NetVista Thin Client Manager V2R1

Using CompactFlash Cards - A Tutorial

August 2000

IBM NetVista Thin Clients
Access for today, flexibility for tomorrow

What is a Flash Card based Solution?

- **What is a CompactFlash Card**
  - A local storage device on which you can store applications and code
- **Flash Card can contain:**
  - Operating system (kernel)
  - Applications
  - Configuration files
- **When to consider a flash solution**
  - Where booting from a local server is not available
  - Where booting across a WAN facility is not practical
  - Where network traffic is already at a maximum
- **Main benefits**
  - Speed up boot time by fetching kernel and applications from a local device
  - Offload network of the download traffic caused by a boot
- **Booting from a Peer Station**
  - Not every station requires its own flash card
  - One station can server as a boot server to other stations

Notes

Objective:
This presentation's objective is to explain and illustrate how a flash card based thin client operates, the components that are involved, and the different ways that it can be configured. This document is meant as a learning vehicle rather than a reference document. For up-to-date information and complete official product reference, use the Advanced Information document available from www.ibm.com/nc/pubs.

A CompactFlash card is a storage device inserted into the workstation on which data can be permanently recorded. It can therefore be used to contain a copy of the operating system and of the applications and have those loaded from the local storage device instead of having to be downloaded across the network.

The flash card can also contain configuration files, in which case it becomes basically a kiosk based solution.

A flash card based solution is a good fit where booting from a local server is not available or where booting across a WAN facility is not practical. It is also a good fit when network traffic is already at a maximum and additional network traffic is to be avoided.

The main benefits of using a flash card are that the boot time can be faster since the load is from a local device instead of the network, and that the network traffic caused by a boot is much reduced, or even eliminated entirely.

Note that not all stations require to have their own flash card since a station does have the ability to boot from a peer station. This requires that the local LAN not be at maximum capacity already since a boot does cause local traffic to increase but it is still a good fit when the local LAN is not at capacity as it lowers the number of flash cards that need to be used.

See www.ibm.com/nc and look for Accessories and Upgrades to see the list of CompactFlash cards that have been tested by IBM with the N2200 and N2800 machines.

Centrally Managed Flash Images

- **Thin Client Manager supports Flash Image management**
- **System Administrator decides what goes on the flash:**
  - Citrix Independent Computing Architecture (ICA) Client
  - ICA Remote Application Manager
  - Emulators
  - Browser
  - Java Virtual Machine (JVM)
  - Peer boot capability
  - Kiosk configuration files
- **Multiple flash images are supported:**
  - TCM configuration for creating flash images for thin client clients
- **Flash image is customizable:**
  - Flash card Bill-of-Materials can use system BOMs and customized BOMs
  - Custom files and applications may be added
- **Automatic update of flash images**
  - System Administrator decides when flash clients are updated
  - Clients automatically update their flash when directed to (at boot time)

Notes

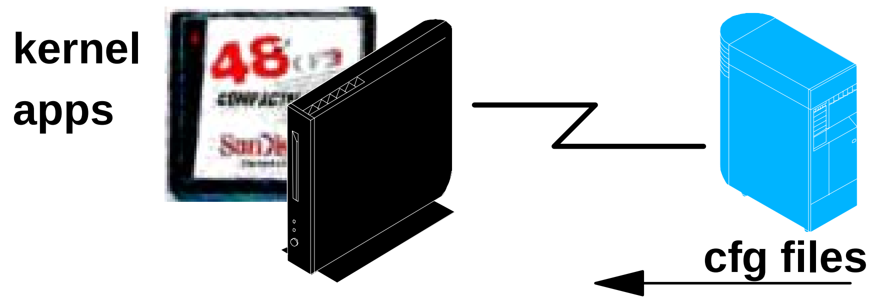
Using the NSM GUI interface, the administrator creates flash images and determines what components are to be included in each flash image.

Multiple images can be created and NSM settings determine which flash image the client gets. This is determined by the "Flash Image Directory" in the Boot Parameters under the Hardware => Workstations Setup task.

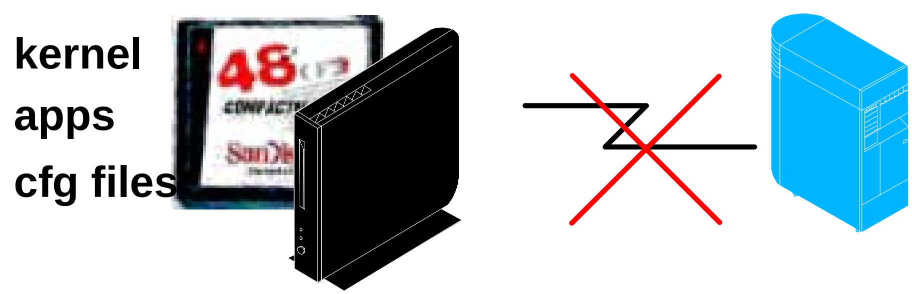
The BOM files determine which components get included in an image. There are system BOM files, which should not be changed if at all possible, and there are customized BOMs that can be created and edited to include custom applications and files in a flash image.

Typical Flash Card Scenarios

- **Scenario 1**
  - Kernel and Applications on the Flash Card
  - Configuration Files on Network Server(s)
  - Users Authenticate with UserID/Password on a server to allow specific user preferences



- **Scenario 2 (Kiosk Mode)**
  - Kernel and Applications on the Flash Card
  - Configuration Files also on the Flash Card
  - No user authentication
  - User home directory in memory (Non-Permanent)



Notes

What are some typical scenarios when using a flash card based solution?

One way of using a flash card is to have the operating system (kernel) and applications (such as the browser) reside on the Flash Card. At boot time, the station loads these components from the flash card instead of from the network. However, the station obtains its terminal configuration and user configuration files from a server on the network and the user authenticates from an authentication server with a userid and password, which permits downloading of user-specific configuration files.

In essence, the operation is similar to a full network based operation except that the larger components that need to be loaded on the station are loaded from the local flash card.

Another typical scenario is to have an operation which is entirely based on the flash card. This means loading even the configuration files from the flash card. The main difference and impact, in this case, is that the configuration files cannot be tailored to a specific user. There is no user authentication and users therefore all use the same user preferences. In fact, there is also no user specific home directories to store user specific data such as bookmarks or color preferences, so every user is the same. This is basically a kiosk mode operation, which is a good solution in many cases when there is no desire or need for specific user preferences to be maintained.

How is a Flash Card Updated?

- **Update from the network**
  - A list of components (BOM file) is kept on an update server and on the station, and they are compared at boot time
    - Can compare BOM files at every boot
    - Can also turn off updates if required
  - Only changed components are downloaded to the flash card
  - Requires network connection to an update server
  - All files can be updated, kernel, applications as well as configuration files
- **Manual Update**
  - Can also update flash cards at a central site
  - Then physically exchange flash cards in the units
  - Not a typical solution
  - Not practical in large networks

Notes

If there are components on the flash card that need updating, such as a new version of the kernel for example, or a new version of the browser, or even changes to the configuration files, how is this accomplished?

Flash cards can always be updated at a central site and then distributed in order to be manually and physically exchanged in the stations, but that is not a practical approach unless there are only a few units concerned. It is also costly in manpower as well as flash cards, the number of which needs to be increased.

The practical solution is to perform the updates via the network. A list of components (BOM file) is maintained on an update server and on the station, and they are compared at boot time, when configured to do so. Each file is identified with a time stamp, and if the time stamp for a file on the server is more recent than the time stamp on the unit, this is an indication that the file is new.
This comparison of files can be done at every boot but it can also be turned off and on via the Thin Client Manager.
If there are changes required, it is not the entire flash card which is updated but only the individual files that are in need of an update.

What is Kiosk Mode?

- **Operating without the user having to logon.**
  - This means operating without user specific preferences
  - User cannot save his own color preferences, bookmarks, ...
- **Designed for lobby units or where all units are the same**
- **Different ways of setting up kiosk**
  - Typically, meant for a single full screen application
  - Can be set up for multiple applications without a desktop
  - Can also be set up for full desktop and multiple applications
- **Can be network based or flash card based**
  - Well suited to flash card operation
- **Two types of kiosk mode**
  - True Kiosk Mode
  - Suppressed Login mode (to retain user preferences capability on a per station basis)

Notes

What is kiosk mode? The typical kiosk mode is a lobby unit for example where any user can walk up to the workstation and use it without having to supply a userid and password. It is therefore a mode of operation where the session is not associated with any specific user and where there are no user preferences, such as bookmarks, that can be stored.

The important characteristic of kiosk mode is that there is no User ID and Password that need to be supplied in order to gain access to the workstation.

There are three ways of setting up for kiosk operation:

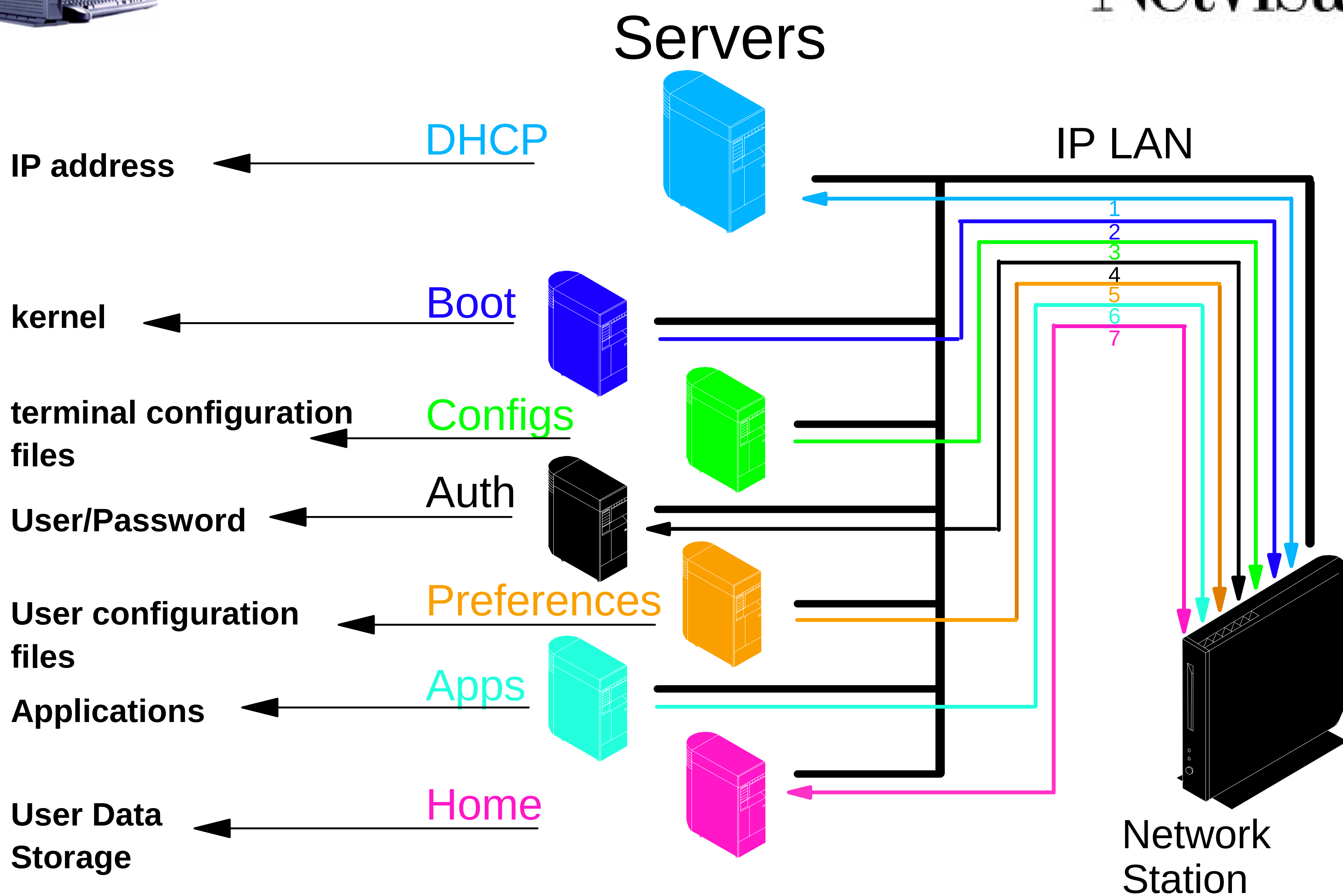
- The most common setup of a kiosk is to bring up a single application, such as a 3270 emulator for example, or a browser, and to bring it up in fullscreen mode as the only application available.
- The unit can also be set up for multiple applications, but without a desktop, where the user can switch from one application to the other using the Alt-Tab key.
- Finally, it can also be set up for full desktop, with a launch bar with folders and icons used to launch multiple applications.

Kiosk mode does not have to be flash card based; it can be either network based or flash based. However, kiosk mode is particularly well suited to flash card based operation because of the absence of specific user configuration files.

