



Guida alla migrazione da Microsoft Java Virtual Machine per sviluppatori

Versione 2.2

16 aprile 2004

Le informazioni contenute in questo documento rappresentano la visione attuale di Microsoft Corporation relativamente ai temi affrontati alla data di pubblicazione. Poiché Microsoft deve rispondere alle mutevoli condizioni di mercato, non si può interpretare come un impegno da parte di Microsoft, la quale non può garantire l'accuratezza di tutte le informazioni fornite successivamente alla data di pubblicazione.

Questo white paper è esclusivamente per scopi informativi. MICROSOFT ESCLUDE OGNI GARANZIA ESPRESSA O IMPLICITA IN QUESTO DOCUMENTO.

Il rispetto di tutte le applicabili leggi in materia di copyright è esclusivamente a carico dell'utente. Fermi restando tutti i diritti coperti da copyright, nessuna parte di questo documento potrà comunque essere riprodotta o inserita in un sistema di riproduzione o trasmessa in qualsiasi forma e con qualsiasi mezzo (in formato elettronico, meccanico, su fotocopia, come registrazione o altro) per qualsiasi scopo, senza il permesso scritto di Microsoft Corporation.

Microsoft può essere titolare di brevetti, domande di brevetto, marchi, copyright o altri diritti di proprietà intellettuale relativi all'oggetto del presente documento. Salvo quanto espressamente previsto in un contratto scritto di licenza Microsoft, la consegna del presente documento non implica la concessione di alcuna licenza su tali brevetti, marchi, copyright o altra proprietà intellettuale.

© 2004 Microsoft Corporation. Tutti i diritti riservati.

Microsoft, Windows, Windows Server, Visual J#, Visual J++, Visual J#, Visual C#, ActiveX, JScript, Visual Basic e Visual Studio sono marchi o marchi registrati di Microsoft Corporation negli Stati Uniti e/o negli altri paesi.

Altri nomi di prodotti e società citati nel presente documento possono essere marchi dei rispettivi proprietari.

Sommario

1	5
	Introduzione.....	5
	Destinatari della guida.....	6
	Competenze richieste.....	6
	Utilizzo della guida.....	6
	Motivi che rendono necessaria la transizione.....	7
	Opzioni di transizione disponibili.....	7
	Migrazione a .NET.....	8
	Scelta di Visual J# o Visual C#.....	8
	Servizio Supporto Tecnico Clienti Microsoft.....	9
2	11
	Migrazione a .NET con Visual J# e J# Browser Controls.....	11
	Migrazione di applicazioni J++ con l'aggiornamento guidato di Visual J#.....	11
	Migrazione di applet Java con J# Browser Controls.....	12
	Compilazione di applet Java in J# Browser Controls.....	12
	Aggiornamento della pagina HTML per l'utilizzo di J# Browser Controls.....	14
	Distribuzione di J# Browser Controls.....	18
	Distribuzione di un controllo browser in più DLL.....	18
	Esecuzione di controlli J# Browser Controls nei computer degli utenti.....	20
	Funzionalità non supportate nella versione corrente.....	21
3	23
	Migrazione a .NET con Conversione guidata linguaggio Java.....	23
	Conversione di progetti di Visual J++ in Visual C#.....	24
	Conversione di progetti in linguaggio Java in Visual C#.....	24
	Aggiornamento manuale del codice non convertito.....	25
	Conversione di applet tramite Conversione guidata linguaggio Java.....	26
	Aggiornamento della pagina HTML.....	26
4	29
	Altre tecnologie di rendering.....	29
	DHTML e script sul lato client.....	29
	ECMAScript e JScript.....	30
	Flash.....	30
5	31
	Passaggio a un altro ambiente di runtime Java.....	31

1

Introduzione

In un accordo raggiunto nel gennaio 2001 con cui si è risolta la vertenza legale con Sun Microsystems in merito alla distribuzione dell'implementazione Microsoft di Java, Sun e Microsoft hanno concordato di limitare il periodo di utilizzo da parte di Microsoft del codice sorgente e dei test di compatibilità di Sun per il supporto di Microsoft® Java Virtual Machine (MSJVM). Nell'ambito dell'accordo di aprile 2004, tuttavia, Sun e Microsoft hanno concordato di estendere la licenza Microsoft per l'utilizzo del codice sorgente e dei test di compatibilità di Java di proprietà di Sun. Questa estensione consente a Microsoft di offrire servizi di supporto per MSJVM e di risolvere potenziali problemi di protezione correlati fino al 31 dicembre 2007.

Nel rispetto degli accordi siglati con Sun, Microsoft ha iniziato a eliminare MSJVM dai propri prodotti subito dopo il raggiungimento dell'accordo. Inoltre, nessun prodotto Microsoft futuro includerà MSJVM. MSJVM viene considerato codice obsoleto e quindi non sarà più esteso o sviluppato. Microsoft continuerà esclusivamente a fornire le correzioni di protezione eventualmente necessarie per mantenere la protezione dei computer degli utenti.

Se si accede a siti e applicazioni dipendenti da MSJVM tramite sistemi in cui non è installato MSJVM potrebbero verificarsi dei problemi. Per i siti Internet o Intranet, i problemi potrebbero interessare gli utenti che dispongono di nuovi sistemi Microsoft che non includono MSJVM (i sistemi aggiornati potrebbero non essere interessati in quanto l'aggiornamento dei sistemi operativi Windows non comporta la rimozione delle copie esistenti di MSJVM).

Microsoft consiglia ai clienti di Microsoft Visual J++® e Microsoft SDK per Java che dispongono di applicazioni incorporate o siti Web che utilizzano MSJVM di iniziare immediatamente l'individuazione delle dipendenze da MSJVM e pianificare tempestivamente la transizione a una soluzione alternativa. I clienti che scelgono di adottare una soluzione supportata da Microsoft possono fare riferimento alle informazioni disponibili per Microsoft Visual J#® .NET e Conversione guidata linguaggio Java. Questi prodotti forniscono supporto agli sviluppatori per la migrazione del codice sorgente esistente a .NET Framework.

Destinatari della guida

La presente guida tecnica è destinata agli sviluppatori responsabili della pianificazione delle attività di migrazione o transizione da MSJVM, che devono determinare gli approcci più appropriati e gestire l'implementazione delle modifiche necessarie.

Competenze richieste

Si presuppone che i lettori di questa guida conoscano Java. Le altre competenze richieste varieranno in base alle soluzioni di transizione scelte, tuttavia, potrebbe essere utile la conoscenza di ECMAScript, HTML dinamico (DHTML) e di alcune soluzioni Microsoft quali .NET Framework e il linguaggio C#.

Utilizzo della guida

Questo documento include informazioni specifiche per la gestione di progetti di migrazione da MSJVM, incluse la migrazione a un ambiente .NET, la riscrittura di applicazioni per altre tecnologie di rendering e la transizione a un altro ambiente di runtime Java (JRE, Java Runtime Environment).

Per informazioni dettagliate sull'individuazione delle dipendenze di un sistema da MSJVM e sulla valutazione delle esigenze aziendali corrispondenti, vedere la *Guida alla transizione da Microsoft Java Virtual Machine per professionisti IT*.

Di seguito è riportato un elenco del contenuto dei capitoli della guida:

Capitolo 1—Introduzione. Questo capitolo include una descrizione della guida e dei motivi per cui Microsoft sospenderà il supporto di MSJVM, oltre a un breve riepilogo delle opzioni disponibili per la transizione.

Capitolo 2—Migrazione a .NET con Visual J# e J# Browser Controls. In questo capitolo vengono illustrati due strumenti per la conversione automatica del codice Java in .NET, ovvero Visual J# e J# Browser Controls.

Capitolo 3—Migrazione a .NET con Conversione guidata linguaggio Java. In questo capitolo viene illustrato lo strumento automatico Conversione guidata linguaggio Java (JLCA) per la conversione di codice Java in .NET.

Capitolo 4—Altre tecnologie di rendering. Questo capitolo include informazioni di riepilogo sulle altre tecnologie di rendering disponibili, come DHTML, ECMAScript e Flash.

