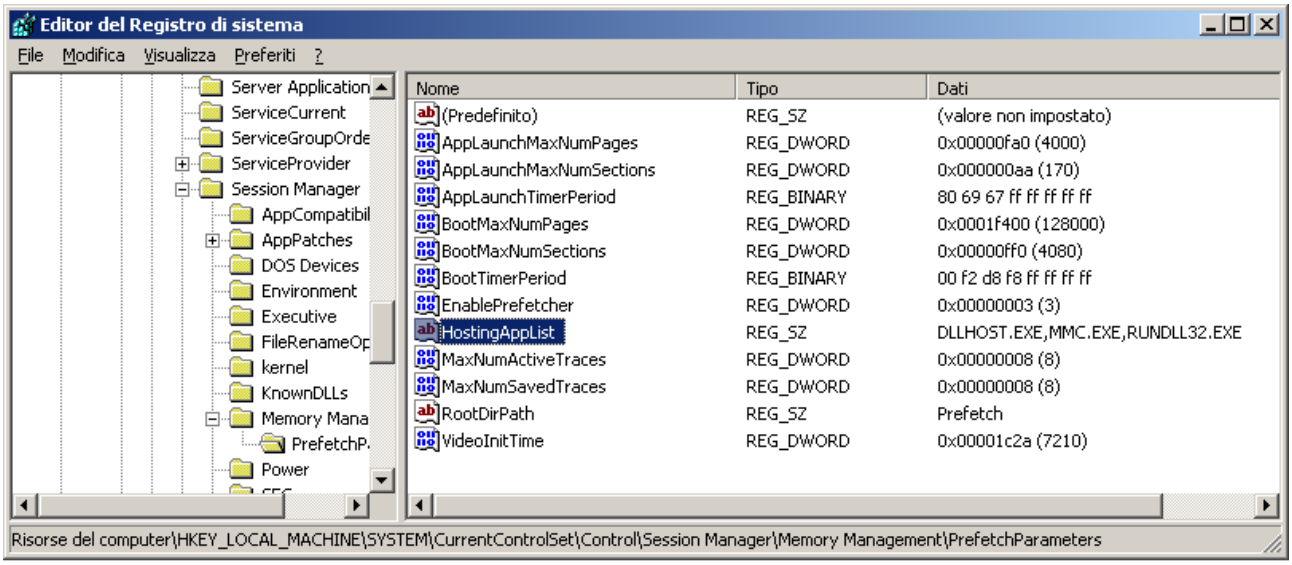
Il **prefetcher**, introdotto in Windows Xp, mira ad accelerare il processo di avvio del sistema, compresa la fase di preparazione degli applicativi.

L'obiettivo viene raggiunto attraverso il monitoraggio dei dati e del codice usati durante l'avvio del sistema e delle applicazioni in modo da sfruttare tali informazioni alla successiva procedura di startup.
Quando il **prefecther** è attivo, il gestore della memoria comunica al codice del **prefetcher,** residente nel Kernel, gli eventuali *page faults* sia quelli che richiedono una lettura dei dati dal disco rigido (*hard faults)* sia quelli che richiedono l'aggiunta di dati già presenti in memoria al *working set* di uno specifico processo (*soft faults*).
Il **prefetcher** tiene monitorati i primi 10 secondi dell'avvio di un applicativo, mentre per la procedura di startup del sistema vengono presi in considerazione 30 secondi a partire dall'inizializzazione della *shell* o 60 secondi a partire dall'inizializzazione dei servizi.
Solo dopo che il prefetcher ha concluso la registrazione della fase di avvio del sistema, comincia a raccogliere informazioni sui *page faults* degli applicativi.

