas cruzadas, 100% compatible con los applets y las aplicaciones Java que se ejecuten en cualquier plataforma que corra una Máquina Virtual Java o en cualquier navegador que permita Java. Puesto que el lenguaje Java está aún evolucionando, Visual J++ está diseñado para evolucionar con él. Visual J++ soporta todas las bibliotecas de clases para Java, incluso aquellas que no se incluyen con el producto, como la biblioteca de clases JDBC.

Extensibilidad de Java con ActiveX

Visual J++ está diseñado para que los desarrolladores tengan la opción de ampliar el código Java utilizando Componentes ActiveX ya existentes, construidos en otros lenguajes. Utilizando Visual J++, el desarrollador puede crear fácilmente Componentes ActiveX en Java que se integren sin costuras con aplicaciones existentes que soporten ActiveX. La extensibilidad de Java hace fácil la creación de aplicaciones en Java que rivalicen con las creadas en lenguajes tradicionales. Ampliando el lenguaje Java con la potencia de ActiveX, se obtiene acceso inmediato a capacidades adicionales sin efectuar cambios en el lenguaje Java. Dichas capacidades adicionales incluyen acceso a bases de datos, programación distribuida, multimedia de alto rendimiento e integración con los lenguajes de programación más populares.

Herramienta de Desarrollo Potente y Profesional

Otro objetivo de diseño para Visual J++ es el incremento de la productividad del desarrollador, reduciendo el tiempo consumido por el ciclo de edición-construcción-depuración. El depurador de Visual J++ ayuda a los desarrolladores a encontrar y solucionar bugs rápidamente, ofreciendo todas las características de depuración con potencia profesional que necesitan los desarrolladores expertos. El diseño del compilador de lenguaje Java de Visual J++ acorta dramáticamente los tiempos de construcción para equipar al desarrollador con la posibilidad de evaluar de forma inmediata las soluciones de errores y las mejoras en el código.

IDE Adaptable y Productivo

Visual J++ está diseñado para incrementar la productividad del desarrollador con un entorno de desarrollo integrado (IDE) adaptable. Los programadores de Java pueden utilizar ahora el entorno Developer Studio (el entorno de desarrollo en C++ preferido durante años) para programar en Java. El IDE Developer Studio es un entorno de desarrollo probado que ofrece abundantes características de integración para la administración de proyectos de desarrollo de software a gran escala. La facilidad de uso de asistentes en Developer Studio mejora la productividad del programador mediante la generación de código para applets y aplicaciones basada en la contestación a sencillas cuestiones sobre el diseño de la aplicación.

Ayuda Global y Amable con el Desarrollador

La reciente y rápida adopción del lenguaje de programación Java significa que muchos desarrolladores se enfrentan al desafío de dominar un nuevo lenguaje. Visual J++ ha sido diseñado para ofrecer a los desarrolladores acceso a la información adecuada, en el momento oportuno y en el formato apropiado para su fácil uso, a fin de que puedan superar los obstáculos iniciales, que encuentran muchos programadores cuando aprenden un nuevo lenguaje.

Herramienta de Desarrollo en Java para Plataformas Cruzadas

Microsoft Visual J++ supera todos los grupos de pruebas de compatibilidad con el Java de Sun Microsystems y lleva el logo de compatibilidad con Java en la caja. Usando el Asistente para Applets de Visual J++, el desarrollador puede crear rápidamente applets que son 100% para plataformas cruzadas y que se ejecutan en cada plataforma que soporte Java en cualquier navegador que permita Java, incluyendo Netscape Navigator.

La Extensibilidad de Java Usando ActiveX

Según se desarrollen las competencias de Java, será cada vez más necesario para las aplicaciones y componentes Java interactuar con sistemas existentes. ActiveX es un estándar binario abierto para la interoperación entre objetos independiente del lenguaje e independiente de la plataforma. Visual J++ ofrece a los desarrolladores la posibilidad tanto de reutilizar Componentes ActiveX existentes con sus aplicaciones Java, como la capacidad de crear Componentes ActiveX reutilizables con Java.

Reutilizando Componentes ActiveX

La habilidad para importar un Componente ActiveX como si estuviese en una clase Java significa que las capacidades de miles de Controles ActiveX y Componentes están disponibles para cualquier programador que utilice Java. Visual J++ 1.1 incluye dos asistentes que aceleran la reutilización de componentes existentes: Type Library Wizard y Database Wizard.

Type Library Wizard para Java

Type Library Wizard para Java crea interfaces Java para los Componentes ActiveX que los desarrolladores puedan cargar en su ordenador. Además de los interfaces Java, este asistente crea un sumario en un archivo de texto que proporciona una visión general completa del interface Java. Utilizando el sumario, el desarrollador puede ver rápidamente las funciones que están disponibles en el Componente ActiveX y analizar la sintaxis Java para dicho componente. Por ejemplo, este es un subconjunto del archivo de sumario para Microsoft Internet Explorer 3.0.

```
public class shdocvw/InternetExplorer extends java.lang.Object
{
}
public interface shdocvw/IWebBrowserApp extends com.ms.com.IUnknown
{
    public abstract boolean getFullScreen();
    public abstract void Stop();
    public abstract void Quit();
    public abstract boolean getTopLevelContainer();
    public abstract boolean getVisible();
    public abstract void putVisible(boolean);
    public abstract int getLeft();
    public abstract void putLeft(int);
    public abstract void GoBack();
    public abstract void Refresh();
    public abstract void Navigate(java.lang.String, com.ms.com.Variant,
```

```

        com.ms.com.Variant, com.ms.com.Variant, com.ms.com.Variant);
    public abstract java.lang.Object getParent();
    public abstract void putStatusBar(boolean);
    public abstract void putFullScreen(boolean);
    public abstract void GoForward();
    public abstract com.ms.com.Variant GetProperty(java.lang.String);
    public abstract int getTop();
    public abstract void PutProperty(java.lang.String,
com.ms.com.Variant);
    public abstract void GoHome();
    public abstract java.lang.String getType();
    public abstract java.lang.String getLocationURL();
    public abstract java.lang.String getStatusText();
    public abstract java.lang.Object getDocument();
}

```

Después de ejecutar el asistente, el desarrollador puede crear un applet Java o un componente que interactúe con Internet Explorer simplemente llamando sus métodos. En otras palabras, el desarrollador puede utilizarlo como si estuviera en otra clase Java. Más adelante, en esta sección aparece una pequeña lista de las aplicaciones que pueden ser utilizadas de esta forma. El código para automatizar Microsoft Internet Explorer y acceder directamente a su página principal es el siguiente:

```

import shdocvw.*;      //import Internet Explorer
class Browse3
{
    public static void main( String args[] )
    {
        IWebBrowserApp testBrowser = (IWebBrowserApp)new
InternetExplorer();
        testBrowser.getApplication();
        testBrowser.putVisible(true);
        testBrowser.GoHome();
    }
}

```

Utilizando ActiveX, los desarrolladores pueden acceder a avanzadas tecnologías multimedia tales como DirectX. Esto significa que los desarrolladores también pueden escribir juegos de alto rendimiento utilizando Java.