Capitolo 5—Passaggio a un altro ambiente di runtime Java. In questo capitolo vengono trattate alcune delle problematiche correlate alla selezione, installazione e distribuzione di un diverso ambiente di runtime Java (JRE, Java Runtime Environment).

Motivi che rendono necessaria la transizione

Il futuro di MSJVM è stato chiaramente definito nell'accordo del gennaio 2001 con Sun Microsystems, che prevede il ritiro dal mercato di MSJVM. Microsoft e Sun hanno stipulato un ulteriore accordo in aprile 2004, in base al quale Sun consente a Microsoft di continuare a supportare MSJVM fino al 31 dicembre 2007. Questo recente accordo, tuttavia, non contempla l'estensione delle licenze di distribuzione della tecnologia Sun in MSJVM. Microsoft continua le operazioni per il ritiro dal mercato di MSJVM, come stabilito dal precedente accordo.

In considerazione di quanto sopra, i clienti hanno la possibilità di non adottare alcuna misura per risolvere eventuali dipendenze da MSJVM. Tuttavia, questa non è l'alternativa consigliata da Microsoft. MSJVM diventerà un componente software non supportato a partire dal 31 dicembre 2007 e pertanto esistono implicazioni serie a livello di protezione nel caso non vengano adottate procedure di transizione per rimuovere le dipendenze esistenti da MSJVM prima di tale data.

L'esecuzione di MSJVM come software non supportato espone il sistema a potenziali vulnerabilità gravi per la protezione. In caso di problemi di protezione individuati dopo la suddetta scadenza, Microsoft non sarà in grado di fornire agli utenti patch di protezione o aggiornamenti funzionali per MSJVM.

È quindi consigliabile adottare tutte le misure necessarie per la migrazione o la transizione da MSJVM, prima di tale scadenza. Nel caso non sia possibile completare correttamente la transizione o si decida di non eseguire la transizione da MSJVM è perlomeno consigliabile impedire l'accesso di MSJVM ai siti attendibili.

Opzioni di transizione disponibili

Microsoft consiglia caldamente di avviare immediatamente il processo di individuazione delle dipendenze dei sistemi esistenti da MSJVM, per stabilirne l'entità. È importante tenere presente che qualsiasi soluzione scelta per risolvere le dipendenze individuate richiederà verifiche estese e un'analisi approfondita di tutti i sistemi interessati.

Dopo aver identificato le dipendenze esistenti da MSJVM, Microsoft consiglia di eseguire una o più delle seguenti operazioni, a seconda delle esigenze specifiche:

- Rimuovere MSJVM o adottare le necessarie misure di protezione per limitare l'accesso ad applet o applicazioni, come indicato nella *Guida alla transizione da Microsoft Java Virtual Machine per professionisti IT*.
- Eseguire la migrazione a un'altra soluzione:
 - Visual J# .NET o J# Browser Controls.
 - C# su .NET mediante l'utilizzo di Conversione guidata linguaggio Java.
 - Altre tecnologie di rendering, come DHTML, ECMAScript o una tecnologia di visualizzazione di terze parti, come Flash.
 - Un ambiente di runtime Java alternativo.

Nota Per informazioni specifiche sulle procedure per l'individuazione delle dipendenze da MSJVM, consultare la *Guida alla transizione da Microsoft Java Virtual Machine per professionisti IT*.

Migrazione a .NET

La migrazione a .NET consente di implementare una soluzione supportata da Microsoft e di riutilizzare il codice Java esistente. Per la migrazione a .NET di applicazioni e applet Visual J++ o in linguaggio Java, le alternative disponibili sono Visual J# e Conversione guidata linguaggio Java. Sebbene entrambi gli strumenti includano procedure guidate e utilità della riga di comando per eseguire automaticamente la migrazione di applet e applicazioni Java in .NET Framework, potrebbero essere necessari alcuni interventi manuali per la conversione del codice.

Grazie all'interoperabilità su linguaggi diversi offerta da .NET Framework, gli sviluppatori hanno la possibilità di utilizzare sia J# che Conversione guidata linguaggio Java in diverse combinazioni per i progetti di migrazione, a seconda degli specifici obiettivi per i vari componenti dei progetti.

Scelta di Visual J# o Visual C#

Se si decide per la migrazione a .NET, è necessario eseguire le opportune valutazioni per individuare la soluzione più appropriata e scegliere se optare per la conversione del codice esistente a Visual J# o Visual C#. Ad esempio, nel caso di investimenti consistenti nella tecnologia J++ e se il know-how disponibile è incentrato su Java, probabilmente la migrazione a Visual J# rappresenta la scelta più appropriata. Visual J# consente infatti agli sviluppatori di trasferire applicazioni J++ o Java in ambiente .NET mantenendo la sintassi del linguaggio Java e, nella maggior parte dei casi, anche le stesse funzionalità JDK. L'utilità J# Browser Controls è un nuovo componente aggiuntivo per J# progettato per la migrazione delle applet.

Se è invece necessaria una ristrutturazione consistente dell'architettura di applet o applicazioni, la migrazione a Visual C# può essere la soluzione ottimale per questi obiettivi. Conversione guidata linguaggio Java per C# consente di convertire sia le applet che le applicazioni Java in C# e .NET Framework.

Per la scelta tra Visual J# and Visual C#, è necessario prendere in considerazione vari altri fattori:

- **Conversione**
Visual J# supporta in genere la conversione di circa il 95% di tutto il codice MS Java o Java JDK 1.1.4 o versioni precedenti. Dopo la conversione a Visual J#, sono in genere necessari interventi manuali in meno del 5% del codice. Conversione guidata linguaggio Java consente di convertire circa il 90% del codice e supporta le funzionalità disponibili in J2SE 1.3 e J2EE 1.3 o versioni precedenti. Per la maggior parte dei progetti, dopo la conversione a Visual C# sono in genere necessari interventi manuali in meno del 10% del codice.
- **Runtime richiesti**
Visual J# richiede l'installazione di .NET Framework e del runtime J# nei computer degli utenti finali. Visual C# richiede unicamente l'installazione del runtime di .NET Framework.

Nota Per ulteriori informazioni sulla migrazione a .NET, inclusi casi aziendali sulla migrazione da J++ a Visual J# e istruzioni per il download di Conversione guidata linguaggio Java, vedere:

Visual J#: <http://www.microsoft.com/italy/msdn/vjsharp/default.asp>

Visual C#: <http://www.microsoft.com/italy/msdn/vcsharp/default.asp>

Servizio Supporto Tecnico Clienti Microsoft

Il Servizio Supporto Tecnico Clienti Microsoft (PSS) è a disposizione dei clienti per assistenza nella conversione di applicazioni e funzionalità esistenti per qualsiasi tecnologia Microsoft, incluso l'ambiente .NET. Il personale del servizio Microsoft PSS può supportare gli sviluppatori nella migrazione del codice Java esistente a Visual J# o Visual C#, per consentire la migrazione delle applicazioni esistenti con le funzionalità correnti. L'assistenza tecnica per lo sviluppo di funzionalità aggiuntive non è inclusa e continuerà a essere un servizio a pagamento. I servizi di supporto per la migrazione da MSJVM sono ora disponibili a livello internazionale. Per ulteriori informazioni, rivolgersi alla filiale locale Microsoft.

Nota Microsoft continuerà a supportare i prodotti esistenti con dipendenze da MSJVM in base alle politiche di supporto esistenti, ma non offrirà più supporto per i problemi correlati a MSJVM dopo il 31 dicembre 2007. Per ulteriori informazioni sui servizi di supporto tecnico nel mondo, visitare:
<http://support.microsoft.com/common/international.aspx>

2

Migrazione a .NET con Visual J# e J# Browser Controls

Microsoft Visual J# può essere utilizzato dagli sviluppatori esperti della sintassi del linguaggio Java per lo sviluppo di applicazioni e servizi in ambiente .NET Framework. Visual J# integra la sintassi del linguaggio Java nell'ambiente di sviluppo integrato (IDE, Integrated Development Environment) di Visual Studio® .NET e supporta inoltre la maggior parte delle funzionalità disponibili in Visual J++ 6.0, incluse le estensioni Microsoft come WFC.

