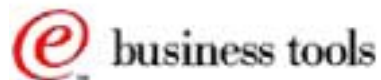


IBM NetVista Thin Client N2200w Windows-Based Terminal



Firmware Update Overview

July 2000

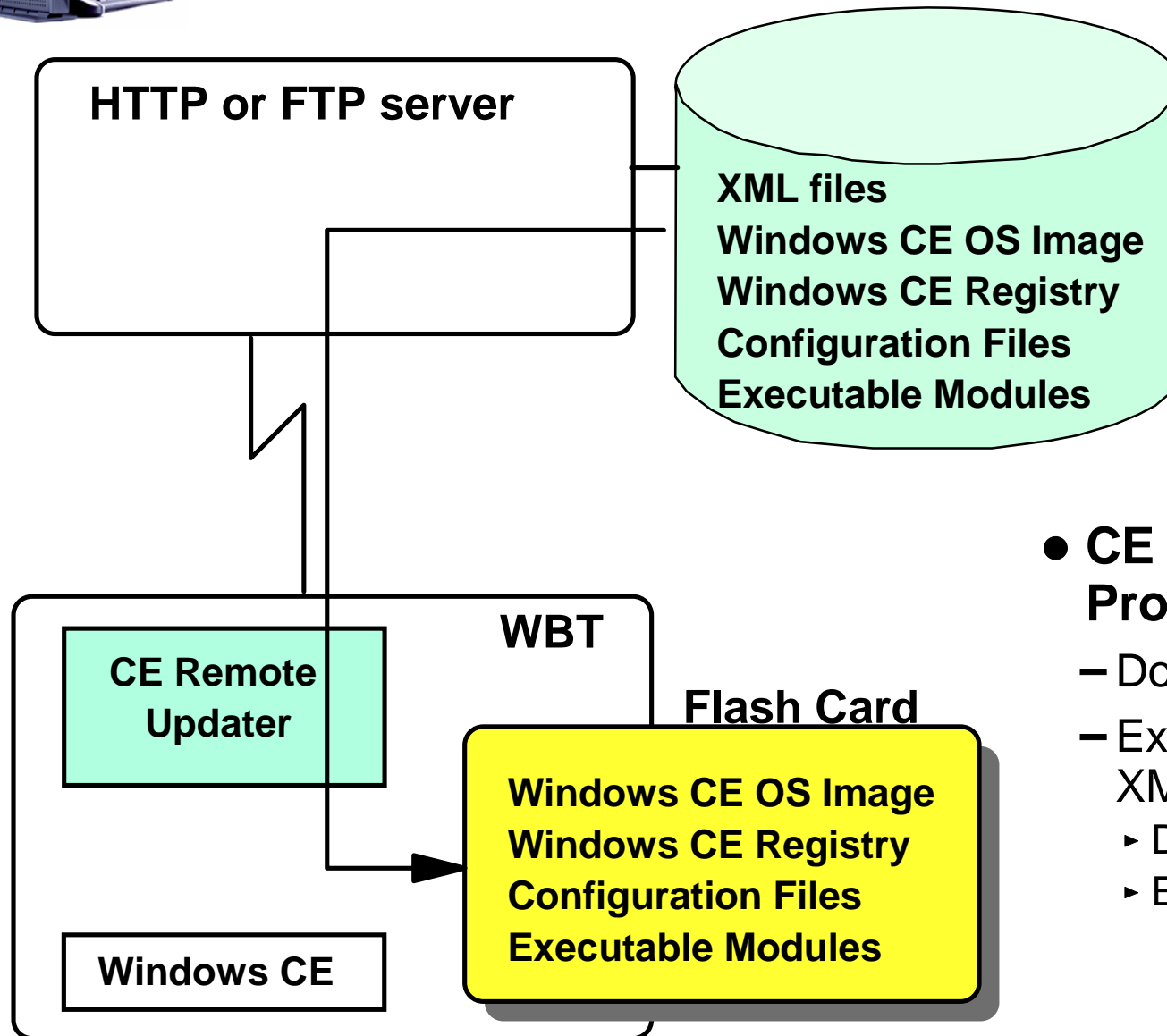


© IBM Corporation

IBM NetVista Thin Clients
Access for today, flexibility for tomorrow



Updating Software Components



- **CE Remote Updater Program**

- Downloads an XML file
- Executes directives in the XML file
 - Downloads components
 - Executes programs



Notes



Objective:

The objective of this presentation is to provide an overview of the mechanism used to update the software components on the IBM NetVista WBT.

At a high level, the process of updating the software components that reside on the flash card is performed by a program called the **CE Remote Updater**.