Las tecnologías ActiveX abarcan tanto a los equipos de escritorio como a los servidores, lo que significa que el desarrollador puede utilizar Microsoft Visual J++ para crear componentes del lado servidor de una forma tan sencilla como crea componentes de escritorio.

Además, usando Microsoft Visual J++, el desarrollador puede construir applets para automatizar cualquier aplicación que ofrezca su funcionalidad a través de interfaces con Modelo de Componentes Objeto (COM). Esta categoría de aplicaciones incluye un gran número de aplicaciones escritas para Windows y UNIX, así como gran parte de las escritas para ordenadores Apple Macintosh, incluyendo Adobe PageMaker, Microsoft Office, Caere Wordscan y Lotus Notes. Un ejemplo típico de lo que se puede alcanzar, es dirigir las funciones gráficas de Microsoft Excel para representar gráficamente los datos creados por un applet.

El Database Wizard

El Database Wizard de Visual J++ 1.1 facilita a las el uso del estándar de conectividad ODBC para acceder a la mayoría de los sistemas de administración de bases de datos, incluyendo Microsoft SQL Server™, DB2, Oracle y otros. El desarrollador puede conectar fácilmente bases de datos populares ODBC y SQL utilizando el Database Wizard, que es útil para aproximarse paso a paso a las conexiones a bases de datos que sigan la norma ODBC utilizando componentes RDO y DAO. En unos pocos pasos sencillos, el desarrollador puede conectar con una base de datos y mostrar los campos de la base de datos seleccionada en una página Web completa con botones de navegación.

Se puede ver el Database Wizard en la Figura 1.

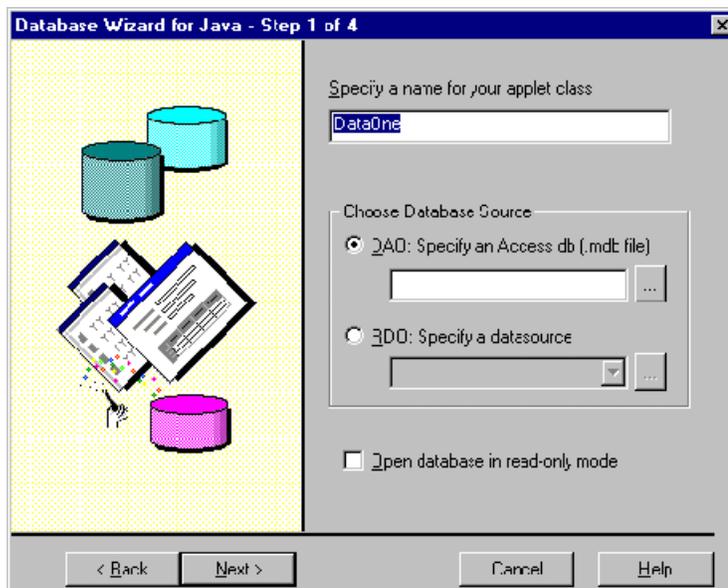


Figura 1. Database Wizard.

Creación de Componentes ActiveX reutilizables con Java

El desarrollador puede ampliar los usos de una aplicación Java proporcionándole funcionalidad en forma de Componentes ActiveX, que estén basados en componentes COM. Con esta capacidad, una aplicación Java puede ser reutilizada por cualquier contenedor ActiveX. Visual J++ 1.1 contiene un nuevo ActiveX Wizard que mejora la facilidad con la que los Componentes ActiveX pueden ser creados a partir de las clases Java normales.

El ActiveX Wizard

Java es uno de los mejores lenguajes para la creación de Componentes ActiveX. La especificación de interface para Java coincide casi exactamente con las especificaciones del interface abierto COM de ActiveX. Con Visual J++ 1.1, el ActiveX Wizard simplifica la creación de Componentes ActiveX a partir de cualquier clase Java. En particular, el asistente elimina el laborioso proceso de creación de un archivo con el Lenguaje de Definición de Interface (IDL), y su compilación dentro de una biblioteca de Tipo que puede ser usada por otros programas. También crea un Identificador Globalmente Único (GUID) y registra la biblioteca de Tipo en el registro del sistema.

Se puede ver el ActiveX Wizard en la Figura 2.



Figura 2. ActiveX Wizard.

Herramienta de Desarrollo Potente y Profesional

Visual J++ está diseñado para incrementar la productividad del programador al proporcionarle un compilador de código fuente Java que ofrece una velocidad récord, y un potente entorno de depuración.

Compilador de Alta Velocidad y Sistema de Construcción Adaptable

El compilador y el sistema de construcción de Microsoft Visual J++ son 100% compatibles con Java y cumplen con todas las pruebas de conformidad con Sun necesarias con el fin de crear código Java para plataformas cruzadas. El compilador de código fuente en Java acorta drásticamente el tiempo de construcción, ofreciendo una velocidad de compilación sin precedentes de 10.000 líneas por segundo.

El desarrollador puede modificar y adaptar las opciones de construcción del proyecto. Los cambios se pueden hacer para configurar diferentes objetivos, tales como Depuración y Detalle, además de archivos individuales dentro de cada configuración. Los desarrolladores también pueden crear sus propias configuraciones o especificar si quieren probar el applet o la aplicación utilizando un navegador o usando una versión run-time de Java autónoma.

Potentes Herramientas de Depuración

Hasta la fecha, una de las características claramente subdesarrolladas en las herramientas de desarrollo en Java, ha sido el soporte de depuración. El completo depurador de Visual J++ ofrece tecnología de depuración para Java que están a la altura de otros lenguajes, tales como Microsoft Visual Basic y Visual C++. Visual J++ contiene todas las características estándares de depuración, tales como la selección de puntos de ruptura en líneas de código fuente. Además, Visual J++ ofrece características de depuración innovadoras, tales como la capacidad de depuración dentro de un navegador Web, la depuración de múltiples applets simultáneamente, la depuración dentro de bytecodes, la selección de puntos de ruptura dentro de bytecodes y el cambio dinámico del valor de una variable durante la depuración

Características Avanzadas de Depuración en Visual J++

Con Visual J++, el desarrollador puede obtener ventajas de las características avanzadas de depuración, no ofrecidas en otros entornos de desarrollo en Java.