Finally, there are also in fact two types of kiosk mode. One is called true kiosk mode, and the second is called suppressed login mode. The second can be used when there is a need to remove some of the limitations of the true kiosk mode and where there is a need to retain user preferences capability on a per station basis.

In this discussion of flash card based operation, we do not need to go into further details of the kiosk operation. For more details, see the separate specific presentation on Using Kiosk Mode.

Review of Boot Sequence



Notes

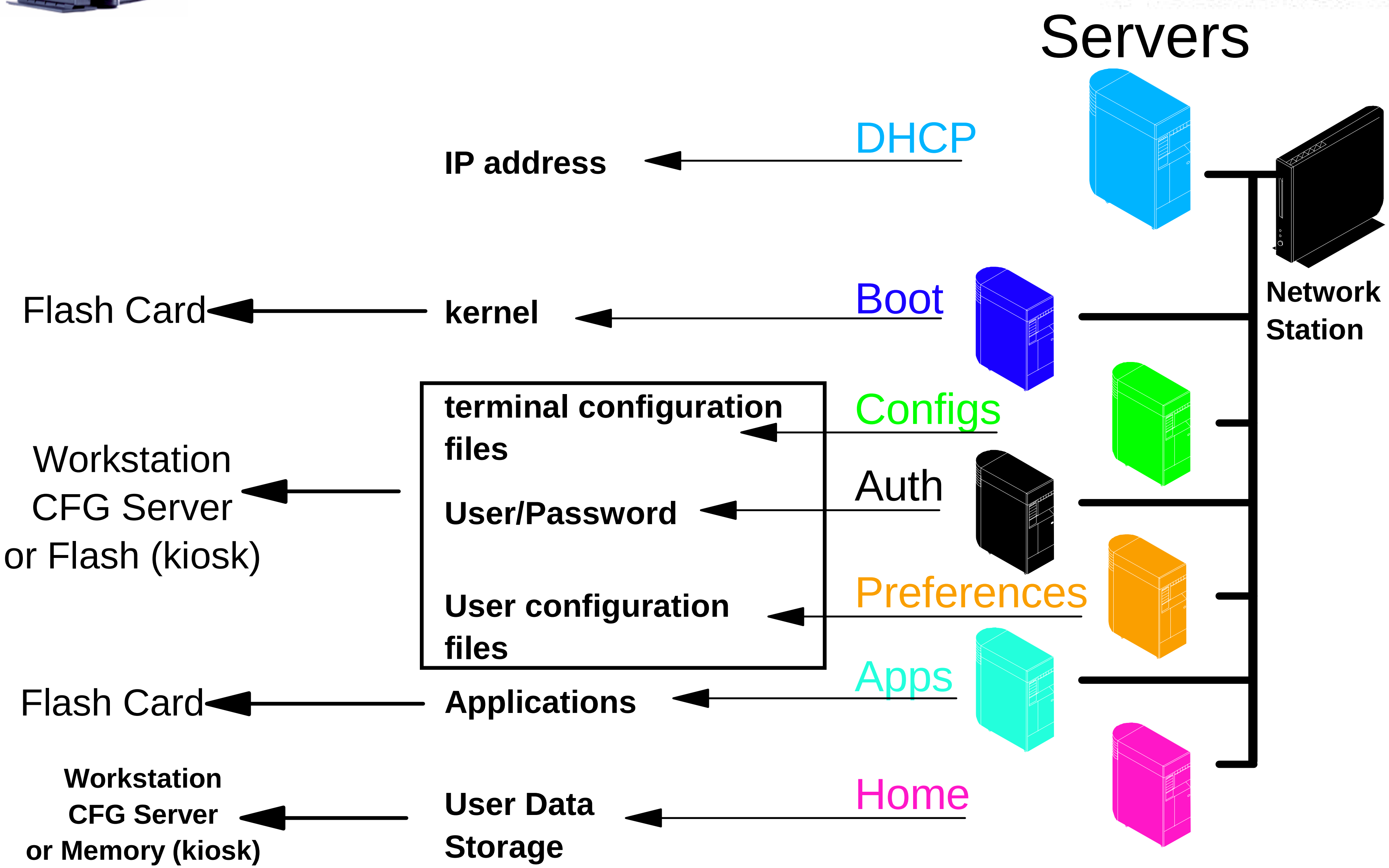
Before we enter into an example of creating a flash card, it is important to review a few concepts in order to refresh our memory regarding the different types of files that are required during boot time, and where these files are obtained from.

This chart is extracted from the Architecture presentation, which we suggest you review if you have not done so in the past. If you did, this is meant as a one page summary.

This diagram illustrates the different logical servers that may be used when booting a workstation. The sequence is as follows:

- A DHCP server is accessed to get the required network data such as the IP address of the station, etc. or the network data can be entered in NVRAM if a DHCP server is not available.
- The operating system (kernel) is loaded from the boot server
- Terminal configuration files (non user specific) are loaded from the workstation configuration server
- The user is authenticated by the authentication server (user ID and password)
- The user configuration files are loaded from a preferences server (today, this is the same as workstation configuration server)
- Then applications are loaded from the application server (today, this is the same as the boot server)
- And finally, the user's home directory (today this is the same as the workstation configuration server) is used to store any preferences that the user may change.

Boot Sequence



Notes

With that in mind then, let us try to simplify the process by talking about two main locations (well, actually three if we count the home directory) where the components required by a thin client can be located and loaded from.

- In basic flash mode, the normal setup is that the kernel and applications are located on the flash card, and the configuration files and home directory are located on a workstation configuration server on the network.

- In kiosk flash mode, all files are located on the flash card, except that the user home directory is actually directed to a file system that is mounted in memory (not on the flash card) and anything that gets stored on the home directory during a session disappears after a reboot.

There may be a few more subtle variations that could be introduced here but that would be for an advanced setup which is outside the scope of this presentation.

User's Home Directory

- **The User's Home Directory is used to store such things as:**
  - User Application Preferences
  - Cache
  - Bookmarks
  - History of URLs accessed
  - Cookies
- **When operating in Network Mode**
  - Located in Userbase/Home/<Username> on the workstation configuration server
- **When Operating in Kiosk Mode**
  - Located in RAM (volatile memory)

Notes

So what is the user's home directory used for?

It is used to store data that is specific to a particular user, such as:

- User Application Preferences
- Cache
- Bookmarks
- History of URLs accessed
- Cookies

When operating in Network Mode, the user's home directory is normally located on the workstation configuration server in the $Userbase/Home/<Username> directory.

When Operating in Kiosk Mode, since there is no actual workstation configuration server because the configuration files are loaded from the flash card, the user's home directory cannot really be placed on the flash card since it is volatile data that does not need to be retained and therefore should not take valuable space on the flash card. It is therefore located in RAM and the information simply disappears after every boot.

Possible Scenarios/Combinations

- **Kiosk Mode can be setup in three possible modes:**
  - Single Application
  - Multiple Applications
  - Full Desktop
  - Each mode can be Network Based or Flash Based
- **Flash Card Based Operation can be setup in two possible modes:**
  - Only boot server files (kernel and applications) on the flash card
    - Configuration files on a network configuration server (and auth server)
  - All files (kernel + configuration files) on the flash card
    - Only possible in kiosk mode since user authentication is not performed
  - Note: Updates to the flash card are possible in all modes, except when booted from DHCP that has been enabled via the simple configuration menu (because that menu uses default values for the boot server and configuration server). If DHCP is enabled via the Advanced Configuration menu, all required options can be configured to specify an update server, even in kiosk mode.

Before we carry on talking about flash and kiosk, it may be a good time to attempt a summary of the different configuration and setup possibilities.

First, it is important to realize that Kiosk Mode can be setup in three possible ways:

- Single Application mode, where only on application (an emulator session for example) comes up fullscreen when the station is booted.
- Multiple Applications mode, where two or more applications are automatically started at boot time. Typically, these are started in fullscreen and the user can toggle between the applications with the Alt-Tab key. Or they can also be minimized (the window manager is still active). However, closing any one of these applications causes a restart of all the applications.
- Full Desktop mode, where the normal desktop with a launch bar is displayed and the user is given one or more folders and icons to select. The user can start and stop applications at will.

For each of the above modes, the station can operate either entirely off the network or entirely off a flash card. In fact, kiosk mode is particularly well suited to flash card based operations since there is no need for user authentication.

As for flash card based operation, it can be setup in two possible modes:

- In the first case, only the boot server files (kernel and applications) are stored on the flash card because they are the largest files. Configuration files remain on a network configuration server and the user authenticates with a userid and password to a network server.
- In the second case, all files, that is kernel, applications and configuration files, reside on the flash card. This is only possible in kiosk mode since user authentication is not performed.
- Note that updates to the flash card are possible in all modes, except when booted from DHCP where DHCP has been enabled using the simple configuration menu (this is because, in that case, the firmware uses standard default values and a second configuration server therefore cannot be specified).

Basic Flash Card Creation - Steps

- **Step 1. Create a Flash Image on the Server**
  - After making all custom configurations changes that are required
- **Step 2. Enable Flash Card Creation and Update in NSM**
- **Step 3. Configure NS Boot for Flash Operation**
  - First boot server as flash
  - Secondary boot server as Network
- **Step 4. Reboot Network Station**
  - Unit fails to boot from flash
  - Unit boots from Network as a backup solution
  - This triggers the flash card creation
- **Step 5. Network Station Reboots as a Flash Unit**

After all this theory, it is now time to move on to more specific examples. The best way to start understanding the components of flash is to follow the simple basic procedure of creating a flash image.

So here are the steps that we will illustrate in the next few charts:

Step 1. Create the Flash Image on the server using NSM. Note that normally, before you create an image, you might want to use NSM to customize certain aspects of NSM, when you will be operating units in kiosk mode, because the files containing these configuration parameters will actually be part of the image you are creating. In this example, we assume that the default configuration is all we need for now. In a later example, we will discuss customization.

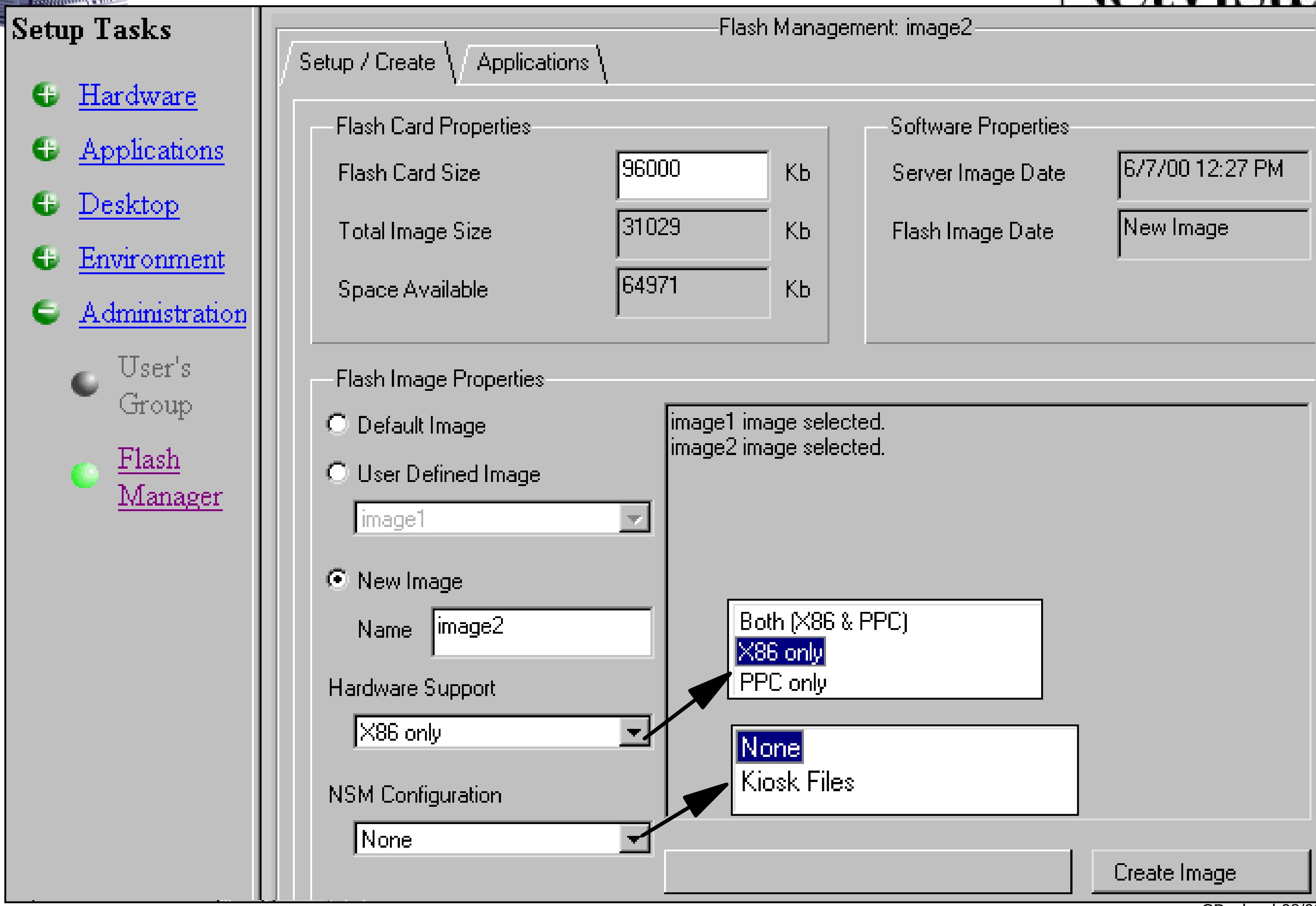
Step 2. Enable Flash Card Creation and Update in NSM. That is, now that we have an image, we must include the configuration parameters that will trigger the writing of the image on the flash card.