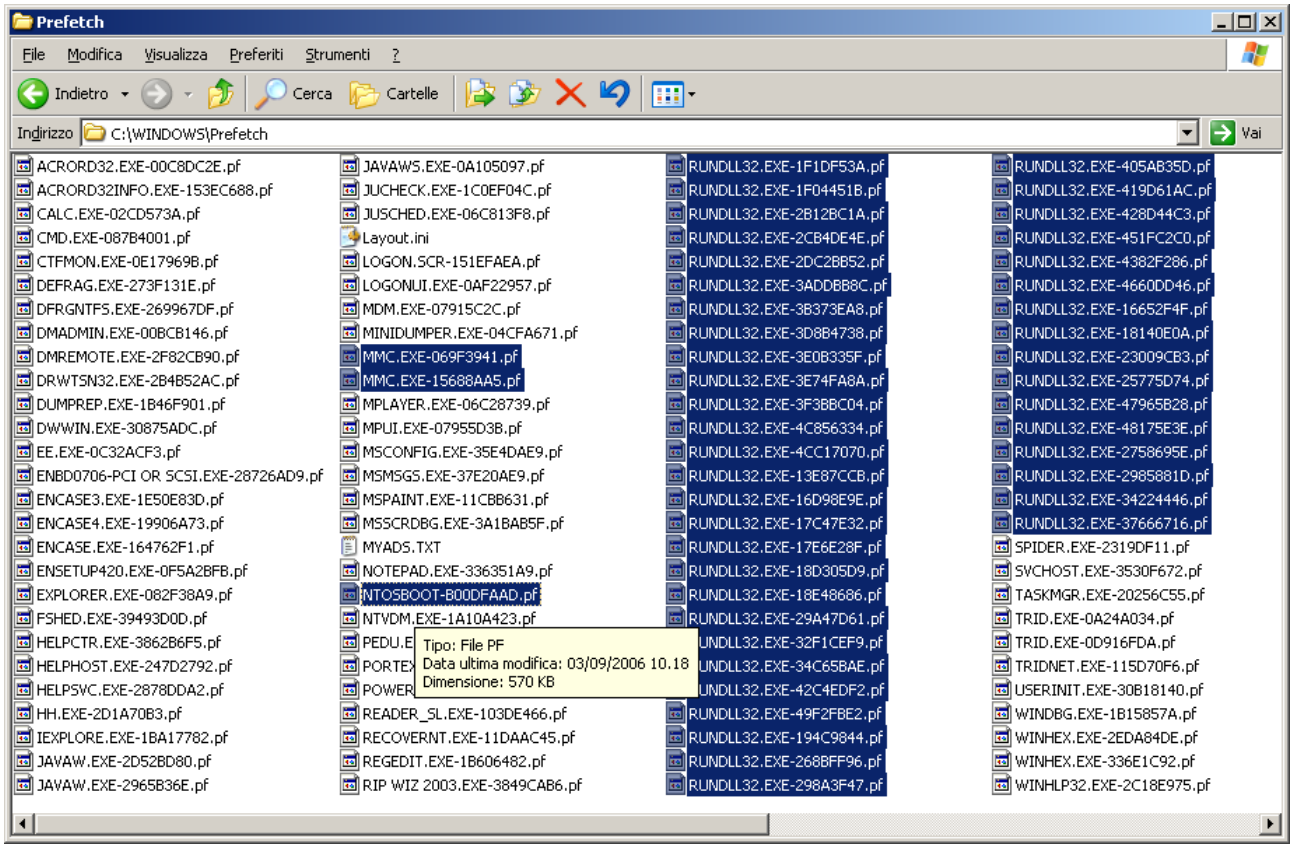
Il nome del file è composto dal nome dell'applicazione monitorata, seguito da un trattino e da una rappresentazione esadecimale dell'hash del percorso del file; inoltre, il file ha estensione **.pf**
Quindi, un esempio potrebbe essere: NOTEPAD.EXE-AF43252301.PF
Ci sono due eccezioni alla suddetta regola per la formazione dei nomi:

1. i file che chiamano in causa altre componenti ad esempio la *Microsot Management Console* (\Windows\System32\Mmc.exe) e *Dllhost* (\Windows\System32\Dllhost.exe). Infatti, in questi casi le componenti aggiunte vengono specificate a linea di comando, e quindi il **prefetcher** indicherà l'hash di tale comando nel nome del file. Avremo perciò, che differenti invocazioni di mmc.exe o dllhost.exe riceveranno dal **prefetcher** nomi diversi a seconda della componente invocata. Il prefetcher conosce la lista degli applicativi che deve trattare come *host* di altre componenti, leggendo il valore della chiave di registro *HostingAppList* localizzata in *HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters.*