Nota Le applicazioni e i servizi creati con Visual J# verranno eseguiti solo nel contesto di .NET Framework. Visual J# è un linguaggio indipendente sviluppato da Microsoft e non è riconosciuto né approvato da Sun Microsystems, Inc. Per ulteriori informazioni, visitare: <http://www.microsoft.com/italy/msdn/vjsharp/default.asp>

Migrazione di applicazioni J++ con l'aggiornamento guidato di Visual J#

Visual J# include una procedura guidata che consente di convertire i progetti di Visual J++ 6.0 in progetti Visual J#. Per utilizzare questa semplice procedura guidata per l'aggiornamento è sufficiente aprire un file di progetto di Visual J++ in Visual J#. È importante sottolineare che non vengono apportate modifiche al codice sorgente Java nel progetto. L'aggiornamento al formato di Visual Studio .NET riguarda solo i file di progetto e di soluzione.

Al termine dell'aggiornamento viene visualizzato un report di riepilogo nel quale sono indicati gli eventuali elementi non totalmente supportati in Visual J#. Il 95% circa di un progetto di Visual J++ viene aggiornato a Visual J# senza che siano necessarie ulteriori modifiche. In ogni caso, per le funzionalità non supportate direttamente in J#, l'ambiente .NET Framework include funzionalità equivalenti a disposizione degli sviluppatori.

Nota La funzionalità applet non è supportata in origine nella versione standard di Visual J#, pertanto il report dell'aggiornamento la indica come funzionalità non supportata. Per supportare la migrazione di questa funzionalità, tuttavia, è ora possibile utilizzare J# Browser Controls.

Migrazione di applet Java con J# Browser Controls

I controlli Microsoft J# Browser Controls consentono agli sviluppatori di eseguire la conversione di applet Java per l'ambiente .NET Framework. Le applet Java convertite in controlli J# Browser Controls mantengono le stesse caratteristiche di runtime senza perdite di funzionalità e possono essere eseguite nei computer degli utenti finali in Internet Explorer nel contesto di .NET Framework. Gli utenti finali che desiderano eseguire i controlli J# Browser Controls ospitati da un sito Web devono prevedere l'installazione di .NET Framework e del runtime di J# Browser Controls nel computer. In modo analogo, gli sviluppatori che desiderano aggiornare le applet Java a # Browser Controls devono prevedere l'installazione del runtime di J# Browser Controls e di .NET Framework SDK o Visual Studio .NET 2003 nei computer di sviluppo.

Nota I controlli J# Browser Controls non sono progettati per l'esecuzione in una Java Virtual Machine, ma solo su Internet Explorer. Per ulteriori informazioni su J# Browser Controls, visitare il sito all'indirizzo <http://msdn.microsoft.com/vjsharp/browsercontrols> (informazioni in lingua inglese).

Per eseguire la migrazione di applet Java a J# Browser Controls sono necessari tre passaggi:

1. Compilare l'applet Java in J# Browser Controls tramite il compilatore Visual J#.
2. Aggiornare la pagina HTML contenente il tag <APPLET> dell'applet.
3. Distribuire J# Browser Controls e la pagina HTML nel Web server.

Compilazione di applet Java in J# Browser Controls

Mediante l'utilizzo del "Compilatore Visual J#" (vjc.exe), è possibile compilare le applet Java in una libreria gestita ed eseguire l'aggiornamento a J# Browser Controls. Per ulteriori informazioni (in lingua inglese) vedere:

http://msdn.microsoft.com/library/en-us/dv_vjsharp/html/vjgrfVisualJCompilerOptions.asp

Le procedure per la compilazione delle applet Java in una libreria gestita sono simili a quelle per la compilazione di qualsiasi altra libreria in Visual J#. Se per l'applet Java esiste solo il bytecode (file class), sarà possibile utilizzare lo strumento per la conversione binaria di Visual J# (JbImp.exe) per convertire l'applet Java in una libreria gestita. Per ulteriori informazioni (in lingua inglese) vedere:

http://msdn.microsoft.com/library/en-us/dv_vjsharp/html/vjgrfJavaBinaryConverter.asp

La compilazione di un'applet Java in un controllo J# Browser Controls non richiede alcuna modifica al codice sorgente dell'applet Java perché l'ambiente di runtime di J# Browser Controls fornisce il supporto per le funzionalità equivalenti nella maggior parte dei pacchetti di livello JDK 1.1.4, incluso il pacchetto **java.applet**.

Le librerie gestite contenenti J# Browser Controls non sono diverse dalle altre librerie gestite in .NET Framework. Quando un utente visita un sito Web che ospita un controllo J# Browser Controls, il runtime di J# Browser Controls scarica la libreria gestita ed esegue la classe che estende **java.applet.Applet** in Internet Explorer.

Le applet Java possono essere compilate come controllo J# Browser Controls dal prompt dei comandi o da Visual Studio .NET.

Compilazione di un'applet Java dal prompt dei comandi

Per compilare un'applet Java dalla riga di comando con il compilatore J# (vjc.exe):

```
C:\OriginiApplet>vjc.exe /target:library /out:Applet.dll *.java
```

Per compilare un'applet Java con lo strumento per la conversione binaria di J#:

```
C:\OriginiApplet>jbimp.exe /target:library /out:Applet.dll *.class
```

In un computer in cui è installato Visual Studio .NET, il compilatore J# (vjc.exe) e lo strumento per la conversione binaria di J# (JbImp.exe) sono accessibili dalla finestra di comando di Visual Studio .NET. Se Visual Studio .NET non è installato, sarà necessario verificare che gli strumenti indicati si trovino nel percorso del prompt dei comandi.

Per la migrazione di applet Java che utilizzano risorse, è necessario attenersi alla procedura descritta nel documento "Upgrading Visual J++ 6.0 Applications That Use Resources" all'indirizzo http://msdn.microsoft.com/library/en-us/dv_vjsharp/html/vjgrfUpgradingVisualJ60ApplicationsThatUseResources.asp e nell'articolo "How To: Resources in Visual J# .NET" all'indirizzo <http://www.gotdotnet.com/team/vjsharp/ResourcesHowTo.htm> (informazioni in lingua inglese).

Compilazione di un'applet Java con Visual Studio .NET

Se l'applet Java è un'applet di Visual J++ 6.0 in un progetto HTML, è inoltre possibile compilarla come controllo J# Browser Controls tramite Visual Studio .NET:

1. Aprire il progetto di Visual J++ 6.0 in Visual Studio .NET. Verrà avviato l'aggiornamento guidato del progetto di Visual J# .NET.
2. Scegliere Avanti in tutti i passaggi dell'aggiornamento guidato. L'aggiornamento guidato esegue la conversione del progetto di Visual J++ in un progetto Libreria di classi di Visual J# .NET.
3. Aprire il report dell'aggiornamento per verificare se sono stati rilevati eventuali problemi durante l'aggiornamento.

Nota Nel report verrà visualizzato un messaggio che indica che i progetti applet non sono supportati. Questo errore può essere ignorato. È invece necessario correggere tutti gli altri errori elencati nel report dell'aggiornamento prima di procedere alla generazione del progetto. Questo errore si verifica perché il componente J# Browser Controls è diventato disponibile dopo il rilascio di Visual Studio .NET 2003.

4. Generare il progetto. L'applet Java verrà compilata in una libreria gestita.
Non è possibile avviare un controllo J# Browser Controls in Internet Explorer da Visual Studio scegliendo **Avvia** dal menu **Debug** oppure premendo F5. Per eseguire il controllo, è necessario copiarlo in una directory virtuale su un server Web.

Nota Non è possibile avviare un controllo J# Browser Controls v1.1 in Internet Explorer da Visual Studio scegliendo **Avvia** dal menu **Debug** oppure premendo F5. Per eseguire il controllo, è necessario copiarlo in una directory virtuale su un server Web. Con J# Browser Controls v1.1b non è più necessario utilizzare un server Web per l'hosting dei controlli.