Step 3. Configure the station for flash operation. That means to set it up so that it boots from the flash card instead of from the network server, but also specify a network server as a second server to use in case the first server cannot be used.

Step 4. Reboot the Workstation. The boot from the first server (the flash card) fails since there is nothing yet on the flash card. That triggers a boot from the second server (network server where the image is located) and subsequently causes the writing of the flash card.

Step 5. After the flash card has been successfully written, a reboot causes the unit to boot from its flash card (first boot server configured).

Step 1a. Creating a Flash Image



Notes

The process of creating a flash image is accomplished via the Flash Manager task located in the Administration group of the Setup tasks in NSM.

This is the main panel (Setup/Create Tab) when you click on Flash Manager.
Under Flash Card Properties, you can enter the size of the flash card (in this example, a 96MB card) in the Flash Card size field. This is optional and only serves to give you an approximation of the amount of space required by the components you select and therefore of the amount of free space remaining on the card.

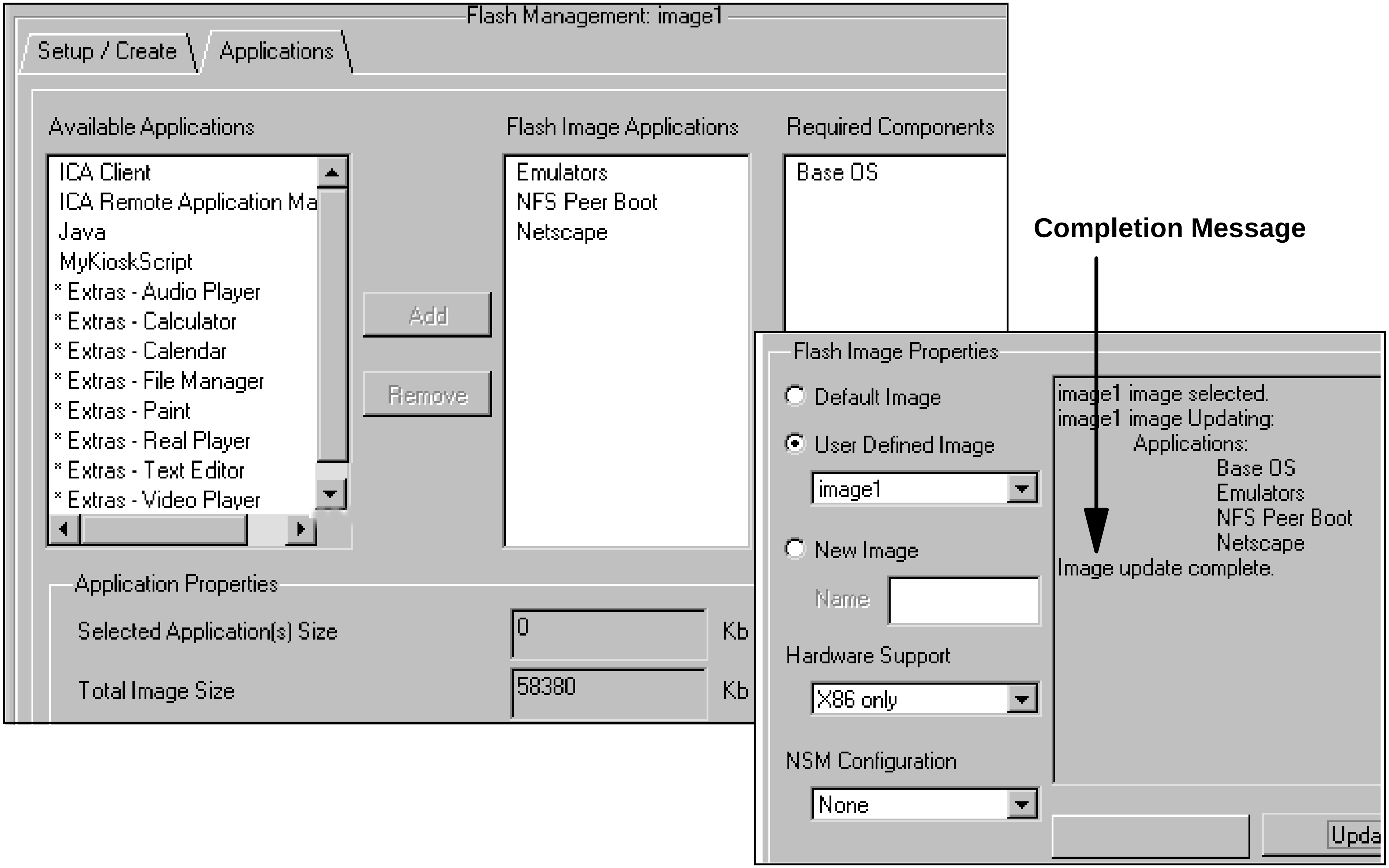
To create a new image, click the New image radio button, and enter a name for the image, for example image2.

In hardware support, chose X86 only , or PPC only, or both, dependent on the hardware platform that the card will be used for.

In the NSM Configuration pulldown, the choice is either None, or Kiosk Files. Select Kiosk files only if you need the kiosk-related files to be included in the flash image, which you would require only if the unit is to be operated in kiosk mode. In this example, we chose None.

Before you click on the Create Image button, you need to go to the Applications tab to select the applications that need to be stored on the card.

Step 1b. Selecting Applications



CBechard-08/00

Notes

The applications that can be selected for inclusion on the card are listed on the left hand side. You can select one or more applications and then click on the **Add** button to move them under the column labeled Flash Image Applications. Notice that the Base Operating System is selected by default as a required component.

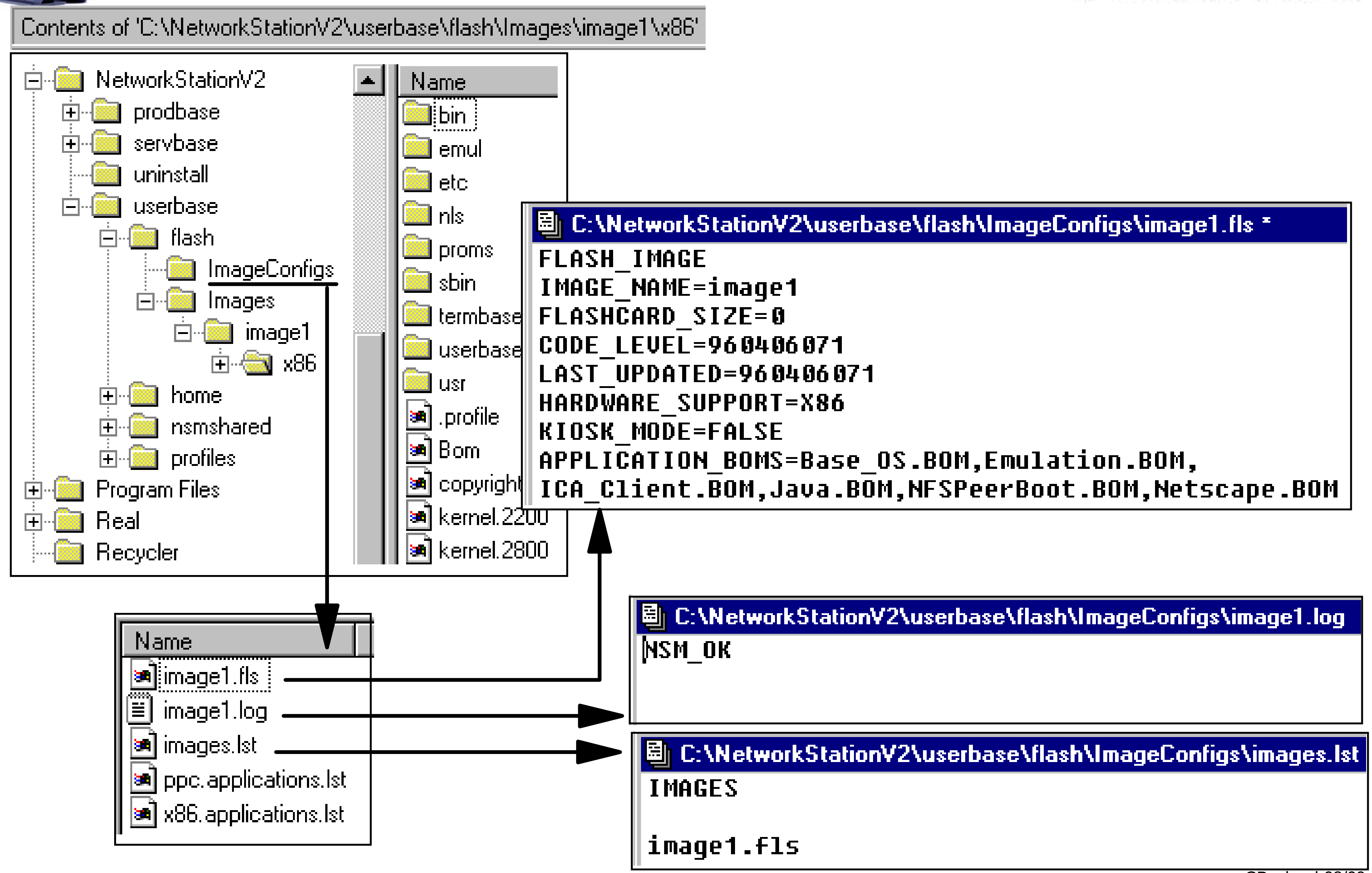
As you select applications, an approximate size is identified at the bottom of the panel to provide a general idea of the size.

Once the applications have been selected, return to the previous panel by clicking on the Setup/Create tab and then click on the **Create Image** button at the bottom of the panel.

A message appears listing the components that are being built and there is a visual indication of the creation process in the form of an oscillating bar. The creation process can take about 5 minutes or so, dependent on the speed of the processor used.
Note: If you get a timeout, you might have to increase the timeout values on your browser.

Success is indicated by a message indicating "Image Update Complete". If there are any errors, these error messages will be listed instead of the completion message.

Where are the Images Stored?



CBechard-08/00

Notes

Once the image is created, where is it stored?

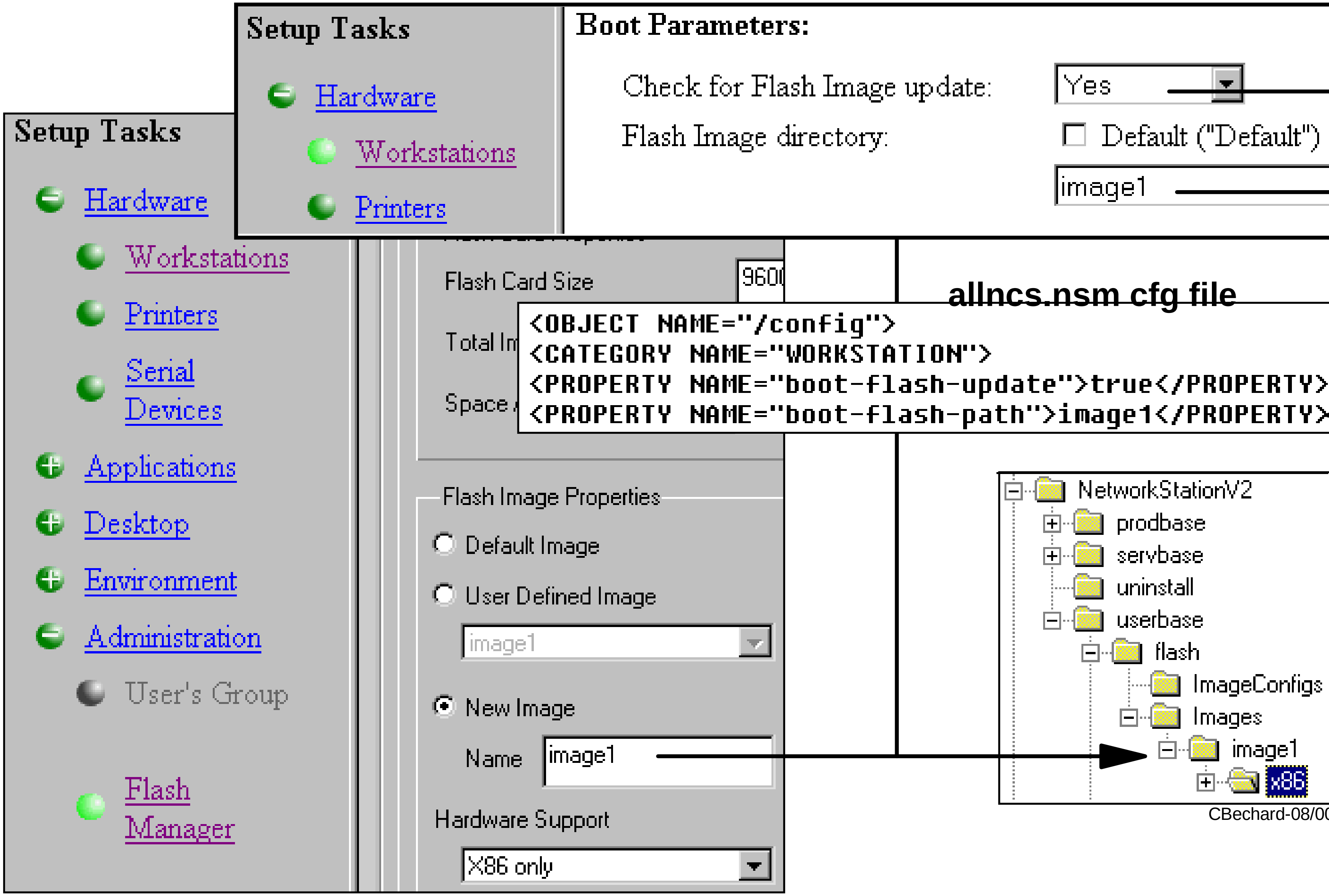
All images are stored on the NSM server in $Userbase/Flash/Images. For each image created, a directory bearing the same name as the name you chose for the image (image1 in this example) is created under the Images directory, as displayed here in the top left hand corner (partial display of a Windows Explorer panel). Notice the directories listed on the right (bin, emul, etc, ...) which are the directories that are part of the flash image.