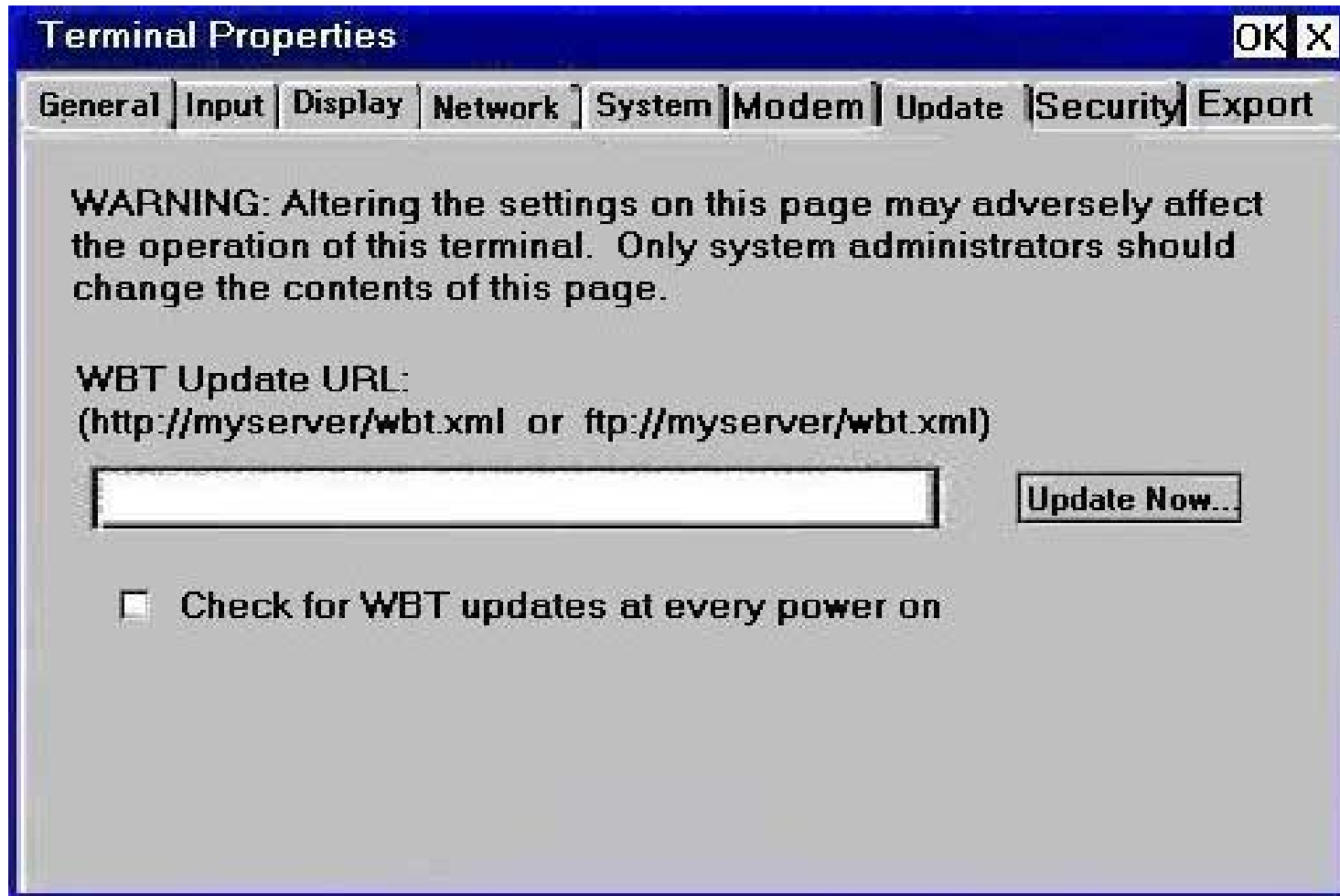
The role of the updater program is to first download from an HTTP or FTP server an XML file which is actually a series of instructions to be executed by the updater program.

These instructions might be to download a file from an HTTP server and write it to the local file system, on the flash card, in a particular directory, or it might be to execute a program already residing on the flash card, or it might be to update the operating system image on the flash card.

So its main purpose is to provide a way of automatically updating the software components but it can also be used to download a set of configuration files to a brand new unit for example (cloning can be performed that way).



How is the Updater Started?





Notes



How is the Updater program started or initiated?

The Updater program can be either manually initiated or can be configured to start at every power on. This is done via the Update Tab of the Terminal Properties panel, as illustrated here, which is accessed by pressing F2 while on the Terminal Connection manager panel.

In either case, an entry must initially be entered in the **WBT Update URL** field, which identifies the HTTP or FTP server to contact, as well as the name of the XML file to process.

Once the entry exists, it can be triggered to execute immediately by a click on the **Update Now...** button, or to be delayed until the next power on by setting on the **Check for WBT updates at every power on** indicator.



What is XML?



- **Extensible Markup Language**
- **HTML is a subset of XML**
- **Both XML and HTML born from SGML**
- **Provides a format for describing structured data**
- **Text based and easy to read**

```
<weather-report>
  <date>March 25, 1998</date>
  <time>08:00</time>
  <area>
    <city>Seattle</city>
    <state>WA</state>
    <region>West Coast</region>
    <country>USA</country>
  </area>
  <measurements>
    <skies>partly cloudy</skies>
    <temperature>46</temperature>
    <wind>
      <direction>SW</direction>
      <windspeed>6</windspeed>
    </wind>
  </measurements>
</weather-report>
```



Notes



What is XML?

