

## 第八章

# NT Registry 管理

### 本章內容：

- \* 替新使用者帳戶建立預設值
- \* 初始檔案對映的使用
- \* 限制遠端登錄存取
- \* 修正登錄安全存取控制列表 (ACLs)
- \* 使用 SYSKEY 替 HKLM\SAM 加密
- \* 各種好東西
- \* 使用 Resource Kit 工具程式
- \* 使用 NTREGMON 監看 Registry

當你管理一群電腦，不論是一台或多台電腦時，你很快會發覺每天做的工作真是林林總總。你得建立新帳戶、移除舊帳戶、判斷備份磁帶機壞掉的原因... 等等。若你的工作只需依序、計劃性地更新、維護、以及整合系統即可，那該有多好，但這些繁瑣的小工作也是很重要的。本章將介紹你幾個關於管理 NT Registry 的幾個小工作；雖然這些工作中不需要全天候操作，但它們可具有相當重要的存在價值。

## 替新使用者帳戶建立預設值

NT 允許多個使用者帳戶共用一台電腦。它和 DOS 及 Windows 3.x 不同的地方是，NT 提供了（經由 Registry）保留每個使用者個別設定的方法。但舊版 NT 並不允許不同電腦共用這些設定，且無法將使用者的設定資料集中在一個位置。

NT 4.0 是第一個支援使用者設定檔 (user profiles) 概念的 NT 版本。和 Win95 裡的設定檔一樣，NT 4.0 的設定檔會抓取使用者的桌面環境設定、應用程式設定，以及其它喜好設定。這些設定檔可以從一台電腦移到另一台電腦上，所以使用者可以在每台登入的電腦裡擁有相同的自訂環境。除此之外，系統管理員還可以制定這些設定檔，預防使用者更改某些設定，也因此使得管理員可以很容易地設定共享電腦教室及其它設備，在這些共享環境裡，保護機器不被任意亂改是相當重要的。

在安裝 NT 時，系統會利用預設設定檔提供設定給使用者帳戶使用。當新建立的帳戶第一次登入時，預設的設定檔會複製到 HKCU 裡，然後這個新帳戶便會繼承預設的設定。很不幸的是，你無法直接更改預設設定檔裡的設定。你可以使用「系統策略編輯器」(如第 6 章「如何使用系統策略編輯器 (System Policy Editor)」裡所述) 來設定「預設使用者」帳戶，但若你希望更改原則樣本裡不存在的設定 - 例如，預設貨幣格式或 Internet Explorer 儲存的 URL 列表 - 你有兩種選擇。第一種是建立新的原則範本，在其中加入想要的設定，另一種是直接編輯預設使用者的設定檔。

NT 會將預設的使用者設定檔儲存在一個檔案裡。在個別的工作站及伺服器上，設定檔會儲存在 %systemroot%\profiles\Default User。你也可以強制將預設設定檔應用到所有的網域登入，只要將預設設定檔放在網域控制站上的 NETLOGON 共用目錄裡即可。【註】這個檔案必須命名為 Ntuser.dat。在這個檔案裡的所有設定都會被應用到新的使用者帳戶，但不會影響到現存的帳戶。Ntuser.dat 實際上只是一個 Registry hive；當新帳戶第一次登入時，NT 會將這個 hive 的內容複製到 HKCU 裡，然後將更動的部份寫入 HKU 下的適當子機碼裡。在改變初始 hive 裡的內容後，會影響到使用者登入機器上的 HKCU 裡的內容。

因為預設的使用者設定檔只是個 Registry hive，所以你可以使用 RegEdit32 來編輯這個檔案。下面是執行的步驟：

1. 啟動 RegEdit32，打開 HKU 視窗並選擇 HKU 主機碼。
2. 使用「登錄」 「載入 Hive...」選單指令來選擇欲編輯的預設使用者設定檔。你可以直接開啟 %systemroot%\profiles\Default User，或是在可以取得 NTuser.man 的狀況下編輯這個檔案。

---

註 你必須使用位於「系統」 「使用者設定檔」 「複製到...」按鈕，將設定檔從本機搬到網域控制站的 NETLOGON 共用目錄裡，才能達到這項功能。

3. 在 RegEdt32 要求機碼名稱時，產生一個可以記憶 hive 作用的名稱。我通常使用「DefaultUserProfile」。RegEdt32 將會匯入這個 hive 並將它附加到你提供的名稱下。
4. 選擇新建立的 hive 機碼，並使用「安全」 「使用權限...」指令將「Everyone:Read」權限增加給這個機碼及其子機碼。這種做法使得設定檔共用機制可以從預設設定檔裡複製機碼到使用者的 HKCU 裡。
5. 利用 RegEdt32 對新 hive 的子機碼產生必要的改變，所有產生的改變都會儲存在這個 hive 檔裡。
6. 在完成編輯 hive 機碼的動作後，使用「登錄」 「卸除 Hive」指令來卸除 hive。然後，其它的電腦或使用者便無法再對存取你先前產生的改變。

## 初始檔案對映的使用

在第 1 章「NT Registry 簡介」裡描述了 Registry 與其前身 INI 檔案的關係。許多 NT 安裝程式仍然執行 16 位元的 Win3.1 應用程式，而這些程式並不支援 Registry，而且也仍有許多 32 位元的應用程式仍然依賴舊有的 INI 檔案架構。除非是那些希望從微軟公司取得「為 Windows 95 而設計 (Designed for Windows 95)」商標的應用程式，因為使用 Registry 是取得商標的必要需求之一。

因為你無法更新舊式的 16 位元程式，將它們改成使用 Registry，【註】這樣一來，你可能會認為永遠無法脫離 INI 檔的掌握，非得不斷地追蹤、備份，以及保護這堆亂七八糟的 INI 檔不可。但事實完全不是這麼回事！NT 引進一項新特性，稱為“初始檔案對映 (initialization file mapping)”，從現在開始我將只稱呼它為“對映 (mapping)”，這項特性允許你強制那些不使用 Registry 的程式可以在 Registry 裡載入及儲存設定資料，而不是透過 INI 檔案來完成這些工作。

---

註 Ron Petrusha 所著的 Inside the Windows 95 Registry 書中的第 5 章解釋了如何在 16 位元或 DOS 的應用程式裡使用 Win95 的 Registry。

NT 已包含幾個系統元件的對映，這些元件包括 Windows 時鐘桌面附屬應用程式 (clock desk accessory) 32 位元磁帶備份應用程式，以及 RegEdit 32！對映並不只是提供給 16 位元的應用程式使用；更正確地說，它可以提供給任何應用程式使用——不論 16 位元或 32 位元——對映裡並不包含用來讀取或寫入 Registry 資料的程式碼。

當然，對映並不是絕對必要；使用 INI 檔案的應用程式仍然可以在 NT 下正常運作，不一定非得使用對映不可。事實上，除非你清楚地對這些檔案進行對映，否則它們會維持在未對映的狀況，而且繼續使用 INI 檔案。

## 對映如何運作？

NT 會抓取第 1 章裡提到的設定檔 API 常式來完成對映。Windows 應用程式及元件通常會使用這些呼叫來取得並設定 INI 檔案裡的資料，但當其中有一個對映項目時，NT 首先會檢查是否存在一個對映機碼 (mapping key)。若存在這麼一個機碼，而且這個機碼指向一個包含資料的機碼，則所包含的資料會傳回給呼叫程式。若沒有對映機碼，或者若它指向一個空的或不存在的 Registry 機碼，則 NT 將會繼續執行並試著從 INI 檔讀取資料。呼叫程式不會注意到要求的檔案以外的資料。

對映通常會發生在存在對映機碼時。這些機碼會儲存在 HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\IniFileMapping 子機碼下。當你檢視這個機碼時，你將會注意到有許多機碼取名為像 Clock.INI、Win.INI 以及 SYSTEM.INI 這類的名稱。這些機碼將舊版 Win 3.1 型式的 INI 檔裡的 section 與 Registry 裡的機碼連繫起來，這麼一來，若舊版的 Windows 3.1 元件像 Clock 或初期的媒體控制介面 (media controller interface; MCI) 便可以繼續在 Registry 裡找到屬於它們的設定資料。

應用程式的設計者及系統管理員可以自行決定是否要建立新的對映來連繫 INI 檔與 Registry 裡的機碼。這項功能允許你在 Registry 裡將資料移動到它歸屬的位置；只要你找到它的位置，便可以利用本書中所提到的技術編輯、儲存、調整，以及管理它的內容。

下面是實例：一個用戶端 (client) 被授權使用電子郵件應用程式管理數百位使用者。這個應用程式是 32 位元的版本，但它並未使用 Registry。我只需要這個程式建立對映，然後建立系統原則範本 (參第 6 章)，這麼一來它便可以集中控制使用者設定其 mail 客戶的方式。從此皆大歡喜。

## 設定你自己的對映

在理想的狀況下，你電腦裡的所有應用程式都應該是 32 位元的、Registry 接受的，以及 NT 原生的程式。但不幸地，只有那些在 RISC 機碼（無法執行 Win95 或 Win 3.x 程式）上執行 NT 的人才擁有如此考究的待遇。雖然對其餘的我們而言，要增加新對映機碼，迫使 NT 使用 Registry 機碼而不是 INI 檔裡的段落並非難事；這個方法的最大優點是，完全不需更動到那些原本使用 INI 檔的應用程式。

若你已經開啟了一個 INI 檔案，你便知道一個 INI 檔分為好幾個段落。段落的名稱使用方括號括起來，而且它們包含「名稱 / 數值」配對。整個的排列方式如本範例所示，這個範例是從一個虛擬資料安全封包的 INI 檔而來：

```
[Encryption]
DefaultSigAlgorithm=RSAShA1
DefaultEncryptionAlgorithm=DES3-EDE-CBC
WipeFilesWhenDone=1
```

在這個範例裡，“Encryption”是段落名稱，而“DefaultSigAlgorithm”、“DefaultEncryptionAlgorithm”，以及“WipeFilesWhenDone”則為數值名稱。

## 新增對映機碼

你可以將 INI 檔裡的所有段落都與 Registry 機碼建立對映。方法很簡單，只要增加新子機碼到 HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping 裡即可。這個子機碼與對映的 INI 檔必須具有相同的名稱；例如，若你希望重新對映一個名為 ccmail.ini 的檔案，請將具備這個名稱的子機碼加入 IniFileMapping 機碼裡。

若你只增加一個新對映的機碼，而未在這個機碼裡有任何其餘的動作，那麼你所做的工作將不會產生任何作用。因為這個子機碼只能告知 NT 注意具有相同名稱的 INI 檔的存取情形；它並未告訴你資料儲存在 Registry 裡的哪個位置。你可以在這個機碼下建立數值，指定一個位置（或多個位置）。這些數值中的每個數值都應有一個與 INI 檔裡的段落對應的名稱。這些段落名稱會與父機碼名稱結合，好讓設定檔的 API 常式可以找出你要求的資料。

你必須建立一個名為 `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Crypto.INI` 的機碼，才能完成前述範例中與機碼的對映工作。請在這個機碼下，新增一個名為 `Encryption` 的數值。這兩個數值結合起來便可以告知 NT 將對 `Crypto.INI` 裡 "Encryption" 段落的存取改為到 Registry 裡查詢。

這些段落機碼的數值會告知 NT，實際資料儲存在 Registry 裡的哪些位置。這裡先假設我們有一個資料安全程式將資料儲存在 `HKLM\Software\Crypto\CurrentVersion\Settings` 裡。我們必須將這個 Registry 機碼的內容指定為 `Encryption` 數值的內容，才能完成從上個段落開始的對映。然後，NT 便有足夠的資訊將 INI 檔案裡的資料轉換到 Registry 機碼。

## 對映機碼技巧

下面將談到幾個建立對映機碼項目的技巧。首先，你可以指定一個預設數值，用來控制那些尚未建立對映的段落。回到我們先前提到的資料安全程式範例，若你新增一個 `Encryption` 機碼，這時 NT 還不會知道要如何將資料對映到 "Signature" 的段落。但你只要增加一個預設數值 ("`<No Name>`" 或 "`Default`") 給這個子機碼，便可以告知 NT 要使用哪個機碼，給哪些尚未定義自己的段落機碼的段落使用。