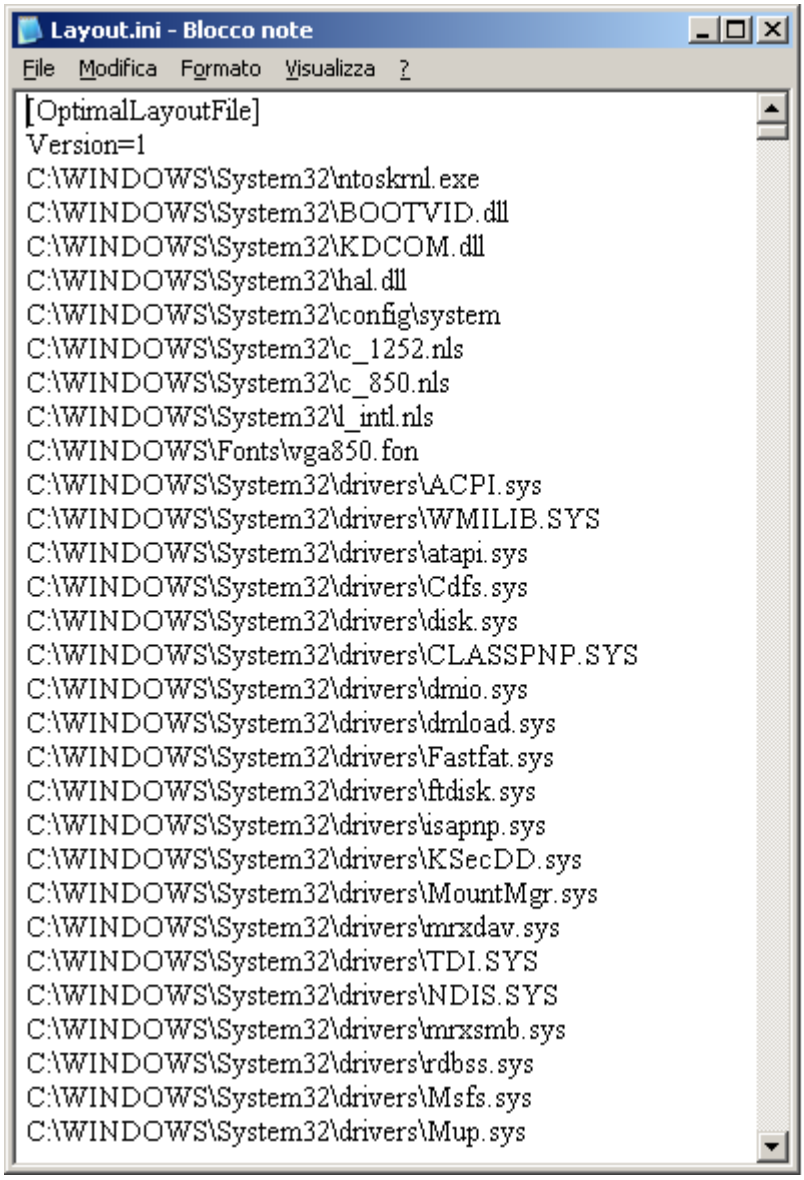
2. il monitoraggio della fase di avvio del sistema, che viene conservato sempre nel file NTOSBOOT-B00DFAAD.PF

I file creati ed utilizzati dal **prefetcher** vengono conservati nel percorso \%systemroot%\prefetch\.
Il **prefetcher** cerca in questa *directory* un file **.pf** per lo scenario in questione in modo da poter preventivamente caricare in memoria dati e codice richiesti dall'applicazione eseguita ed evitare così situazioni di *page fault*.
I file vengono aggiornati di volta in volta apportando i cambiamenti necessari per rappresentare, sempre fedelmente, la situazione di *startup* dell'applicativo.