XML stands for Extensible Markup Language. Both XML and HTML originate from SGML (Standard Generalized Markup Language) and HTML is a subset of XML.

XML provides a format for describing structured data that is text based and easy to read.

The example on the right here of a weather report illustrates how data can be structured in an easy to read and understand fashion.



XML Validation



- **"Well formed" means the XML is syntactically correct**
- **"Validated" means the XML data is well formed and adheres to a specific representation**
- **Document Type Definitions (DTD) and XML schemas are ways of formally declaring XML representations**

Sample SoftwareUpdate.DTD file

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Package
(CopyFile*,DiskOnChipOsUpdate*,Execute*,UpdateRegistry*,Reboot*)>

<!ATTLIST Package
  ID CDATA #IMPLIED>

<!ELEMENT CopyFile
(Description?,Source,Destination,CommandLine?)>
<!ELEMENT DiskOnChipOsUpdate (Description?,Source)>
<!ELEMENT Execute (Description?,FileName,CommandLine)>
<!ELEMENT UpdateRegistry (Description?,FileName)>
<!ELEMENT Reboot (#PCDATA)>

<!ATTLIST CopyFile
  OSImage (Yes|No) "No"
  OSImageSize CDATA "0"
  Register (Yes|No) "No"
  Execute (Yes|No) "No">

