

COMPUTER

GRAPHICS

 & PUBLISHING

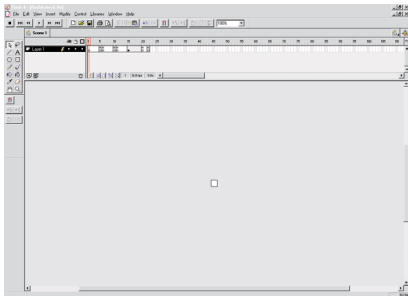
Uno dei più grossi problemi nel caso di plug-in come *Flash* è il riconoscimento delle capacità software dei navigatori che si connettono. La Macromedia ha parzialmente risolto questo problema studiando soluzioni *Javascript* per interrogare il browser degli utenti, che vengono automaticamente scritte da *Flash* nel momento in cui si esporta un filmato all'interno di una pagina *HTML*. Purtroppo, sebbene lo sforzo della Macromedia sia apprezzabile, le tecnologie

è impossibile, ma con *Flash* si può. Il meccanismo è quasi banale: si inseriscono delle azioni all'interno dei filmati *Flash* e se queste azioni vengono eseguite allora significa che il plug-in è presente, altrimenti vuol dire che il navigatore ne è sprovvisto. In realtà, vedremo che la cosa è molto più elaborata e interessante. In caso di presenza del plug-in, si può infatti capire di che versione si tratta e si possono quindi redirezionare le persone a delle pagine che contengono le informazioni appropriate per l'upgra-

Scoprire la presenza di Flash con Flash

Realizziamo un filmato per interrogare i browser degli utenti sulla presenza di plug-in

di scripting hanno sempre avuto parecchi problemi. L'esecuzione di codice *Javascript* può infatti essere disabilitata dagli utenti e poi ci sono tantissime tipologie di navigatori che rispondono in modo molto diverso agli script. Per questo motivo, il controllo dell'ambiente di navigazione degli utenti è sempre stato un argomento che ogni tanto faceva la sua apparizione sui forum/newsgroup della Macromedia. Il "lavoro" di tutte queste persone è stato raccolto e condensato in diversi articoli tecnici e dai gestori di alcuni siti dedicati a *Flash*, quindi si può affermare che ciò che state per leggere è il risultato di un interessante lavoro di collaborazione.



▲ **Figura 1:** filmato di base di dimensioni 18 x 18 pixel con tutti i keyframe necessari

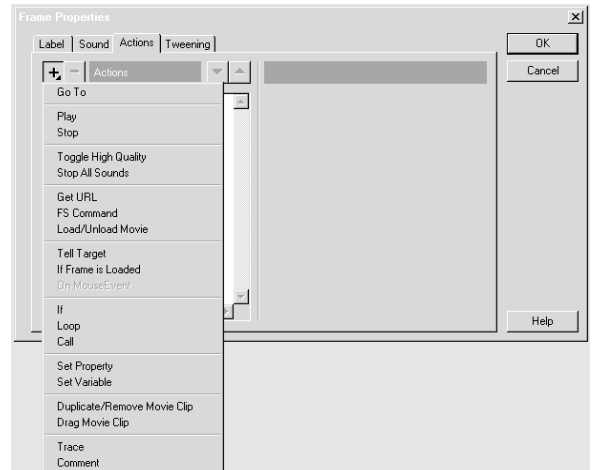
Per approcciare questo tutorial, bisogna necessariamente avere una conoscenza di base delle caratteristiche di *Flash*; per questo potete fare tranquillamente riferimento ai numerosi articoli apparsi sui precedenti numeri della nostra rivista, sia sotto forma di recensioni sia di tutorial. L'ultima premessa riguarda il fatto che prima d'iniziare vi consiglio vivamente di scaricare il materiale di supporto per questo tutorial dal CD-ROM allegato a questo numero oppure dalla pagina *Internet* www.ihf.it/supporto2.htm, questo renderà sicuramente più chiari tutti i passaggi e più fluida la lettura.