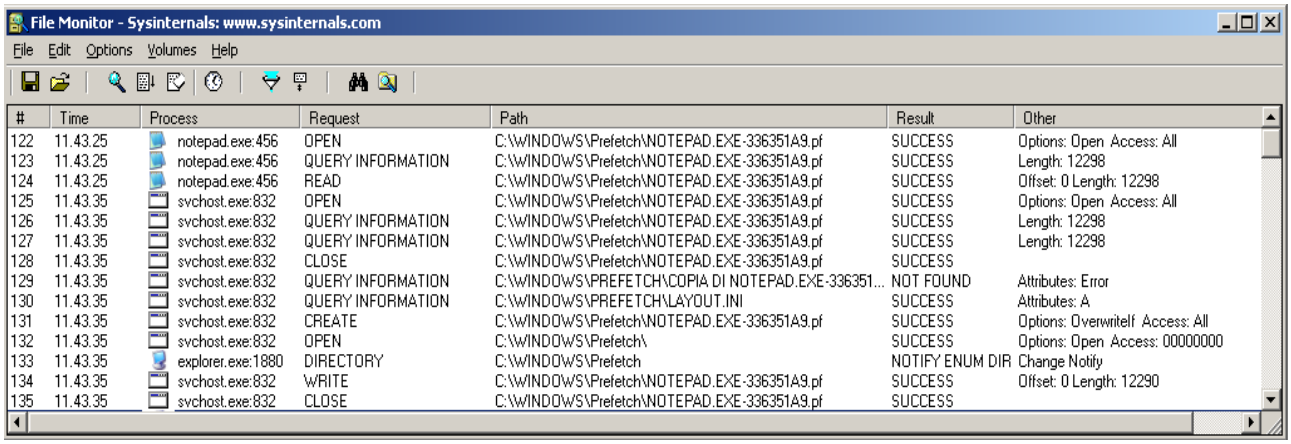
Nell'immagine sopra riportata vediamo il contenuto di una *directory* di **prefetch,** con selezionati i file di cui abbiamo precedentemente trattato.

Un altro file degno di nota è il ***Layout.ini*** (presente nel percorso *\%systemroot%\prefetch*). Infatti, ogni due o tre giorni, durante i periodi di scarso carico del processore, il ***Task Scheduler*** organizza una lista di file e directory in modo che siano richiamate durante la fase di avvio del sistema o degli applicativi. Tale lista viene posta proprio nel file ***Layout.ini***, di cui riportiamo un estratto nell'immagine seguente.



A questo punto, il ***Task Scheduler*** lancia *l'utility* di deframmentazione con una opzione che procede a deframmentare secondo il contenuto dei file. In questo modo, il ***defragmenter*** cerca su ogni volume del disco rigido, aree sufficientemente grandi per ospitare tutti i file elencati nel Layout.ini, in modo che essi si trovino in settori contigui, consentendo così una maggiore efficienza del *prefetching* di Windows.

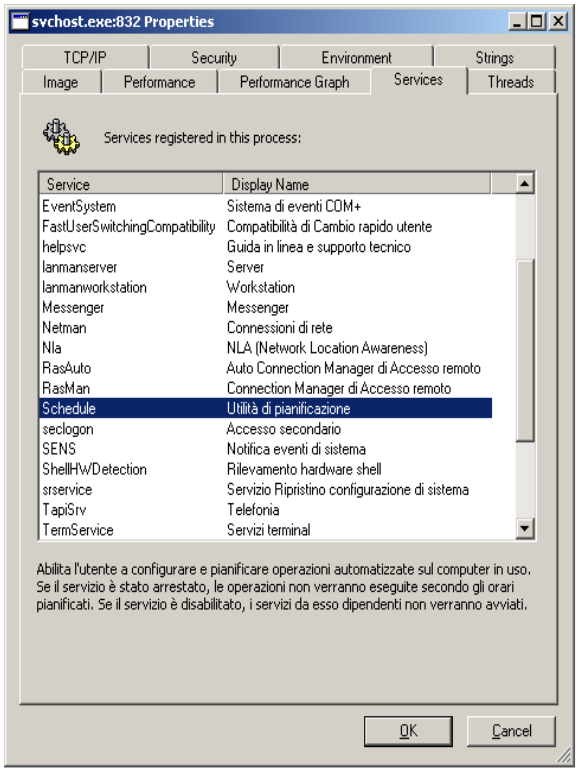
Non ci rimane che vedere all'opera il funzionamento del prefetcher di Windows.
Per fare ciò useremo due *tools* scaricabili gratuitamente da www.sysinternals.com: **Filemon** e **Process Explorer**. Il primo ci permetterà di monitorare quali siano le operazioni effetuate sul *file system* dal prefetcher mentre il secondo verrà usato solamente per analizzare in dettaglio i servizi richiamati dalle operazioni.
Eseguendo Filemon, il programma comincerà a registrare un'enorme quantità di operazioni; sarà perciò necessario impostare un filtro di *include* (CTRL+L) uguale a 'prefetch' in modo da vedere solo ciò che ci interessa.
Lanceremo poi un applicativo qualunque (nel nostro caso Notepad.exe), ed attenderemo una decina di secondi. Quello che otterremo sarà qualcosa di simile all'immagine seguente.