<!ELEMENT Description (#PCDATA)>
<!ELEMENT Source (#PCDATA)>
<!ELEMENT Destination (#PCDATA)>
<!ELEMENT CommandLine (#PCDATA)>
<!ELEMENT FileName (#PCDATA)>
```




Notes



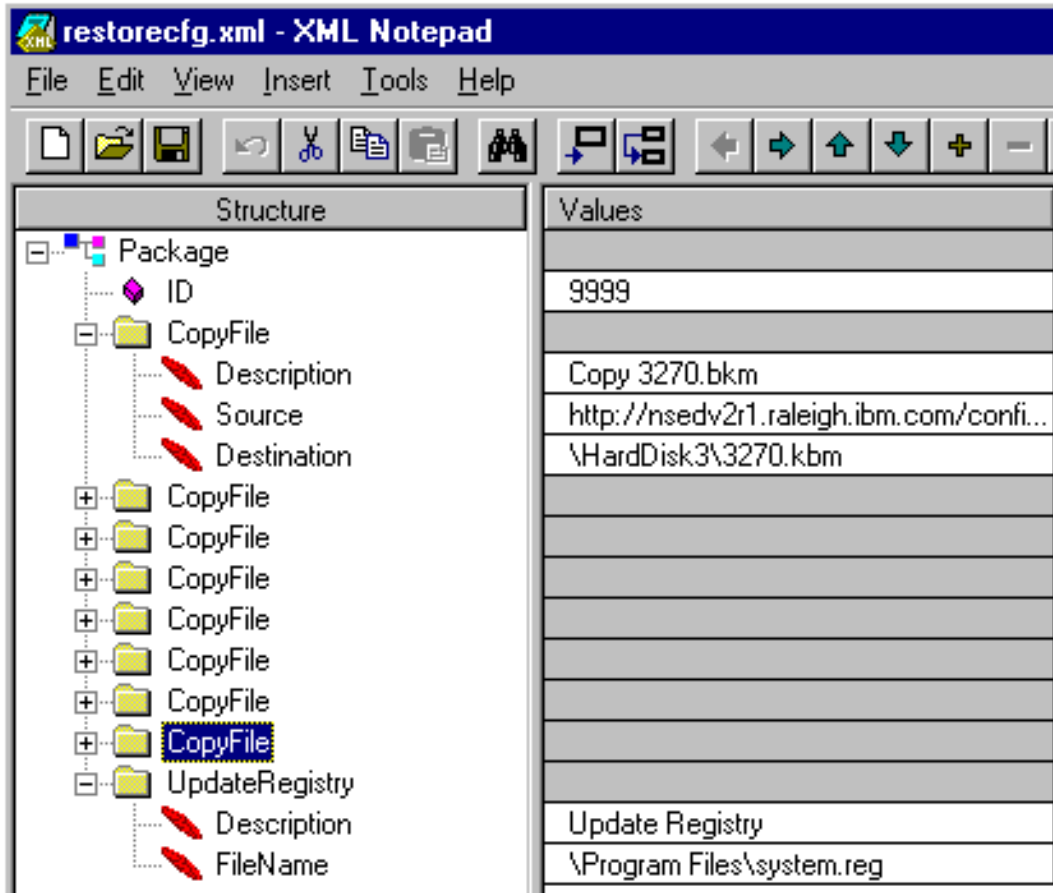
How do we know if the XML data is correctly stated? How do we validate it?

Validating XML means verifying that the XML data is well formed (meaning syntactically correct) and that it adheres to a specific representation. That representation is defined using Document Type Definitions (DTD) and XML schemas to formally declare XML representations.

Shown on the right here is a sample DTD that we use with the WBT for software update XML files.



XML NotePad



- Edits and Validates XML files against a DTD
- Free Editor
- IE 5.0 can also be used

- Available from <http://msdn.microsoft.com/xml/notepad/xpsetup.exe>



Notes



XML can be written any text editor.

However, there is an XML Notepad editor, which is available free from Microsoft, that can be used to work with XML and to validate the XML definitions against a specific DTD.

Other XML resources are:

- XML Developer Center: <http://msdn.microsoft.com/xml/default.asp>
- XML Magazine: <http://www.xmlmag.com/>
- XML.com: <http://www.xml.com>
- DevX: <http://www.xml-zone.com/>



How the Updater Works?



- **User enters the URL of the XML command file and clicks on Update Now**
- **CE Remote Updater fetches the XML command file from the remote server**
- **The XML command file is parsed using the XML Parser**
- **Each update command in the command file is executed**
- **Binary files are pulled from the server as indicated by the XML command**
- **CE Remote Updater indicates the operation as complete**



Notes



What is the update process?

From the Terminal Connection Manager, the user brings up the Terminal Properties by using PF2 and selects the Update tab. The user then enters the URL of the XML command file that needs to be executed.

- That XML file name might be the same every time, or it might be different. If we do not want to user to have to change the entry in the Update tab on every update, the administrator can change the content of the XML file at the host while always maintaining the same name
- Or the user might be instructed to enter a different XML file name dependent on the update that needs to be performed

The user then clicks on the Update Now button

This causes the CE Remote Updater program to fetch the XML command file from the remote server and to parse it (interpret its contents).

Each update command in the XML file is executed in sequence, and according to the instructions contained in the file.

Most of these commands consist of pulling (downloading) binary files from the server to be added to existing files or to replace existing files on the flash card.

The Updater program indicates the operation as complete when all the commands (directives) have been processed.



XML Command File Elements



- **Description of software files to be downloaded**
 - New or replacement executables (EXE)
 - New or replacement libraries (DLL)
 - Registry update files (REG)
 - Whole new Windows CE binary (NK.BIN)
- **Pointer to the network location of those files**
- **Other directive to be executed**
 - Execute, wait for execution, reboot, etc.
 - (Package ID to check against local history log)



Notes



What actually do the directives in an XML file consist of?

They essentially describe the software files to be downloaded. These files can be, for example:

- New or replacement executables (EXE)
- New or replacement libraries (DLL)
- Registry update files (REG)
- Whole new Windows CE binary (NK.BIN)

The command directives contain pointers to the network location of these files and have therefore everything needed to fetch these files and download them to the terminal.

Other directives that can be performed are program executions, wait for the completion of a program before going on to the next directive, and reboot the terminal.



Command File Update Directives



- **<Package>**
 - Root tag, identifies the file as an Updater update package file
- **<CopyFile>**
 - Copies down any file from network <Source> to terminal <Destination> with attributes
- **<Execute>**
 - Runs a program that already exists on the terminal
- **<UpdateRegistry>**
 - Uses contents of a .reg file to update terminal's Windows CE registry
- **<Reboot>**
 - Reboots the terminal
- **<DiskOnChipOsUpdate>**
 - Performs a progressive update of the Windows CE operating system



Notes



This is a summary of the actual directives that can be specified. The details for these can be found in the documentation.

<Package>

- This is the root tag that identifies the file as an Updater update package file

<CopyFile>

- This is to copy any file from a network location <Source> to the terminal <Destination> with certain attributes

<Execute>

- This runs a program that already exists on the terminal and usually indicates that the updater should wait for completion before processing the next directive.

<UpdateRegistry>

- Uses contents of a .reg file to update terminal's Windows CE registry.

<Reboot>

- Reboots the terminal

<DiskOnChipOsUpdate>

- Performs a progressive update of the Windows CE operating system. This particular directive causes the update to occur directly from the network. If there is enough RAM available on the station, it is preferable to use the staged update, using <copyfile>, in order to diminish the time window where the flash card is vulnerable (in write mode).



SP1 Install XML Example



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Package SYSTEM "SoftwareUpdate.dtd">
<!-- Upgrad to IBM WBT SP1 -->
<Package ID="0181">
  <CopyFile>
    <Description>Copy saveconfig.exe</Description>
    <Source>http://nsedv2r1.raleigh.ibm.com/sp1/getconfig.exe</Source>
    <Destination>getconfig.exe</Destination>
  </CopyFile>
  <Execute WaitForExit="Yes">
    <Description>Save Configuration</Description>
    <FileName>getconfig.exe</FileName>
    <CommandLine> </CommandLine>
  </Execute>
  <CopyFile OSImage="Yes" OSImageSize="0xA1018F">
    <Description>SW Update to SP1</Description>
    <Source>http://nsedv2r1.raleigh.ibm.com/sp1/nk_0181a.bin</Source>
    <Destination>\Program Files\nk.bin</Destination>
  </CopyFile>
  <Reboot>Yes</Reboot>
</Package>
```



Notes



Here is a real example of an XML file. This is the file used to install the WBT's Service Pack 1. This update consists only of an update of the Windows CE kernel image, and this XML file was supplied as part of the Service Pack 1 Update package.