Algunos ejemplos son: depuración de applets en el navegador, depuración de múltiples applets simultáneamente en la misma página Web, depuración a nivel de bytecode y depuración de Componentes ActiveX y objetos COM escritos en Java.

Depuración en el navegador

Utilizando Visual J++, el desarrollador puede depurar un applet en una página Web desde dentro del navegador Microsoft Internet Explorer 3.0. Esto significa que el desarrollador puede probar la forma en la que el applet interactúa con otros objetos en la página Web. Los desarrolladores también pueden invocar métodos del applet desde otros objetos de la página Web y observar la respuesta del applet. Cuando se depura un applet de navegador sin un navegador, los desarrolladores no pueden probar muchas de las características del applet.

Se puede ver un ejemplo de la depuración de un applet en un navegador en la Figura 3.

Figura 3. Depuración de Múltiples Applets usando el Navegador Microsoft Explorer 3.0.

Depuración de múltiples applets simultáneamente

Usando Visual J++, el desarrollador puede depurar simultáneamente múltiples applets que se ejecutan en la misma página Web, como es el caso de una página con múltiples marcos (frames), cada uno de ellos con un applet Java que se comunica con los demás.

Depuración multihilada

Empleando Visual J++, el desarrollador puede depurar applets y aplicaciones Java multihiladas. Por medio de la caja de diálogo Thread Manager, el desarrollador puede cambiarse de un hilo a otro, parar y a continuación lanzar la ejecución de cualquier hilo.

Depurar Componentes Activos

Usando Visual J++, el desarrollador puede depurar Componentes ActiveX escritos en el lenguaje Java.

Examinar el Estado de un Programa

Utilizando las ventanas Call Stack y Watch, el desarrollador puede examinar el estado de la aplicación que está depurando y controlar su flujo. Esta funcionalidad básica también está disponible cuando se usan otras herramientas de Desarrollo en Java. El depurador de Visual J++ también ofrece características únicas como información en Ventanas de Variables y DataTips™ que ponen la información crítica al alcance de los desarrolladores, ayudando así a estimular la productividad.

- **Ventana de pila de llamadas.** El desarrollador utiliza la ventana Call Stack para ver la pila de llamadas a métodos que está en proceso en cualquier momento. Usando la ventana Call Stack, los desarrolladores pueden ver también los parámetros y los valores pasados a cualquier otro método. Los desarrolladores pueden establecer puntos de ruptura en la ventana Call Stack y utilizan la posición actual del cursor como un punto de ruptura temporal.

Se puede ver la ventana Call Stack en la Figura 4.

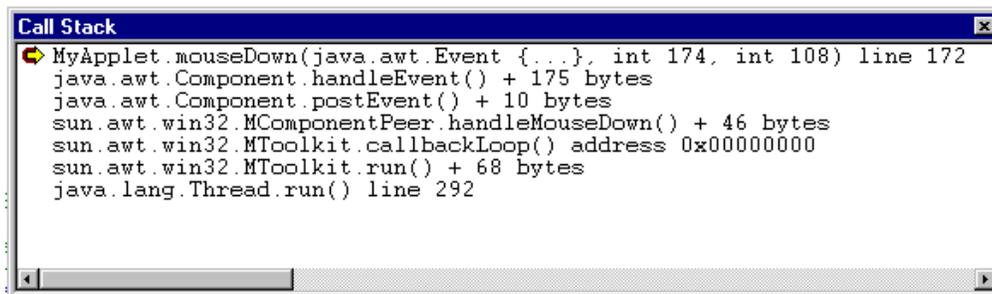


Figura 4. La ventana Call Stack.

- **Ventana de supervisión.** Los desarrolladores pueden utilizar la ventana Watch para monitorizar variables de interés. Se puede arrastrar una variable desde la ventana Source a la ventana Watch. Se pueden establecer grupos de variables a monitorizar en diferentes momentos, establecer los valores de las variables y de este modo, alterar el flujo del programa. Los desarrolladores pueden usar la vista en árbol de la jerarquía en Visual J++ para examinar los objetos de las clases y sus valores. Se puede usar la jerarquía además de para ver las variables de una clase derivada, para ver también las variables de todas sus clases base.

Ventana Variables

La ventana Variables es única en Visual J++. Utilizándola, los desarrolladores pueden ver los valores de las variables que les interesen. La ventana Variables contiene tres paneles: **Locals**, **Auto** y **This**.

- **Locals** visualiza la información de todas las variables del ámbito en un momento dado.
- **Auto** visualiza expresiones de interés en las líneas de código actuales y anteriores.
- **This** visualiza la información de todos los parámetros del objeto de la clase actual.

La ventana Variables también dispone de la misma característica de vista en árbol de la jerarquía de clases que la ventana Watch. Es importante resaltar que las variables que hayan cambiado desde la última ruptura en el código, aparecen destacadas en la ventana Variables. De este modo, el desarrollador puede ver si ha cambiado algo en el programa.

DataTips

En Visual J++ se dispone de DataTips de información. El desarrollador los utiliza simplemente apuntando con el cursor a una variable o marcando una expresión en la ventana Source. Sus valores aparecen en el formato FastTips habitual. Se pueden ver los DataTips en acción en la Figura 5.

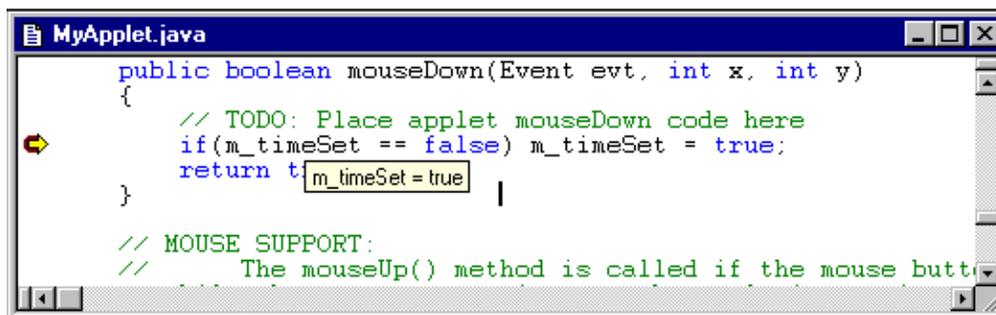


Figura 5. Los DataTips en acción.

Control del Flujo del Programa

El control de flujo del programa incluye la posibilidad de seleccionar puntos de ruptura, de modo que el desarrollador pueda moverse entre el código y parar la

ejecución del programa a voluntad. Se puede ver el entorno de desarrollo integrado (IDE) de Developer Studio en el modo de punto de ruptura en la Figura 6.