你可以在段落機碼的數值裡使用幾個特殊符號。表 8-1 顯示這些符號；下個段落將說明它們的運作情形。

表 8-1：初始檔案對映裡使用的特殊字串

符號	代表意義
SYS	將資料儲存在 <code>HKLM\Software</code> 路徑下；例如， <code>SYS:Netscape</code> 便是指 <code>HKLM\Software\Netscape</code>
USR	將資料儲存在 <code>HKCU</code> 路徑下；例如， <code>USR:Software\Qualcomm\Eudora</code> 便是指 <code>HKCU\Software\Qualcomm\Eudora</code>
!	將資料儲存在 Registry 及 INI 檔案裡命名的段落。當資料寫入這兩者其中之一時，資料也會被寫入另一項
@	不允許從 INI 檔案裡讀取資料，即使 Registry 裡沒有發現符合的資料亦然
#	當新使用者登入時，從 INI 檔將段落的設定複製到指定的 Registry 位置

## 對映範本

Entrust Technologies(<http://www.entrust.com/>) 所開發的 Entrust 資料安全封包有 16 位元及 32 位元的版本。為了保持程式碼的一致，Entrust 公司的工程師決定採用 INI 檔案而不使用 Registry。你只要跟隨下面的程序便可以建立一組對映，利用 Registry 資料來取代 Entrust 的 INI 檔案。

1. 在 `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping` 下建立一個新子機碼，名稱為 `entrust.ini`。
2. 因為 Entrust 設定是使用者自訂的，所以我建立一個新機碼，`HKCU\Software\Entrust`，用來控制設定資料。另外還新增 `Other` 及 `Entrust Settings` 兩個子機碼來控制資料。
3. Entrust.INI 裡的使用者自訂資料全位於“Entrust Settings”這個段落。我們在 `entrust.ini` 下新增一個名為 `Entrust Settings` 的子機碼，並將它預設數值給為 `@USR:Software\Entrust\EntrustSettings`，用來對映段落裡的資料。如此一來，NT 便會將儲存在“Entrust Settings”這個段落裡的資料對映到具有相同名稱的機碼；@ 符號的作用是預防對映程式從 INI 檔案裡讀取資料。
4. 將 `entrust.ini` 子機碼的預設數值給定為 `#USR:Software\Entrust\Other`。這種作法會迫使 NT 替新使用者複製 INI 檔案的資料，並將 `entrust.ini` 其它段落的資料儲存在 `HKCU\Software\Entrust\Other` 裡。

這些步驟的執行結果如圖 8-1 所示。最後，我利用 RegEdit 將對映機碼儲存成

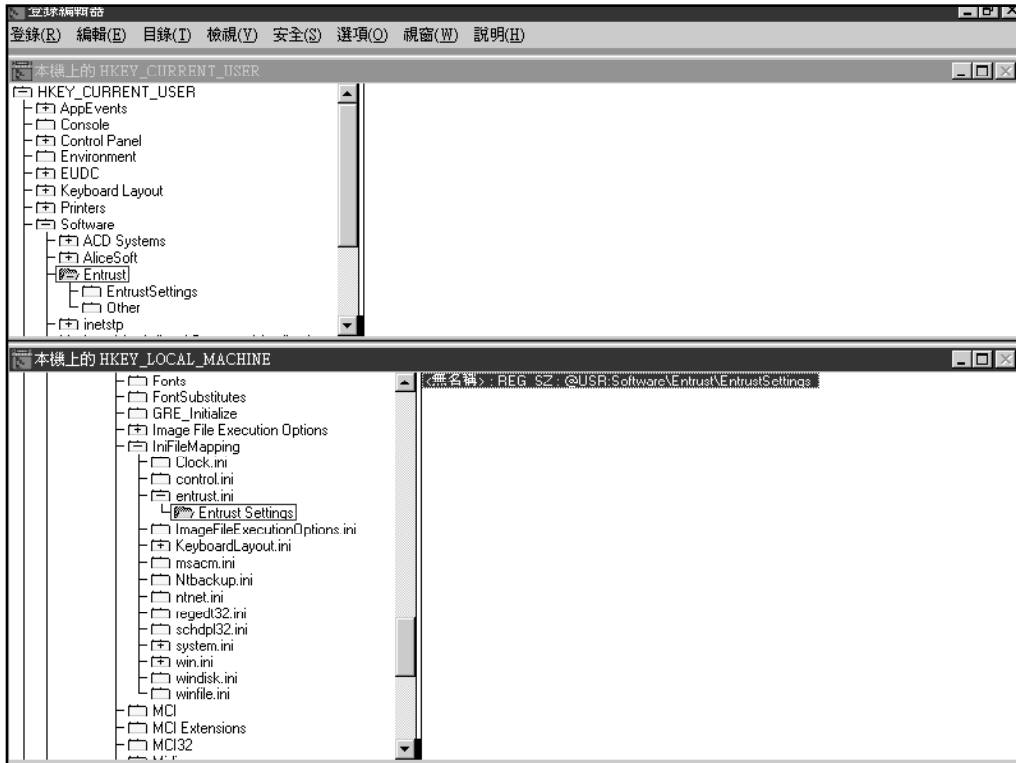


圖 8-1：新增 INI 檔案對映

## 限制遠端登錄存取

在 Windows NT 3.51 及其早期的版本裡，任何使用者都可以透過網路存取 Registry 的資料。從安全性的考量上看來，這相當不妥；NT 4.0（以及安裝了 SP4 或 SP5 後的 3.51 版本）預設情形是只允許「系統管理員」群組可以從遠端存取 Registry。這比原來的做法要來得安全一些。

但這種設定不可能適合你的環境。有時允許所有系統管理員存取仍然是有點危險，某些高級機器可能會批准增加安全特性，只允許單一帳戶或群組透過網路存取其 Registry 資料。相對的，你也可能希望其它的使用者及群組可以從遠端連線上某台

機器，並編輯這台機器上的 Registry 資料。

你可以自行決定要讓哪些使用者、群組，以及服務存取特定機碼上的 Registry，只要設定一個 Registry 機碼的 ACL 即可，這個機碼的名稱為 HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg。NT Shell 會授權對這台機器 Registry 的遠端存取權限，並只將這些權限授權給這個機碼上的存取控制列表 (access control list) 裡的項目。

在開始著手進行之前，我必須先指出這個限制機碼 (restriction key) 決定了整體 Registry 的存取控制。但你仍可以對個別機碼進行更嚴格的控制。例如，你可以設定限制機碼的權限，授權一群使用者可以對 Registry 進行存取動作，但若你在其它的使用者可見機碼上也放入存取控制，則會採用最具限制性的控制。

## 建立限制機碼

在開始利用這個限制特性之前，先決條件是你必須擁有限制機碼。預設情形是，NT Server 4.0 在安裝後便有了這個機碼，但 NT Workstation 4.0 則否。而 NT 更早之前的版本也沒有；若你希望限制存取情形，你必須在 Registry 裡手動增加限制機碼。下面是尚未擁有這個機碼時的執行步驟：

1. 使用「系統管理員」(或具備系統管理員權限的帳戶)身份登入並執行 RegEdit32 程式。移動至 HKLM\SYSTEM\CurrentControlSet\Control。
2. 使用「編輯」→「新增機碼...」指令來增加新機碼，機碼名稱為 SecurePipeServers，然後選擇 SecurePipeServers 這個機碼並再度使用「編輯」→「新增機碼...」指令來建立新子機碼，新子機碼名稱為 winreg。
3. 新增一個名為 Description 的 REG\_SZ 數值給 winreg 子機碼。微軟建議你將 description 的內容給定為“Registry Server”，但實際內容你可以自行決定。

由於使用的機器不同，你可能會發覺你只擁有限制機碼的某些部份；例如，NT Workstation 4.0 在不安裝 service pack 之前本身便擁有 HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers 機碼，但它沒有 winreg 子機碼，而你必須先要有這個子機碼才能運作限制功能。

## 在限制機碼上設定權限

只要 winreg 機碼存在，你便可以利用「安全」→「使用權限...」指令給予它存取控制列表。這個機碼所應用的權限可以決定哪些使用者及群組能透過網路存取你的 Registry。

「登錄鍵使用權限」對話方塊（如圖 8-2 所示）允許你更改可存取這個機碼的使用者及群組，你也可以修改這些使用者及群組的使用權限。（若你需要複習一下相關部份，可以看看第 5 章「如何使用 RegEdt32 指令」裡的「設定 Registry 權限」段落。）

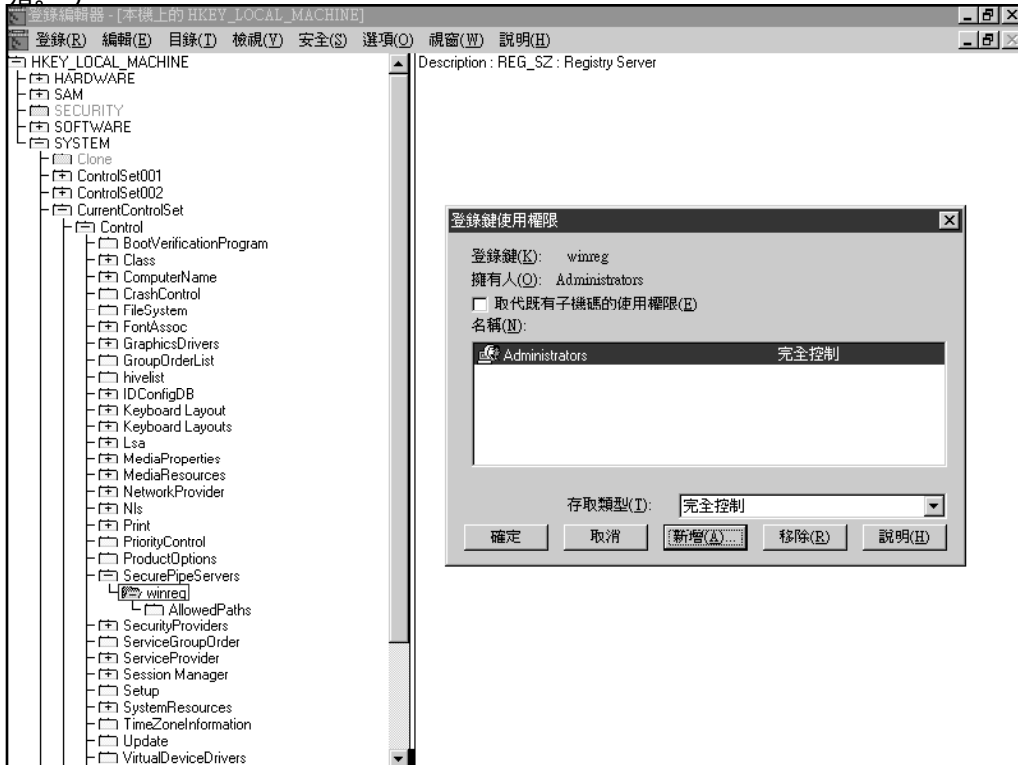


圖 8-2：設定 Registry 機碼權限

NT 4.0 的安裝程式裡的預設情形會擁有如圖 8-2 所示的使用權限；「Administrators」或「Domain Admins」群組擁有「完全控制」權限，而系統及建立這個機碼的帳戶也擁有相同的權限。你也可以將新使用者或群組加入這個清單裡並給予它們適當的權限；例如，你可能授權「讀取」權限給所有網域裡的使用者，但卻只將「完全控制」使用權限給予某個帳戶。另外，請不要更動到「SYSTEM」及「CREATOR OWNER」這兩個使用權限；kernel 及 Registry 子系統會根據這些使用權限對機碼進行存取動作。

---

### 警告

某些系統服務，像 directory replicator 及 print spooler，需要對 Registry 具有遠端存取權限。若你更改了位於 winreg 機碼上的存取控制項目，可能會導致這些服務停止運作。為了避免這種問題，請確定用來執行這些 replication 服務及 print spooler 的帳戶在 winreg 機碼上的 ACL 裡具有明確的使用權限。

---

## 允許例外

你也可能選擇將 Registry 裡的限制放鬆，允許發生存取控制規則裡（在 winreg 機碼上指定）的例外。這些例外可以用兩種方式顯示：你可以提供一列機碼清單，裡面的機碼不具備存取控制，或者你可以指定一列使用者清單，裡面的使用者可自由存取特定的機碼及數值。

你必須在 `HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths` 裡增加下面的數值，才能控制這兩種方式。