È arrivata l'ora d'iniziare

Come si fa a scoprire l'esistenza di un plug-in senza chiedere attraverso linguaggi di scripting la presenza dello stesso? Nella maggioranza dei casi, questa operazione

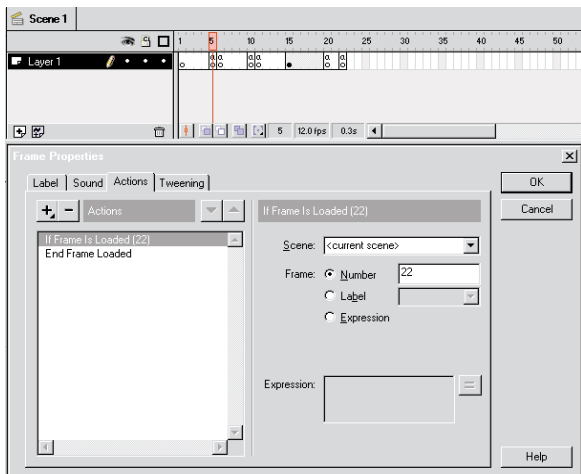
de. Il "ridirezionamento", che in *Flash* si realizza attraverso l'utilizzo dell'azione "GetURL", sarà proprio il tipo di azione che utilizzeremo; ogni azione GetURL sarà chiamata utilizzando una caratteristica specifica di una versione di *Flash*.

Vediamo l'ordine logico del filmato che stiamo per creare:



▲ **Figura 2:** il menu delle azioni. Per raggiungerlo, fate doppio click sul frame di vostro interesse ed entrate nella tab "Actions". Selezionate l'azione di vostro interesse, ne dovrete poi impostare le opzioni in base alle vostre esigenze

1) Preload. Inseriamo all'inizio del filmato un precaricamento (frame 5 e 6). Questo ci assicura che tutti i frame con le rispettive azioni siano presenti prima dell'esecuzione del filmato. Questo passo è indispensabile perché è da questo che deriva l'affidabilità di tutto il filmato. La sequenza delle azioni di ridirezionamento dev'essere infatti esattamente ordinata perché abbia pieno successo (**Figura 2, 3 e 4**).



◀ **Figura 3:** frame 5, questa è l'azione "If frame is Loaded" necessaria a gestire il precaricamento del filmato

2) Controllo della presenza di *Flash 4*. Inseriamo un'azione GetURL sulla timeline principale del filmato (frame 10 e 11), impostiamo il valore della url a un'espressione. Le espressioni non sono interpretate da *Flash 3* o *Flash 2*, quindi questa azione GetURL non sarà presa in considerazione da questi player.

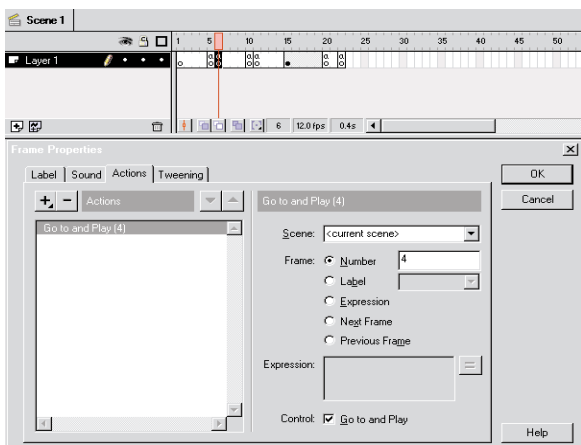
3) Controllo della presenza di *Flash 3*. Inseriamo una seconda azione GetURL all'interno di un simbolo filmato (movie clip) e inseriamo il simbolo all'interno della timeline del filmato principale (frame 15). I simboli di tipo filmato sono ignorati da *Flash 2*.

4) Controllo della presenza di *Flash 2*. Inseriamo una terza azione GetURL sulla timeline del filmato principale (frame 2) che sarà eseguita se il plug-in *Flash 2* è installato e se i primi due ridirezionamenti hanno fallito.