Per ulteriori informazioni (in lingua inglese) vedere "Deploying J# Browser Controls":

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dv_vstechart/html/vjtskMigratingJavaAppletsToMicrosoftJBrowserControls.asp

e "How to: Debug J# Browser Controls": http://msdn.microsoft.com/library/en-us/dv_vstechart/html/vjtskhowtodebugjbrowsercontrols.asp

Per ulteriori informazioni (in lingua inglese) sull'aggiornamento dei progetti di Visual J++ 6.0 a Visual J# .NET, vedere "Upgrading from Visual J++ 6.0":

http://msdn.microsoft.com/library/en-us/dv_vjsharp/html/vjsamUpgradingFromVisualJ60.asp

Aggiornamento della pagina HTML per l'utilizzo di J# Browser Controls

Dopo aver compilato l'applet Java in una libreria gestita, è necessario aggiornare la pagina HTML che ospita l'applet Java. Il tag <APPLET> o il tag <OBJECT> dell'applet Java nella pagina HTML deve essere convertito nella sintassi del tag <OBJECT> supportata da J# Browser Controls.

Il runtime di # Browser Controls include uno strumento per la conversione di applet HTML in tag Object (TagConvert.exe) che aggiorna automaticamente le pagine HTML per l'utilizzo dei controlli J# Browser Controls. TagConvert.exe è disponibile nella directory di installazione di J# Browser Controls. La sintassi per l'utilizzo dello strumento è la seguente:

```
TagConvert [opzioni] <file di origine>
```

Ad esempio:

```
TagConvert.exe PaginaApplet.html
```

L'input per lo strumento può essere rappresentato da qualsiasi file di testo, inclusi tutti i file con estensione html, htm, asp e aspx.

Lo strumento converte il tag <APPLET> o il tag <OBJECT> dell'applet Java nel seguente tag <OBJECT> di J# Browser Controls:

```
<OBJECT
  CLASSID="clsid:a399591c-0fd0-41f8-9d25-bd76f632415f"
  WIDTH= pixel
  HEIGHT= pixel
  ID=nomeControlloBrowser
  ALIGN= allineamento
  HSPACE= pixel
  VSPACE= pixel
  VJSCODEBASE = URLcodebase
>
  <PARAM NAME = attributo1 VALUE = valore>
  <PARAM NAME = attributo2 VALUE = valore>
  . . .
  alternateHTML
</OBJECT>
```

Nell'esempio precedente, **CLASSID** è l'ID di classe del controllo ActiveX che scarica ed esegue J# Browser Controls. Il valore CLASSID indicato nell'esempio è quello che deve essere utilizzato nel tag <OBJECT> di J# Browser Controls.

VJSCODEBASE è l'URL della classe del controllo J# Browser Controls e della libreria gestita contenente la classe. Il token "#" separa il nome del file della libreria gestita dal nome della classe del controllo J# Browser Controls. Il nome di file deve includere anche l'estensione. Ad esempio:

```
VJSCODEBASE =  
http://www.SitoWeb.com/Applet/ClasseApplet.dll#ClasseApplet
```

Processo di conversione

Quando si esegue la conversione delle pagine HTML, il tag <APPLET> originale o il tag <OBJECT> dell'applet Java viene eliminato e sostituito con il tag <OBJECT> di J# Browser Controls. Questo strumento genera inoltre un file JavaScript denominato jbctagconvert.js in tutte le directory in cui viene eseguito. Quando si carica la pagina HTML aggiornata in un browser, il rendering del controllo J# Browser Controls nella finestra del browser viene eseguito utilizzando la pagina HTML aggiornata in combinazione con i file JavaScript. Lo strumento genera il file JavaScript per offrire agli utenti maggiori potenzialità per l'esplorazione con J# Browser Controls in un prossimo aggiornamento di Internet Explorer.

Ulteriori informazioni (in lingua inglese) sul potenziale aggiornamento di Internet Explorer sono disponibili all'indirizzo:

<http://msdn.microsoft.com/ieupdate/default.asp>. Anche se questi aggiornamenti non dovessero essere implementati, il codice aggiuntivo sopra menzionato non comprometterà il funzionamento di J# Browser Controls e non sarà quindi necessario rimuoverlo.

Nota Il tag <APPLET> o il tag <OBJECT> dell'applet Java viene sostituito dal tag <OBJECT> del controllo J# Browser Controls racchiuso tra commenti e preceduto dal tag . Questo blocco di commento e il tag associati non devono essere eliminati perché sono necessari per la visualizzazione del controllo J# Browser Controls. Anche il file JavaScript generato dallo strumento, ovvero jbctagconvert.js, non deve essere eliminato ed è necessario che sia incluso in ogni directory in cui è presente una pagina HTML che contiene un controllo J# Browser Controls.

Prima della conversione dei tag, viene creata una copia di backup dei file originali. Le copie di backup dei file hanno l'estensione vjsbak e vengono memorizzate nella directory del file originale. Ad esempio, il nome della copia di backup di index.htm sarà index.htm.vjsbak.

Per creare il valore dell'attributo VJSCODEBASE, lo strumento utilizza i valori degli attributi CODE e CODEBASE del tag <APPLET> originale (o il tag <OBJECT> dell'applet Java). Ad esempio, il blocco di codice:

```
CODE = "ClasseApplet"  
CODEBASE = http://www.SitoWeb.com/Applet
```

viene modificato in:

```
VJSCODEBASE =  
http://www.SitoWeb.com/Applet/ClasseApplet.dll#ClasseApplet
```

Per impostazione predefinita, lo strumento presuppone che il nome della classe del controllo J# Browser Controls (ad esempio, ClasseApplet) sia uguale al nome della DLL (ClasseApplet.dll). Se i nomi della classe del controllo J# Browser Controls e della DLL sono diversi, sarà necessario modificare corrispondentemente il valore dell'attributo VJSCODEBASE.

Quando si utilizza questo strumento, è consigliabile compilare l'applet Java in una libreria gestita con lo stesso nome della classe applet. Ad esempio:

```
C:\OriginiClasseApplet>vjc /target:library /out:ClasseApplet.dll *.java
```

Il runtime di J# Browser Controls v1.1 supporta esclusivamente i protocolli HTTP e HTTPS nell'attributo VJSCODEBASE. Tutti i percorsi assoluti specificati in VJSCODEBASE devono quindi iniziare con http:// o https://. Se si specifica un percorso relativo, il controllo J# Browser Controls viene caricato utilizzando il protocollo HTTP. J# Browser Controls non supporta il caricamento dei controlli da una posizione diversa da DOCTYPE, ovvero la posizione da cui viene caricata la

pagina HTML. Il valore indicato in VJSCODEBASE deve corrispondere alla posizione DOCTYPE oppure rappresentare una delle sottodirectory di DOCTYPE.

Nota Con il runtime di J# Browser Controls v1.1b questo requisito non esiste più e viene introdotto il supporto diretto dei file, senza che sia necessario un server Web.

Molti attributi del tag OBJECT rimangono invariati durante la conversione. Nella sezione seguente vengono descritti in dettaglio gli attributi convertiti dallo strumento TagConvert. Molti parametri del tag <OBJECT> di J# Browser Controls sono identici ai parametri corrispondenti del tag <APPLET> o del tag <OBJECT> dell'applet Java.

Opzioni della riga di comando per TagConvert.exe

Di seguito sono riportate le opzioni della riga di comando supportate per TagConvert.exe:

`/recurse:<carattere jolly>`

Lo strumento cerca i file da convertire nella directory corrente e in tutte le sottodirectory in base ai caratteri jolly specificati. Ad esempio, per aggiornare tutti i file con estensione htm e html inclusi nella directory corrente e nelle relative sottodirectory:

```
TagConvert /recurse *.htm *.html
```

`/verbose`

Lo strumento stampa i nomi dei file modificati durante la conversione. Il nome di file include il percorso completo del file. Viene inoltre stampato il numero totale di file analizzati e il numero totale di file convertiti. Ad esempio, per aggiornare tutti i file con estensione htm nella directory specificata e copiare tutti i nomi dei file modificati nel file filemodificati.txt:

```
TagConvert /verbose \OriginiApplet\Pagine\*.htm >
filemodificati.txt
```

Mapping degli attributi tra il tag <APPLET> e il tag <OBJECT> di J# Browser Controls

Nella tabella seguente vengono indicate le corrispondenze tra gli attributi nel tag <APPLET> e gli attributi nel tag <OBJECT> di J# Browser Controls.