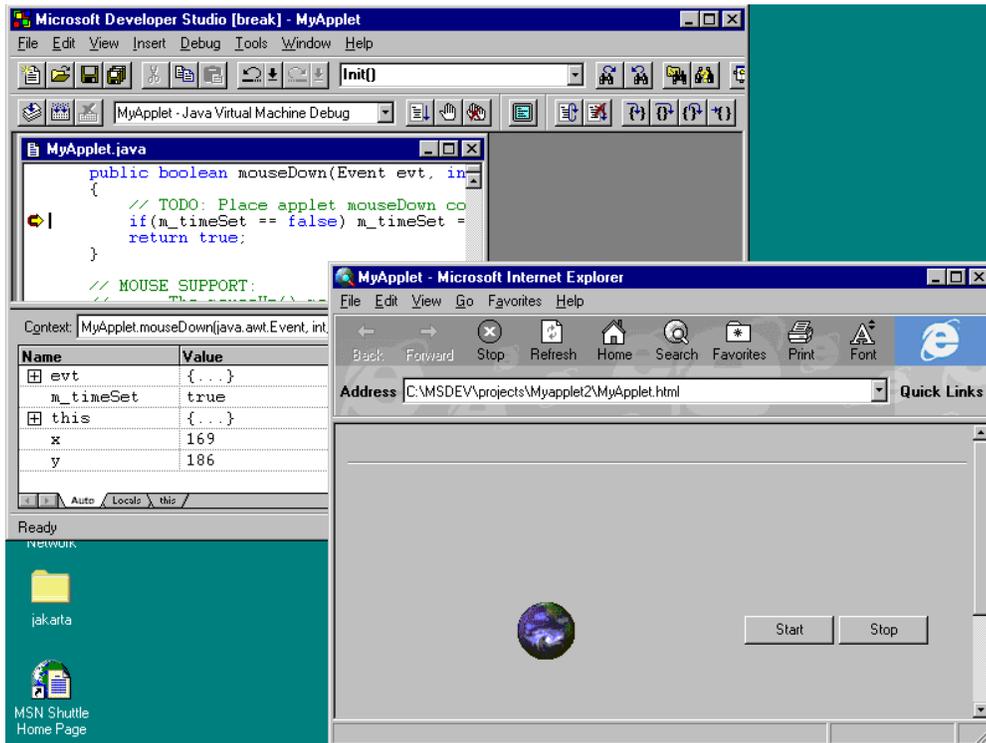


Figura 6. El interface de Developer Studio en modo de punto de ruptura.

Puntos de Ruptura

Cuando se escribe código en Java, sin tener en cuenta la herramienta, el desarrollador puede establecer puntos de ruptura en las líneas de código fuente.

Además, cuando se usa Visual J++ los desarrolladores también pueden:

- Establecer puntos de ruptura en métodos.
- Establecer puntos de ruptura en bytecodes. El desarrollador puede usar esta técnica durante la depuración de programas que incluyen otro código en Java, que no es código fuente.
- Establecer puntos de ruptura en bytecodes que no está directamente mapeado a líneas de código fuente. El desarrollador puede utilizar esta técnica cuando trata con múltiples sentencias en la misma línea de código fuente, por ejemplo, `int j=1; int k=2; int l=3.`

- Establecer puntos de ruptura para las herramientas de depuración con el fin de romper únicamente bajo ciertas condiciones, como por ejemplo si una variable alcanza un valor específico. El desarrollador puede utilizar esta técnica para pasar a través de un bucle en el código, y comprobar rápidamente que ocurre, solamente cuando sucede un evento de interés.
- Establecer un punto de ruptura mientras se está ejecutando una aplicación.
- Parar la aplicación asincrónicamente, sin importar qué paso se estaba ejecutando en ese momento. Los desarrolladores pueden utilizar esto cuando una aplicación entre en un bucle sin fin.
- Establecer puntos de ruptura en la mayoría de las ventanas de depuración, incluyendo la ventana Call Stack, la ventana Source, la ventana Disassembly, la barra de tareas y en el Visor de Clases.

Los desarrolladores pueden utilizar todas estas posibilidades dentro de Visual J++ con aplicaciones multihilo y applets escritos con Java.

Depuración Paso a Paso

Como en otras herramientas Java, también se dispone de esta funcionalidad básica.

El desarrollador puede:

- Saltar a la siguiente línea de código fuente.
- Pasar sobre la llamada a un método sin ejecutarlo paso a paso.
- “Ejecutar hasta el cursor”, o usar la posición actual del cursor como un punto de ruptura temporal.

Además de lo básico, con Visual J++ los desarrolladores también pueden:

- Saltar fuera de las llamadas a métodos. El desarrollador puede utilizar esta opción cuando, habiendo entrado en el método, se da cuenta que el error no está en ese método.
- Entrar en el código desensamblado. Los desarrolladores pueden utilizar esta opción para recorrer el código instrucción a instrucción, o depurar el código cuando no se dispone de código fuente.

- Ejecutar hasta el cursor en la ventana Call Stack. El desarrollador puede utilizar esto para volver a un método concreto con el fin de inspeccionarlo.

Igual que antes, los desarrolladores pueden utilizar todas estas posibilidades con aplicaciones multihilo y applets creados en Java.

IDE Adaptable y Productivo

Visual J++ permite a los desarrolladores escribir más código en menos tiempo, ya que les proporciona un Entorno de Desarrollo Integrado (IDE) a medida.

Entorno de Desarrollo Integrado

El desarrollador que utilizan Microsoft Visual J++ obtiene ventajas del fácil uso y la productividad del IDE de Developer Studio cuando construyen código Java. El IDE de Developer Studio es un entorno de desarrollo reconocido como líder que comparten Microsoft Visual C++, Microsoft Visual InterDev y el sistema de control de versiones Microsoft Visual SourceSafe. La Biblioteca Microsoft Developer Network (MSDN) en CD se puede integrar, como referencia, en el IDE de Developer Studio. Por consiguiente, la mayoría de los desarrolladores se sentirán automáticamente confortables utilizando Visual J++.

Integración del IDE de Developer Studio

Microsoft Visual Studio 97 incluye Microsoft Visual C++, Microsoft Visual Basic, Microsoft Visual InterDev, la Biblioteca Microsoft Developer Network en CD y Microsoft Visual J++. Utilizándolo, el programador puede desarrollar código en Java mientras usa otras potentes herramientas en el mismo entorno. Por ejemplo, los desarrolladores pueden añadir control de versiones con sistemas de control de versiones habituales, tales como PVCS o Microsoft Visual SourceSafe, que serán estrechamente integrados. Además, los desarrolladores pueden integrar la Biblioteca Microsoft Developer Network en CD, que contiene la documentación del producto, directamente dentro del IDE.