#	Time	Process	Request	Path	Result	Other
122	11.43.25	notepad.exe:456	OPEN	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Options: Open Access: All
123	11.43.25	notepad.exe:456	QUERY INFORMATION	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Length: 12298
124	11.43.25	notepad.exe:456	READ	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Offset: 0 Length: 12298
125	11.43.35	svchost.exe:832	OPEN	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Options: Open Access: All
126	11.43.35	svchost.exe:832	QUERY INFORMATION	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Length: 12298
127	11.43.35	svchost.exe:832	QUERY INFORMATION	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Length: 12298
128	11.43.35	svchost.exe:832	CLOSE	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	
129	11.43.35	svchost.exe:832	QUERY INFORMATION	C:\WINDOWS\PREFETCH\COPIA DI NOTEPAD.EXE-336351...	NOT FOUND	Attributes: Error
130	11.43.35	svchost.exe:832	QUERY INFORMATION	C:\WINDOWS\PREFETCH\LAYOUT.INI	SUCCESS	Attributes: A
131	11.43.35	svchost.exe:832	CREATE	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Options: OverwriteIf Access: All
132	11.43.35	svchost.exe:832	OPEN	C:\WINDOWS\Prefetch\	SUCCESS	Options: Open Access: 00000000
133	11.43.35	explorer.exe:1880	DIRECTORY	C:\WINDOWS\Prefetch	NOTIFY ENUM DIR	Change Notify
134	11.43.35	svchost.exe:832	WRITE	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	Offset: 0 Length: 12290
135	11.43.35	svchost.exe:832	CLOSE	C:\WINDOWS\Prefetch\NOTEPAD.EXE-336351A9.pf	SUCCESS	

Le prime tre voci riportate da Filemon, ci riportano operazioni di **OPEN**, **QUERY INFORMATION** e **READ** sul file NOTEPAD.EXE-336351A9.pf che contiene le indicazioni per il prefetcher al fine di ottimizzare l'avvio di Notepad.
Le rimanenti voci, comparse una decina di secondi più tardi sono richieste di **READ**, **OPEN**, **QUERY**, **CREATE**, **WRITE** e **CLOSE** sempre a carico del file NOTEPAD.EXE-336351A9.pf, ma questa volta il processo richiedente è un non meglio precisato **svchost.exe**.
Sappiamo che su un sistema Microsoft, solitamente si trovano contemporaneamente in memoria più copie (istanze) di **svchost** ognuna però con riferimento ad uno o più servizi.

Per verificare quale sia il servizio operante nel nostro caso (abbiamo visto in precedenza che dovrebbe trattarsi del **Task Scheduler**), utilizziamo Process Explorer, il quale, selezionando il **PID** (Process Identifier) del svchost in questione (832), ci fornisce le seguenti informazioni.



In conclusione, vediamo che tra i servizi che girano sotto il svchost con PID=832, abbiamo proprio il **Task Scheduler** – **Utilità di Pianificazione** che, come spiegato, si occupa delle operazioni di aggoirnamento dei file, oltre che della manutenzione del **Layout.ini**