- **Machine** 數值（類型為 `REG_MULTI_SZ`），接受一系列 Registry 路徑清單。這裡列出的路徑可以被網路上的任何機器看到。預設情形是，NT 載入一組路徑，驅動（enable）replicator、print spooler、event logger 和 kernel，使各個功能運作正常：`System\CurrentControlSet\Control\ProductOptions` 以及 `Software\Microsoft\Windows NT\CurrentVersion` 是 kernel 使用的，`System\CurrentControlSet\Control\Print\Printers` 是 print spooler 使用的，`System\CurrentControlSet\Services\Eventlog` 是 event logging service 使用的，而 `System\CurrentControlSet\Services\Replicator` 則是 directory replicator 使用。
- **Users** 數值（類型也是 `REG_MULTI_SZ`）列出「User」及「Domain Users」群組裡的成員可以取得的 Registry 路徑。這個機碼的預設值是空值，也請你將它維持為空值，除非你有強烈的理由要讓個別使用者免於受到限制機碼的影響。通常，若你擁有一個使用者，而且這個使用者需要特殊的存取權限，最好的作法是將這個使用者帳戶放入一個群組，並在限制機碼上指定使用權限給這個群組。

經由這兩種方式所授權的存取權限仍然比不上你將使用權限直接授予機碼。例如，若你使用「安全」 「使用權限...」指令將「Everyone: 讀取」的存取權限給予 `HKLM\Software\Netscape\Netscape Navigator`，然後增加相同的路徑給 Machine 數值，遠端的使用者將無法改變子樹下的數值：新增的 ACL 將會覆寫 Machine 項目的存取權限。

## 修正登錄安全存取控制列表 (Registry Security ACLs)

Registry 裡的每個機碼都有 ACL。很不幸地，這些 ACLs 有許多權限過大。例如，預設情形下，「Everyone」這個帳戶對幾個機碼擁有寫入權限，這些機碼允許使用

者執行任何程式 - 這樣一來造成了相當大的安全問題。而你只要多花些心思，執行幾個簡單的步驟，便可以大大地改善 NT 安全狀況。

首先，稍微離題一下：所有認證過的使用者都會自動成為「Everyone」群組裡的一員。在執行 NT 4.0 SP3 或更後面版本的機器上，這些使用者也會是「Authenticated User」群組的成員。「Everyone」也包括「anonymous」及「guest」帳戶，因此，在通常的情形下，最好的做法是，「絕不」授權「Everyone: 完全控制」這個存取權限給「任何事物」。

回到實際上的步驟來。首先，請應用「限制遠端存取」段落裡所建議的改變。在你完成這些動作後，請確定「Everyone」在 HKLM\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths 上只有讀取權限。這種作法可以預防有干涉者插入他自己才允許的路徑做為匿名存取用途。