La integración con Visual C++ y Visual InterDev dentro del mismo entorno significa que los desarrolladores pueden construir múltiples proyectos en la misma área de trabajo, utilizando cada uno de ellos un lenguaje diferente. Esto es útil para proyectos donde el código del lado cliente se escribe en Java y el código del lado servidor se escribe en un lenguaje diferente; o cuando el proyecto de desarrollo contiene dos programas de sintaxis de código que se comunican a través de interfaces COM.

Entorno de Desarrollo Integrado Adaptable

Adaptando el entorno Visual J++ completo, el desarrollador puede ser mucho más productivo ajustándolo a sus preferencias personales. Los desarrolladores pueden adaptar cualquier ventana del entorno de desarrollo integrado (IDE). Pueden elegir **Hide** en cualquier menú de contexto para ocultar una ventana y liberar así espacio adicional de pantalla, mientras se está editando código, por ejemplo, se puede hacer clic con el botón derecho del ratón sobre la ventana Output que tiene un uso poco frecuente, para cerrarla. Los desarrolladores también pueden ocultar el área de trabajo al hacer clic en **Hide** en la barra de herramientas Workspace. Para aquellos desarrolladores que necesiten todos los píxeles de edición, Visual J++ dispone del modo de pantalla completa. Todas las ventanas son totalmente acoplables y ofrecen menús de contexto para facilitar su adecuación.

De igual modo, todas las barras de herramientas del IDE son adaptables. Los desarrolladores pueden elegir cualquiera de las ocho barras de herramientas estándares, todas ellas acoplables y adaptables. El desarrollador puede añadir botones a una barra de herramientas estándar o crear una barra de herramientas a medida para definir un estilo de trabajo personal.

Los desarrolladores pueden añadir sus herramientas favoritas de entre las herramientas add-on en el menú **Tools**. Una vez añadidas, pueden pasar los parámetros esenciales de las herramientas desde otros proyectos. El IDE de Developer Studio se ha convertido en el entorno de desarrollo preferido para la programación en Java, ya que se puede adaptar fácilmente el menú **Tools** para añadirle el compilador Java y Appletviewer. Los desarrolladores pueden añadir al IDE de Developer Studio combinaciones de teclas rápidas para cada función particular, incluyendo combinaciones de teclas rápidas para los comandos **File** y **Window**, entre otros.

Escritura de Macros con Visual Basic, Edición para Scripts

Dentro del entorno de desarrollo Visual J++ 1.1 está disponible el popular lenguaje de scripts Microsoft Visual Basic, Edición para Scripts (VBScript). De este modo, los desarrolladores pueden automatizar las tareas habituales y aumentar la velocidad de desarrollo. Por ejemplo, las ventanas y los documentos son objetos que permiten que el desarrollador pueda activarlos y redimensionarlos mediante programación. El desarrollador puede escribir una macro para automatizar estas tareas, o

simplemente almacenar sus pasos y a continuación añadir la macro a un menú o a una barra de herramientas.

Ventana Project Workspace

La ventana Project Workspace de Visual J++ ofrece tres vistas principales: ClassView, FileView y InfoView. Usando la ventana Project Workspace el desarrollador navega rápidamente al código o información deseados.

Se puede ver la ventana Project Workspace en la Figura 7.

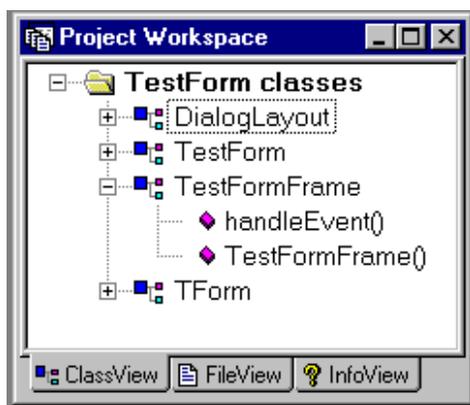


Figura 7. La ventana Project Workspace de Visual J++.

ClassView

El programador utiliza ClassView para examinar como grupos de clases, métodos y variables relacionados, del mismo modo en que piensa el desarrollador. ClassView es, esencialmente, un navegador de código no compilado que mantiene activo el código cambiado para que el desarrollador pueda navegar a través de todo el código actual. Cuando el navegador hace doble clic en cualquier método o variable, aparece el código relacionado en el editor. El desarrollador puede agrupar también métodos y variables por nivel de acceso, según sean privados, protegidos o públicos. Esto es particularmente útil cuando se crean applets para ser manejados en scripts.

Sólo se exponen para ser utilizados en lenguajes para scripts, como Visual Basic, Edición para Scripts, y JavaScript, los métodos y variables declarados como públicos. El desarrollador los agrupa según su acceso, para ver rápidamente cuáles están disponibles para utilizarlos en scripts. Se muestra un símbolo junto a cada método o variable para indicar si está accesible. Por ejemplo, el dibujo de un

candado indica que es privado. El desarrollador puede establecer puntos de ruptura en cualquier declaración de variable o en cualquier método, utilizando de panel ClassView.

La capacidad de añadir clases, métodos y variables es una característica especialmente útil de ClassView. Añadir una nueva capacidad a un programa es tan fácil como acceder a un método o clase en ClassView con un clic en el botón derecho del ratón. Las características de ClassView prueban el código según es escrito. Por ejemplo, cuando el desarrollador selecciona métodos modificados con cajas de verificación (check boxes), ClassView previene la creación de modificadores erróneos que no son conformes con las especificaciones del lenguaje Java. Esto ayuda a que el desarrollador sea productivo, impidiendo la introducción de errores potenciales.

FileView

El desarrollador usa FileView para examinar los archivos que conforman el proyecto. Los archivos examinados pueden ser archivos Java, archivos en lenguaje etiquetado para hipertexto (HTML), archivos gráficos u otros formatos de archivos. Son el uso del botón derecho del ratón, el desarrollador puede cambiar también de una forma fácil y rápida las opciones de construcción del proyecto. El desarrollador también puede construir el proyecto utilizando Visual J++ desde FileView.

InfoView

El desarrollador usa InfoView para hojear la documentación en línea. La forma de utilizar InfoView está cubierta en detalle en la documentación. Utilizando un clic del botón derecho del ratón, sobre un libro de la documentación, el desarrollador puede buscar detalles, crear subconjuntos e imprimir páginas relevantes.