5) No-Flash: se l'utente non ha *Flash* installato le azioni di ridirezionamento non avvengono e il browser farà una delle seguenti cose: le versioni 4.0 di *Netscape* e di *IE* avvertono l'utente che c'è del contenuto *Flash* e gli chiedono d'installare il plug-in (in *Netscape* viene fornito un link al Download Center, mentre *IE* esegue un'autoinstallazione); nei browser più vecchi, il movie può essere ignorato oppure viene visualizzata al suo posto un'icona di plug-in rotta, per evitare che l'utente rimanga su una pagina senza contenuti. All'interno della pagina *HTML*, utilizziamo una tag di refresh per portarli su una pagina non-Flash.

Come tradurre tutto in pratica

Preparate un filmato *Flash* con 22 frame e impostate la



▲ **Figura 4:** frame 5, l'azione Go To and Play(7) esce dal loop in cui si trova il filmato finché non è stato caricato completamente

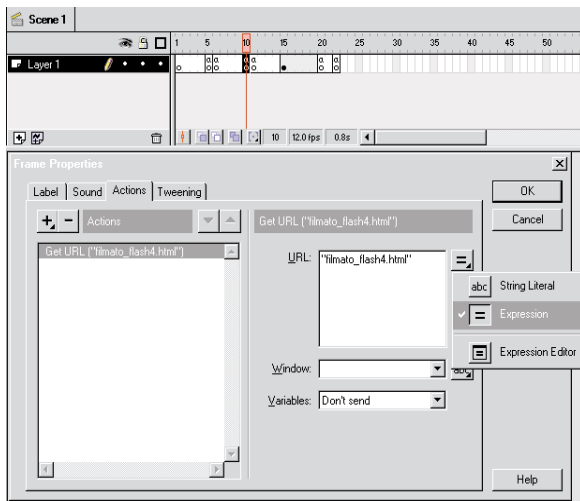
grandezza del filmato alle dimensioni più piccole possibili (quadrato di 18 pixel). Create quindi un keyframe vuoto (F6) sui frame 5, 6, 10, 11, 15, 20, e 22 del filmato. Sull'ultimo keyframe del filmato (frame 22) aggiungete un'azione di stop, questo previene il loop infinito del filmato in alcuni player. Con queste operazioni gli elementi di base del filmato sono pronti (**Figura 1**). Per tutte le operazioni che seguiranno, si lavora principalmente con le azioni dei frame. Per inserire queste azioni basta fare un doppio click sul frame nel quale si vogliono inserire le azioni, selezionare l'azione desiderata dal menu in alto a sinistra della Tab actions delle Frame properties (quello con l'icona "+"), e inserire i valori come spiegato facendo riferimento anche alle figure (**Figura 2**). Adesso possiamo iniziare a ricostruire in pratica la struttura logica già spiegata:

1) Preload. Come abbiamo già detto, il precaricamento di tutto il filmato assicura che le varie azioni getURL siano eseguite in modo corretto. Al frame 5 della timeline aggiungiamo un'azione "If frame is Loaded" (**Figura 3**). Nel campo testo "numero" delle opzioni del frame per l'azione inseriamo il numero 22; in questo modo, le azioni scritte all'interno del blocco "If Frame is Loaded" saranno eseguite solo dopo che l'intero filmato sarà caricato. Con la riga "if frame is loaded (22)" selezionata, aggiungete un'azione Goto, checkate "Go to and Play" per l'opzione "control" e nel campo testo "numero" delle opzioni del frame per l'azione inserite il numero 7 (**Figura 4**). Per finire il preloader, aggiungete un'azione "go to" sul frame 6 del filmato, checkate "Go to and Play" dell'opzione "control", nel campo testo "numero" delle opzioni del frame per l'azione inserite il numero 4 (**Figura 5**). Questa azione sul frame 6 rimanda indietro il filmato al frame numero 4, precedente a quello di controllo del caricamento (numero 5); quest'ultimo quando il filmato è completamente caricato spedisce al frame 7, successivo a quello che manda in loop il filmato (numero 6).