接著，遵照微軟在 knowledge base 網站上文章 Q127813 裡的建議，限制「Everyone」只能在下面這三個機碼上有「讀取」權限：

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
```

這些機碼指明當 NT 啟動時 (Run 及 RunOnce) 要執行的程式或程式移除 (Uninstall) 時要執行的程式，所以這些機碼應該是別人無法任意更改的。

同樣地，請在 HKLM\SYSTEM\CurrentControlSet\Service\Schedule 上移除「Server Operators」群組的「寫入」使用權限。一般情形下，「Server Operators」群組的成員對工作有排程的使用權限，但這些工作可以在 SYSTEM 帳戶下執行 - 好讓「Server Operators」成員取得「Administrator」權限。【註】同樣的道理，請在 HKLM\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon 上移除「Server Operators」群組的寫入權限可以預防 UserInit 及 Boot Verification Program 數值上有類似的風險。

下一個步驟可以自由決定：盡可能地限制 Registry 的存取控制。而這種情形視應用程式不同而有所差異。例如，Office 97 需要在 HKLM\Software 及 HKCU\Software 下對它自己的機碼有「Everyone: 讀取」權限。移除這個使用權限將會導致 Office 的特性停止運作。Internet Explorer 4.0 也是相同的情況，其它的產品也差不多都是

---

註 揭露這些缺點的責任就交給 Internet Security Systems, Inc. 的 David LeBlanc(<http://www.iss.net>) 了。

如此。當你更改 Registry 機碼的 ACLs 時，請在還未將改變散播到整個網路之前，先測試一下應用程式，確定這些應用程式仍可正常運作。

由 Coopers & Lybrand【註】【譯註】所撰寫的關於微軟的白皮書建議改變 HKCR、HKLM\Software\Microsoft\RPC，以及 HKLM\Software\Microsoft\Windows NT\CurrentVersion，允許「Everyone: 特殊存取」的權限。他們特別建議授權「Everyone」下列權限：查詢數值、子機碼計數、通知，以及讀取控制。我將延用這些建議並建議你採用「Authenticated Users」取代「Everyone」。這項改變所造成的缺點在 Kirill Ermakov 的安全警告（[ekv@comp.chem.msu.su](mailto:ekv@comp.chem.msu.su)）裡有提到。它注意到 HKLM\Software\Microsoft 下的大多數機碼都有「Everyone: 刪除」及「Everyone: 設定數值」的存取權限，好讓使用者可以刪除必要的機碼。例如，你可以很容易地刪除 LanmanWorkStation 及 LanmanServer 機碼，這兩個機碼將會讓你不必使用到 SMB 共用目錄！

總是有些旁門左道的方法。ISS 公司的 David LeBlanc 曾設計過一個名為 everyone2user 的工具程式（可以從 <ftp://ftp.iss.net/everyone2user.exe> 取得），這個工具程式會遞迴地往下處理 Registry 的六個主機碼，並將存取權限「Everyone」群組改為「Domain User」或「Users」。這種方法就我的經驗而言還蠻安全的，但它還是要冒一點兒險，就是 Registry 可能會全被改掉。在使用 everyone2user 之前還是請你記得做好備份。

還有一些要注意的地方。在 knowledge base 網站上的文章 Q139342 裡警告了在 LanmanServer 上設定不當的使用權限的後果。目前尚未有文章提醒你有關改變其它機碼所造成的影響，但這並不表示安全無虞。還是請你小心地保持做良好備份的習慣。

## 使用 SYSKEY 替 HKLM\SAM 加密

就像 Unix 一樣，NT 不會儲存使用者或機器的密碼。但它會抓取密碼並透過一個稱為「one-way function(OWF)」的系統傳遞。OWF 會取得密碼並產生相關的新資料區塊，但這個區塊不包含密碼。在 OWF 裡的 "OW" 命名由來是因為它無法取得

---

註 它可以從網站 <http://www.microsoft.com/ntserver/guide/cooperswp.asp> 取得。

譯註 Coopers & Lybrand 是國際著名的會計師聯合事務所。

OWF 的輸出並「回溯」產生原本的密碼。NT 會儲存 OWF 的輸出而不是儲存密碼，所以你無法偷取 OWF 的資料並用它取代密碼。

1997 年春天，一群來自 L0pht Heavy Industries(<http://www.l0pht.com>) 的駭客公開發表了一項聲明，聲稱可以從 NT 的 SAM 資料庫中取得 OWF 產生的密碼數值，並將這些數值導向密碼破解工具來破解密碼。這類危害在 Unix 世界裡已廣為人知多年了，但它出現在 NT 的世界裡卻仍值得大書特書。但就實際而言，真正的危險性仍相當低，因為只有系統管理員有權對 SAM 進行存取，取得 OWF 所產生的密碼。而因為系統管理員本身對系統有相當大的存取權限，所以對他們而言，要破解密碼本來就不是什麼難事。

無論如何，微軟的確在 Internet 上及 SAM 密碼資料的安全性上遭遇了打擊。因此，微軟引進了一個保護 SAM 資料的方法，就是對資料進行嚴格的加密；SYSKEY 工具程式會安裝並控制這項額外的保護層。SYSKEY 可以在 NT 4.0 SP3 裡取得；它同時也是 NT Server Enterprise Editor 的一部份。因為 NT 5.0 會在 Active Directory 裡儲存使用者資料，所以在 NT 5.0 裡既不會包含也不會支援 SYSKEY 【註】。

## SYSKEY 的作用

SYSKEY 增加了一層安全保障，利用 128 位元的「系統代碼」將 SAM 資料庫裡的密碼資料加密。這個機碼（微軟稱呼它為「password encryption key」，簡稱 PEK；我也將這麼稱呼它）會在你安裝 SYSKEY 時隨機（randomly）產生。在 PEK 產生之後，NT 會使用 PEK 將所有 SAM 裡的密碼資料（但不是一般的帳戶資料）加密及解密。因為資料會被加密處理，所以其它的偷竊者或駭客就算取得這個資料也沒有用處（而且要取得這個資料仍必須對網域控制站具有系統管理員的權限）。此外，因為資料以加密的格式儲存，所以當它被備份到 ERD 或磁帶時資料仍維持在保護狀態。

資料加密後，便會被存回 SAM 資料庫，而且那些存取到這些密碼資料的系統服務（包括本機的 security authority 或 LSA）必須依賴殼層才能將資料解密。因為

---

註            在 NT 5 beta 1 版本裡確是如此，但誰知道將來這是否仍會改變呢？

這項功能的緣故，kernel 必須在開機時便知道何謂 PEK：SAM 密碼資訊會包含系統服務的密碼資料，還會包含目前使用者的密碼資料，當機器開機時，這項服務便會開始執行。

NT 會儲存 PEK 來完成這項功能。你可能會懷疑為何儲存 PEK 會提升安全性；它看來有點笨，竟然用儲存密碼的作法來加密要保護的資料！答案很簡單：有 " 另一個 " 的機碼用來加密 PEK。這個多出來的機碼便是系統代碼，SYSKEY 也是根據這個機碼命名的。SYSKEY 支援三種儲存系統代碼的選項，當有需要解密 PEK 時，系統便可取得這些資訊。

第一種，也是最安全的選項是允許你將系統代碼儲存在磁片上。當機器開機時，這個磁片必須放入電腦磁碟機裡，讓 kernel 取得系統代碼並用來解密 PEK。若你缺乏正確磁片的話，機器將無法開機進入磁片所保護的 NT 系統【註】。這種情形導致了一種新型式的開機失敗，所以，請記得將這張磁片備份。此外，這張磁片允許你對 SAM 資料進行存取動作，所以你必須控制哪些人對它有存取權限。

第二個選項是儲存系統代碼，這個機碼會由另一個機碼加密。這個多出來的機碼是你選擇產生的。它並不會在開機時要求插入系統代碼磁片。系統代碼的加密版本會儲存在電腦上，所以只需要 *passphrase*，而不需要任何磁片或主磁碟。

最後，你可以選擇是否系統代碼儲存在本機上。NT 會使用微軟稱為「complex obfuscation algorithm」的方式隱藏這個機碼。這種方式讓組合系統代碼這項工程變得更為困難。「security through obscurity」( 隱晦的安全性 ) 與其它的方法比較起來，它提供了較低的安全性，但它有個優點，就是允許自動重新開機。因為 kernel 可以在需要時取得 PEK，不需要任何人力操作。這對某些應用程式而言相當重要；你可以決定它是否能成為伺服器的最佳選擇。

## 安裝 SYSKEY 前須知

和大多數其它的 NT 元件一樣，在你知道它可以替 Registry 資料增加安全性之後，

---

註 每一個 NT 的安裝程式都有自己的 PEK。若你在一台電腦上安裝多種版本的 NT，則每一個版本都會有一個 PEK。若你在一台機器上安裝兩個 NT，但遺失了其中一個的系統代碼磁碟片，你仍然可以利用另外一個開機。

想必你一定迫不及待地想要安裝 SYSKEY。這裡有一點要注意的是，你必須小心並正確地透過網路配置 SYSKEY。還有，你必須了解 SYSKEY 可以保護哪些項目，以及它可能會引起的問題。SYSKEY 並不保證你可以很容易上手。

每台 NT 4.0 工作站及伺服器都可以執行 SYSKEY，也都可以使用前面所提到的三種系統代碼儲存選項。若你選擇使用系統代碼磁碟片 (key disk) 或 passphrases，請記得它們就像 ERD 一樣：只能在建立它們的機器上使用，所以你必須替每一台要保護的機器製作一片磁片。(你也可以採用偷懶的方式，在所有機器上使用相同的 passphrases。)

首先，讓我們從最嚴重的問題開始：SYSKEY 可以讓系統更安全，但它的功能是單向的，一去便無法回頭。只要你啟動加密功能給 SAM 帳戶資料庫使用，便自此無法關閉加密的功能並返回舊有的未加密版本 (雖然你可以使用 ERD 復原，如同下面所述)。就實際情形而言，只要你將 ERD 保持在最新狀態，便不需要回到原本未加密的版本。

下一個問題是安全專家稱為「鎖是夠堅固，但門的材質挺脆弱的」問題。假設你在一個網域裡有多個網域控制站，而且其中有一個使用 SYSKEY，但其它則不使用，你便無法在網域裡增加任何安全。雖然有一台機器的 SAM 資料庫被保護，但因為其它的控制站都有重複的備份，你想保護的資料仍然有洩露的可能。理想的情形下，你應該在每一台有帳戶資料庫的機器上執行 SYSKEY。也就是說，所有的網域控制站及任何有本機帳戶的 NT 工作站機器都應該執行這項功能，才能達到足夠的安全。

SYSKEY 造成的最後困難度在於，它必須將機器上的資料加密，才能提高安全性。只要你保有對系統代碼的存取權限，NT 便可以解密 PEK 及使用 PEK 存取密碼。若你選擇使用 passphrase 或 key floppy，但卻遺失或忘記其中內容的話，你將必須利用 ERD 恢復原狀。若你所擁有的 ERD 是在加密之後製作的，那麼你將無法登入系統！所以，請好好保護 key floppy 並製作備份，這樣才能避免磁片壞掉會遺失所造成的危險。

" 我告訴你三次的話，就是真理 "

微軟建議你在安裝 SYSKEY 時製作三份 ERD：安裝 SYSKEY hotfix 或 service

pack 之前製作一份、安裝完 SYSKEY 但尚未啟動 SYSKEY 之前製作一份，以及安裝完 SYSKEY 並重新開機後再製作一份。雖然這麼做實在是有夠麻煩，但製作這三個備份才能提供你最大的安全保障，讓你在未來當機時可以復原：

- post-SYSKEY ERD 包含帳戶資料庫加密之後的版本。當你新增或移除帳戶時，將 ERD 保持在夠新的狀態。這樣一來，只要你擁有系統代碼（不論是儲存在電腦或磁片上），便可以利用 ERD 復原帳戶資料庫，以及 Registry 的其它部份。
- pre-SYSKEY ERD 會包含 Registry 未加密之前的完整記錄。若你需要復原機器但缺乏系統代碼時，你有兩個選擇：重新安裝 NT 並遺失目前所有帳戶資料，或者是使用 ERD 復原 Registry，但遺失啟動 SYSKEY 之後產生的所有改變。
- preinstall-SYSKEY ERD 可以讓你避免遭受 hotfix 或 service pack 造成的問題。通常，在安裝任何 service pack 或 hotfix 之前，你應該「總是」更新 ERD。這樣才能在必要復原機器時有個退路。

請在每台機器上保留這三種 ERD 以備不時之需。最起碼，網路上的每一台網域控制站以及上一組擁有重要帳戶的 NT 工作站機器都有這樣的需求，。

## 更新網域控制站

微軟也警告你在主網域控制站上安裝 SYSKEY 的後果：若你在安裝 SYSKEY 時產生問題，或遺失了那台機器的系統代碼，將會導致無人可登入你的網域！對那些具有不少使用者的網域而言，你實在應該有備份的網域控制站；若你目前沒有，請考慮增加一個。

在多重網域或多重控制站裡執行 SYSKEY 最安全的方法如下：

1. 選擇網域。確定這個網域至少擁有兩個網域控制站（一個「主要網域控制站」及一個「備份網域控制站」）。
2. 將「主要網域控制站」降級為「備份」狀態；這項動作會將一台備份控制

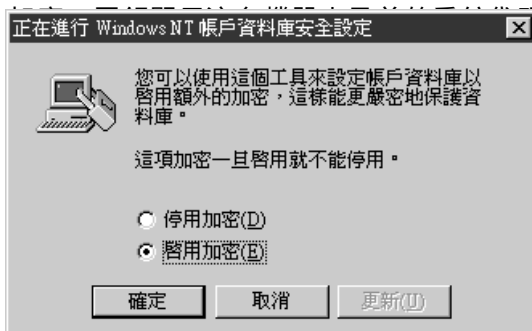
站自動提升為主要控制站。（若你需要參考資料，可以看看 AEleen Frisch 所著的 Essential windows NT System Administration。）如此一來，新提升的控制站便接管了認證網域登入要求的工作，而且新的主控制站也必定擁有網域帳戶資料庫的完整複製。

3. 在原本的主網域控制站上啟動 SYSKEY。當你覺得萬事俱備時，將步驟 2 裡提升的機器降級。這樣便可以將原本的網域控制站還原成主要狀態，也使得網路恢復開始時的樣子。
4. 在網域裡的其它網域控制站上啟動 SYSKEY。若網路裡有一個以上的網域，請回到步驟 1 並選擇另一個網域繼續處理。

若你的網域只擁有一台控制站，你可以直接在這個控制站上啟動 SYSKEY，但前提是你必須擁有前面建議的三片 ERD。

## 啟用 SYSKEY 保護功能

只要 SYSKEY.EXE 可以執行，你便可以控制 SAM 資料庫的加密情形。如同你所預期的，只有系統管理員可以開啟「系統代碼」的保護功能。在你第一次執行 SYSKEY 時，會看到如圖 8-3 顯示的對話方塊。這個對話方塊會警告你這項加密功能一旦啟用之後就不能停用。這個視窗裡只有五個控制項：「停用加密」及「啟用



加密狀態。你可以將狀態由停用改為啟用。在下一個段落「改變代碼的儲存方式」

圖 8-3：初始 SYSKEY 的對話方塊

啟動 SYSKEY 的第一個步驟很簡單：就是按下「啟用加密」圖鈕，再按下「確定」按鈕。然後，你會看到一個警告對話方塊，提醒你將無法還原這項轉換並建議你在「確定」後，你會看到「帳戶資料庫識別碼」對話方塊告知 SYSKEY 你要將產生的「系統代

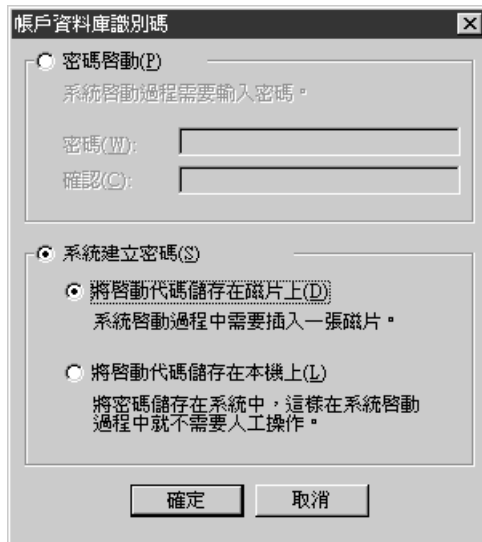


圖 8-4：指定 SYSKEY 的位置

- 若你希望使用 passphrase 解開系統代碼，請按下「密碼啟動」圖鈕，並在「密碼」及「確認」欄位輸入密碼。最高可以輸入 128 字元的 passphrase，而長一點的文字較佳。很不幸地，SYSKEY 對密碼並沒有最少字數的限制。微軟建議至少 12 個字元，但這很容易便可達成 — 挑選兩個容易記憶

的形容詞及一個名詞，然後將它們使用介系詞或特殊字元連接在一起即可（像「exploding!friendly\*holiday」或「galloping\_sleepy#motorhome」）。NT 會將你輸入的 passphrase 交由一項特殊的演算法處理，這個演算法會根據你輸入的字元產生 128 位元的代碼。

- 若你希望系統自行產生密碼，請按下「系統建立密碼」圓鈕。在這個模式裡，NT 會使用自己的虛擬亂數產生器【註】產生一個 128 位元的亂數系統代碼。就如同你所知道的，這個代碼必須儲存在某個位置。請選擇要將它儲存在何處：

- 「將啟動代碼儲存在磁片上」按鈕指示 NT 將加密的系統代碼儲存在磁片。代碼本身將會被儲存在一個名為 StartKey.key 的檔案裡。當你選擇這個選項時，請準備一片磁片來儲存這個代碼。請切記，不要使用 ERD 磁片來存放這個代碼 — 這樣一來那些覬覦密碼的人只要取得一片磁片便取得了這兩個片段的資料（就跟不要把所有的雞蛋放在同一個籃子裡是同樣的道理）。
- 「將啟動代碼儲存在本機上」按鈕會將系統代碼儲存在 HKLM\SYSTEM 裡。當你選擇這個選項時，系統便可以無人自行重開機。

在你選擇好要使用的方法後，請按下「確定」按鈕。若你選擇將代碼儲存在磁片上，SYSKEY 將會提示你插入一片磁片，然後它會確定將代碼寫入磁片裡。否則，代碼將會靜靜地更新並結束 SYSKEY 程式。

在你下次開機時，SYSKEY 保護功能便會開始發生效用。這代表說，除非你在本機儲存系統代碼，否則往後每次開機時，你都必須輸入 passphrase 或放入 key floppy。

---

## 建議

若你趕時間的話，你可以在執行 SYSKEY 時使用  $\downarrow$  旗標；這種作法會指定程式安靜地產生系統代碼並將它儲存在本機上。這是一項易於使用

---

註 偉大的數學家兼電腦學家 John von Neumann 曾說過，依賴軟體來產生亂數根本不可靠。但是，用虛擬亂數產生器（就像 NT 所使用的）應該是還好啦！

的技巧，用來設定新的工作站及伺服器時特別好用；你可以將這個指令加入日常使用的設定指令檔裡，然後在空閒時，再改變代碼的儲存方式。這種做法可以讓你很有效地提供立即的保護。

## 改變代碼的儲存方式

在安裝並啟動 SYSKEY 後，你便有機會更改原本儲存代碼的方式。你可以執行 SYSKEY 並改變儲存的方式。當你改變方式時，SYSKEY 會產生一個新的系統代碼並將它儲存起來，而不是重新使用舊的代碼；這可以保護密碼資料不受危害。

為了改變機器上的代碼儲存方式，你必須執行 SYSKEY 並按下「更新」按鈕。然後「帳戶資料庫識別碼」對話方塊（圖 8-4 所示）將會出現；與目前啟用的儲存方式相對應的圓鈕將會位於啟動狀態。你只要按下其它的圓鈕，並視需要填入密碼，便可以將儲存代碼的方式更改成其它儲存方式了。

因為 SYSKEY 會在更改儲存方式時產生新的代碼，所以你必須在更改的過程中提供舊的代碼。這表示在你按下「確定」後所發生的事主要取決於你「先前」使用的儲存方式。若你原本使用「將啟動代碼儲存在本機上」，而現在卻想將它改為其它方式時，SYSKEY 會從 HKLM\SYSTEM 取得舊代碼，所以你不需要另外做任何處理。若你原本的儲存方式是將密碼儲存在磁片上或使用 passphrase 將資料保護起來，則 SYSKEY 會要求你提供 key disk 或 passphrase。這種作法可以預防下列情形所造成的危害：更改代碼、儲存在磁片而磁片被偷走，因而造成機器無法開機。



，而圖 8-6 顯示了要求 passphrase 的對話方塊，SYSKEY 將會顯示確認對話方塊，提示你用你選擇的方式儲存。若你沒有提供正確的



圖 8-6：「密碼」對話方塊

## 復原 SYSKEY 保護的登錄資料

第 3 章「緊急處理」描述了利用 ERD 復原已毀損登錄資料的技巧。你可以遵照相同的步驟來復原 SYSKEY 保護的機器。但 SYSKEY 會造成一些不同的地方。復原 SYSKEY 保護的機器的規則很簡單：就是使用正確的 ERD。

### 回存 SYSTEM 及 SAM hive

雖然加密的帳戶資料儲存在 HKLM\SAM 子目錄裡，但實際的 PEK，以及其它 SYSKEY 用來判斷系統代碼儲存位置的資料則位於 HKLM\SYSTEM。你「必須」在同一時間回存 SYSTEM 及 SAM hive，才能將加密的帳戶資料庫恢復 - 並不是只有 SAM 就夠了。若你無法完成這項工作，NT 便無法將登錄資料解密，可能來是由於它無法找到系統代碼（若你沒有回存 SYSTEM），或是因為代碼沒有解密資料（若你沒有回存 SAM）。當然，你必須從相同的 ERD 裡回存這些 hive 資料。

## 取得正確的系統元件

若你在安裝 NT 4.0 service pack 後取得 SYSKEY 程式，你一定不會注意到有三個系統檔案被取代：winlogon.exe、samsrv.dll，以及 samlib.dll。這三個檔案和 syskey.exe 會完成帳戶資料庫的保護工作。你必須擁有這些檔案才能啟動 SAM 資料的加密及解密。

在第一次安裝 NT 時，系統會將安裝的所有元件版本記錄在 system.log 裡。在安裝 service pack、hitfix 或類似 Internet Explorer 的軟體時，這些軟體會取代一個或多個系統檔案，並更新 system.log 裡的項目，好讓這個檔案反應出目前所有的 DLLs、驅動程式，以及其它作業系統元件的版本。

這代表若你安裝 SYSKEY 做為 hotfix 及 service pack 的一部份，則 system.log 裡的 winlogon.exe、samsrv.dll 和 samlib.dll 項目將會反映出安裝完 SYSKEY 的版本，而非原本安裝的版本。若你希望將機器復原成尚未安裝 SYSKEY 之前的狀態，請使用 NT 安裝程式裡的「修復系統檔案」選項來還原成原來的版本。然而，你必須確定可以從安裝 SYSKEY 之前的 ERD 裡回存 SAM 及 SYSTEM hive：若你還原舊有的系統元件但卻將沒有將加密的登錄資料還原，一定會產生很多問題。

---

### 警告

若你安裝了 SYSKEY 但未將它啟動，則 winlogon.exe、samsrv.dll 以及 samlib.dll 將不會符合原本的安裝情形。在安裝這些檔案的新版本時，即使系統處於停用加密的狀態，它們仍會改變登錄格式 (Registry format)。若你使用 NT 的安裝程式，搭配尚未啟用 SYSKEY 前製作的 ERD，將這三個檔案復原成原本的狀態，你「必須」也利用這些 ERD 將 SAM 及 SYSTEM 的資料復原：否則，舊元件將無法讀取新的登錄格式。

---

應該使用哪一個 ERD？

三個 ERD 的確是比大多數 NT 系統擁有的 ERD 來得多，對你而言，決定採用哪一個 ERD 可能需要全盤的考量。但這並不困難：每個 ERD 都可以控制要將系統回復到哪種狀態。使用哪一個 ERD 要看欲復原的系統上有哪些內容而定。表 8-2 顯示你可以選擇的選項。

## 各種好東西

到目前為止，你已經學到了如何使用工具修改、備份，以及復原登錄資料。你可能會懷疑可以將這些工具應用到怎樣的地步！關於登錄有一些共通且必要的管理工作；認識這些工作的操作方式，將可確保機器在你的管理下穩定而安全地運作。

表 8-2：ERD 的還原表

還原成...	使用這個 ERD	不要忘了...
安裝 SYSKEY 以前的系統	preinstall ERD	你可能會遺失從安裝 SYSKEY 後所改變的帳戶資料庫。 請選擇 NT 安裝程式裡的「修復系統檔案」復原原本的 winlogon.exe、samsrv.dll 和 samlib.dll。 你隨時可以回到這個階段，即使沒有系統代碼也行。
安裝完 SYSKEY 但尚未啟動 SYSKEY 之前的系統	pre-SYSKEY ERD	你可能會遺失從安裝 SYSKEY 後所改變的帳戶資料庫。 你隨時可以回到這個階段，即使沒有系統代碼也行。 使用這片 ERD 時，「不要」從 CD 裡選擇「修復

		系統檔案」。
啟動 SYSKEY 以後的系統	postSYSKEY ERD	從 ERD 更新後的所有帳戶資料庫改變都會被保留。 磁片或機器上必須有系統代碼 /passphrase。 使用這片 ERD 時，「不要」從 CD 裡選擇「修復系統檔案」。

## 改變 Registry 大小

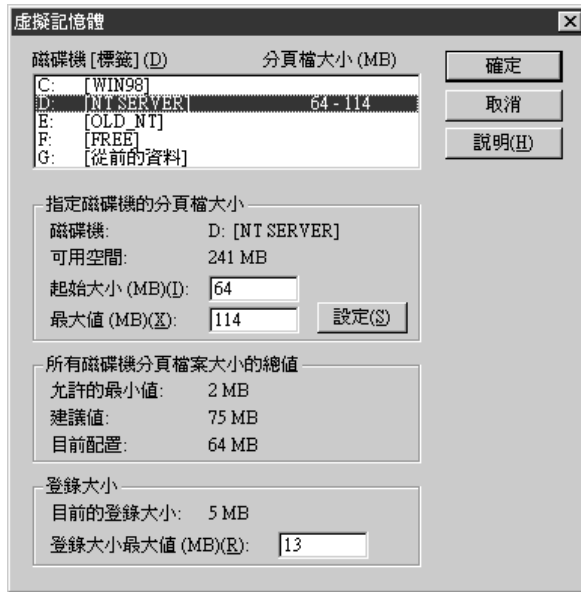
因為 Registry 是一堆 hive 的集合，而 hive 大多數都是磁碟檔案，所以你無從得知 NT 實際上如何將整個 Registry 對映到記憶體。這種作法讓 Registry 呼叫顯得更有效率；但也代表了隨著 Registry 的增長，會隨之花費相當比例的虛擬記憶體空間。為了預防 Registry 在系統分頁檔案裡佔據太多空間，NT 提供了一個內部參數稱為 Registry Size Limit(RSL)。RSL 會設定登錄資料可使用的位址空間上限；但是，隨著你新增軟體及使用者到機器上，Registry 也會隨之變大。若它變得太大以致與 RSL 相衝突時便會產生問題。（你可以到微軟的 knowledge base 網站 <http://www.microsoft.com/kb/default.asp> 上搜尋「Maximun Registry Size」，可以得到一系列關於此類問題的清單，但其中大多數都不是很好懂。）

預設情形下，RSL 會設定成系統的虛擬記憶體配置的 20-25% 左右。這個限制只是個最大值，並不是絕對，而且 RSL 設定的限制並不代表要保留這麼大的空間，只是代表系統不能使用超過這個數量的空間而已。在可使用的虛擬記憶體大小及 RSL 之間有一個複雜的關係存在；通常，與全部虛擬記憶體配置相比，你應該將 RSL 保持在 80% 或更少。否則的話，可能會導致效能不彰。

「虛擬記憶體」對話方塊（見圖 8-7）會顯示目前的 RSL 及登錄所使用的空間。若目前的大小比 RSL 的 80% 還大，請增加 RSL 的大小。在選擇新的 RSL 時，請確定將它保持在低於全部虛擬記憶體的 80% 以下；通常，你不必將它增加到大於虛擬記憶體的 33% 以上。若你需要更多的空間，可以考慮增加虛擬記憶體的大小，然後再將 RSL 加大。

在 NT 4.0 裡，你可以透過圖 8-7 裡的「虛擬記憶體」對話方塊調整 RSL。下面是改變 RSL 的步驟：

1. 按下「控制台」→「系統」→「效能」索引標織。



「虛擬記憶體」對話方塊。

的數值。

對話方塊，再關閉「系統」控制台後才會發生效用。

圖 8-7：「虛擬記憶體」對話方塊

## 稽核 Registry 存取情形

在第 5 章「稽核 Registry 機碼活動」裡，你學到如何將稽核控制應用到 Registry 裡的所有機碼。既然 NT 在 Registry 儲存了這麼多重要的資料，稽核其中部份的資料似乎是個不錯的主意——你可以稽核其中的部份機碼，預防可能的安全問題，或在使用者執行不該執行的動作時及早發現。

## 建議

NT 本身常使用讀取機碼來判斷機碼是否存在。這種作法很常見，所以我建議你避免稽核下列動作 — 讀取、查詢數值、子機碼計數 — 這些動作會產生許多不必要的稽核記錄項目。

一旦你啟動稽核功能，指定的事件將會儲存在系統的事件記錄裡。因為這些檔案是用來記錄發生哪些稽核事件，所以請妥善地保護這些檔案！設定它們的使用權限，將「完全控制」權限給「CreatorOwner」、「SYSTEM」和「Administrator」，而且將指定「讀取」權限給「Everyone」，然後確定沒有其它的使用者對記錄檔有寫入的存取權限。

## 利用稽核記錄

當你啟動稽核功能後，每當指定的事件發生時，NT 都會在安全事件記錄裡寫入一個稽核項目。這裡是一個範例項目：