Notice as well the ImageConfigs directory, that contains some of the control information files. For example, the image1.log file contains a single line (NSM_OK) indicating the successful completion of the image creation process. The Images.lst file contains the list of all images currently in existence (these names are displayed in a drop down list on the main panel when you choose User defined image to select an image that you may want to update).

The image1.fls file contains the characteristics of the image, and in particular the list of BOM files that contain a list of all the components that are part of the image. The resulting (consolidated) single BOM file for this image is located in the image1/x86 directory.

The x86.applications.lst files (contents not listed here) lists the applications that appear in the flash management creation window where you select applications to be included in the image.

Step 2. Enable Flash Image Update



Notes

Now that we have a flash image on the NSM server, the next task is to write this image onto the flash card.

This is accomplished by enabling image creation and update using the Hardware => Workstations task in NSM. On that panel, scroll down to the Boot Parameters portion of the display, which we show here in the top panel (We have only shown the fields that are relevant in order to save space).

First, enable flash image update by selecting Yes in the **Check for Flash Image update** box.

Then specify the Flash Image directory where the image resides. This is simply the name of the image (image1 in our example). NSM knows to look for the image1 directory under $Userbase\Flash\Images. If you leave Default as the selection, then you should have created an image called default.
Important Note: Note that this is a very important step because, when you will later boot the station on which you are writing the flash card, the name that you enter here is the name of the image that will be downloaded to the flash card.

NSM writes two statements in the allncs.nsm file (if this is done at the system level): one indicating that the system should check for updates at boot time and the other specifying the name of the image directory on the server to use for either download (when creating) or for updates. **Note**: If the image being created is an image containing configuration files for operating in kiosk mode, it is important to perform this step before creating the image so that the image name is written in the allncs.nsm file before allncs.nsm gets copied to the image during the image creation process.

Step 3. Configure the Station for Flash Boot

Network Priority	MENU 06.x
DHCP	Disabled
BOOTP	Disabled
Local (NVRAM)	First
Boot File Source	Flash
Boot File Server Settings	**MENU 08.x**
First Server IP Address	0.0.0.0
First server Directory/file	/kernel.2800
Second Server IP Address	9.67.239.49
Second server directory/file	/NetworkStationV2/prodbase/x86/kernel.2800
File Server protocol	NFS
Workstation Configuration Server Settings	**MENU 09**
First Server IP Address	9.67.239.49
First Server Directory/File	/NetworkStationV2/userbase/profiles
First Server Protocol	NFS
Authentication Server Settings	**MENU 05**
First Server IP Address	9.67.239.49
First Server protocol	RAP

Notes

In this example (a Windows NT server example), let us assume that we are configuring for flash boot for the kernel and applications but that we are using a Workstation Configuration server and User Authentication for the configuration files, and we are using NVRAM.

We are configuring this unit to boot from flash first, and from the network in second place.

First set the boot file source to Flash in the network settings menu (06.x).

For the boot file server, set the first server to the flash card (0.0.0.0) and the path to /kernel.2800 (or 2200). This is the location that the unit will try to boot from first.

For the second boot file server, specify the location of the server where the flash image resides and the directory as the standard network directory for the platform you are using (in this case /networkstationv2/prodbase/x86/kernel.2800).

The Workstation Configuration server is the server where the flash image resides as well, and the authentication server is the same as the Workstation Configuration server.

Let us now take a look at what happens when this unit is rebooted.

Step 4. Reboot

- **Look for kernel.xxxx on the flash card**
  - Not found => Boot from the network (second boot file server)
- **Load kernel from the network boot server and the cfg files from the network cfg server**
- **Is Flash Image Update flag is ON?**
  - Yes => Check if there is an image on the flash card
    - No image present => Copy image from workstation configuration server to flash card
    - Reboot
- **Locate kernel.xxxx on the flash card**
  - Found => Continue booting, load kernel from flash
- **Load configuration files from Workstation Configuration server**
- **Authenticate user**
- **Done**

Notes

This is the simplified process that results in the initial creation of the flash card.

When the station is booted, it looks for the kernel on the flash card, since that is the first boot server specified.

Because it fails to find a kernel on the flash card (the flash card is still blank) the unit boots from the second boot server, which is the network server. It loads the kernel from the boot server and the configuration files from the configuration server, and then checks if the Flash Image Update flag is turned ON (which it should be since we did this in Step 2).

Since the flag is on, the unit looks for an image on the flash card to update. Because there is no image present, this is an indication that this is an initial creation, and it downloads the image that we identified in Step 2 from the configuration server and writes the image on the flash card. Then it reboots the unit.

On reboot, the unit looks for the kernel on flash card (which this time should be found) and booting continues, loading the kernel from the flash card. Then configuration files are loaded from the Workstation Configuration server, the user is authenticated and the unit becomes operational.

How the Image Gets Updated

- **Look for kernel.xxxx on the flash card**
  - Found => Boot from flash
- **Load kernel from the flash card**
- **Load configuration files from the network cfg server**
- **Is the Flash Image Update flag is ON?**
  - Yes => Check if there is an image on the flash card
- **An Image is present on the flash card**
  - Yes => Load BOMSYNC
  - Access the network configuration server
  - Compare the local BOM with the BOM on the cfg server
    - If components all have same time stamp => Continue booting
    - If time difference => Synchronize the BOM by updating local components from the server
- **Reboot**

Notes

What happens when an update to the contents of the flash card is required?

Let us assume that we are now 6 months after the initial creation of the image and a new release of the operating system is installed on the server to update the components on the production system (but not the components in the image). To update the image, the administrator uses the Flash Manager, selects image1 and clicks on **Update Image**.

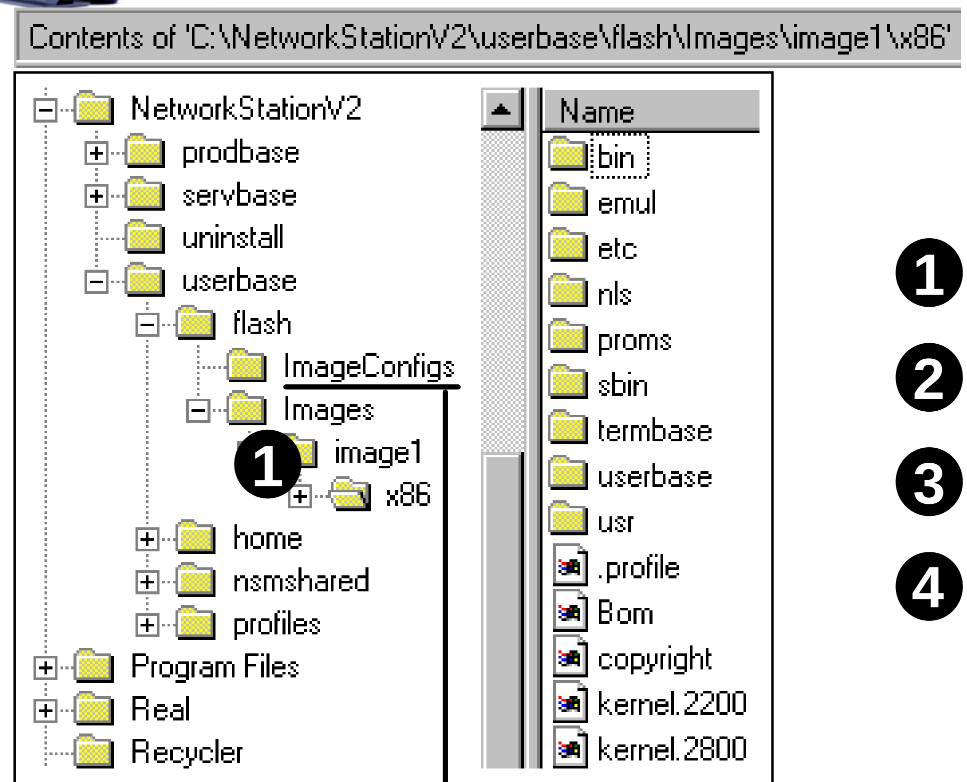
This causes the files that are part of the image and the corresponding BOM files on the server to be updated with the new components that are now on the production system (that is, the normal files in the $Prodbase directories). The next time the station reboots, the following (simplified) events take place:

The unit looks for the kernel on the flash card, finds it, loads the kernel from flash, then loads the configuration files from the network configuration server.

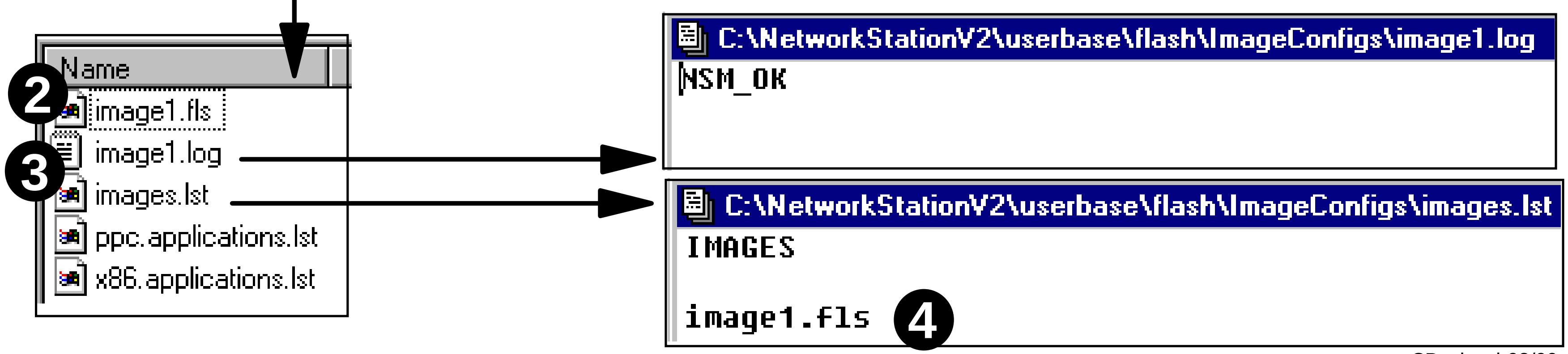
It then verifies if the Flash Image Update flag is ON? Since the flag is on, it verifies that there is an image on the flash card. Since there is one, it accesses the network configuration server and compares the local BOM file against the corresponding BOM file in the flash image on the server.

It finds that there are differences in the time stamps of many components, so it replaces the local files with the more recent files that are on the server (but only those files that show a different time stamp). The station is then rebooted in order to use the newly updated components.

Deleting a Flash Image from the Server



1. Delete the image directory itself
2. Delete the image.fls file
3. Delete the image.log file
4. Delete the image.fls entry from the image.lst file



CBechard-08/00

Notes

What happens if an image on the server is no longer required and needs to be deleted?

There is no GUI function that allows the deletion of obsolete images but these images can easily be erased manually.

(1) First erase the image directory (for example image1 in our example) from the $Userbase/Flash/Images directory.