Sintassi tag <APPLET> (con le estensioni di Internet Explorer)	Sintassi tag <OBJECT> di J# Browser Controls
CODEBASE	VJSCODEBASE
CODE	VJSCODEBASE
WIDTH	WIDTH
HEIGHT	HEIGHT
NAME	ID
ID	ID
ALIGN	ALIGN
VSPACE	VSPACE
HSPACE	HSPACE
ARCHIVE	Non supportato nella versione corrente
ALT	ALT
<PARAM>	<PARAM>
alternateHTML	alternateHTML
<PARAM NAME = FireScriptEvents VALUE = True>	Non modificato, ma non supportato nella versione corrente
<PARAM NAME = cabbase VALUE = nomeFileCab> <PARAM NAME = cabinets VALUE = nomiFileCab>	Non modificato, ma non supportato nella versione corrente
<PARAM NAME = useslibrary VALUE = NomeDescrittivoDU> <PARAM NAME = useslibrarycodebase VALUE = NomeFileDU> <PARAM NAME = useslibraryversion VALUE = NumeroVersioneDU>	Non modificato, ma non supportato nella versione corrente
<PARAM NAME = namespace VALUE = spaziodeinomiApplicazione>	Non modificato, ma non supportato nella versione corrente

Nella versione corrente i seguenti attributi del tag <APPLET> non sono supportati e non esistono attributi equivalenti nel tag <OBJECT> di J# Browser Controls:

- Gli attributi **archive**, **cabbase** e **cabinets**. L'assemblaggio di controlli J# Browser Controls in file cab, zip o jar non è supportato nella versione corrente. I controlli J# Browser Controls devono essere distribuiti nel server Web come file DLL autonomi.
- L'attributo **FireScriptEvents**. Nella versione corrente non è supportato il sink degli eventi generati da J# Browser Controls negli script della pagina HTML. Le funzionalità di script sono state introdotte in J# Browser Controls v1.1b, attualmente in versione beta.
- Gli attributi **useslibrary**, **useslibrarycodebase**, **useslibraryversion** e **namespace**. L'ambiente di runtime di J# Browser Controls non supporta la semantica di gestione dei pacchetti Java di MSJVM.

Gli attributi nel tag <APPLET> per i quali non esistono equivalenti diretti nel tag <OBJECT> possono essere copiati così come sono. Questi attributi hanno lo stesso significato nel tag <OBJECT> di J# Browser Controls e nel tag <APPLET> originale.

Distribuzione di J# Browser Controls

Per distribuire controlli J# Browser Controls, è necessario copiare la libreria gestita e le pagine HTML aggiornate nella directory appropriata del server Web. La libreria gestita deve essere copiata nella stessa directory della pagina HTML o in una sottodirectory.

Quando si utilizza il server Web IIS, è necessario impostare in modo corretto le autorizzazioni per la directory virtuale in cui viene copiato il controllo J# Browser Controls. Il campo **Autorizzazioni di esecuzione** della directory virtuale deve essere impostato su **Solo script**, ovvero il livello di autorizzazione predefinito di una directory virtuale in IIS. Anche tutti i file di risorse utilizzati nel controllo J# Browser Controls, ad esempio file di immagini, file audio o file di dati, devono essere copiati nella stessa posizione della libreria gestita.

Nota Con J# Browser Controls v1.1b non è più necessario utilizzare un server Web per l'hosting dei controlli.

Distribuzione di un controllo browser in più DLL

Un controllo J# Browser Controls può essere suddiviso in più file DLL. In questi casi, la pagina HTML deve fare riferimento alla DLL contenente la classe principale del controllo J# Browser Controls. Il runtime di J# Browser Controls scaricherà le DLL aggiuntive richieste durante la fase di esecuzione.

Per la distribuzione di un controllo suddiviso in più DLL in un server Web, tutti i file correlati allo stesso controllo J# Browser Controls devono essere copiati nella stessa directory come file autonomi e non possono essere assemblati come file cab, zip o jar.

Assemblaggio di più controlli J# Browser Controls nella stessa libreria

È possibile assemblare più controlli J# Browser Controls nella stessa libreria gestita. In questo caso, il tag <OBJECT> di ogni controllo browser deve fare riferimento alla stessa libreria gestita ma con nomi di classi differenti.

Ad esempio, se la libreria Applet.dll nella directory DirApplet contiene i controlli J# Browser Controls denominati Applet1 e Applet2, è possibile fare riferimento ai due controlli come illustrato nell'esempio seguente:

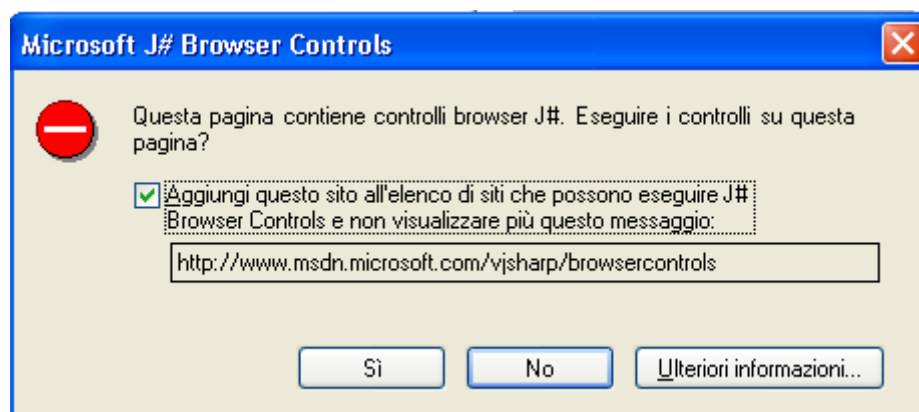
```
VJSCODEBASE = http://www.microsoft.com/DirApplet/Applet.dll#Applet1
```

VJSCODEBASE = <http://www.microsoft.com/DirApplet/Applet.dll#Applet2>

Poiché i controlli J# Browser Controls possono essere scaricati solo dalla stessa posizione della pagina HTML o da una delle sue sottodirectory, potrebbe essere necessario copiare la libreria gestita in più posizioni se le directory delle pagine HTML sono diverse.

Esecuzione di controlli J# Browser Controls nei computer degli utenti

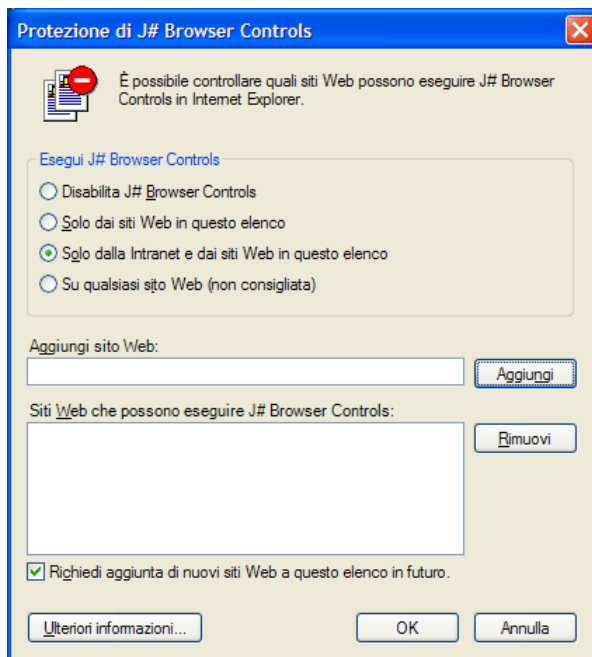
Per eseguire i controlli J# Browser Controls in Internet Explorer, è necessario che nei computer degli utenti sia installato il runtime di J# Browser Controls. Per impostazione predefinita, il runtime di J# Browser Controls richiede agli utenti finali di confermare l'esecuzione dei controlli ospitati nelle pagine Web. Quando un utente accede a un sito Web contenente un controllo J# Browser Controls, verrà visualizzata la finestra di dialogo seguente:



Se l'utente sceglie **Sì**, i controlli J# Browser Controls vengono scaricati ed eseguiti. Scegliendo **No** si impedisce l'esecuzione del controllo browser.

Se si seleziona la casella di controllo **Aggiungi questo sito all'elenco di siti che possono eseguire J# Browser Controls e non visualizzare più questo messaggio** e si sceglie **Sì**, il sito Web verrà aggiunto all'elenco dei siti per cui è consentita l'esecuzione dei controlli J# Browser Controls e all'utente non verrà più richiesta una conferma per le pagine del sito Web.