```
12/2/97 11:27:19 PM Security Success Audit Object Access 560 Administrator BOOMBOX
Object Open:
Object Server: Security
Object Type: Key
Object Name: \REGISTRY\USER\S-1-5-21-34824712-245319459-1244863647-500
New Handle ID: 240
Operation ID: {0,47947}
Process ID: 2161664032
Primary User Name: Administrator
Primary Domain: BOOMBOX
Primary Logon ID: (0x0,0x1E35)
Client User Name: -
Client Domain: -
Client Logon ID: -
Accesses Create sub-key

Privileges -
```

在「使用者管理員」裡啟動檔案 / 物件存取稽核便可以取得這些資料，然後使用 RegEdt32 稽核 HKCU 裡是否有成功的「建立子機碼」存取要求。之後，每當我在

H K C U 下建立子機碼時，都會得到一個新的稽核記錄。

就如你所看到的，這個記錄會告知哪一個機碼是要求的目標機碼（Object Name 欄位），哪個使用者名稱發出要求（伴隨使用者的網域），以及所要求的是哪種存取權限。若你希望掃描事件記錄做為 Registry 存取之用，我建議你使用工具程式像 SomarSoft 公司的 DumpEvt (<http://www.somasoft.com>)，或撰寫自己的 Perl 指令檔（script），利用 Win32::EventLog 模組裡的函式分析事件記錄（Event Log）。

### 追蹤軟體的安裝或重新安裝

任何使用 Registry 的軟體（具有「Designed for Windows 95」或「Designed for Windows NT」標籤）都會追蹤 H K L M \ S O F T W A R E 或 H K C U \ S O F T W A R E 的情形。微軟建議軟體研發者在這兩個機碼下建立自己的子機碼，所以你會看到許多類似 H K L M \ S O F T W A R E \ N E T S C A P E 或是 H K L M \ S O F T W A R E \ Q u a l c o m m 的項目。你可以直接稽核這些機碼，或者只稽核有興趣的機碼。例如，若你希望有人在機器上增加新軟體時看到稽核通知，你便可以在 H K L M \ S O F T W A R E 上增加「建立子機碼」這個稽核項目。若你希望知道何時有人安裝 Netscape 公司生產的軟體，你可以在 H K L M \ S O F T W A R E \ N e t s c a p e 及 H K C U \ S O F T W A R E \ N e t s c a p e 下稽核「建立子機碼」項目。

### 避免內部隱憂

NT 允許系統管理員安裝一個或多個 DLL 認證密碼，然後再將密碼傳送到登入的子系統。NT 所運用的 NetWare gateway 工具便會使用這樣的 DLL 來完成工作，而關於這個 DLL 作用的文件說明可以從微軟取得。這使得使用者可以安裝抓取密碼的 DLL，這個 DLL 只會將密碼儲存在檔案裡，而不會更改密碼，然後再將密碼傳送到登入子系統。NT 會在 H K L M \ S Y S T E M \ C u r r e n t C o n t r o l S e t \ C o n t r o l \ L s a \ N o t i f i c a t i o n P a c k a g e s 裡保留這些 DLL 的清單。我強烈建議你對這個機碼啟動稽核功能。

---

註 當然，你可以爭論說這些工具應該包含在 NT 裡。我很同意這項觀點，但這樣一來，Microsoft 公司就必須整理、說明，以及測試這些工具，而這些工具中大多數只是支援專家、系統管理員... 等等的人們使用。

## 使用 Resource Kit 工具程式

Resource Kit 有兩種版本：NT Workstation Resource Kit 包含與管理 NT Workstation 相關的工具，NT Server Resource Kit 則包含伺服器專用的內容，再加上 Workstation 版本裡可以取得的內容。即使這些工具當中某些只具有部份功能（像“beta”版本的 telnet server 維持在 beta 長達兩年），而大多數都很缺乏說明文件，Resource Kit 大約價值 US\$150 左右，之所以這麼貴，是因為它裡面有許多工具不易從其它來源取得。【註】

### Resource Kit 裡有哪些內容

NT 4.0 Resource Kit CD 裡有相當多的工具及文件。表 8-3 簡單列出與 Registry 有關的項目。這些工具中大多數在 NT 3.1 Resource Kit 裡便已經存在；在 1997 年夏天，微軟推出 resource kit 裡的更新部份（可以從 <ftp://ftp.microsoft.com/busys/winnt/winnt-public/reskit/nt40> 取得），它增加了 reg.exe 這個新工具。reg 取代了先前幾個工具，但這些工具現仍存於 resource kit CD 裡。我會在表中標明這些已被取代掉的工具，好讓你知道可以跳過哪些項目。

表 8-3：Resource Kit Registry 工具

工具	作用	注意
comp reg.exe	比對兩個 Registry 數值內容 - 就像 diff 一樣。	參本章稍後的「機碼及數值的比較」
reg.exe	可以做到許多功能：新增、移除，或改變機碼；載入及卸除 hive，以及其它功能。	參本章稍後的「REG: 瑞士刀」
regback.exe regrest.exe	備份並還原 Registry 機碼、數值，以及 hive。可以用來復原全部或部份毀損的 Registry 資料。	在第 3 章中有介紹
regchg.exe	從命令列裡改變某個數值。	被 reg.exe 所取代
regdel.exe	從命令列刪除 HKLM 的某個子機碼。	被 reg.exe 所取代
regdir.exe	提供目錄類型的清單，列出某個特定的目錄或子機碼。	
regdmp.exe	利用文件格式 Dump 指定機碼的內容，再加它機碼的子機碼及數值。	
regentry.hlp	許多 NT 的機碼及數值的文件說明。	

reg find.exe	搜尋 Registry，找出特定的數值；運作情形就像 grep 及 RegEdit 裡的 search 函式一樣	參本章稍後的「搜尋機碼」
regini.exe	根據你撰寫的 command script 內容新增、移除，或改變機碼	
regkey.exe	提供用來設定幾個日常參數（自動登入，快速获取使用者設定檔數量... 等等。）的使用者介面 (GUI)。	最好使用系統原則
regread.exe	讀取 HKLM 裡的某個子機碼並傳回機碼的數值。	被 reg.exe 所取代

表 8-3 (續)

工具	作用	注意
regsec.exe	設定機碼及其子機碼上的安全描述。	請參考本章稍早的「修正登錄安全存取控制列表」
reskey.exe	還原使用 SAVEKEY 儲存的機碼。	被 reg.exe 所取代
rktools.hlp	對 Resource Kit 裡的每項工具給予簡短的描述。	
regchg.exe	改變遠端機碼上的機碼數值。	被 reg.exe 所取代
savekey.exe	將機碼的數值儲存起來等待稍後重新載入。	被 reg.exe 所取代

## REG: 瑞士刀

我曾經聽過將 reg.exe 工具程式描述成「罐裝 RegEdit32」。它幾乎可以做到 RegEdit32 的所有功能，但它允許你從命令列【註】執行指令。這樣一來，當你想對 Registry 做些小變動時，你便可以利用這項工具，不需啟動 RegEdit32 便快速完成變動；它也允許你在登入的指令檔及批次檔裡嵌入 Registry 操作。（當然，你在第 7 章「如何使用 Registry 撰寫程式」裡學過如何使用 Registry，但對那些不太會使用 Perl 的人而言，reg 是一個較簡單的替代方案。）

若你曾經使用過 net 指令，你便會很快喜歡上 reg 的運作方式。就像 net 一樣，在使用 reg 時，你必須給它一個指令，這個指令可以從選項清單裡指定（query、

註 它相當有用，我可以經由它了解許多文法錯誤及拼字錯誤的情形。有了高品質，才能事半功倍，不是嗎？

add、update、delete、copy、save、backup、restore、load、以及 unload)，後面跟著一個或多個命令可以解譯的選擇性參數。這裡是一個小範例，在這個範例裡 reg 將會取得 query 指令給 HKLM 裡某個子機碼：