(2) From the $Userbase/Flash/ImageConfigs directory, erase the image1.fls and (3) image1.log files. The log file usually contains only one line, stating NSM_OK if the image creation was successful. The image1.fls file contains the list of BOMs which make up the image.

Finally, the images.lst file contains one line for each image that has been created. It is the contents of this file that is displayed in the drop down list when listing the existing images. (4) Edit the file and remove the line image1.fls.

Kiosk Boot - Full Desktop Example

- **Step 1. Use NSM to configure user preferences**
  - In this example, assume this is for all stations so we do this at the system level
- **Step 2. Use NSM to enable flash card create/update**
  - We must specify a name for the flash image here BEFORE we create the flash image
- **Step 3. Manually create an allkiosk.nsm file from the allusers.nsm file**
- **Step 4. Use NSM Flash Manager to create the flash image**
- **Step 5. Configure the Station for flash boot**
  - But specify a second configuration server to be used as the update server for the flash image
- **Step 6. Reboot the station**
  - Flash card will be created and station will reboot in kiosk mode

Notes

Now that we have seen a basic example of creating a flash card, and since kiosk mode is particularly well suited to a flash card based operation, we now look at an example of creating a flash card for a kiosk mode where all files are loaded form the flash card.

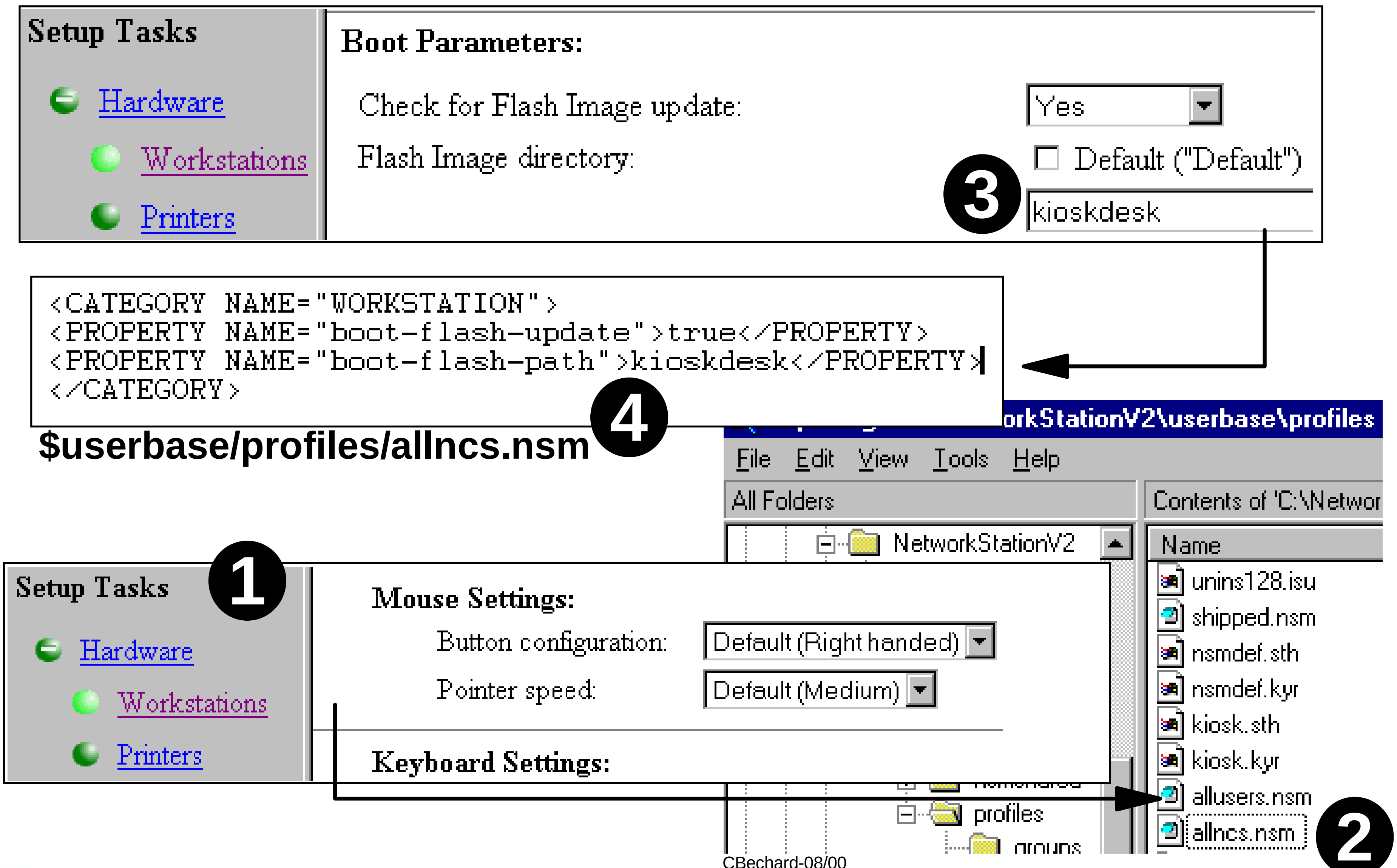
In kiosk mode, the unit never needs to access the network except when the time comes to update the flash card.

For this example, it does not really matter what type of kiosk operation we choose (single application, multiple applications or full desktop), so we use the full desktop case. When the station boots, we want the full desktop to show up, without the user having to authenticate, and we want the ability to update the flash card when changes are required.

The steps required are listed here and we will provide details on each step in the following charts.

Notice that in this case, we have to enable the flash image update and specify the image name BEFORE we create the image.

Steps 1 and 2. Configure Preferences and Enable Flash Update



Notes

Since we have decided, in our example, that one configuration will be used for all users, we configure everything at the system level. In (1), we first use NSM to configure all the characteristics that we need in order to customize the desktop as we want it to appear. For example, we add default printers, set the mouse to be left handed or right handed, set proxies for the browser, set emulator defaults, and so on.

These configuration changes are written by NSM to the allusers.nsm file and the allncs.nsm file (2), dependent on whether these parameters are workstation related parameters or user related parameters.

Then we use NSM Hardware => Workstations to enable flash update and specify the name of the image that we use for the kiosk image. In this example, we called it kioskdesk (3).

The result is that NSM adds to the allncs.nsm file the two statements shown in this example in (4). Notice that this sets the boot-flash-update property to true and the boot-flash-path property to the name of the image (which is also the name of the directory where the image resides on the server).

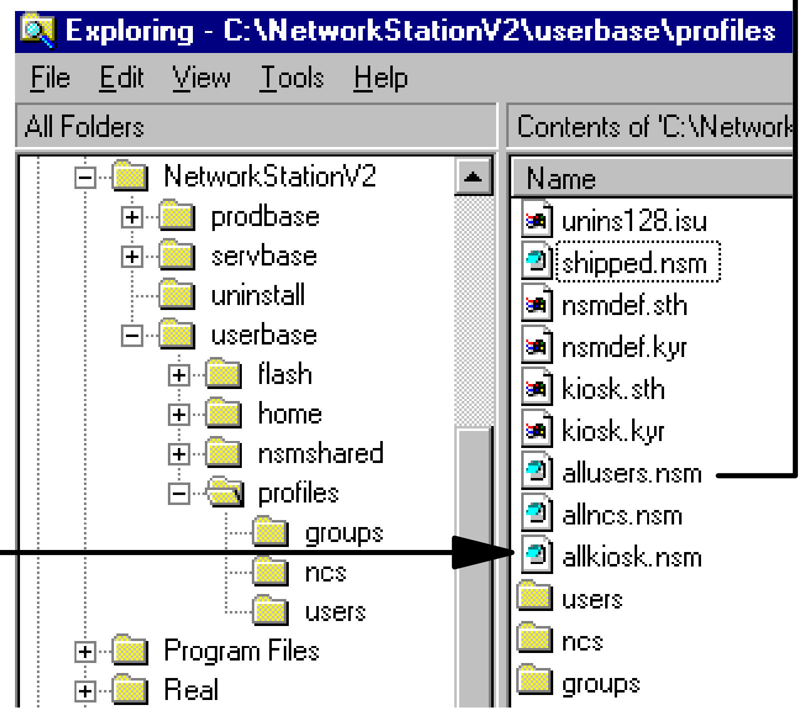
At this point in time, if a station boots from the network using this server, the allncs.nsm and the allusers.nsm would be read and the desktop would display with all the characteristics we just configured, in non-kiosk mode because we have not enabled kiosk mode yet.

Step 3. Create allkiosk.nsm

```
<PACKAGE NAME="/startup">
<CATEGORY NAME="STARTUP">
</CATEGORY>
</PACKAGE>
<OBJECT NAME="/login/session">
<CATEGORY NAME="INTERNET">
<ELEMENT>
<FIELD NAME="op">SET</FIELD>
<FIELD NAME="arg1">SOCKS_HOST</FIELD>
<FIELD NAME="arg2">socks.raleigh.ibm.com</FIELD>
</ELEMENT>
<ELEMENT>
<FIELD NAME="op">SET</FIELD>
<FIELD NAME="arg1">SOCKS_PORT</FIELD>
<FIELD NAME="arg2">1080</FIELD>
</ELEMENT>
</PROPERTY>
</CATEGORY>
<CATEGORY NAME="KIOSK">
<PROPERTY NAME="commands" TYPE="LIST" ACTION="APPEND">
<ELEMENT>
<FIELD NAME="op">SET</FIELD>
<FIELD NAME="arg1">NSM_KIOSK_MODE</FIELD>
<FIELD NAME="arg2">ON</FIELD>
</ELEMENT>
</PROPERTY>
</CATEGORY>
```

CBechard-08/00

- **1. Edit allusers.nsm in $userbase/profiles**
- **2. Add the KIOSK Category name**
- **3. Save as allkiosk.nsm in $userbase/profiles**



Notes

The next important step is to enable kiosk mode. This is done by setting an environment variable called NSM_KIOSK_MODE to the value ON.

In this specific example, the easiest way to proceed is to first to make a copy of the allusers.nsm file, save it under the name allkiosk.nsm, and then modify allkiosk.nsm to include the NSM_KIOSK_MODE environment variable. (When the system reads the configuration files, if allkiosk.nsm exists, it will read that file first before reading the allusers.nsm, and if the allkiosk.nsm file contains statements that sets the kiosk flag ON, it then bypasses reading the allusers.nsm).

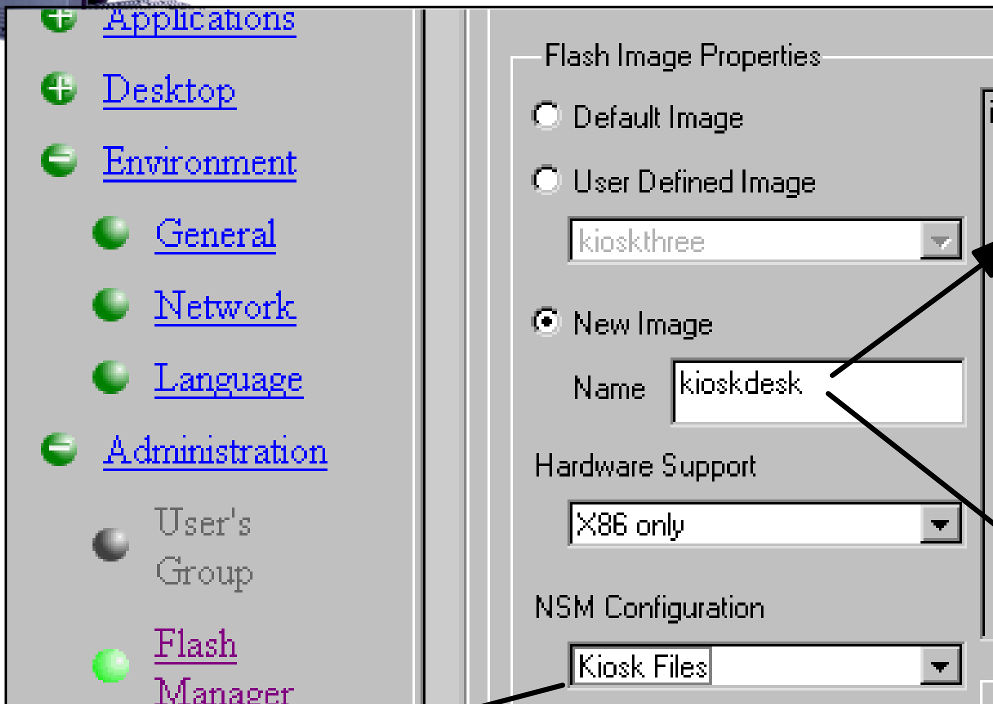
In allkiosk.nsm, add the statements that are shown on this chart in bold, usually at the end of the file after the package name "/startup". The best way to do this is to copy and paste these statements from one of the default kiosk files (*.ksk) that can be found in the $Servbase/default directory on the server.