Per gestire l'elenco dei siti in cui è consentita l'esecuzione dei controlli J# Browser Controls, è inoltre possibile utilizzare la finestra di dialogo con le opzioni di protezione di J# Browser Controls disponibile in Strumenti di amministrazione nel Pannello di controllo. In Windows XP, è possibile accedere a questa finestra di dialogo da **Prestazioni e manutenzione** nel Pannello di controllo. Fare doppio clic sull'icona **Protezione di J# Browser Controls** per aprire la finestra di dialogo seguente:



Le opzioni disponibili sono:

- **Disabilita J# Browser Controls**
Impedisce l'esecuzione dei controlli J# Browser Controls nel computer da qualsiasi sito Web.
- **Solo dai siti Web in questo elenco**
Consente l'esecuzione dei controlli J# Browser Controls solo dai siti Web aggiunti all'elenco. **Questa è l'impostazione predefinita.**
- **Solo dalla Intranet e dai siti Web in questo elenco**
Consente l'esecuzione solo dei controlli J# Browser Controls nella rete Intranet o dai siti Web aggiunti all'elenco.
- **Su qualsiasi sito Web**
Consente l'esecuzione dei controlli J# Browser Controls da tutti i siti.

Se si immette l'indirizzo di un sito Web nella casella di testo **Aggiungi sito Web** e si sceglie **Aggiungi**, il sito Web verrà aggiunto all'elenco dei siti in cui è consentita l'esecuzione dei controlli J# Browser Controls. In modo analogo, se si seleziona un sito Web nell'elenco **Siti Web che possono eseguire J# Browser Controls** e si sceglie **Rimuovi**, il sito verrà rimosso dall'elenco. Quando si aggiunge un sito Web all'elenco dei siti in cui è consentita l'esecuzione dei controlli J# Browser Controls, all'utente non verrà chiesta alcuna conferma prima dell'esecuzione dei controlli nel sito Web.

Per impostazione predefinita, la casella di controllo **Richiedi aggiunta di nuovi siti Web a questo elenco in futuro** è selezionata, il che determina una richiesta di conferma all'utente prima dell'esecuzione dei controlli J# Browser Controls dai siti Web non inclusi nell'elenco. Quando la casella di controllo è deselezionata, i controlli J# Browser Controls non verranno eseguiti dai siti Web non inclusi nell'elenco e all'utente non verrà richiesta alcuna conferma.

Funzionalità non supportate nella versione corrente

Le seguenti funzionalità non sono supportate nella versione corrente di J# Browser Controls:

- **Protezione basata sull'attendibilità**
La semantica di protezione basata sull'attendibilità non è supportata come in MSJVM.
- **Java Package Manager**
La funzionalità per la gestione dei pacchetti Java supportata in MSJVM non è supportata per i controlli J# Browser Controls. I controlli J# Browser Controls non supportano una cache degli oggetti che possa essere utilizzata per installare le classi localmente ed eseguirle con un set di autorizzazioni limitato tramite firma digitale. Pertanto, non sono supportate neanche l'installazione e l'esecuzione di unità di distribuzione.
- **File di archivio**
I file di archivio non sono supportati. Un controllo J# Browser Controls può essere suddiviso in più librerie gestite. Tuttavia, quando si distribuisce un controllo suddiviso in più librerie in un server Web, i file devono essere copiati come file autonomi e non possono essere assemblati in file cab, zip o jar.
- **Nessuna finestra di progettazione supportata**
Per i controlli J# Browser Controls non è supportata alcuna finestra di progettazione in Visual Studio .NET.

3

Migrazione a .NET con Conversione guidata linguaggio Java

Conversione guidata linguaggio Java è uno strumento che consente la conversione in Visual C# dei progetti di Visual J++ 6.0 e dei file in linguaggio Java. Eseguendo la conversione di questi file in Visual C#, è possibile riutilizzare la base di codice esistente e sfruttare i vantaggi offerti da .NET Framework. Conversione guidata linguaggio Java non modifica il progetto esistente, ma crea un nuovo progetto di Visual C# basato sul progetto originale. Lo strumento Conversione guidata linguaggio Java non è destinato unicamente alla conversione di codice Java per MSJVM, ma consente di convertire anche programmi Java sul lato server, tra cui Java Server Pages (JSP) e Servlet, oltre a librerie J2SE e J2EE. Il progetto convertito di Visual C# contiene tutto il nuovo codice Visual C# generato automaticamente dal codice esistente in linguaggio Java o Visual J++.

Per convertire alcune funzionalità del progetto originale che non sono disponibili in Microsoft Visual C#®, Conversione guidata linguaggio Java crea *classi di supporto* che duplicano le funzionalità originali. Nonostante vengano eseguiti tutti i tentativi possibili per preservare l'architettura originale dell'applicazione nel progetto convertito, l'architettura delle classi di supporto è a volte notevolmente diversa rispetto alle classi duplicate.

È possibile che parte del codice del progetto non possa essere convertito automaticamente. Al termine della generazione del nuovo progetto, tutti gli errori, gli avvisi o i problemi rilevati nel corso della conversione vengono visualizzati in un report. Il codice non convertito viene evidenziato nel codice del nuovo progetto tramite commenti con l'etichetta `UPGRADE_TODO`. I commenti relativi alla conversione possono essere visualizzati nell'elenco delle attività. Ogni commento sulla conversione contiene un collegamento a un argomento della Guida con informazioni per la conversione manuale del codice.

Nota Se si utilizza Visual C# per la prima volta, è consigliabile effettuare alcune prove per acquisire familiarità con questo linguaggio prima di eseguire la procedura di conversione. Per ulteriori informazioni (in lingua inglese), vedere "C# Language Tour": http://msdn.microsoft.com/library/en-us/cskon/html/vclrfGettingStarted_PG.asp e "Converting Visual J++ or Java-language Project to Visual C#": http://msdn.microsoft.com/library/en-us/dv_jlca/html/vbtskusingjavavalanguageconversionassistant.asp

Conversione di progetti di Visual J++ in Visual C#

Per convertire un progetto di Visual J++ tramite Conversione guidata linguaggio Java:

1. Avviare Visual Studio .NET.
2. Scegliere **Apri** dal menu **File** e quindi fare clic su **Converti**.
3. Selezionare **Conversione guidata linguaggio Java** e scegliere **OK**.
4. Nella pagina **File di origine** selezionare **Un progetto VJ++ 6.0**.
5. Nella pagina **Selezionare un file di progetto Visual J++** scegliere **Sfoggia**.
6. Cercare il file vjp corretto e selezionarlo.

Nota Se nella direttiva CLASSPATH del file vjp vengono specificati file jar o class, tali file verranno ignorati.

7. Nella pagina **Specificare il percorso del nuovo progetto** specificare il nome e la directory del nuovo progetto da creare.
8. Nella pagina **Pronto per la conversione** scegliere **Avanti**.
9. Correggere il codice che non può essere convertito automaticamente.

Conversione di progetti in linguaggio Java in Visual C#

Per convertire un progetto in linguaggio Java tramite Conversione guidata linguaggio Java:

1. Avviare Visual Studio .NET.
2. Scegliere **Apri** dal menu **File** e quindi fare clic su **Converti**.
3. Selezionare **Conversione guidata linguaggio Java** e scegliere **OK**.
4. Nella pagina **File di origine** selezionare **Una directory contenente i file del progetto**.
5. Nella pagina **Seleziona directory di origine** scegliere **Sfoggia**.
6. Cercare il progetto corretto e selezionarlo.

Nota Nella directory selezionata non verranno visualizzati i file, ma tutti i file jav e java in essa contenuti saranno convertiti. Tutti gli altri file inclusi nella directory vengono ignorati.

7. Nella pagina **Informazioni nuovo progetto** specificare le informazioni seguenti:
 - Il nome del progetto da creare.
 - La directory in cui sono disponibili eventuali file aggiuntivi per il progetto.
 - Il tipo di output del progetto.
8. Nella pagina **Specificare il percorso del nuovo progetto** specificare il nome e la directory del nuovo progetto da creare.
9. Nella pagina **Pronto per la conversione** scegliere **Avanti**.
10. Correggere il codice che non può essere convertito automaticamente.