```
C:\ntreskit>reg query HKLM\Software\Qualcomm /s
```

```
Listing of [Software\Qualcomm]
```

```
[Eudora]
```

```
[Eudora\3.0.1]
```

## 查詢機碼

reg query 允許你查詢某個機碼的數值，或某個範圍機碼的所有數值。經由這個方法，你可以快速地檢查機碼是否擁有某個數值：

```
REG QUERY [rootKey\] key [\value] [machine] [/S]
```

### rootKey

選擇性的；指定以做為查詢基準的主機碼。可以是 HKLM、HKCU、HKCR、HKU 或 HKCC。若省略的話，預設值是 HKLM。

### key

指定 rootKey 下的機碼完整名稱。

### value

指定 key 下要查詢的數值。若省略的話，會顯示 key 下所有的機碼及數值。

### machine

要查詢的遠端機器名稱；若省略的話，預設值是本機。你只能查詢遠端機器上 HKLM 及 HKU。

### /s

查詢 key 的所有子機碼。

## 增加新機碼

reg add 會在 Registry 裡新增機碼和數值。你可以在現存機碼裡增加數值、增加不具有數值的新機碼，或是在某個機碼下建立新機碼和數值。若你要增加的機碼或數值已經存在，reg 將會發出警告。

```
REG ADD [rootKey\] key [\value=newValue] [machine] [dataType]
```

### rootKey

選擇性的；指定做為查詢基準的主機碼。可以是 HKLM、HKCU、HKCR、HKU 或 HKCC。若省略的話，預設值是 HKLM。

### key

指定 rootKey 下要新增機碼的完整名稱。

### value

選擇性地指定 key 下新增的數值名稱。若省略的話，在建立這個機碼時將不會隨之產生數值。

### newValue

新建立數值的內容。這個字串數值可以包含空白及特殊字元，但必須使用方括號將內容括起來。

### machine

要查詢的遠端機器名稱；若省略的話，預設值是本機。你只能查詢遠端機器上的 HKLM 及 HKU。

### dataType

要增加的新數值類型。可以是 REG\_SZ、REG\_MULTI\_SZ、REG\_EXPAND\_SZ 或 REG\_DWORD。若省略的話，預設值是 REG\_SZ。若你指定 REG\_DWORD，請將 newV alue 指定為十進位的數字。

例如，若你希望增加數值，以停用「撥號網路」的「儲存密碼」核取方塊 (checkbox)，請使用這個指令：

```
reg add SYSTEM\CurrentControlSet\Services\RasMan\Parameters\DisableSavePasswordValue=1
```

## 更新目前機碼

reg update 會更新現存機碼的數值。只要你有足夠的使用權限都可以更新任何數值，你可以根據父機碼的 ACL 判斷是否有足夠權限；若你想修改遠端機器的 Registry 資料，你必須對那台機器具有存取權限。若你修改不存在的數值，reg 將會發出警告。

```
REG UPDATE [rootKey\] key [\value=newValue] [machine]
```

### rootKey

選擇性的；指定做為查詢基準的主機碼。可以是本機上的 HKLM、HKCU、HKCR、HKU 或 HKCC，或者是遠端機器上的 HKLM 和 HKU。若省略的話，預設值是 HKLM。

### key

指定 rootKey 下要更新的機碼的完整名稱。

### value

指定 key 下的哪一個數值要更新。

### newValue

用來取代目前數值的內容。這個字串數值可以包含空白及特殊字元，但必須使用方括號將內容括起來。請使用十進位的數字來指定 DWORD 數值。

### machine

要查詢的遠端機器名稱；若省略的話，預設值是本機。你只能查詢遠端機器上的 HKLM 及 HKU 項目。

## 移除機碼

reg delete 會移除機碼或數值。在移除機碼時，這個指令會移除目標機碼下的所有子機碼及數值；但它在刪除任何內容之前，會詢問你是否確定要刪除。當你使用這個命令時，請多加小心。和 reg update 一樣，你只能刪除 ACLs（和 / 或遠端 Registry 設定）允許進行存取動作的機碼。

```
REG DELETE [rootKey\] key [\value] [machine]
```

### rootKey

選擇性的；指定做為查詢基準的主機碼。可以是本機上的 HKLM、HKCU、HKCR、HKU 或 HKCC，或者是遠端機器上的 HKLM 和 HKU。若省略的話，預設值是 HKLM。

### key

指定 rootKey 下要更新的機碼的完整名稱。

#### value

指定 key 下的哪個數值要移除。若省略的話，在 key 下的所有機碼及數值都會被刪除。

#### machine

要查詢的遠端機器名稱；若省略的話，預設值是本機。你只能查詢遠端機器上的 HKLM 及 HKU 項目。

### 複製機碼及數值

reg copy 可能是所有 reg 指令裡我最喜歡的一個項目，它可以大大地減少將設定從一個地方複製到另一個地方所花費的工夫。你可以利用這個指令，將一個數值或整個 hive 從原本位置複製到網路上的其它機器！這個指令可以很容易地將複製工作完成，用途不少，例如將檔案關聯集合複製到新的機器，或是調整一台機器好讓其設定符合其它的機器。

```
REG COPY [srcRootKey\] srcKey [\srcValue] [srcMachine] [destRootKey\] destKey  
[\destValue] [destMachine]
```

#### srcRootKey

選擇性的；指定要使用哪個主機碼來包含來源機碼。可以是 HKLM、HKCU、HKCR、HKU 或 HKCC。若省略的話，預設值是 HKLM。

#### srcKey

指定來源機碼的完整名稱。

#### srcValue

選擇性地指定 srcKey 下要複製的數值。若省略的話，srcKey 下的所有機碼及數值都會被複製。

#### srcMachine

遠端機器的名稱，做為要複製的來源；若省略的話，預設值是本機。你只能使用遠端機器的 HKLM 及 HKU 做為來源。

#### destRootKey

選擇性的；指定要將複製的機碼要位於哪個主機碼下。可以是 HKLM 或 HKU；若省略的話，預設值是 HKLM。

#### destKey

指定存放複製資料的機碼的完整名稱。

#### destValue

選擇性地指定要複製的數值的名稱；若沒有指定 srcValue 的話，這個值也可以忽略。

#### destMachine

做為複製目標的遠端機器名稱；若省略的話，預設值是本機。

在我安裝一個有名的網路郵件封包的試用版時，我相當擔心新版的程式會毀損到舊版的 Registry 設定。你可以使用一個快速的指令避免這個麻煩：

```
reg copy software\qualcomm\eudora software\qualcomm\eudora-3.0
```

這個指令會製作現存設定的備份，好讓我可以安心地安裝新版的程式，有了這層認知，我便可以在需要時很容易地回復先前的版本。

## 儲存及回存機碼

REGBACK 及 REGREST 這兩個工具程式允許你備份及還原整個 hive，但 reg 提供了一對類似的函式，增加了儲存及重新載入個別機碼的功能，達到與 RegEdit32 指令一樣的功能。你可以使用 reg save 或是 reg backup（這兩個有同樣的作用）儲存機碼及其數值，：

```
REG SAVE [rootKey\] key fileName [machine]
```

### rootKey

選擇性的；指定做為查詢基準的主機碼。可以是本機上的 HKLM、HKCU、HKCR、HKU 或 HKCC，或者是遠端機器上的 HKLM 和 HKU。若省略的話，預設值是 HKLM。

### key

指定 rootKey 下要更新的機碼的完整名稱。若省略的話，rootKey 的全部內容都會被儲存起來。

### fileName

用來容納儲存資料的檔案名稱。fileName 不能指定副檔名。

### machine

要查詢的遠端機器的名稱；若省略的話，預設值是本機。

你可以使用這個指令，很快地將目前的所有設定儲存一份複製：

```
reg save HKLM my-profile
```

然後你便可以在任何可以使用 hive 檔案的地方使用這個檔案。

你也可能會使用 reg restore 指令回存 hive。這個指令會使用一組新的數值集合覆寫現存的機碼，所以你在使用時必須特別小心（reg 會在覆寫到任何內容時，詢問你是否確定要執行這個指令）：

```
REG RESTORE fileName [rootKey\] key [machine]
```

### fileName

容納回存資料的檔案名稱，不具副檔名。

### rootKey

選擇性的；指定做為查詢基準的主機碼。可以是本機上的 HKLM、HKCU、HKCR、HKU 或 HKCC，或者是遠端機器上的 HKLM 和 HKU。若省略的

話，預設值是 HKLM。

#### key

指定機碼的完整名稱，所指定的機碼的子機碼及數值都將被取代。

#### machine

要查詢的遠端機器名稱；若省略的話，預設值是本機。你只能查詢遠端機器上的 HKLM 及 HKU 項目。

## 載入及卸除 hive

在第 5 章的「儲存及載入 Registry 機碼」段落裡解釋了如何使用 RegEdit32，將已儲存的機碼視為一般 hive 載入及卸除，而這些動作必須發生在 HKLM 或 HKU 下。reg 工具程式提供相同的功能，但也仍具有相同的限制。

你可以使用 reg load 指令載入 hive。和 reg restore 不同的是，reg load 會載入 hive，並將它增加到指定的機碼裡，而不是覆寫指定機碼的內容。所以你可以使用 reg load 載入 hive、編輯內容，以及卸除 hive，這期間完全不會影響到 Registry 的其它部份。（若你懷疑你怎麼會想這麼做，可以回去重讀本章開頭的「替新使用者帳戶建立預設值」段落。）下面是指令的形式：