This is all that needs to be done in this particular example of a full desktop. If instead we had elected to use a single application mode, we would have used one of the sample kiosk files in the $servbase/default directory, and renamed it to allkiosk.nsm file, instead of building the allkiosk.nsm from the allusers.nsm file.

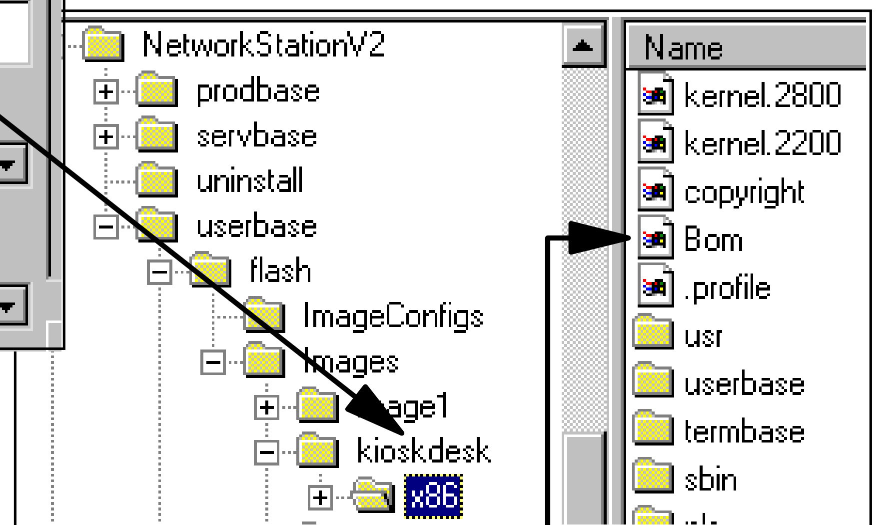
Finally, if we wanted to prepare kiosk mode files for specific workstations (as opposed to doing it at the system level for all workstations), we would not name the file allkiosk.nsm, but we would name it ipaddress.nsm, or hostname.nsm or macaddress.nsm. At boot time, the system checks first if there is an *.nsm file that exists with the ip address, hostname or mac address of the station, and if not, if an allkiosk.nsm exists.

IMPORTANT CAUTION: It is imperative, when editing these files, that you do not use an editor that automatically ends lines with a Carriage Return (CR) and Line Feed (LF). These are hexadecimal X'OD' and X'OA'. **DO NOT USE NOTEPAD** on Windows NT, but WordPad appears to function correctly and so does WinVI, which is a free editor on the internet.

Step 4. Create the Flash Image



Specify same name as in Step 2.



Causes the files below to be copied from $userbase/profiles and to be added to the image's BOM file

F - P -rw-r--r-- 0 0 319958 Aug 07 14:45:01 2000 termbase/profiles/shipped.nsm
F - P -rw-r--r-- 0 0 14642 Aug 07 14:45:01 2000 userbase/profiles/nsmdef.kyr
F - P -rw-r--r-- 0 0 129 Aug 07 14:45:01 2000 userbase/profiles/nsmdef.sth
F - P -rw-r--r-- 0 0 13746 Aug 07 14:45:01 2000 termbase/profiles/kiosk.kyr
F - P -rw-r--r-- 0 0 129 Aug 07 14:45:01 2000 termbase/profiles/kiosk.sth
F - P -rw-r--r-- 0 0 2922 Aug 07 14:47:03 2000 termbase/profiles/allncs.nsm
F - P -rw-r--r-- 0 0 18024 Aug 07 14:47:01 2000 termbase/profiles/allkiosk.nsm

CBechard-08/00

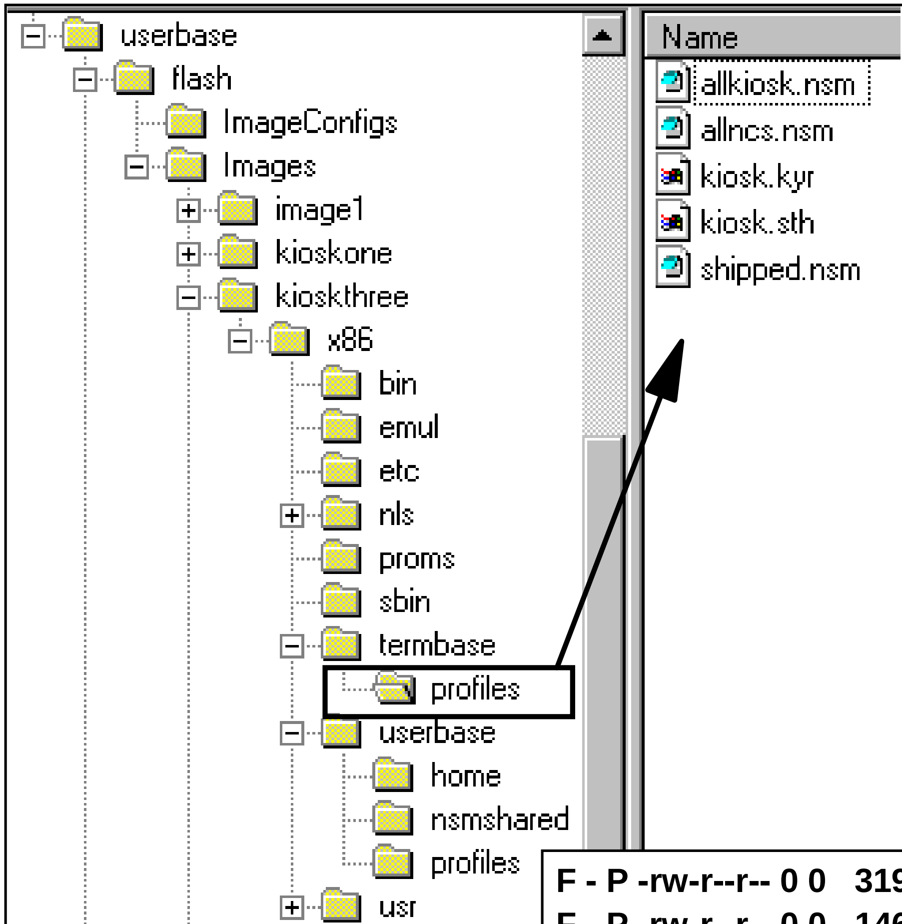
Notes

We are now ready to create the actual flash image on the, using the NSM Flash Manager application.

In the **New Image** field, enter the name that was used in step 2 when the flash image updated flag was enabled (because this is the name that is currently set in the allncs.nsm configuration file in the boot-flash-path property). In this example, we used kioskdesk.

Then choose the platform (x86, ppc or both) in **Hardware Support** and make sure to select **Kiosk Files** in the **NSM Configuration Files** field. This causes the system to copy the files shown here at the bottom of the chart (in particular allkiosk.nsm and allncs.nsm) from the $Userbase/profiles directory to the flash image being created and to add these files to the image's BOM file.

Step 4. Create the Flash Image (2)




```
F - P -rw-r--r-- 0 0   319958 Aug 07 14:45:01 2000 termbase/profiles/shipped.nsm
F - P -rw-r--r-- 0 0   14642 Aug 07 14:45:01 2000 userbase/profiles/nsmdef.kyr
F - P -rw-r--r-- 0 0   129 Aug 07 14:45:01 2000 userbase/profiles/nsmdef.sth
F - P -rw-r--r-- 0 0   13746 Aug 07 14:45:01 2000 termbase/profiles/kiosk.kyr
F - P -rw-r--r-- 0 0   129 Aug 07 14:45:01 2000 termbase/profiles/kiosk.sth
F - P -rw-r--r-- 0 0   2922 Aug 07 14:47:03 2000 termbase/profiles/allncs.nsm
F - P -rw-r--r-- 0 0   18024 Aug 07 14:47:01 2000 termbase/profiles/allkiosk.nsm
```

Notes

Notice that these kiosk-related configuration files that are pulled in from the production system's files are copied from $Userbase/profiles to the image's /termbase/profiles (as shown on this chart) since this is the directory that is used, once the image is on the flash card.

The configuration files must be copied to the /termbase/profiles directory because when the station boots, that is the location where it expects to find the configuration files.

When the image creation completes, the image should then contain all the files required for the station to operate entirely from the flash card.

Step 5. Configure for Kiosk Flash Boot

NetVista

Network Priority		**MENU 06.x**
DHCP		Disabled
BOOTP		Disabled
Local(NVRAM)		First
Boot File Source		Flash
Boot File Server Settings		**MENU 08.x**
First Server IP Address		0.0.0.0
First server Directory/file	1	/kernel.2800
File Server protocol		nfs (or rfs)
Second Server IP Address		9.67.239.49
Second server Directory/file	2	/NetworkStationV2/prodbase/x86/kernel.xxxx
Second Server protocol		nfs
Workstation Configuration Server Settings		**MENU 09**
First Server IP Address		0.0.0.0
First Server Directory/File	3	/termbase/profiles
First Server Protocol		local
Second Server IP Address		9.67.239.49
Second Server Directory/File	4	/NetworkStationV2/userbase/profiles
Second Server Protocol		nfs
Authentication Server Settings		**MENU 05**
First Server IP Address		0.0.0.0
First Server protocol		RAP

Notes

The last required step is to configure the station to boot from flash and to also get its configuration files from the flash card, and we also want to configure it so that it can create its flash card initially and then update its flash card from the network server in the future.

This is similar to the first example we used, which is to set the boot file source to Flash, and to set the first boot file server to 0.0.0.0, the first server boot file to /kernel.xxxx and the protocol to nfs (1).

Since we are creating a new flash card, we know that the boot to the first server will fail initially (since the flash card is now blank) so we need to configure the second boot server to point to the network server, as in (2).

The first workstation configuration server is now also set to the flash card by specifying 0.0.0.0 as the server address, /termbase/profiles as the configuration path, and local as the protocol (actually, the words flash in the NS Boot menu) as shown in (3).

As we also want to set the workstation to be able to retrieve updates to the flash card, the second configuration server should be set to the network server, the path to the normal $userbase/profiles entry and the protocol to nfs (or rfs) (4). The station knows to use the parameters specified in the second configuration server as the location of the update server.

When the station boots, it fails to boot from flash the first time (blank flash card), so it boots from the network (second server), copies the flash image from the server to the flash card, then reboots from the newly created flash card (first server) and also loads its configuration files from the flash card (first workstation configuration server).

Syncing BOM Files - Example

NetVista

- Example of BOM sync when operating entirely off the flash card
- Configuration files are on /termbase/profiles
- Notice the "Failing over to server 9.67.239.49" which is the second configuration server

```
Doing bomsync ...
Flash image server : 0.0.0.0
Flash image root : /termbase/flash/Images/kioskthree/x86/
Server protocol : NFS/DEFAULT
Flash image server : 9.67.239.49
bomsync in free(): warning: modified (chunk-) pointer.
Flash image root : /networkstationv2/userbase/flash/Images/ki
Server protocol : NFS/DEFAULT
Failing over to server 9.67.239.49
Server filesystem reached
Timestamps of BOMs differ.
syncing server BOM and distribution
```

CBechard-08/00

e business tools

Notes

Later on, if at some point in time one or more of the components of the flash image on the server is changed, an update to the flash card would take place, and here is an example of an update taking place.

With the parameters specified as shown in the previous chart, the messages displayed here are those that appear on the workstation's display, at boot time, when the station verifies if an update is required.

As displayed by the messages, "doing bomsync" indicates that the station is verifying whether an update is required or not.

The message indicates that the station is aware that the local server is 0.0.0.0 (flash card) but that the update server is 9.67.239.49, which is the server we configured as the second workstation configuration server.

The message "time stamps of BOMS differ" indicates that the station found out that the timestamps of some components in the BOM file of the flash image on the server are different from the time stamps of the same components in the BOM file that resides on the flash card, and so it proceeds to make the required updates by replacing only those files that do have a different time stamp.

If a second configuration server was not configured, the update could not take place because the station would not know which server to contact in order to get to the BOM file on the server.