AppWizards

El Asistente para Applets de Java

El Asistente para Applets de Java ofrece gran variedad de opciones para la construcción de las atribuciones de un applet. El desarrollador puede incluir comentarios que documenten algunas partes del código o detalles sobre los parámetros pasados al applet desde la página Web. Los desarrolladores pueden obtener ventajas del soporte para multihilo y el soporte para animaciones sencillas. Pueden ejecutar el applet como una aplicación o como un applet, seleccionar el

tamaño del applet y manejar los eventos habituales, tales como los clics o los movimientos del ratón.

Cuando el desarrollador utiliza el Asistente para Applets de Java con el fin de crear código, éste puede ser compilado y ejecutado inmediatamente, de modo que el código pueda ser usado como un primer bloque de construcción en la creación de código único.

Se puede analizar un paso del Asistente para Applets de Java en la Figura 8.

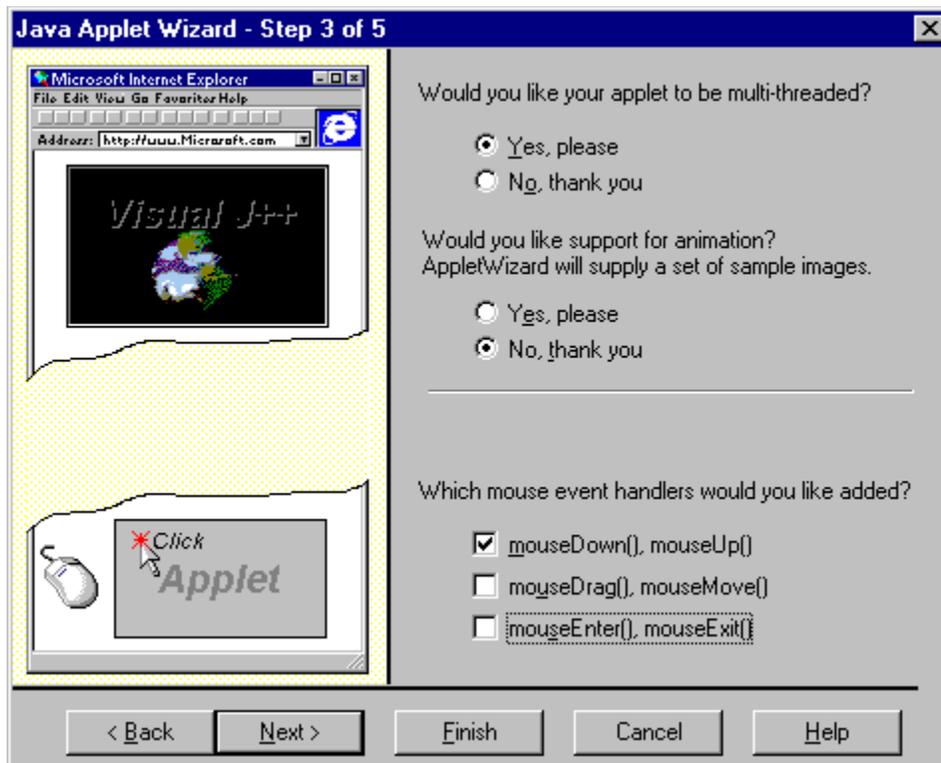


Figura 8. El Asistente para Applets de Java.

Asistente para el área de trabajo de Java

Usando el Asistente para el área de trabajo de Java el desarrollador puede migrar programas existentes creados con Java a Visual J++ fácilmente, ya que establece el área de un proyecto en la que los desarrolladores pueden insertar archivos de otros programas Java.

Asistentes para Aplicaciones Add-in

El desarrollador, utilizando Visual J++ puede añadir un asistente implementado a medida para crear applets y aplicaciones para propósitos específicos y para bibliotecas de clases concretas. Los AppWizards a medida de otros vendedores incluidos con la primera versión de Visual J++ provienen de compañías como AimTech, Neural Applications, Corp., y Jdesigner Pro. Con Visual J++ 1.1, los enlaces con estas y otras compañías se suministran en el sitio Web de Visual J++ www.microsoft.com/VisualJ. Esto pone a disposición de los desarrolladores las últimas versiones de estos asistentes tan pronto como están disponibles, en lugar de hacerlo cuando aparezca la siguiente versión de Visual J++.

Sistema de Proyectos

El sistema de Proyectos ofrece una forma sencilla e intuitiva para que el desarrollador administre el trabajo. Cuando el desarrollador inicia la programación, el Asistente para Applets Java de Visual J++ crea un área de trabajo del proyecto. Usando esta área de trabajo, un desarrollador puede trabajar en múltiples proyectos al mismo tiempo. Esto es útil cuando se desarrollan dos aplicaciones diseñadas para trabajar juntas. Por ejemplo, construyendo dos applets en la misma página Web o construyendo un applet en una página Web y una aplicación de Interface de Acceso Común (CGI) para ser ejecutada en el servidor. Alternativamente, el desarrollador puede construir un proyecto como un subproyecto, de forma que aun proyecto dependa de otro. Cuando el proyecto principal es recompilado, Visual J++ mantiene ambos proyectos actualizados.

Los desarrolladores pueden especificar y seleccionar entre múltiples configuraciones finales. De esta forma, pueden elegir las opciones de depuración y compilación, y la plataforma (establecer cuando se trata de una aplicación independiente o un applet destinado a correr en un navegador). También pueden elegir el navegador en el que probar el applet.

Además, el Java Workspace AppWizard crea áreas de trabajo en blanco en Java, dentro de las cuales el desarrollador puede añadir código Java a proyectos ya existentes.

Editor de Código Fuente

La elección de un editor de código fuente puede ser muy personal. Muchos desarrolladores relacionan combinaciones de teclas con funciones específicas y

características de un editor de código fuente concreto, directamente con su productividad. Debido a que el interface de Visual J++ es adaptable, el desarrollador puede elegir como adecuar el editor de código fuente. El editor estándar del IDE de Developer Studio ofrece todas las opciones que un desarrollador puede buscar, incluyendo auto-indentación, selección por columnas, soporte del botón derecho del ratón, edición mediante arrastrar-y-soltar y búsqueda. Para leer rápidamente el código, el desarrollador puede utilizar la sintaxis resaltada por color, no solamente para Java, sino también para código HTML. También están disponibles versiones de emulación de los famosos editores BRIEF y Epsilon para el desarrollador que quiera usar un editor de código fuente similar al que utilizan en otros productos, o para usarlo con otro sistema operativo.