The Service Pack 1 consisted of a new OS image (a file called nk_0181a.bin) and an executable called saveconfig.exe that is used to save the previous configuration and restore it after the update.

The xml file contains four directives to instruct the Updater program to do the following:

- **Copy** the file saveconfig.exe from the sp1 directory on the nsedv2r1 server to the root directory (of the file system on the flash card)
- Then **execute** the saveconfig.exe executable, and wait for its completion before proceeding to the next instruction
- **Copy** the file nk_0181a.bin file from the sp1 directory on the nsedv2r1 server to the /Program Files directory (of the file system on the flash card) under the name nk.bin
- Then **reboot** the unit



Export



Terminal Properties [OK] [X]

General | Input | Display | Network | System | Modem | Update | Security | **Export**

Export Configuration URL:
ftp://myserver/export_directory)

ftp://nsedv2r1.raleigh.ibm.com/configurations [Export Now...]

Export Configuration

Export device configuration to
"ftp://nsedv2r1.raleigh.ibm.com/configurations" is com

Contents of 'D:\WBT\code\configurations'

Name	Size	Type
3270.kbm	26KB	KBM File
3270euro.kbm	23KB	KBM File
3270euro.trf	2KB	TRF File
3885abf1.ses	2KB	SES File
5250.kbm	27KB	KBM File
5250euro.kbm	23KB	KBM File
5250euro.trf	2KB	TRF File
logon.mcr	1KB	MCR File
myattrib.acm	1KB	ACM File
Myhotspots.hot	5KB	HOT File
SaveConfig.reg	9KB	Registratio...
special.trf	2KB	TRF File



Notes



What we have discussed so far is the role of the Remote Updater program in updating the software components on the WBT. However, the updater program can also be used to download new components to the flash card, or to restore components, as we indicated earlier on when we have talked about cloning and about saving the configuration files.

So how is that accomplished?

The official way is to use the Export Tab of the Terminal Properties panel, illustrated here, and supply the name of an FTP server and a directory where the files are to be stored.

A click on Export causes the upload to the FTP server of the Registry file and all the configuration files in the `harddisk3` subdirectory.

A confirmation message is issued indicating that the export has completed. The image in the bottom right hand corner is an image of Windows Explorer on the FTP server showing the configuration directory (`D:\;WBT\code\configurations`) after the upload.

Notice the `SaveConfig.reg` at the bottom of the list, which is a copy of the Windows CE Registry. But what are these other files that have been uploaded?



What types of Components?



- **The Operating System (nkxxxxxx.bin)**
- **Windows CE Registry (system.reg)**
- **Applications (*.exe)**
- **Dynamic Link Libraries (*.dll)**
- **Emulator Configuration Files (3885abf*.ses)**
- **Keyboard Map Files (*.kbm)**
- **HotSpots Map Files (*.hot)**
- **Attributes Map (*.acm)**
- **Character Translation Files (*.trf)**
- **Macros (*.mcr)**



Notes



Usually, the of components that need to be updated are the operating system and the applications, but there are other components as well, such as keyboard maps files, that may also be in need of update.

What type of components exist on the flash card that might need to be downloaded or updated or saved and restored? This chart summarizes most of the types of files that may exist.

- The Operating System image file is an nkxxxxxx.bin file, where xxxxxx is usually a date format (such as 051500)
- The Windows CE Registry (system.reg) that contains most of the configuration parameters
- Applications (*.exe) like emulators
- Dynamic Link Libraries (*.dll)
- The Emulator Configuration parameters are not maintained in the Registry but in files named 3885abf*.ses. The Registry contains a minimal entry pointing to these files
- The Keyboard Map Files (*.kbm) which can be customized user maps or standard supplied maps
- HotSpots Map Files (*.hot)
- Attributes Map (*.acm)
- Character Translation Files (*.trf)
- Macros (*.mcr), such as an automated logon macro for an emulator session for example.