Aggiornamento manuale del codice non convertito

Al termine della conversione del progetto di Visual J++ o dei file in linguaggio Java tramite Conversione guidata linguaggio Java, è possibile che il nuovo progetto di Visual C# contenga parti di codice che non sono state convertite automaticamente. Per consentire la conversione manuale di queste parti di codice in Visual C#, vengono inseriti alcuni commenti nel codice del nuovo progetto.

Tipo di commento	Descrizione
UPGRADE_TODO	Codice che non può essere convertito automaticamente.
UPGRADE_WARNING	Codice che potrebbe presentare dei problemi.
UPGRADE_NOTE	Codice che potrebbe presentare un comportamento diverso dal codice originale.

È inoltre possibile che siano presenti alcuni errori del compilatore che devono essere corretti prima della compilazione dell'applicazione. Ogni commento include una breve descrizione del problema e un collegamento a un argomento della Guida con informazioni sulle procedure per la conversione del codice. Per aggiornare manualmente il codice non convertito:

1. Dopo aver eseguito Conversione guidata linguaggio Java, aprire il progetto in Visual Studio .NET.
2. Scegliere **Mostra attività** dal menu **Visualizza** e quindi **Commenti** o **Tutte**. Le note relative alla conversione verranno visualizzate nell'**Elenco attività**.
3. Nell'**Elenco attività** fare doppio clic sulla prima nota sulla conversione. Lo stato attivo si sposta nella posizione corrispondente al commento nel codice.
4. Esaminare la nota e il codice a cui fa riferimento il commento. Per ulteriori informazioni sulla correzione manuale di un errore, fare clic sul collegamento incluso nella nota per visualizzare l'argomento della Guida relativo all'errore.

Alcune note relative alla conversione nel codice convertito per le pagine JSP non sono collegamenti. In questo caso, è possibile tagliare la nota e incollarla nella casella **Cerca** della Guida.

5. Aggiornare il codice manualmente.
6. Ripetere i passaggi da 3 a 5 fino a risolvere tutte le note relative alla conversione.

Nota Per ulteriori informazioni (in lingua inglese), vedere "Manually Upgrading Unconverted Code" all'indirizzo

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dv_jlca/html/vbtskmanuallyupgradingnonconvertedcode.asp

Per un elenco degli errori, avvisi e problemi, vedere "Java Language Errors, Warnings, and Issues" all'indirizzo http://msdn.microsoft.com/library/en-us/dv_jlca/html/vberrjavallanguageerrorswarningsissues.asp (informazioni in lingua inglese).

Per un elenco degli argomenti della Guida relativi ai metodi e alle classi JSP che non possono essere convertiti automaticamente, vedere "Javax.servlet.jsp Error Messages" all'indirizzo http://msdn.microsoft.com/library/en-us/dv_jlca/html/vberrjavaservletjsperrormessages.asp (informazioni in lingua inglese).

Conversione di applet tramite Conversione guidata linguaggio Java

Il processo di conversione delle applet tramite Conversione guidata linguaggio Java è suddiviso in due fasi. La prima prevede la conversione del codice dell'applet vera e propria in C# tramite Conversione guidata linguaggio Java. La seconda, invece, prevede la conversione della pagina HTML che fa riferimento all'applet nella versione .NET equivalente. Per convertire il codice dell'applet in C#, è possibile utilizzare la stessa procedura descritta per la conversione delle applicazioni. Il codice dell'applet verrà convertito in un progetto libreria di classi.

Dopo aver corretto eventuali problemi di conversione rilevati nell'applet convertita, si otterrà un controllo Windows Forms (WFC). È possibile ospitare i controlli utente convertiti in Internet Explorer proprio come se si trattasse di un'applet. I controlli ospitati vengono dichiarati nelle pagine HTML con il tag <OBJECT> anziché con il tag <APPLET>. Per identificare il controllo, è necessario utilizzare l'attributo CLASSID specificando il percorso e il nome completo del controllo separati dal simbolo di cancelletto (#), come indicato nell'esempio seguente:

```
<OBJECT id="mioControllo"
classid="http:LibreriaControlli1.dll#LibreriaControlli1.mioControllo"
VIEWASTEXT></OBJECT>
```

Per una visualizzazione corretta del controllo, è necessario che il file DLL che lo contiene si trovi nella stessa directory virtuale della pagina Web in cui viene visualizzato il controllo oppure che il file DLL sia installato nella cache dell'assembly globale.

Aggiornamento della pagina HTML

I controlli WFC non supportano il tag <APPLET> obsoleto o la sintassi del tag <OBJECT> utilizzata da MSJVM e da altre macchine virtuali per l'esecuzione delle applet Java in Internet Explorer. Per eseguire l'applet convertita nel browser, è necessario innanzitutto modificare la pagina Web per utilizzare la sintassi del tag <Object> .NET. Questa operazione può essere eseguita in modo automatico o manuale.

Il modo più semplice per aggiornare i riferimenti alle pagine Web è quello di impostare Conversione guidata linguaggio Java per l'aggiornamento automatico dei riferimenti. Se si dispone di un'applet di Visual J++ in un progetto HTML e si decide di eseguirne la conversione tramite il file di Visual J++, il file HTML associato verrà analizzato automaticamente da Conversione guidata linguaggio Java e, se necessario, verranno modificati tutti i riferimenti ai tag <APPLET>. Sarà quindi possibile copiare e incollare il testo dei tag in tutte le altre pagine che fanno riferimento alla stessa applet.

Se non si dispone di un progetto di Visual J++ per l'applet o non si intende utilizzarlo per la conversione, è possibile convertire i riferimenti HTML manualmente. Quello che segue è un esempio della sintassi del tag <APPLET> utilizzata nelle pagine HTML che includono estensioni MSJVM:

```
<APPLET
CODE = fileApplet
CODEBASE = URLcodebase
WIDTH= pixel
HEIGHT= pixel
VSPACE= pixel
HSPACE= pixel
ALIGN= allineamento
```

```

ID = nomeIstanzaApplet
ARCHIVE = elencoArchivio
ALT = testoAlternativo
>
<PARAM NAME = FireScriptEvents VALUE = True>
<PARAM NAME = cabbase VALUE = nomeFileCab>
<PARAM NAME = cabinets VALUE = nomiFileCab>
<PARAM NAME = useslibrary VALUE = NomeDescrittivoDU>
<PARAM NAME = useslibrarycodebase VALUE = NomeFileDU>
<PARAM NAME = useslibraryversion VALUE= NumeroVersioneDU>
<PARAM NAME = namespace VALUE = spaziodeinomiApplicazione>
<PARAM NAME = attributoApplet1 VALUE = valore>
<PARAM NAME = attributoApplet2 VALUE = valore>
. . .
alternateHTML
</APPLET>

```

La sintassi supportata dai controlli Windows Form è la seguente:

```

<OBJECT
CLASSID="http:<NomeFileOutput>#<NomeClasse>"
WIDTH= pixel
HEIGHT= pixel
ID=nomeIstanzaControlloBrowser
ALIGN= allineamento
HSPACE= pixel
VSPACE= pixel
>
<PARAM NAME = attributo1 VALUE = valore>
<PARAM NAME = attributo2 VALUE = valore>
. . .
alternateHTML
</OBJECT>

```

Nell'esempio precedente di sintassi WFC, **CLASSID** rappresenta la stringa CLASSID contenente il nome del file di output e il nome della classe dei controlli Windows Forms. È possibile trovare entrambe queste impostazioni nella finestra di dialogo Proprietà progetto. Per aprire la finestra di dialogo Proprietà progetto, fare clic con il pulsante destro del mouse sul progetto applet convertito in Esplora soluzioni e scegliere "Proprietà".

4

Altre tecnologie di rendering

Per la conversione sono disponibili anche altre opzioni, tra le quali tecnologie di rendering alternative come DHTML, ECMAScript, VBScript, JScript e Flash. Queste tecnologie consentono di implementare gli elementi di esplorazione più semplici (ad esempio menu, effetti di attivazione, calcolatrici semplici e client di chat) e possono rappresentare un'alternativa valida per la conversione di applet di complessità medio-bassa progettate per operazioni di esplorazione e di interfaccia utente.