Configuring for Kiosk Flash Boot w/DHCP

NetVista

Network Priority	MENU 06.x	DHCP Options
DHCP	First	
BOOTP	Disabled	
Local(NVRAM)	Disabled	
Boot File Server Settings		
First Server IP Address	0.0.0.0	66
First server Directory/file	/kernel.2800	67
File Server protocol	local	211
Workstation Configuration Server Settings		
First Server IP Address	0.0.0.0	212a
First Server Directory/File	/termbase/profiles	213a
First Server Protocol	local	214a
Second Server IP Address	9.67.239.49	212b
Second Server Directory/File	/NetworkStationV2/userbase/profiles	213b
Second Server Protocol	NFS	214b
Authentication Server Settings		
First Server IP Address	RAP://0.0.0.0	98

Notes

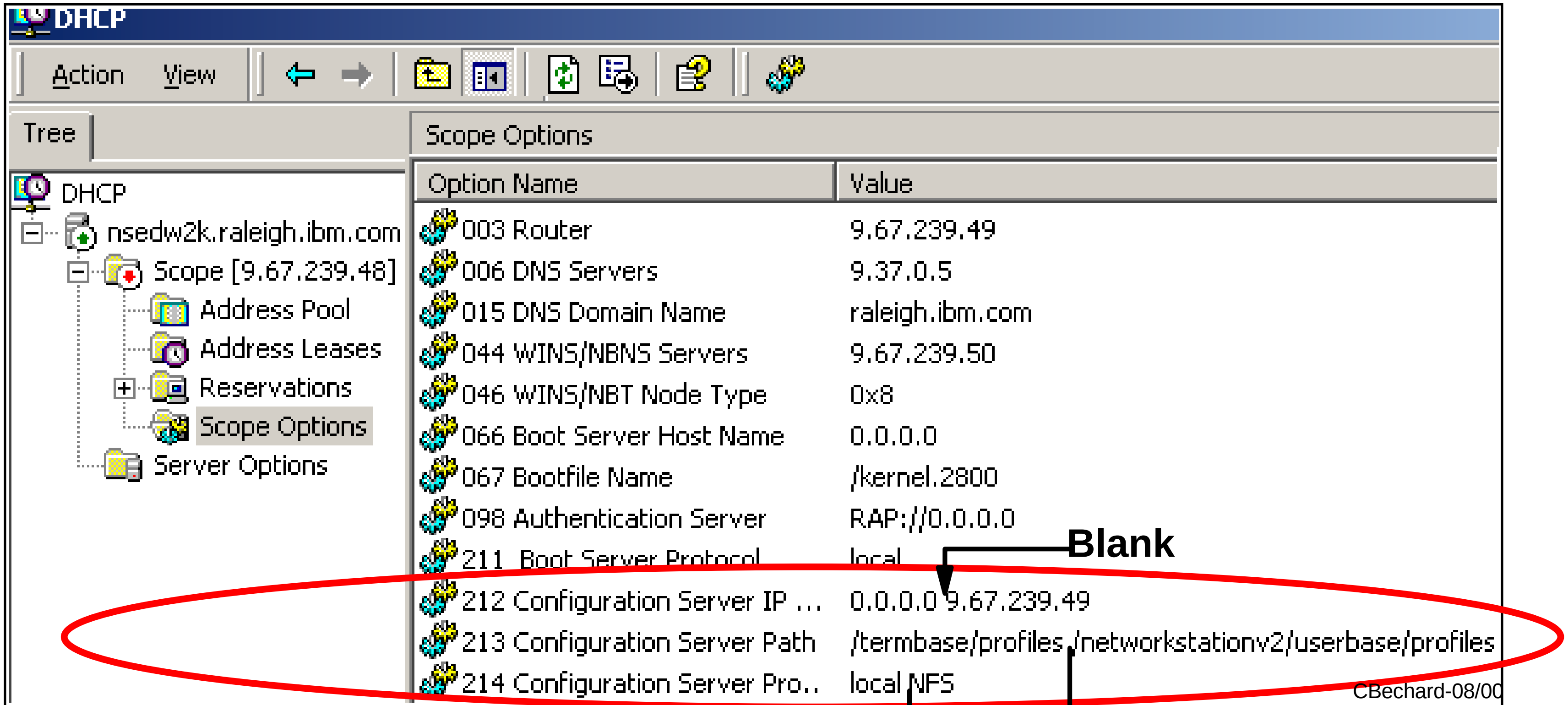
What about the case where we are getting the network configuration data from a DHCP server instead of from NVRAM?

This chart shows the DHCP options that need to be configured in order to configure the station with the same parameters that we previously entered in NVRAM.

Notice that the second workstation configuration server parameters are entered in DHCP options 212, 213 and 214 as the second entry (we use 212b, 213b and 214b to indicate that this is the second entry in the option filed). That is, two values, separated by a blank, can be entered in these DHCP options.

The next chart illustrates the DHCP options that we configured in a Windows 2000 server DHCP server for this particular example.

DHCP Server Option



These three options each contain two values, separated by a blank.

Blank **Blank**

Notes

This chart displays the panel from a DHCP server on a Windows 2000 server system that we used for the example described in the previous charts.

Notice how options 212, 213 and 214 contain two entries, the first one corresponding to the first workstation configuration server, and the second one to the second workstation configuration server which is used as the update server to update the flash image.

Summary

NetVista

	DHCP Option	Network Based	Typical Flash (User Authentication)	All Flash (Kiosk) with Flash Update
Boot Server 1	66	x.x.x.x	0.0.0.0	0.0.0.0
Boot file 1	67	$prodbase/x86/kernel.x	/kernel.x	/kernel.x
Boot protocol	211	nfs (or rfs)	local	local
Boot Server 2	219	Note 1	Note 1	Note 1
Boot File 2	None	Note 1	Note 1	Note 1
Cfg server 1	212a	x.x.x.x	x.x.x.x	0.0.0.0
Cfg Path 1	213a	$userbase/profiles	$userbase/profiles	/termbase/profiles
Cfg Protocol 1	214a	nfs (or rfs)	nfs (or rfs)	local
Cfg Server 2	212b	Note 1	Note 1	x.x.x.x
Cfg Path 2	213b	Note 1	Note 1	$userbase/profiles
Cfg Protocol 2	214b	Note 1	Note 1	nfs or rfs
Auth server	98	RAP://x.x.x.x	RAP://x.x.x.x	RAP://0.0.0.0
Comments		All operations done from network servers. Can be used in either normal mode or kiosk mode.	Kernel and application loaded from flash but configuration files from a network server.	Everything loaded from flash (kiosk mode) but flash updates from network when required

CBechard-08/00

Note 1 - Optionally specified as backup server

Notes

This chart is a summary of the parameters that are required for the three most common scenarios.

For example, the first column (Network Based) shows that a first boot server must be configured, that a second is optional as a backup server, and that a first workstation configuration server must be specified (with again a second as backup if required) and finally an authentication server since this is a mode where the user identifies himself specifically with userid and password.

In the typical flash column, the only change is that the boot server is now on the flash card. In this case, if an update to the flash card contents is required, the update server is the server configured as the first configuration server.

The All Flash column shows that first boot server and first workstation configuration server are now all on the flash card. However, in order to allow updates to the flash card, workstation configuration server number 2 is configured and will be used as the update server.

The DHCP Options column indicates the DHCP options corresponding to the values that can be entered in the NSBOOT menus.

Using DHCP without special options

	DHCP Option	All Flash (Kiosk) using DHCP enabled via simple configuration menu These values are not supplied by DHCP but are the defaults used by the firmware
Boot Server 1	66	0.0.0.0
Boot file 1	67	/kernel.xxxx
Boot protocol	211	local
Boot Server 2	219	N/A
Boot File 2	None	N/A
Cfg server 1	212a	0.0.0.0
Cfg Path 1	213a	/termbase/profiles
Cfg Protocol 1	214a	local
Auth server	98	RAP://0.0.0.0
Comments		Everything loaded from flash (kiosk mode) but no flash updates possible. CBechard-08/00

Notes

In the case where the DHCP server does not support all the options required by thin clients (66, 67, 211, 212, ...) or where the DHCP server serves out only the options normally required by PCs, it is still possible to use this DHCP server to support flash card based stations. However, in that case, the parameters required fro allowing flash card updates cannot be specified.

This is accomplished by enabling DHCP, on the workstation, by using the simple configuration menu instead of the advanced menu and booting while in that simple configuration mode.

The effect of using the simple configuration menu is that the firmware will accept DHCP offers that do not contain the normally required options for thin clients, so it will accept the IP address, gateway address and network mask, and it will then supply default values for the other parameters that are expected. This can be done because the values for flash card based kiosk operation are fixed values.

However, there is one limitation to this scenario, and it is that since a second workstation configuration server cannot be specified, it is not possible then to set up the flash card for automatic updates.

Erasing/Viewing Flash Card

- **To Erase Flash Card Contents**
  - Boot from the network (with the flash card installed)
    - login as nsm_nfsroot (Windows NT), root (AIX) or QSECOFR (AS/400)
  - Mount the flash card
    - mount /dev/wd0a /mnt (mounts with read/write access)
  - Issue the following commands
    - cd /mnt (sets flash card root as current directory)
    - rm .profile (removes the .profile file)
    - rm -R * (removes all files except .profile)
    - cd /
    - fsck_ffs /dev/wd0a
    - umount /mnt

- **To view Flash Card Contents**
  - If booted from flash, the flash card is visible as the root directory
  - If booted from the network with the flash card installed
    - login as nsm_nfsroot (Windows NT), root (AIX) or QSECOFR (AS/400)
    - Mount the flash card **in read mode only** (mount **-r** /dev/wd0a /mnt)
    - cd /mnt
    - the flash card contents are now the current directory

Notes

What if we want to completely erase the contents of a flash card in order to start from scratch, as if the card was a brand new card?

With the flash card installed in the station, boot from the network and login as nsm_nfsroot (Windows NT), root (AIX) or QSECOFR (AS/400) because these users have the required authority to be able to write the flash card.

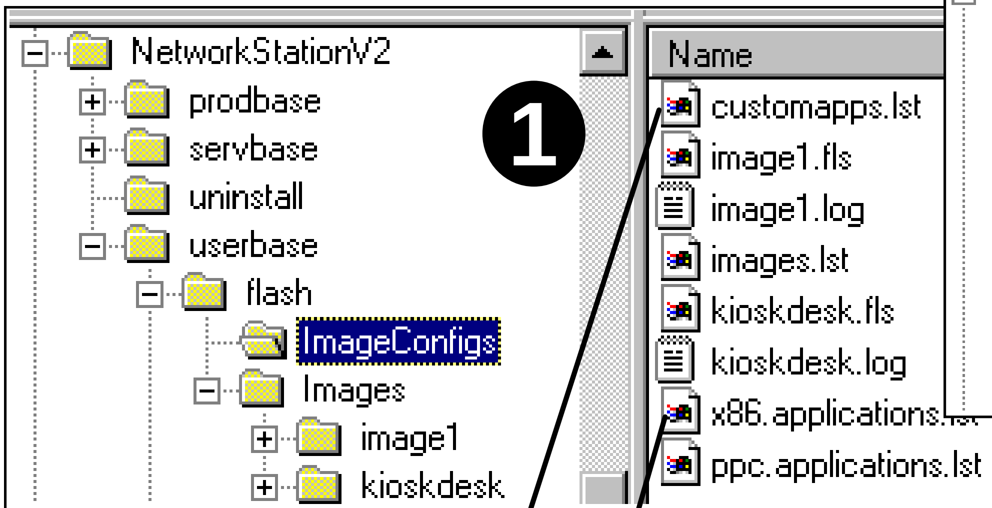
Then mount the flash card in read/write mode with the mount command, as illustrated here, and then issue the commands to erase all the files.

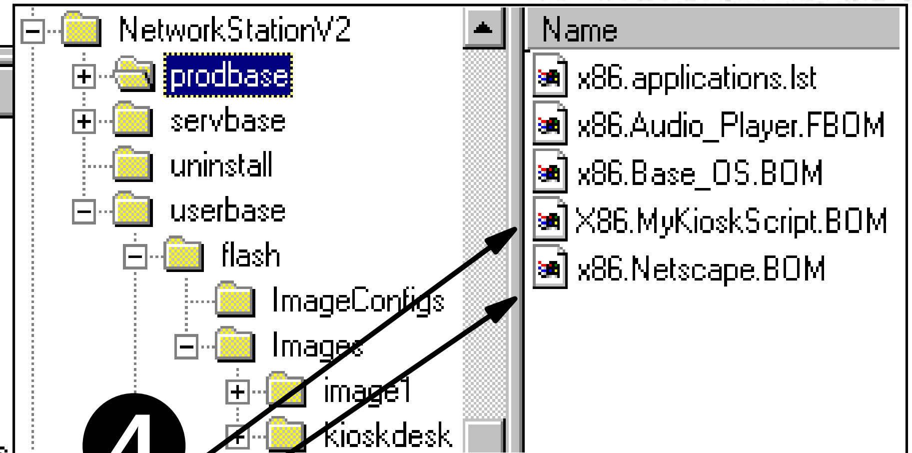
The card can then be rebuild as a new card.

If all you want to do is to view the flash card contents, there are two cases:

- If the station was booted from flash, the flash card is simply visible as the root directory since it was automatically mounted
- If the station booted from the network, login as nsm_nfsroot (Windows NT), root (AIX) or QSECOFR (AS/400) and then mount the flash card **in read mode only.** Once mounted, change the directory to /mnt to view the contents.