Editor de Recursos y Asistente de Recursos

Usando Visual J++, los desarrolladores pueden crear y editar forms, menús e imágenes. Pueden reutilizar recursos de otras aplicaciones basadas en Windows cuando utilizan el administrador de disposición Java basado en Dialog Layout Unit (DLU) para crear forms con el mismo aspecto y la misma operabilidad que los familiares forms basados en Windows.

Editor de Forms y Menús

El editor que un desarrollador utiliza en Visual J++ para crear forms, menús e imágenes es el mismo que el popular editor de recursos utilizado en Visual C++. Las características del editor de forms incluyen líneas y bordes adherentes para alinear controles y soporte para la técnica de arrastrar-y-soltar que permite añadir de una paleta a una caja de diálogo. Apuntar y hacer clic es la técnica que utiliza el desarrollador en el editor de Menús para crear menús sencillos. Además, el desarrollador utiliza el administrador de disposición Java basado en DLU para situar controles Java en una caja de diálogo de un modo familiar.

Asistente de Recursos

El Resource Wizard convierte recursos basados en Windows en código Java independiente de la plataforma. De esta forma, el desarrollador puede reutilizar el interface de una aplicación existente y ofrecer al usuario final actual el mismo interface de usuario habitual. El código creado por el Resource Wizard es situado en un archivo separado del código que crea el desarrollador, facilitando así la actualización de un form.

Cuando se utiliza el Resource Wizard, el desarrollador puede especificar métodos únicos para manejar los eventos de un diálogo en lugar de forzar el empleo de un único método. El código generado de esta forma es además no propietario y, por consiguiente, completamente portable a otros entornos de desarrollo.

Editor de Imágenes

El Image Editor ofrece las opciones que los desarrolladores esperan de un editor de imágenes o de bitmaps comercial. Por ejemplo, pueden crear archivos .GIF o .JPEG compatibles con Java y convertir bitmaps existentes en archivos con formato .GIF o .JPEG.

Ayuda Global y Amable con el Desarrollador

Visual J++ ha sido diseñado como un entorno para aprender Java, tanto a nivel de principiante como a nivel avanzado. Visual J++ incluye un completo conjunto de información “para comenzar”, por el que un programador recién llegado a Java puede ser productivo rápidamente. Este sistema de información global no sólo incluye material para aprender la sintaxis del lenguaje Java, sino también una amplia Ayuda en línea, documentación impresa y en línea, tutoriales paso a paso y muchos programas de ejemplo documentados. Y lo mejor de todo, el propietario de Visual J++ puede hojear y buscar en la documentación en línea por contenidos, incluyendo las versiones completas de las especificaciones del lenguaje Java de Sun y del API.

Documentación y Otras Características Útiles para la Productividad

Debido a que la programación en Java es relativamente nueva, existe una necesidad de herramientas e información que permitan a un programador adquirir velocidad rápidamente. Además, los desarrolladores necesitan aprender recomendaciones y trucos sobre cómo codificar algo más que aplicaciones rudimentarias utilizando el lenguaje Java. Visual J++ ofrece tanto información sobre programación avanzada, como ayuda a los principiantes.

Visual J++ ofrece también características de productividad como el Asistente para Applet Java y la opción Add Method en Class View, para ayudar al desarrollador a eludir los errores de programación más comunes y a concentrarse en el aprendizaje y la programación.

La Tarjeta “Lea Esto Primero”

Desde que se abre la caja, los desarrolladores pueden empezar a ganar velocidad. Una tarjeta de referencia rápida de una sola página, etiquetada “Lea Esto Primero” ayuda a instalar a partir de ahí Visual J++ y a crear el primer applet. Y lo más importante, en ese momento se dice donde encontrar más información.

Aprenda Java Ahora

Visual J++ ofrece información conceptual y de cómo hacer determinadas cosas en la caja del producto, por medio de un libro de 320 páginas titulado *Aprenda Java Ahora* del autor de best sellers Randy Davis, publicado por Microsoft Press®. Randy

Davis, el célebre autor de best sellers de muchos de los libros ... *Para Torpes*, ofrece información clara y concisa sobre el lenguaje de programación Java.

Póster con la Jerarquía de Clases

Para una referencia rápida, también se incluye en la caja del producto, un póster de la librería de clases Java. El desarrollador puede remitirse al póster para aprender rápidamente las posibilidades de una clase específica y ver qué hereda de otras clases. En la documentación en línea se incluye una versión electrónica del dicho póster.

Asistente para Applets Java

El desarrollador utiliza el Asistente para Applets Java con el fin de construir las líneas generales de applets y aplicaciones que son completamente funcionales y a continuación construir las a partir de ellas. Debido a que el Asistente para Applets Java genera comentarios en el código que construye, el desarrollador principiante puede utilizar dichos comentarios para comprender como trabaja una aplicación Java bien diseñada. El Asistente para Applets Java también construye una página HTML con el applet Java embebido en ella, de modo que el programador puede comenzar a ejecutar y depurar un applet inmediatamente.

Class View

Las posibilidades Add Method, Add Variable y Add Class de Class View prueba el trabajo mientras que el desarrollador escribe el código. Las cajas de pruebas previene al programador de escribir nuevos métodos, variables o clases, utilizando parámetros que violen las especificaciones del lenguaje Java. Esto sirve de gran ayuda a los programadores noveles en Java que están aprendiendo la sintaxis Java. Se puede ver la caja de diálogo Add Method en la Figura 9.

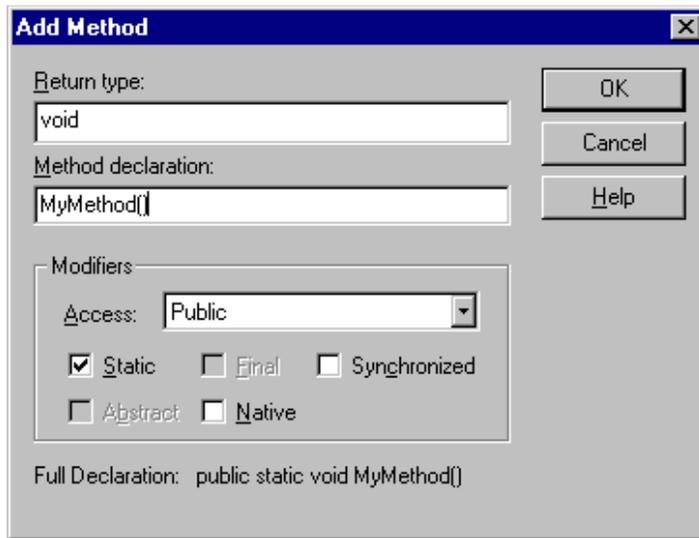


Figura 9. La caja de diálogo Add Method.