Restoring a Configuration



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Package SYSTEM "SoftwareUpdate.dtd">
<!-- Restore Configuration files -->
<Package ID="9999">
  <CopyFile>
    <Description>Copy 3270.bkm</Description>
    <Source>http://nsedv2r1.raleigh.ibm.com/configurations/3270.kbm</Source>
    <Destination>\HardDisk3\3270.kbm</Destination>
  </CopyFile>
  <CopyFile>
    <Description>Copy 3885abf1.ses</Description>
    <Source>http://nsedv2r1.raleigh.ibm.com/configurations/3885abf1.ses</Source>
    <Destination>\HardDisk3\3885abf1.ses</Destination>
  </CopyFile>
  <CopyFile>
    <Description>Copy Registry File</Description>
    <Source>http://nsedv2r1.raleigh.ibm.com/configurations/SaveConfig.reg</Source>
    <Destination>\Program Files\system.reg</Destination>
  </CopyFile>
  <UpdateRegistry>
    <Description>Update Registry</Description>
    <FileName>\Program Files\system.reg</FileName>
  </UpdateRegistry>
</Package>
```




Notes



Now that we have uploaded and saved the configuration data from one workstation, we can use these saved configuration in two ways:

- In the event that the configurations are lost or corrupted on this particular station, these files can be downloaded to the same station in order to restore the configuration files as they were at the time they were saved.
- Or, if we want to duplicate this configuration on many other units, we can have these same files downloaded to other units using the Remote Updater tool.

Illustrated here is a sample XML file that we could use for that purpose. In this example, we have shortened the file to download just a keyboard map, an emulator session and the registry. If the name of this file is `restorecfg.xml` and the alias of the directory where it resides is "configurations", then we would, on the Update tab, specify the command `http://nsedv2r1.raleigh.ibm.com/configurations/restorecfg.xml`.

Even if you are unfamiliar with XML, this is pretty easy to read and to recognize the actions that are being requested.

- Copy the `3270.kbm` file from the `/configurations` directory on the `nsedv2r1.raleigh.ibm.com` server to the `harddisk3` directory
- Copy the `3885abf1.ses` file from the `/configurations` directory on the `nsedv2r1.raleigh.ibm.com` server to the `harddisk3` directory
- Copy the `SaveConfig.reg` file from the `/configurations` directory on the `nsedv2r1.raleigh.ibm.com` server to the `/programs` directory and store as `system.reg`
- Update the Registry (meaning the active registry which is in memory at this time) with this new registry information

Note that the Registry information may need to be edited before being downloaded to a station other than the one it came from if the IP configuration data is different, which would be the case if static IP is used. Exact instructions how to do this are given in the product reference publication.



Viewing Flash Card Contents



The screenshot shows a Windows Explorer window titled "Exploring - V:\Windows". The left pane shows the "All Folders" tree with "C\$ on 'Client' (V:)" selected and circled in red. Below it, the "Windows" folder is also circled in red. The right pane shows the "Contents of 'V:\Windows'" table with the following data:

Name	Size	Type
Desktop		File Folder
Programs		File Folder
Recent		File Folder
StartUp		File Folder
BootTimeUpdate.exe	78KB	Application
ceicamgr.exe	181KB	Application
device.exe	15KB	Application
drvcnfg.exe	81KB	Application
dynace.exe	19KB	Application
filesys.exe	85KB	Application
gwes.exe	389KB	Application
mstsc.exe	180KB	Application
nk.exe	287KB	Application
ReflashBoot.exe	5KB	Application
remnet.exe	46KB	Application
sesswiz.exe	40KB	Application



Notes



We have been talking about many components up to now, all of which are located on the WBT's flash card in a local file system.

How does one take a peek at what is on the flash card?

Actually, there isn't a function on the WBT's user interface that allows a display of the files on the flash card.

However, there is a way to do this with an ICA session. When defining an ICA session to a MetaFrame server, a client drive is automatically defined when the client connects in (assuming these options have not been turned off in the MetaFrame configuration), and once the drive is defined, the user can use Windows Explorer in the ICA session to see the directory structure of the information on the flash card.

This is illustrated here where the V drive has been mapped to the flash card on the WBT. The contents of the Windows directory displayed here shows some of the .exe modules.



Sample HardDisk3 Contents



The screenshot shows two overlapping Windows Explorer windows. The top window, titled 'Exploring - V:\Program Files', displays the contents of the Program Files directory. The bottom window, titled 'Exploring - V:\HardDisk3', displays the contents of the HardDisk3 directory.

Exploring - V:\Program Files

File Edit View Tools Help

All Folders

- Desktop
- My Computer
- 3½ Floppy (A:)
- (C:)
- (D:)
- ddrive on 'Nsedv2r1' (E:)
- C\$ on 'Client' (V:)
 - HardDisk
 - HardDisk2
 - HardDisk3
 - My Documents
 - Program Files**
 - Temp
 - Windows

Contents of 'V:\Program Files'

Name	Size	Type
system.reg	9KB	Registration Entries

Exploring - V:\HardDisk3

File Edit View Tools Help

All Folders

- Desktop
- My Computer
- 3½ Floppy (A:)
- (C:)
- (D:)
- ddrive on 'Nsedv2r1' (E:)
- C\$ on 'Client' (V:)
 - HardDisk
 - HardDisk2
 - HardDisk3**
 - My Documents
 - Program Files
 - Temp
 - Windows

Contents of 'V:\HardDisk3'

Name	Size	Type
3270.kbm	26KB	KBM File
3270euro.kbm	23KB	KBM File
3270euro.trf	2KB	TRF File
3885abf1.ses	2KB	SES File
5250.kbm	27KB	KBM File
5250euro.kbm	23KB	KBM File
5250euro.trf	2KB	TRF File
logon.mcr	1KB	MCR File
myattrib.acm	1KB	ACM File
Myhotspots.hot	5KB	HOT File
special.trf	2KB	TRF File



Notes



Of more interest to us at this time, when we discuss configuration parameters, is the content of the Programs subdirectory, which contains the Registry file, and the contents of the HardDisk3 subdirectory, that contains the configuration files that we have discussed before, such as the emulator 3885*.ses files, and the keyboard remap files, etc.

We have illustrated those here, in these two bitmaps.

CAUTION: The official way to modify these files is to download them using the Updater function we have discussed earlier. However, since it is possible to manipulate these files via an ICA session, it is possible to erase files, and replace them, and modify them just like any other file system so one must be very cautious in the use of this tool.

So if you copied these to another location, say the D drive for example, they could be erased from the V drive and recopied back from the D drive just like any other files.



Sample (Partial) Registry



REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Clients\Registration\TEC\Connections\DETVMIC1]
"DeleteCmdLineArgs"="/Delete:"DETVMIC1""
"DeleteExe"="\Windows\sesswiz.exe"
"PropertiesCmdLineArgs"="/Properties:"DETVMIC1""
"PropertiesExe"="\Windows\sesswiz.exe"
"StartCmdLineArgs"="\HardDisk3\3885abf1.ses"
"StartExe"="dynace.exe"
"Description"="DETVMIC1"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Clients\Registration\ICA\Connections\NSEDWTSE]
"LkPid"=dword:1F0F776
"DeleteCmdLineArgs"="/D:NSEDWTSE"
"DeleteExe"="CEICAMGR"
"PropertiesCmdLineArgs"="/P:NSEDWTSE"
"PropertiesExe"="CEICAMGR"
"StartCmdLineArgs"="/L "NSEDWTSE""
"StartExe"="WFICACE"
"Description"="NSEDWTSE"
```