```
REG LOAD fileName [rootKey\] key [machine]
```

#### fileName

指定要載入的 hive 檔案名稱，不具副檔名。你可以指定完整的本機路徑或 UNC 路徑。

#### rootKey

選擇性地指定要在哪一個主機碼下建立新 hive。可以是 HKLM 或 HKU。若省略的話，預設值是 HKLM。

#### key

指定接收新 hive 的機碼名稱；這個機碼必須是目前不存在的機碼，它將在隨

後建立。key 必須是 HKLM 或 HKU 下立即的子機碼 (即 HKLM 及 HKU 下第一層的子機碼)。

machine

要載入 hive 的遠端機器名稱。

例如，載入「替新使用者帳戶建立預設值」裡所建議的 ntuser.dat 這個 hive 檔，便會將 ntuser.dat 複製到 ntuser-default，接著使用這個指令：

```
reg load ntuser-default DefaultProfile
```

然後再視需要修改 hive。

在你處理完載入的 hive 後，你可以使用 reg unload 卸除這個 hive。它的指令文法相當簡單：

```
REG UNLOAD [rootKey\] key [machine]
```

rootKey

選擇性的；指定做為查詢基準的主機碼。可以是本機上的 HKLM、HKCU、HKCR、HKU 或 HKCC，或者是遠端機器上的 HKLM 和 HKU。若省略的話，預設值是 HKLM。

key

指定要卸除的機碼的完整名稱。key 必須是 HKLM 或 HKU 下立即的子機碼

machine

要卸除 hive 的遠端機器名稱；若省略的話，預設值是本機。

## 機碼及數值的比較

當你試著解決設定問題所造成的麻煩時，不妨檢查一下當掉的機器及一台正常運作的機器之間有何不同，這對於你解決問題是很有幫助的。若你沒有 resource kit 的

話，你可以將相關的 Registry 部份儲存成文字檔，然後使用工具如 windiff 來顯示這兩個檔案之間的不同。compreg 工具（在 NT 4.0 resource kit 包含）則提供了在命令列下參數的方式來比較 Registry 機碼的不同。這裡是它的運作方式：

```
COMPREG key1 key2 [-v] [-r] [-d] [-g] [-n] [-h] [-?]
```

- key1 指定要比較的第一個機碼的完整路徑。這個路徑可以包含機碼名稱（例如，\\ENIGMA\HK\KEY\_LOCAL\_MACHINE\SOFTWARE\LJL）。你可以不用拼出 Registry 主機碼的完整名稱，只要利用我們在本書中使用的簡稱法並去掉原本的「HK」即可；用上例來說明的話，你也可以指定 \\ENIGMA\lm\SOFTWARE\LJL 路徑，這樣可以少打一些字。若不指定主機碼的話，預設值便是 HKCU。
- key2 指定要比對的第二個機碼的完整路徑。可以是不同機器上與 key1 相同的路徑，或者是完全不同的路徑。若你只指定機器名稱，compreg 會使用與 key1 相同的路徑，但根據 key2 指定的電腦名稱來搜尋。
- v verbose 模式；列印出數值內容不同的機碼及內容相符合的機碼。
- r 遞迴處理那些只有一個子機碼的機碼。
- e 在結束時，將 errorlevel 設定給最後遇到的錯誤。在指令檔或批次檔裡使用這個參數（switch）可以測試 compreg 傳回的數值。
- d 不印出機碼的數值內容不同的數值；只印出機碼。
- n 黑白輸出（預設情形是使用多重顏色顯示）。
- ? 顯示簡短的 help 訊息

在兩台機器間找出不同之處的功能有時相當有用。雖然解決這些問題的討論會在第 9 章「Registry 小技巧」裡提供，我還是希望模仿目前的磁碟限制並將它加以修改。很不幸地，在我修改完它後它不會運作，而且我無法看出哪裡做錯了。下面的

```
compreg software\Microsoft\Windows\CurrentVersion\Policies\Explorer \\armory
```

會顯示我犯了錯誤，而我可以在它尚未對 Registry 造成損害之前將它修正。

## 機碼搜尋

有時你不得不採用暴力搜尋法。若你曾經使用過 `grep` 或 `findstr` 來找出你「已知」藏在你磁碟裡某處的資料，你一定會喜歡 `regfind` 這項工具程式。它在使用上非常具有彈性：它可以搜尋數值及機碼名稱或內容，可以只搜尋或搜尋到後取代，而且它熟知所有常用的 Registry 資料類型。這項彈性使得它比其它 resource kit 的工具程式要來得複雜一些：

```
REGFIND [-h hiveFile hiveRoot | -w win95Dir | -m \\machine]
        [-i tabStop] [-o outputWidth]
        [-p keyPath] [-z | -t dataType] [-b | -B] [-y] [-n]
        [searchString [-r replacementString]]
```

### -h hiveFile hiveRoot

指定到本機碼 hive 檔案（可使用 `reg save` 或 `RegEdt32` 產生檔案）的完整路徑。

### -w win95Dir

告知 `regfind`，在 `win95Dir` 目錄裡搜尋 Windows 95 使用的 `user.dat` 及 `system.dat` 這兩個 hive 檔案。

### -m machine

指定 `regfind` 要進行搜尋動作的 NT 機器。

### -i tabStop

設定 `tabstop` 的寬度；預設值是 4。

### -o outputWidth

告知 `regfind`，要使用何種寬度輸出。預設值是螢幕的寬度，或 240（若輸出會重新導入檔案的話）。

### -p keyPath

指定 `regfind` 從 `keyPath` 開始搜尋。你可以指定 `HKEY_LOCAL_MACHINE`、

HKEY\_USERS、HKEY\_CURRENT\_USER 或 USER；因為 HKCR 及 HKCC 會連結到 HKLM，所以這項要求並不會有什麼太大的遺漏。若你省略這個參數 ( switch ) 的話，regfind 會搜尋整個 Registry。

-z 搜尋遺漏 0 結束符號或擁有不合法長度的 REG\_MULTI\_SZ 或 REG\_EXPAND\_SZ 字串。

-t dataType

強制 regfind 只搜尋這個資料類型的數值。你可以指定下列類型中的一項：REG\_SZ、REG\_MULTI\_SZ、REG\_EXPAND\_SZ、REG\_DWORD、REG\_BINARY 和 REG\_NONE。若不指定類型的話，regfind 會搜尋所有的 SZ 類型。

-b 告知 regfind 搜尋指定字串時，除了搜尋 -t 指定的 SZ 類型外，還要搜尋位於 REG\_BINARY 數值裡的字串。

-B 與 -b 相同，但除了 Unicode 外還會搜尋 ANSI 字串。

-y 在 SZ 搜尋中使用這項參數的話，會強制 regfind 執行大小寫不同的搜尋。若搜尋類型為 REG\_DWORD、REG\_BINARY 和 REG\_NONE 的話，忽略這項參數。

-n 不只搜尋內容，還搜尋機碼及數值名稱。-n 及 -t 這兩個參數一次只能選用一個。

searchString

要搜尋的字串。將字串將雙重括號包裝起來，便可以搜尋內部具有空白、括號... 的字串。若不指定搜尋字串的話，搜尋結果將會尋找所有指定類型的數值。在搜尋 REG\_DWORD 時，你可以指定要使用十進位或是十六進位，前端放置 0x。在搜尋二進位數值時，你必須提供一個長度位元，選擇性地跟著一連串 DWORD，在裡面放入要搜尋的資料。

### -r replacementString

使用 replacementString 取代 searchString。searchString 及 replacementString 必須具備相同類型，但長度可以不同。應用 -r 有幾個限制：

- 你可以利用與 searchString 相同的方式指定 replacementString。但若你的 searchString 只具有 REG\_BINARY 的長度，你便不可能使用 -r。
- 若你同時指定 -z 和 -r，取代的字串（replacement string）將會被忽略。regfind 會修正遺漏結束符號或長度錯誤的字串，而不會取代任何字串。
- -r 代表不再詢問是否確定要取代，所以除非你確定要進行取代工作，否則最好不要使用這項參數。

因為這個指令有點複雜，來個範例應該有助於了解這個指令的運作方式。讓我們試著找出所有內容或名稱包含 Mac 字串的機碼：

```
C:\ntreskit>regfind -y -n Mac
Scanning \Registry registry tree
Case Insensitive Search for 'Mac'
Will match values of type: REG_SZ REG_EXPAND_SZ REG_MULTI_SZ
Search will include key or value names
\Registry
  Machine
    SOFTWARE
      Microsoft
        AsyncMac
        Exchange
        Client
          Mac File Types
        Shared Tools
          Text Converters
            Export
              MSWordMac4
              MSWordMac5
              MSWordMac51
            Import
              MSWordMac
    SYSTEM
```

```
ControlSet001
  Services
    AsyncMac
    AsyncMac2
    EventLog
    System
      AsyncMac
ControlSet003
  Services
    AsyncMac
    AsyncMac2
    EventLog
    System
      AsyncMac
Users
  S-1-5-21-1944135612-1199777195-24521265-500
    Software
      Microsoft
        Ntbackup
          Backup Engine
            Process Macintosh files = 1
            Machine Type = 0
        Telnet
          Machine1 = fly.hiwaay.net
          LastMachine = hq
          Machine2 = hq
```

regfind 的唯一缺點是，它無法接受運算式或萬用字元，做到與 findstr 及 grep 一樣的功能。撇開這項限制不談，當你必須要找出一個已數值卻不知道路徑的機碼時，它倒是個很好的工具。

## 使用 NTREGMON 監看 Registry

若你請教一位私家偵探，收集證據最好的方法是什麼，他可能會告訴你一個很簡單的答案：靜靜地觀察以及等待。很不幸地，使用 RegEdt32 及 RegEdit 來觀察 Registry 改變的情形實在是件吃力不討好的工作。除非你在事前已知道要觀察哪些機碼及數值，否則你很難監視個別機碼及數值的改變情形，而且也很難判斷哪些應用程式、程序或驅動程式改變了目前觀看的設定。

Mark Russinovich 及 Bryce Cogswell 解決了這個問題，造福了各地的系統管理員及程式設計師。他們撰寫了一個工具程式稱為 NTREGMON (你可以從網站 <http://www.ntinternals.com/> 取得)，這個工具程式提供了監看系統裡的 Registry 存取情形的方法。它可以監看的項目包括讀取、寫入以及查詢，除此之外，它還會將這些項目記錄在記錄檔 (log) 裡，方便你在需要時閱讀這些內容；它也可以根據你提供的過濾方式來限制所記錄的 Registry 存取情形。NTREGMON 會處理並判斷誰修改了某個機碼或數值，也可以用來觀察 NT 處理 Registry 資料的方式。

在執行這個應用程式時，NTREGMON 會先安裝一個小小的設備驅動程式；這個驅動程式會替所有的 Registry API 安裝 hook，所以它可以看到參數呼叫是如何傳進系統以及系統傳回了怎樣的回應。NTREGMON 應用程式本身的工作只是開啟這個設備驅動程式並等待它送回擷取到的資料而已。

## 利用 RegEdt32 偷看

REGMON 並不是監看 Registry 的唯一方法。這裡是另一個方便的技巧，允許你觀看 SAM 及 SECURITY 這兩個 hive (在 NT 3.51 及 NT 4 的機器上)，而這兩個 hive 通常是無法存取的。

1. 啟動「排程」服務，並使用 SYSTEM 帳戶登入，在「服務」對話方塊裡選擇「系統帳戶」圓鈕。
2. 在「排程」開始執行後，開啟 NT 的命令列視窗並使用 at 指令排定 RegEdt32 的時間表。例如，若你決定要在 1:35 P.M. 動身，就將 RegEdt32 排定在 1:36 P.M. 好了。就像這樣：