Ayuda en F1

El programador novel en un lenguaje a menudo tiene problemas para recordar como se usan algunas palabras claves específicas, además de las posibilidades y los parámetros de nuevas clases y métodos. Utilizando la Ayuda en F1, los programadores hacen doble clic en las palabras claves, clases, métodos o variables de interés, y el sistema de Ayuda visualiza los detalles necesarios.

Ayuda sobre Errores de Construcción

Visual J++ ofrece información específica para ayudar a los desarrolladores a encontrar y corregir errores de compilación. Por ejemplo, simplemente haciendo doble clic en un error de construcción se visualiza la línea de código que contiene dicho error. Entonces, presionando la tecla F1, el sistema de Ayuda visualiza un sumario del error y sus causas más frecuentes. Usando esta función de Ayuda, el programador inexperto no tiene por qué quedarse bloqueado al tratar de comprender un error de construcción.

Consejo del Día

Cada vez que el programador empieza Visual J++, el Consejo del Día destaca una característica poco conocida del lenguaje Java y del IDE de Developer Studio.

Qué contiene la Documentación de los Libros en Línea

La documentación contenida en los Libros en Línea de Visual J++ ofrece información global sobre cualquier cosa que necesite saber un programador que utiliza Java, incluyendo consejos para buscar de forma sencilla, hojear y adaptar los Libros en Línea.

En Visual J++, el sistema de ayuda está basado en el navegador, de modo que el desarrollador puede obtener fácilmente cualquier parte de la documentación facilitada y dentro de la información basada en Web.

Prueba Controlada

La Prueba Controlada de los Libros en Línea de Visual J++ está diseñada para demostrar a los nuevos usuarios algunas de las características principales del IDE de Developer Studio y de Visual J++.

Guía para Usuarios

La Guía para Usuarios de los Libros en Línea de Visual J++ ofrece información global y detallada acerca de todas las características de Visual J++.

Especificaciones del Lenguaje Java de Sun

En los Libros en Línea se incluyen e integran las especificaciones completas del lenguaje Java de Sun Microsystems, Inc. fácilmente localizable y accesible usando la Ayuda en F1.

Interface de Programación para Aplicaciones Java de Sun

En los Libros en Línea se incluye e integra el *Interface de Programación para Aplicaciones Java* completo, escrito por James Gosling y Frank Yellin, fácilmente localizable y accesible usando la Ayuda en F1.

Ejemplos, Ejemplos y más Ejemplos

Trabajar con ejemplos es una de las principales formas en que un programador aprende un nuevo lenguaje. Al examinar ejemplos de código, el programador ve como varias partes de una aplicación en ejecución trabajan juntas. Los ejemplos de Visual J++ están contruidos dentro de los Libros en Línea de modo que el programador puede buscar con el fin de encontrar ejemplos, obtener información sobre éstos e instalar y ejecutar los ejemplos si lo desea.

Sitios Web Favoritos

En Visual J++ se incluyen los enlaces a sitios Web habituales donde el desarrollador puede obtener más información sobre Java y programación en Java. El desarrollador que quiera más documentación, ejemplos y actualizaciones de productos, puede usar un enlace a la página principal de Microsoft Visual J++.

Cómo se utiliza la Documentación de los Libros en Línea

El desarrollador puede buscar, explorar y adaptar la información que desee, utilizando las características integradas de los Libros en Línea de Visual J++. Los desarrolladores pueden encontrar rápidamente la información que necesitan y a continuación volver a la programación.

F1

Haciendo clic en la palabra clave o en el mensaje de error de construcción, el usuario puede obtener el lugar apropiado de la documentación.

Búsqueda

El desarrollador puede buscar palabras específicas o combinaciones de palabras, y utilizar en los comandos de búsqueda operadores lógicos tales como AND y OR.

Examinar el Historial del Sistema

Los desarrolladores pueden localizar rápidamente y utilizar una lista de historiales de lo que ellos ya han mirado en los Libros en Línea, para volver a una parte útil de la Ayuda.

Revisar los Resultados de la Última Interrogación

El desarrollador puede examinar los resultados completos de la última interrogación de búsqueda, en caso de que el primer tema no contestase por completo a la pregunta.

Definir Subconjuntos

Los desarrolladores pueden construir, para una referencia rápida, una biblioteca a medida con los libros y los temas de Ayuda más frecuentemente utilizados.

Entonces, los desarrolladores pueden hojear y buscar en la biblioteca del mismo modo que lo hacen en todos los Libros en Línea.

Añadir Marcadores

Los desarrolladores pueden insertar un marcador con el fin de señalar una parte específica de los Libros en Línea que deseen recordar y encontrar rápidamente la próxima vez.

Incluir Anotaciones

Pueden añadir sus propias anotaciones a las secciones de los Libros en Línea. Personalizando, de este modo, la documentación según van aprendiendo.

Ver el Código y la Ayuda Juntos

Se puede adaptar el entorno de Visual J++, de forma que se puedan ver tanto el código como el tema de Ayuda al mismo tiempo. Ya no será necesario nunca más cambiar adelante y atrás entre el código y la ventana de documentación.

Incluye la Biblioteca Microsoft Developer Network en CD

Usando el entorno de desarrollo (IDE) de Developer Studio para desarrollar código en Visual J++, el desarrollador puede incluir la Biblioteca Microsoft Developer Network en CD con el conjunto de documentación. Microsoft Developer Network (MSDN) ofrece más de 600 Mb de datos de referencia, incluyendo los libros blancos del API de Win32, material de referencia relacionado con Java y consejos y trucos para desarrolladores de cualquier nivel. Todo el material MSDN es completamente localizable, y puede ser referenciado, anotado y usado como cualquier otro componente de los Libros en Línea.

Servicios de Soporte

Todos los desarrolladores que utilizan Visual J++ se benefician de la robustez de los servicios de soporte de Microsoft, incluyendo Microsoft Developer Network (MSDN), las Series Mastering basadas en el ordenador, tutoriales que se adaptan al ritmo del usuario y el programa Component Builder. Las series Mastering incluyen el Mastering Internet Development, que cubre la integración de Java con sitios Web, y Mastering Visual J++ que entra en el detalle del desarrollo de aplicaciones utilizando Visual J++. Además, Java y Visual J++ son el núcleo de destacados eventos de Microsoft como TechEd, Microsoft Professional Developer Conference (PDC) y otras conferencias.

Cuando lo necesita, el desarrollador puede acceder al soporte telefónico estándar de Microsoft para Microsoft Visual J++ y conectar con los populares foros en línea relativos a Java.