What does the Registry look like?

Here is a partial sample of the Windows CE Registry file, and it looks just like any Windows Registry.

Like all registries, it can be edited but with great care.

One case where there is a need to edit the registry is when an administrator wants to pull configuration files from one WBT unit and push it back down to multiple WBTs in a cloning process.



Some Registry Keys



REGEDIT4

```
[HKEY_CURRENT_USER\Software\FutureSoft\DCS\WBT\Options\PrnSetup]
[HKEY_CURRENT_USER\Software\FutureSoft\DCS\WBT\Recent URL List\Hotspots Mappers]
[HKEY_CURRENT_USER\Software\FutureSoft\DCS\WBT\Recent URL List\Keyboard Mappers]
[HKEY_CURRENT_USER\Software\FutureSoft\DCS\WBT\Recent URL List\Character Mappers]
[HKEY_CURRENT_USER\Software\FutureSoft\DCS\WBT\Recent URL List\Attribute Mappers]
[HKEY_CURRENT_USER\Software\FutureSoft\DCS\WBT\Connectors\Telnet\Settings\HostName]
[HKEY_CURRENT_USER\Software\DCS\WBT\DCS\WBT\Recent File List]
[HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client]
[HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\nsedwtse]
[HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default\AddIns\RDPPDR]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Clients\Registration\TEC\Connections\DETVMIC1]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Clients\Registration\ICA\Connections\NSEDWTSE]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Clients\Registration\ICA\Global\WFCClient]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Clients\Registration\RDP\Connections\nsedwtse]
[HKEY_LOCAL_MACHINE\HARDWARE\Microsoft\WBT\INPUT]
[HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\KEYBD]
[HKEY_LOCAL_MACHINE\Comm\Tcpip\Parms]
[HKEY_LOCAL_MACHINE\Drivers\Cyxdisp]
[HKEY_LOCAL_MACHINE\Drivers\Cyxdisp\DisplaySettings]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Keyboard Layout]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Shell]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WBT\Shell\Startup\10]
[HKEY_LOCAL_MACHINE\SOFTWARE\BSQUARE\WBT\SoftwareUpdate]
[HKEY_LOCAL_MACHINE\SOFTWARE\BSQUARE\TerminalProperties\SecurityConfiguration]
[HKEY_LOCAL_MACHINE\Printers]
[HKEY_LOCAL_MACHINE\Printers\Ports]
[HKEY_LOCAL_MACHINE\Printers\PCL Laser]
[HKEY_LOCAL_MACHINE\Printers\PCL Inkjet]
[HKEY_LOCAL_MACHINE\Printers\Vprinter]
```




Notes



If you define only an RDP session, an ICA session and one emulator session, the list in this chart shows all the Registry keys that the Registry would contain.

So there is a small number of keys only that are required.



Sample 3885abf*.ses file



[GENERAL TAB]

DConnectMessages=1
ConnectMessages=1
HideMouseCursor=0
AutoConnect=1
FullScreenMode=0
ErrorMessage=1
CommadbarState=1
SaveMessages=1

[CONNECTOR]

LineMode=1
HardClose=0
LocalEcho=0
CustomTerm=1
WaitData=0
SocketPort=0
Host=detvmic1.lexington.ibm.com
TermType=
CRPairing=2
Service=
Network Type=0
BreakType=0
AutoReconnect=0
BinaryMode=0
SubSys=telnet.con
SPX Socket=
AllTimingMarks=0
Port=23

[MACRO]

bAutoStart=0
Macrofile=\\HardDisk3\

[DISPLAY]