Nota Benché Microsoft non sia in grado di garantire la sicurezza e l'affidabilità delle soluzioni offerte da altre società, sono disponibili anche soluzioni alternative non Microsoft. Ai clienti che scelgono di valutare queste soluzioni si consiglia di eseguire test accurati prima di scegliere diversi percorsi di migrazione.

DHTML e script sul lato client

Nella presente guida, l'acronimo DHTML (Dynamic HTML) viene utilizzato in senso ampio per fare riferimento a utilizzi combinati di CSS, script e HTML nel modello DOM (Document Object Model). Se utilizzata in modo accurato, la tecnologia DHTML offre diversi vantaggi in quanto rappresenta una soluzione flessibile e a basso overhead, adattabile a browser e piattaforme diverse. Poiché non è richiesto alcun plug-in o download, l'interpretazione e il rendering degli strumenti Web vengono eseguiti nel browser anziché in una macchina virtuale o in un plug-in. Molti siti Web offrono già codice a basso costo o disponibile pubblicamente per l'implementazione di semplici funzionalità di interfaccia quali menu a discesa, dispositivi di scorrimento del testo, orologi, pulsanti di attivazione, nonché utilità più complesse quali calcolatrici, client chat ed editor.

Per quanto riguarda gli svantaggi, la tecnologia DHTML richiede un uso accurato per garantirne la compatibilità in browser diversi ed è pertanto necessario eseguire test dettagliati nelle piattaforme e nei browser appropriati. Poiché la tecnologia DHTML si basa principalmente sulle funzionalità del linguaggio di script e sull'implementazione dei fogli di stile, potrebbero verificarsi dei problemi di compatibilità con il browser e la piattaforma. I browser della versione 6.0 gestiscono i fogli di stile abbastanza bene e il linguaggio di script (ECMAScript ed estensioni) include un gruppo di funzioni di base che risultano affidabili nella maggior parte dei browser.

Nei siti elencati di seguito sono disponibili alcuni esempi dei vari effetti che possono essere implementati in DHTML. Se gli esempi includono effetti simili alle proprie esigenze, sarà possibile utilizzarli come punto di partenza per le implementazioni.

Discussione sull'utilizzo di DHTML in Internet Explorer:
<http://msdn.microsoft.com/ie> (informazioni in lingua inglese)

Vari esempi, inclusi un orologio e un client chat:

<http://msdn.microsoft.com/downloads/samples/internet/default.asp> (informazioni in lingua inglese)

Siti non Microsoft (informazioni in lingua inglese):

<http://www.codetoad.com/dhtml/>

<http://simplythebest.net>

<http://www.dynamicdrive.com/dynamicindex1>

Nota Microsoft non offre alcuna garanzia del corretto funzionamento di questi esempi nell'ambiente degli utenti. Per implementare soluzioni simili, è consigliabile testare accuratamente tutti gli elementi per essere certi che corrispondano alle effettive esigenze.

ECMAScript e JScript

JavaScript era in origine la versione di Netscape di ECMAScript (European Computer Manufacturers Association), una specifica di linguaggio standard, mentre JScript è la versione Microsoft di ECMAScript. JScript .NET è la nuova generazione dell'implementazione Microsoft del linguaggio ECMA 262 e riunisce le funzionalità esistenti di JScript con le migliori funzionalità dei linguaggi basati su classi.

In molti casi, le funzionalità delle applet Java possono essere realizzate tramite il linguaggio ECMAScript. È sufficiente eseguire una ricerca di JavaScript in Internet per ottenere una vasta gamma di risorse, molte delle quali includono script pronti per l'uso che potrebbero rappresentare validi sostituti del codice di applet Java.

Nota Non è disponibile un metodo di conversione diretto per i nuovi linguaggi di script. In questo caso è necessario riscrivere il codice Java nel linguaggio appropriato. Per ulteriori informazioni (in lingua inglese) sulle tecnologie di scripting Windows, inclusi JScript e VBScript, vedere:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vtoriMicrosoftWindowsScriptTechnologies.asp>

Flash

Flash è un altro esempio di tecnologia di rendering in grado di offrire le funzionalità di determinate applet Java che attualmente dipendono da MSJVM. In alcuni casi, Flash può anche consentire un'estensione delle funzionalità di queste applet. Flash è un plug-in client progettato per il rendering di grafica e animazioni interattive.

Come per qualsiasi altra tecnologia di rendering, la migrazione a Flash non è un processo automatico e richiede la riscrittura del codice esistente oltre a conoscenze approfondite del linguaggio di script associato (ActionScript). Flash non è un prodotto supportato da Microsoft. Pertanto, se si sceglie questa tecnologia come soluzione per la migrazione sarà necessario valutare le opzioni di supporto disponibili dai fornitori appropriati.

Nota Per ulteriori informazioni (in lingua inglese) su Flash, le funzionalità del prodotto e altre considerazioni per l'implementazione, visitare il sito Web all'indirizzo: <http://www.macromedia.com/software/flash/>

5

Passaggio a un altro ambiente di runtime Java

Nota Microsoft non offre alcuna garanzia del corretto funzionamento di ambienti di runtime Java alternativi nell'ambiente degli utenti. È consigliabile testare accuratamente tutti gli ambienti di runtime Java durante le fasi di implementazione per assicurarsi che soddisfino tutte le esigenze specifiche.

Molti fornitori di computer hanno annunciato la distribuzione di ambienti di runtime Java (JRE) alternativi quando Microsoft non potrà più supportare MSJVM. La scelta di questo tipo di soluzione dipende dalle esigenze specifiche di ogni azienda, a livello di compatibilità, affidabilità, versioni e manutenzione.

Anche se molti fornitori dichiarano che i propri ambienti di runtime Java sono conformi alle specifiche Java pertinenti, ciò non implica che tutti gli ambienti di runtime Java siano necessariamente adatti a soddisfare le proprie esigenze. Non tutte le implementazioni delle specifiche Java sono identiche e le differenze potrebbero causare difficoltà.

È possibile che sia necessario trasferire il codice Visual J++ esistente per poterlo utilizzare con un altro ambiente di runtime Java. L'utilizzo delle estensioni Microsoft può essere identificato facilmente nei file di origine J++ correnti. Per trovare i file di origine Java che utilizzano la direttiva @com, è possibile utilizzare il comando CMD.EXE del form:

```
find /I "@com" c:\src\*.*
```

Con alcuni comandi simili è possibile trovare altre estensioni, ad esempio le direttive @dll e @security. Dopo aver identificato l'utilizzo delle estensioni, è possibile valutare gli effetti della loro rimozione.

L'utilizzo o la rimozione delle estensioni Microsoft non comporta alcuna differenza sintattica nel codice Java perché le direttive sono incluse nei commenti. È possibile disattivare le estensioni con le opzioni del comando javac, in modo da poter confrontare facilmente i risultati dell'esecuzione dell'applicazione con e senza le estensioni.

Per la distribuzione di ambienti di runtime Java alternativi in un'organizzazione si utilizzano gli stessi strumenti normalmente utilizzati per installare nuove applicazioni nei computer degli utenti. Per ulteriori informazioni sulla corretta installazione di un ambiente di runtime Java alternativo, consultare la documentazione del fornitore. Nel caso si scelga di installare un ambiente di runtime Java alternativo, eventuali problemi di supporto Java non potranno essere risolti da Microsoft e dovranno essere gestiti in base agli specifici accordi di licenza e assistenza stipulati con il fornitore dell'ambiente di runtime Java alternativo.

Quello che segue è un elenco parziale degli ambienti di runtime Java alternativi disponibili, che non intende essere in alcun modo un elenco esaustivo di **tutti** gli ambienti alternativi disponibili:

Ambiente di runtime Java Sun per Windows:

http://www.java.com/en/download/windows_automatic.jsp (informazioni in lingua inglese)

Ambiente di runtime Java IBM (incluso in IBM WebSphere SDK per servizi Web):

<http://www-106.ibm.com/developerworks/webservices/wsdk/> (informazioni in lingua inglese)

BEA WebLogic JRockit:

<http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/jrockit>
(informazioni in lingua inglese)