2) Controllo della presenza di *Flash 4*. Sul frame 10 del filmato aggiungete un'azione getURL che punti alla pagina che contiene i vostri contenuti *Flash 4* (filmato_flash4.html) e assicuratevi d'impostare attraverso il menu a destra della finestra di testo l'icona "=" rappresentativa delle espressioni (**Figura 6**). È un'operazione molto importante, perché solo *Flash 4* è in grado di valutare le espressioni. Gli altri player non possono eseguire l'operazione e quindi la logica di "investigazione" del filmato

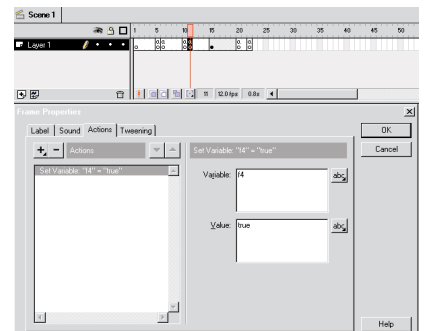
◀ **Figura 5:** frame 6, l'azione Go To and Play(4) è quella che manda il filmato in loop

► **Figura 6:** frame 10, ecco la prima azione di ridirezionamento GetURL ("filmato_flash4.html"). Serve a controllare la presenza di Flash 4, se il plug-in è presente l'azione viene letta e spedisce l'utente alla pagina dedicata al plug-in versione 4



mente tutto il meccanismo. Aggiungiamo quindi del codice alle righe appena scritte (Figura 9) in modo che appaiano così:

```
Set Variable:
"f4" = "true"
If (f4 eq "true")
    Stop
Else
    Play
End If
```

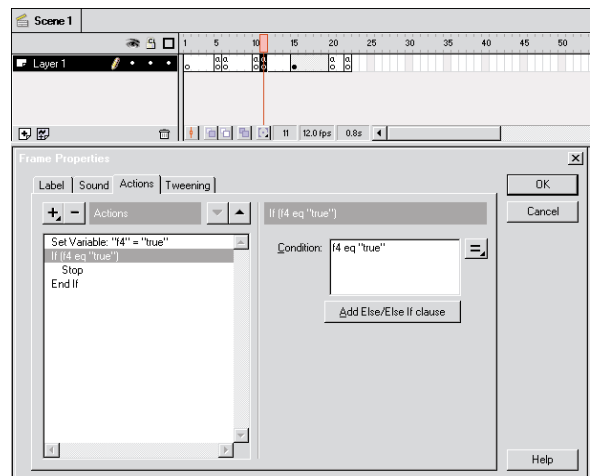


▲ **Figura 7:** frame 11, alla variabile f4 viene assegnato il valore true (vero). Questo assegnamento funziona solo se il plug-in installato sulla macchina dell'utente è in versione 4

► **Figura 8:** frame 11, se la variabile f4 è uguale a true, quindi il plug-in dell'utente è in versione 4, allora il filmato si ferma per dare tempo all'azione GetURL del frame 10 di fare il suo dovere

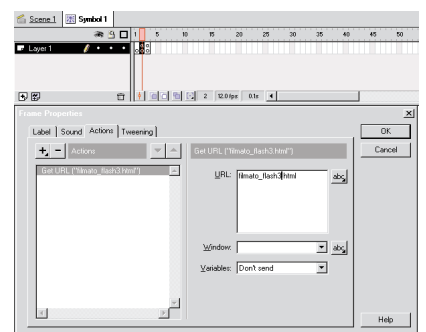
è salva. Inserite il testo della vostra url fra virgolette in modo che sia interpretato come stringa e non come nome di variabile. Le operazioni non sono finite qui, bisogna aggiungere infatti dell'altro codice actionscript che blocchi il filmato nel caso che Flash 4 sia attivo. Se non lo fate, è possibile che il filmato esegua le successive azioni getURL prima che il filmato Flash 4 sia caricato. Nel frame 11 impostiamo una variabile chiamata f4 a True (Figura 7) e se la variabile viene valutata True eseguiamo un'azione di stop attraverso action script (Figura 8). È abbastanza ovvio che se il plug-in è versione 4 (quindi il setup della variabile funziona) la condizione (f4 eq "true") sarà sempre vera. Questo è il codice da aggiungere.

```
Set Variable: "f4" = "true"
If (f4 eq "true")
    Stop
End If
```



In questo modo, Flash 3 eseguirà eventualmente l'azione di play nell'istruzione else che diventa l'ultima azione del blocco actionscript.