CursorType=0
SessionToolbar=0
MinFontSizeX=2
MinFontSizeY=2
VerticalScrollbarMode=2
BottomStatusLine=1
UserFontFacename=Courier New
TopStatusLine=0
EnableBlinking=1
HistorySize=100
ScrollRate=5
HorizontalScrollbarMode=0
TerminalBorder=1
UserFont=0
CursorVisible=1
EnableHotspots=0
SubSys=tdisplay.dis
UserFontSizeX=8
CursorBlink=1
ReverseHistory=1
UserFontSizeY=13
Autosize=1
ErasedPageSave=1
AutoscrollToCursor=0

[EMULATION]

IBM3270_ATTKEYSEQ=Telnet Method
IBM3270_SHORTNAME=
IBM3270_RESPECTNUMERIC=0
HotspotsKeyString=
IBM3270_CHARACTERSET=English-US C370 CECV V2
IBM3270_TERMTYPE=0
KeyboardMapFile=
CharTransFile=Default
IBM3270_AUTOSKIP=0
IBM3270_LUNAME=
IBM3270_CONVERTNULLS=0
IBM3270_TYPEAHEADDELAY=300
IBM3270_ALTSIZE=0
IBM3270_SOUND=0
IBM3270_UPPERCASE=0
IBM3270_TN3270E=1
IBM3270_NOTIFY=0
IBM3270_EXTATTRIB=0
IBM3270_HLLAPINULLS=1
IBM3270_HLLAPIENABLE=0
IBM3270_LONGNAME=
SubSys=ibm3270.emu
IBM3270_GRAPHICALOIA=1
IBM3270_OEMREPLY=0
IBM3270_TERMMODEL=0
AttributeMapFile=
IBM3270_DISPLAYNULLS=0
IBM3270_TYPEAHEAD=1



Notes



The emulators on the WBT do not store their configuration information in the Registry (for some reason) but in a separate file called 3885abf*.ses.

There is one file for each emulator session defined and the file is very similar to an *.ini file.

What does a 3885abf*.ses configuration look like?

Here is a partial sample file to give you an idea of the types of parameters contained in these files.

Many of these parameters contain a binary value of 0 or 1 indicating the absence or presence of a certain feature or function, and some, like the host address (illustrated in red here) are self explanatory.

These files can also be edited, modified and downloaded. However, one must be careful as it is sometimes difficult to know which file represent which connection definition. The connection name is entered in the registry and points to one of these 3885* files.



WBT Code Download Site



Address: <http://techsupport.services.ibm.com/nc/>

IBM United States ShopIBM Support

Home Products Consulting Industries News About IBM Call 1-888-SH

Search

ShopIBM

IBM Network Station™ and IBM NetVista™ Software that brings network computing to life.

New!
NetVista ThinClient Software
 → NetVista, WBT
 → Thin Client Utilities

Downloads

- [Thin Client Express Service Utility V1R1](#)
Product ID 5648-D83
File name: tceServiceUtil.exe size: ~16MB
- [Readme](#)
- [Thin Client Manager Operations Utility V1R1](#)
Product ID 5648-D84
File name: tcmOpUtil.exe, size: ~13MB
- [Readme](#)

CD Order Information

- [Ordering Thin Client Utilities on CD](#)

Downloads

- NetVista N2200w, [Service pack 1](#)
File name sp1.exe size: ~3.5MB

Documentation

- [Service pack 1 Installation notes](#)

Registration

- Register to download software

Software on CD

- V1R3 Software
- V2R1 Software
- NetVista

V2R1 and V1R3 Downloads

- Linux™
- PC Server

Documentation

- Publications
- Redbooks



Notes



Where was the Service Pack 1 obtained from?

From the standard NetVista Thin Client Download site (techsupport.services.ibm.com/nc) as illustrated in this composite image where we have rearranged some of the components.

The NetVista Thin Client box appears actually at the bottom of this panel, and from there, the NetVista, WBT displays the Service Pack 1 download. This is also where you can have access to the NetVista utilities and to the product publications.



Documentation - Web Site



The screenshot shows a web browser window with the address bar containing `http://www.ibm.com/nc/pubs`. The page header features the IBM logo, navigation links for Home, News, Products, Services, Solutions, and About IBM, and buttons for Shop IBM and Support. A search bar is present below the navigation. The main content area is titled "Thin Client Publications" and lists two product categories:

- N2200e Thin Client Express**
 - [Reference Information](#): This information supports the IBM NetVista N2200e.
- N2200w Thin Client for Windows-based Terminal Standard 1.5**
 - [Reference Information](#): This information supports the IBM NetVista N2200w.
 - [Emulation Help Files](#): This information includes self-extracting emulator help files for use with the IBM NetVista N2200w.



Notes



The publications can also be accessed via the standard NetVista Thin Client publications web site, www.ibm.com/nc/pubs.

This is also where the online help files for the emulators and for the NetPrint server can be downloaded.

Note that these emulator help files are not installed and available on the WBT itself, but can be installed on a the Windows Terminal Server to which the WBTs have access via an RDP or ICA session.