Adding Custom BOM Files





2

customapps.lst

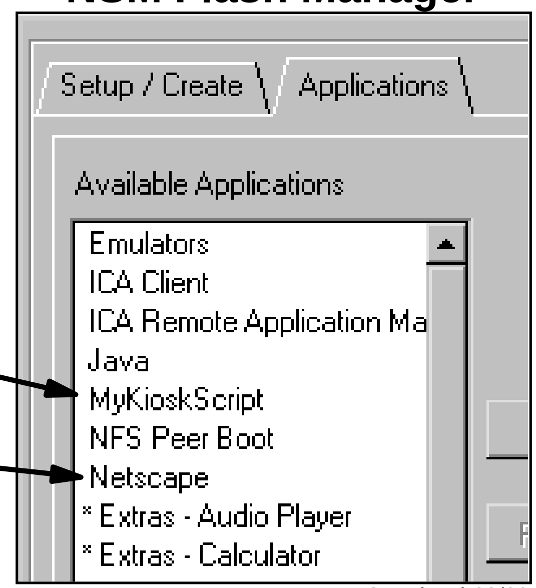
APPLICATIONS
MyKioskScript&AVAILABLE&MyKioskScript.BOM&1024

x86.applications.lst

960406071
APPLICATIONS
Base OS&REQUIRED&Base_OS.BOM&31029821
Netscape&AVAILABLE&Netscape.BOM&19278018
Emulators&AVAILABLE&Emulation.BOM&5563931
ICA Client&AVAILABLE&ICA_Client.BOM&1487155
ICA Remote Application Manager&AVAILABLE&ICA_RemoteAppMgr.BOM&598532
Java&AVAILABLE&Java.BOM&10522978
NFS Peer Boot&AVAILABLE&NFSPeerBoot.BOM&2510126
* Extras - Audio Player&AVAILABLE&Audio_Player.FBOM&73291

3

NSM Flash Manager



CBechard-08/00

Notes

What about the case where we want to include files on the flash card that are not part of the normal system files, that is our own custom file or applications.

One way would be to edit the system BOM files to add these components, but that is not a good strategy because these changes would disappear as soon as a new PTF or release of the product came out since the system BOM files would be replaced with new files.

For that purpose therefore, there is a special file called **customapps.lst** that can be manually created to contain any custom files that we want to include.

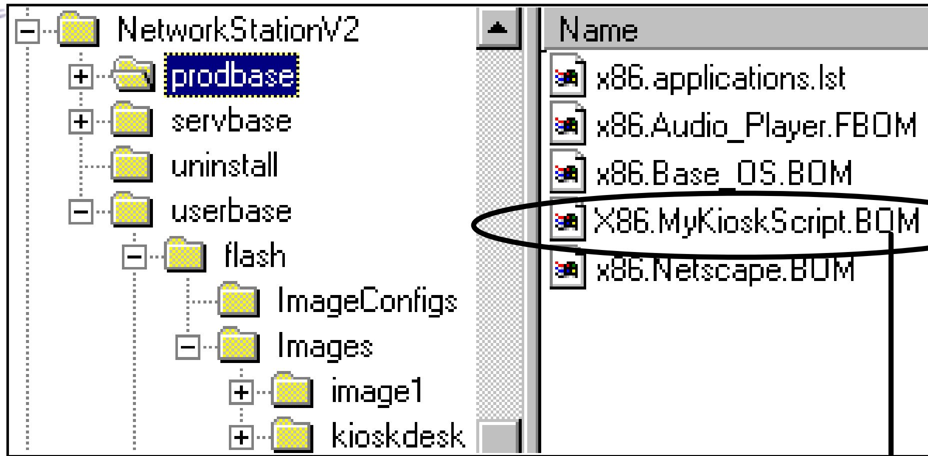
For the purpose of this example, we have created a custom component called MyKioskScript which we want to be able to select when we create an image. It is used to include a script file that starts two applications in kiosk mode.

We first create a customapps.lst file, in the format indicate in (2) and store it in the $userbase/flash/ImageConfigs directory, as shown in (1). Notice that in the NSM Flash Manager, when you use the Applications tab, the list of applications displayed are taken from the x86.applications.lst (in the case of x86) but if a customapps.lst file exists, it also list its contents (3), which is why our custom MyKioskScript entry appears in the list.

As a result, this causes the flash manager to retrieve the components listed in the BOM file(s) listed in the MyKioskScript.BOM (4) the same way that it gets the Netscape components from the netscape system BOM file

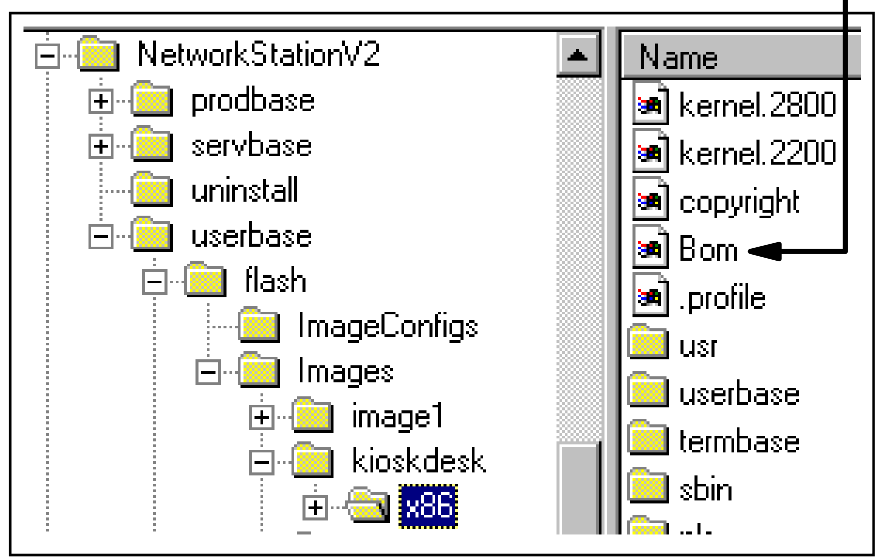
The sample contents of the BOM files are shown in the next chart.

Adding Custom BOM Files(2)



x86.MyKioskScript.BOM

```
F T P -r-xr-xr-x 0 0        246272 May 23 09:17:33 2000 x86/usr/local/bin/kiosktwo
```



CBechard-08/00

Notes

As we indicated before, we created this custom component in order to be able to include a single file, which is a script that starts two applications while in multi-application kiosk mode.

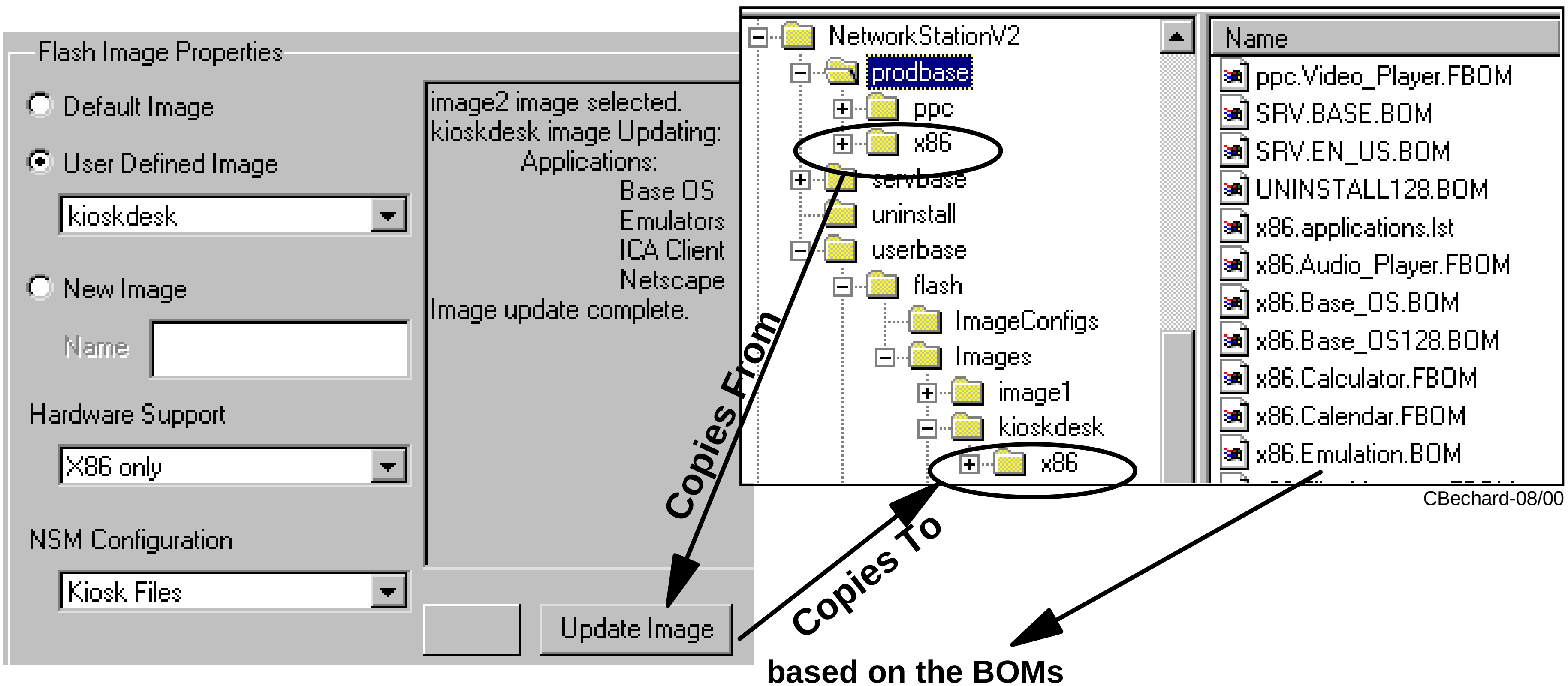
This chart shows what the BOM file for our custom component might look like, where we include a file called kiosktwo in the /usr/local/bin directory.

This file is initially stored in the $prodbase/x86/usr/local/bin directory, and the BOM file directs the flash management utility to copy this file to the flash image it is creating, in the <imagename>/x86/usr/local/bin directory.

Since this is a custom component and a custom BOM file, it is not replaced when a system update is done and therefore remains available for any other flash image creations.

Updating to a New Release

- When a new release or PTF is applied, the $Prodbase/x86 components are updated
- The $userbase/flash/images/kioskdesk/x86 are NOT updated until the Update Image in NSM is used by the administrator



Notes

What happens when a new release or PTF is applied to the NSM server?

As this chart attempts to show, many of the components that are located in the many directories under $Prodbase could be replaced with new versions, and all the BOM files in the $Prodbase would also be replaced, with many of the entries in these BOM files having changed.

Once the system update has been done, none of the images that were stored in the flash/images directories would have been changed in any way. To update these images, the administrator selects the image name in the NSM flash management utility and clicks on the "Update Image" button.

This causes the flash management utility to compare the BOM file of the image with the new BOM files in $Prodbase and to copy all the files that have been updated.

Once the image on the server is updated, the corresponding flash images on the workstations can be updated when they boot the next time, provided that they have been configured for automatic updates.

Editing a BOM file

NetVista

```
F T P -r-xr-xr-x 0 0       246272 May 23 09:17:33 2000 x86/usr/local/bin/kiosktwo
```

Permissions

User ID

Group ID

Size

Date/Time

Path/Filename

P = Path is relative to $Prodbase
C = Path is complete, relative to root of server

T = Marked for Service
- = New Entry

F = In Flash
- = Not In Flash

e business tools

Notes

To create custom images and BOM files, one must be able to manipulate and edit BOM files. This chart illustrates the format of the BOM file.

The F in the first column indicates that this file is to be written to the flash card. Replacing the F with an hyphen (-) indicates that the component is not to be written.
The T indicates that the file is marked for Service.
The P indicates that the base path is $prodbase. If it is different, a C is used to indicate that the path is complete, that is, from the root of the server.
Then follow the permissions, user ID , Group Id, and size of the component.

Next is the date/time stamp, which is the essential control field in BOM files, and finally the path and filename.

DNS Considerations

- **The $Prodbase/etc/resolv.conf from the server is copied to the flash**
  - This file is based on the server's DNS info, which may not be correct for all the clients
- **Either customize the resolv.conf file BEFORE creating each flash image**
- **Or create a symbolic link from /etc/resolv.conf to /tmp/resolv.conf to dynamically use the NVRAM or DHCP DNS information**
- **See the Advanced Information document on www.ibm.com/nc/pubs for more details on how to set via customized BOMs**
- **Also see www.pc.ibm.com/qtechinfo/MIGR-4M4RG3.html for a document entitled "Domain Name Server Resolution in V2R1". This document explains the bypass procedure to use the DNS inform from NVRAM or DHCP.**

Notes

If the DNS information required by the station is different from the DNS information on the server, or if the station must use the DNS information supplied via DHCP or NVRAM, changes are required in the way the resolv.conf file is used.

See the references listed here for details.

There are also other miscellaneous documents related to DNS so we suggest a search on that web site with the words DNS.

Note that the official reference documentation for flash based operation can be found in Chapter 5 of the Advanced Information document available from www.ibm.com/nc/pubs. This information will be updated soon to follow the different releases and PTFs being issued over time.

That reference document should therefore be your official source of information whereas this presentation should be only used to get illustrated examples of how it works and an understanding of the configuration possibilities.