```
at 13:36 /interactive regedt32.exe
```

在約定的時間到了時，RegEdt32 將會啟動，但它必須在 SYSTEM 帳戶下才會執行，而不是一般帳戶即可。HKLM\SYSTEM 及 HKLM\SAM 將會被啟動，所以你可以開啟並檢查它們的內容。請不要期望看到什麼有意義的內容，因為它們的內容全都是二進位資料。要注意的是，「不要在這些 hive 裡編輯任何項目」。

#	Process	Request	Path	Result	Other
588	tspsrv.exe	QueryValue	HKLM\SYSTEM\CurrentControlSet...	SUCCESS	"AT"
589	tspsrv.exe	QueryValue	HKLM\SYSTEM\CurrentControlSet...	NOTPOU...	
590	tspsrv.exe	CloseKey	HKLM\SYSTEM\CurrentControlSet...	SUCCESS	Key: 0xE11CCA20
591	Explorer.exe	OpenKey	HKCU\Control Panel\Input Method	SUCCESS	Key: 0xE11CCA20
592	Explorer.exe	QueryValue	HKCU\Control Panel\Input Method\sh...	SUCCESS	"1"
593	Explorer.exe	CloseKey	HKCU\Control Panel\Input Method	SUCCESS	Key: 0xE11CCA20
594	Explorer.exe	OpenKey	HKCU\Control Panel\Input Method	SUCCESS	Key: 0xE1BF1DA0
595	Explorer.exe	QueryValue	HKCU\Control Panel\Input Method\sh...	SUCCESS	"1"
596	Explorer.exe	CloseKey	HKCU\Control Panel\Input Method	SUCCESS	Key: 0xE1BF1DA0
597	Explorer.exe	OpenKey	HKCU\Control Panel\Input Method	SUCCESS	Key: 0xE11CCA20
598	Explorer.exe	QueryValue	HKCU\Control Panel\Input Method\sh...	SUCCESS	"1"
599	Explorer.exe	CloseKey	HKCU\Control Panel\Input Method	SUCCESS	Key: 0xE11CCA20
600	regedt32.EXE	CloseKey	HKCR\CLSID	SUCCESS	Key: 0xE17C66C0
601	regedt32.EXE	CloseKey	HKCU\Software\Microsoft\Windows...	SUCCESS	Key: 0xE1BE2720
602	regedt32.EXE	CloseKey	HKLM\SYSTEM\CurrentControlSet...	SUCCESS	Key: 0xE138C5A0
603	csrss.exe	OpenKey	HKCU	SUCCESS	Key: 0xE138C5A0
604	csrss.exe	OpenKey	HKCU\AppDataEvents\Schemes	SUCCESS	Key: 0xE1BE2720
605	csrss.exe	QueryValue	HKCU\AppDataEvents\Schemes(Default)	SUCCESS	"Current"
606	csrss.exe	CloseKey	HKCU\AppDataEvents\Schemes	SUCCESS	Key: 0xE1BE2720
607	csrss.exe	CloseKey	HKCU	SUCCESS	Key: 0xE138C5A0
608	csrss.exe	OpenKey	HKCU	SUCCESS	Key: 0xE138C5A0
609	csrss.exe	OpenKey	HKCU\AppDataEvents\Schemes\Apps.Def...	SUCCESS	Key: 0xE1BE2720
610	csrss.exe	QueryValue	HKCU\AppDataEvents\Schemes\Apps.Def...	SUCCESS	""
611	csrss.exe	CloseKey	HKCU\AppDataEvents\Schemes\Apps.Def...	SUCCESS	Key: 0xE1BE2720
612	csrss.exe	CloseKey	HKCU	SUCCESS	Key: 0xE138C5A0
613	csrss.exe	OpenKey	HKCU	SUCCESS	Key: 0xE138C5A0
614	csrss.exe	OpenKey	HKCU\AppDataEvents\Schemes\Apps.Def...	NOTPOU...	
615	csrss.exe	QueryValue	HKCU(Default)	NOTPOU...	
616	csrss.exe	CloseKey	HKCU	SUCCESS	Key: 0xE138C5A0
617	csrss.exe	OpenKey	HKLM\Software\Microsoft\Windows...	SUCCESS	Key: 0xE138C5A0
618	csrss.exe	OpenKey	HKLM\Software\Microsoft\Windows...	NOTPOU...	
619	csrss.exe	QueryValue	HKLM\Software\Microsoft\Windows...	SUCCESS	"D:\WINNT\Media"
620	csrss.exe	CloseKey	HKLM\Software\Microsoft\Windows...	SUCCESS	Key: 0xE138C5A0
621	regedt32.EXE	CloseKey	HKCU\Software\Microsoft\RegEdt32...	SUCCESS	Key: 0xE13A0FE0
622	regedt32.EXE	CloseKey	HKCR	SUCCESS	Key: 0xE135C5E0
623	regedt32.EXE	CloseKey	HKCU	SUCCESS	Key: 0xE171A860
624	regedt32.EXE	CloseKey	HKCU\Software\Microsoft\Internet Ex...	SUCCESS	Key: 0xE12E3EE0
625	regedt32.EXE	CloseKey	HKLM\System\ControlSet001\Hardwa...	SUCCESS	Key: 0xE13583C0
626	regedt32.EXE	CloseKey	HKU	SUCCESS	Key: 0xE17AC2A0

圖 8-8 : NTREGMON 主要介面

在清單裡的每一個項目都會顯示六個欄位資料：

- ID 是 NTREGMON 給定的序號。第一個登入的項目會取得 ID #1，而 ID 從此開始遞增。但這些 ID 是設備驅動程式給定的。若事件發生的速度快過 NTREGMON 將它們放入清單的速度，序號裡將會發生不連號的情形。
- Process 顯示產生要求的程序 (process) 名稱。因為 DLL 會被載入程序的位址空間裡，所以 NTREGMON 只會顯示程序的名稱，而不是提出要求的 DLL 的名稱。
- Request 顯示程序要求的動作。通常會看到 QueryValue、OpenKey、CloseKey 和 SetValue，但 NTREGMON 也會列舉 (enumeration)、安全改變 (security change)，以及其它透過第 7 章所說明的 Registry API 可以取得的 Registry 服務。
- Path 顯示提供的路徑，這也是要求的一部份。NTREGMON 會連同最頂層的主機碼一起將路徑顯示出來。
- Result 顯示呼叫的 Registry API 所傳回的數字結果碼。通常，你將會在這裡看到 "SUCCESS" 或 "NOTFOUND" 這兩個結果碼。
- Other 是一個概括性的欄位。對 Registry 呼叫而言，它會傳回資料，NTREGMON 會在這裡顯示符合類型的資料。你將會看到字串數值被用括號括起來，但 DWORD、HKEY、以及其它二進位資料則會顯示成一個十六進位數字的區塊。你可以自行決定是否要解譯二進位資料並將它改為有意義的資料。

### 控制你所見的畫面

除了改變每一個的大小位置外，NTREGMON 並沒有提供其它改變使用者介面的方法。「Events」、「Clear Display」指令會消除目前畫面上登入的 Registry 存取記

錄，而「Events」 「Auto Scroll」指令則會切換 NTREG MON 是否要自動捲動顯示清單，以顯示目前新增加的項目。撇開這兩個指令不談，NTREGMON 可以說是一個平凡的應用程式 — 它遵循微軟的 UI 指示，而且完全不提供任何鈴聲或笛聲的音效。

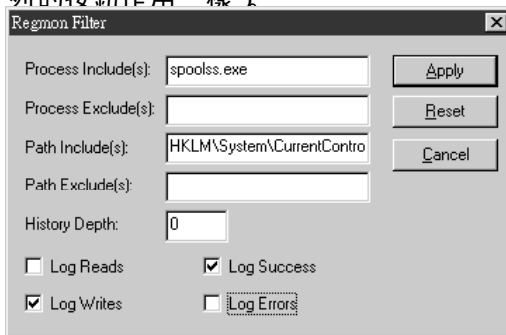
## 擷取及過濾

只需要兩個程序，便可以利用 NTREGMON 判斷 Registry 裡發生了什麼事。第一個步驟是選擇性的：你可以利用 capture filter 選擇欲觀察（或不希望觀察）的事件。NTREG MON 會在第二個步驟 - 真正去擷取事件時應用這個 filter。

### 啟動及關閉擷取功能

在你第一次啟動 NTREGMON 時，它處於 capture 模式。若你只等了一會兒便讓它執行，將會看到偶爾會有 Registry 的存取出現在它的視窗裡；若你切換到「檔案管理員」並開啟一個檔案，甚至只要在「我的電腦」視窗裡按下一個圖示，都可以看到更多的 Registry 存取記錄情形。若你離開 NTREG MON 的 capture 模式，它很可能會擷取大量的資料，遠超過你真正想要尋找的資料量。

最好的減少資料負載量的方法很簡單：在你不需要時，關閉 capture 模式。「Events」 「Capture Events」指令會切換 capture 模式的關閉 / 開啟狀態（如工具列的按鈕作用一樣）。



」選單裡的第一個指令是大有來由的：因「Regmon Filter」對話方塊（見圖 8-9）讓你選擇欲觀察的事件，以及不希望看到哪些事件。下面

- Process 過濾要求的程序名稱。在圖 8-9 裡，我只希望看到 spoolss.exe 造成的呼叫以及它載入其程序空間裡的 DLL。你可以在這個欄位使用萬用字元，而大小寫的不同在這裡是被忽略的。

圖 8-9 : NTREGMON 的過濾對話方塊

- Path Include 及 Path Exclude 結合起來便可以允許你相當精確地控制要記錄的機碼路徑。在 Path Include 裡指定的路徑，去掉 Path Exclude 裡列出的路徑，這個路徑以下的子機碼都會被監視。

例如，若你在「Path Include」裡輸入 HKLM\SOFTWARE，而在「Path Exclude」裡輸入 HKLM\SOFTWARE\Microsoft 的話，記錄將會顯示所以位於 HKLM\SOFTWARE 裡的機碼，去掉位於 Microsoft 下的機碼。這些路徑也可以包含萬用字元。

- History Depth 是一個錯誤的標識。它應該說是「Number of entries to keep」或其它類似的功能；它實際上會決定 NTREGMON 在清除其內容清單之前保留多少行資料。預設的數值是 0，代表新資料會直接加到結尾，但你可以提供一個特殊數值（如 "10" 表示 NTREGMON 只保留最後 10 個稽核事件）。
- Log 的核取方塊（checkbox）決定 NTREGMON 要記錄哪些動作。預設情形下，它會記錄 Read、Writes、Successes 和 Errors，但你可以調整這些項目，選出真正想觀察的資料。

---

註 正解：沒錯，它的確如此。事實上，它甚至會寫出不是印表機的內容集合裡的資料。

圖 8-9 顯示一組 filter 標準，而圖 8-10 則顯示利用這個 filter 標準擷取到的事件；我正在嘗試若只改變印表機中的一項內容，printer spooler 是否會將所有的印表機內容寫出。【註】

ID	Process	Request	Path	Result	Time
1	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\ChangeID	SUCCESS	6/25/05 10:10:00
2	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\Name	SUCCESS	6/25/05 10:10:00
3	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\ShareName	SUCCESS	6/25/05 10:10:00
4	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\Status	SUCCESS	6/25/05 10:10:00
5	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultPriority	SUCCESS	6/25/05 10:10:00
6	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultPrinter	SUCCESS	6/25/05 10:10:00
7	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterName	SUCCESS	6/25/05 10:10:00
8	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterType	SUCCESS	6/25/05 10:10:00
9	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterDriver	SUCCESS	6/25/05 10:10:00
10	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultDevMode	SUCCESS	6/25/05 10:10:00
11	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\FriendlyName	SUCCESS	6/25/05 10:10:00
12	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\Priority	SUCCESS	6/25/05 10:10:00
13	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultPriority	SUCCESS	6/25/05 10:10:00
14	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultPrinter	SUCCESS	6/25/05 10:10:00
15	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterName	SUCCESS	6/25/05 10:10:00
16	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterType	SUCCESS	6/25/05 10:10:00
17	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterDriver	SUCCESS	6/25/05 10:10:00
18	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultDevMode	SUCCESS	6/25/05 10:10:00
19	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\FriendlyName	SUCCESS	6/25/05 10:10:00
20	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\Priority	SUCCESS	6/25/05 10:10:00
21	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultPriority	SUCCESS	6/25/05 10:10:00
22	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultPrinter	SUCCESS	6/25/05 10:10:00
23	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterName	SUCCESS	6/25/05 10:10:00
24	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterType	SUCCESS	6/25/05 10:10:00
25	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\PrinterDriver	SUCCESS	6/25/05 10:10:00
26	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\DefaultDevMode	SUCCESS	6/25/05 10:10:00
27	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\FriendlyName	SUCCESS	6/25/05 10:10:00
28	spoolsv.exe	SetValue	HKLM\System\CurrentControlSet\Control\Print\Printers\Epson\Priority	SUCCESS	6/25/05 10:10:00
29	Explorer.exe	SetValue	HKLM\Printer\DevMode\Epson	SUCCESS	6/25/05 10:10:00

法只  
存。就  
不  
成檔  
習之  
些儲

圖 8-10：過濾之後的事件