3) Controllo della presenza di Flash 3: Se la prima azione leggibile solo dal plug-in versione 4 viene saltata, allora il filmato continuerà ad andare avanti e raggiungerà lo stadio successivo, quello in cui tenteremo di vedere se gli utenti



▲ **Figura 10:** filmato Symbol 1, sul secondo frame contiene l'azione GetURL ("filmato_flash3.html"); se l'azione di ridirezionamento del 4 non ha funzionato si tenta di vedere se c'è il plug-in versione 3, se così è l'azione viene letta e spedisce l'utente alla pagina dedicata a Flash 3

hanno la versione 3 del plug-in. Per fare ciò, creiamo un simbolo filmato (Insert/New Symbol...) con tre keyframe, sul secondo frame di questo filmato aggiungiamo un'azione getURL che punta a una pagina che contiene il vostro filmato Flash 3 (Figura 10). Come abbiamo detto precedentemente, inseriamo questa azione getURL al-

Adesso accade una cosa curiosa. Flash 3 non dovrebbe riuscire a leggere queste istruzioni quindi dovrebbe semplicemente saltarle. Ma non è così. Infatti, per motivi di evoluzione del formato (dalla versione 3 alla 4) succede che Flash 3 attualmente esegue l'ultima azione trovata all'interno di un blocco d'istruzioni actionscript. Cosa significa? Semplicemente che l'azione di stop nell'istruzione if viene letta anche da Flash 3 bloccando inevitabil-



l'interno di questo filmato, cosicché sia visibile solo ai player successivi a *Flash 3*; ciò previene la possibilità che utenti *Flash 2* siano ridirezionati ai contenuti per il plug-in *versione 3*. Sul frame 3 del filmato aggiungete un'azione Tell Target per dire al filmato principale di fermarsi (**Figura 11**). Ciò evita che sia eseguita l'ultima azione `getUrl` prima che il contenuto del filmato per *Flash3* sia caricato. Il codice che vediamo è:

```
Begin Tell Target ("")
    Stop
End Tell Target
```

Piazzate il piccolo filmato appena realizzato sullo stage del filmato principale al quindicesimo frame (**Figura 12**).

4) Controllo della presenza di *Flash 2*. Dopo che il filmato ha superato i ridirezionamenti precedenti, è arriva-

una pagina non *Flash* dopo 20 secondi (il tempo può variare in base alla lunghezza del filmato "investigatore" e alla velocità del server su cui è ospitato, per sicurezza conviene quindi fare un po' di prove). Tutti i più conosciuti editor *HTML* hanno degli strumenti dedicati all'inserimento di queste tag. Se siete sprovvisti di uno di questi software, oppure se volete sapere ciò che state scrivendo nella pagina, andate nella sezione `head` della pagina (quella compresa fra le tag `<head>` e `</head>`) e inserite questo codice:

```
<META HTTP-EQUIV="REFRESH" CONTENT="20;
    URL=pagina_senza_flash.html">
```

potete scegliere la quantità di tempo che la meta tag deve aspettare prima di rispedire l'utente alla pagina non-*Flash*, sostituendo il valore 20 con il numero desiderato.

Avete bisogno di controllare ancora un paio di cose nel codice *HTML*: in *Netscape* fate attenzione a non includere

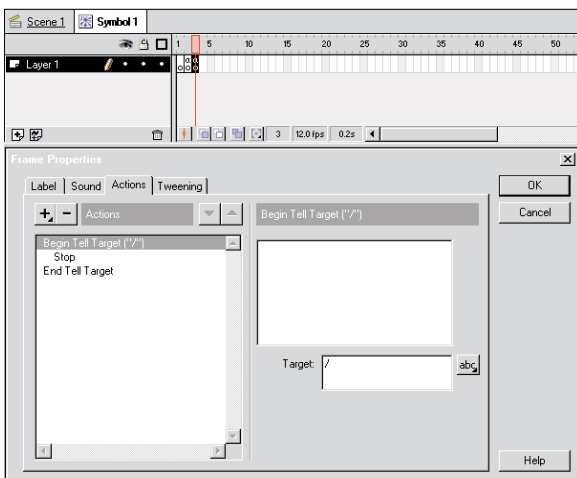
il parametro `"swLiveConnect=true"` nella tag di `embed` del vostro `movie`, altrimenti i navigatori avranno una pausa variabile all'incirca tra i 5 e i 20 secondi mentre caricano *Java*; se nel vostro sito utilizzate gli `FSCOMMANDS` di *Flash*, potrete abilitare `liveconnect` sulle pagine seguenti.

Concludendo

Ci sono ancora un paio di cose da ricordare. Innanzitutto, assicuratevi che il colore del vostro filmato *Flash* e del vostro documento *HTML* siano gli stessi, poi rendete il processo più naturale agli occhi degli utenti aggiungendo un messaggio "loading...". Per finire, non dimenticate di spiegare il processo di upgrade agli utenti *Flash 2* e *3* nei filmati *Flash* alternativi e ricordatevi di descrivere sulla pagina non-*Flash* come ottenere il plug-in *Flash*.

Come potete notare, questi ultimi passaggi del tutorial sono sprovvisti d'immagini, questo perché per essere chiari sarebbero servite troppe figure che non sarebbero comunque state sufficientemente esplicative. Rimando quindi nuovamente ai file di supporto per vedere le pagine *HTML* e tutti gli altri elementi qui spiegati, ma non visualizzati.

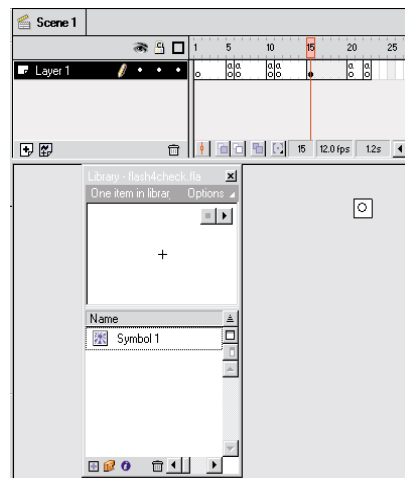
Dopo aver completato tutti questi passi, il vostro filmato per il controllo della presenza dei plug-in è pronto per essere messo online.



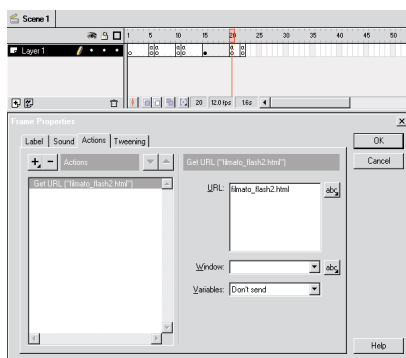
▲ Figura 11: filmato *Symbol 1*, in modo simile a quanto fatto nel frame 11, diciamo al filmato principale di fermarsi per permettere eventualmente all'azione `GetURL` per *Flash 3* di essere completata

to il momento di vedere se c'è almeno il plug-in della *versione 2*. Sul frame 20 del filmato principale aggiungiamo un'azione `getUrl` contenente l'indirizzo della pagina che contiene il filmato *Flash2* (**Figura 13**). In questo caso, diversamente da quanto abbiamo fatto per controllare la presenza del plug-in *versione 4*, nel menu a destra della finestra per l'inserimento dell'indirizzo lasciamo l'icona "abc", che serve per dire a *Flash* che ciò che è scritto nella finestra è testo puro.

5) Non-*Flash*. Completato il filmato che cerca di scoprire la presenza di *Flash*, l'unica operazione che rimane è l'inserimento dello stesso in una pagina *HTML* che contenga una tag di refresh necessaria per spedire l'utente a



◀ Figura 12: frame 15, questo è il frame dove viene inserito il filmato dedicato al ridirezionamento per *Flash 3*



◀ Figura 13: frame 20, l'ultima azione di ridirezionamento avviene solo se le prime due non hanno funzionato. In questo caso, si tratta della versione più semplice dell'azione, `GetURL` ("filmato_flash2.html")