View 4.0.0

VMware View Architecture Planning Guide

- View Components:
  - **Client devices** - Windows PCs use View Client. A Mac or Linux PC uses a Web browser and View Portal. Windows devices can also use View Portal, but some functionality is not supported. Thin client devices use View thin client software.
  - **Connection server** – acts as a broker for client connections. Authenticates users through Active Directory and directs the request to the appropriate VM, physical or blade PC or Windows Terminal Services server.
    - Authenticating users
    - Entitling users to specific desktops and pools
    - Managing desktop sessions
    - Establishing secure connections between users and desktops
    - Single sign-on
    - Setting and applying policies

    Configuration data is stored in an embedded LDAP directory. In the DMZ, you can install and configure View Connection Server as a security server. Security servers are not required to be in an Active Directory domain.
  - **Client** - software for accessing View desktops. Authorization can require Active Directory credentials, a UPN (User Principle Name), a smart card PIN or an RSA SecurID token. Protocols include PCoIP, Microsoft RDP and HP RGS.
  - **Portal** – uses a web browser to display their View desktop. On Linux clients View Portal requires rdesktop, and on Mac OS/X View Portal requires Microsoft Remote Desktop Connection Client for Mac.
  - **Agent** - install the View Agent service on all VMs, physical systems and Terminal Service servers that you use as sources for View desktops.
  - **Administrator** - Web-based application allows administrators to configure View Connection Server, deploy and manage View desktops, control user authentication and troubleshoot end user issues.
  - **Composer** - A software service on a vCenter Server instance that manages VMs. You can then create a pool of linked clones from a specified parent VM.
  - **vCenter Server**
- Client support:

	RDP	PCoIP	HP RGS	USB access	Wyse MMR	Virtual Printing	Offline Desktop
Win 2000	Yes					Yes	
Win XP Pro	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Win XP Home	Yes	Yes		Yes	Yes	Yes	
Vista Business SP1 & SP2	Yes	SP2 only	Yes	Yes			
Vista Ultimate SP1 & SP2	Yes	SP2 only	SP2 only	Yes	SP1 only	SP1 only	
Vista Enterprise SP2	Yes	Yes	Yes	Yes			
RHEL 5.1 – portal only	Yes						
SLED 10 – portal only	Yes						
Ubuntu 8.04 – portal only	Yes						
Mac OS X 10.4 – portal only	Yes						
Mac OS X 10.5 – portal only	Yes						

- PCoIP is a new high-performance remote display protocol provided by VMware. This protocol is available for View desktops that are sourced from VMs, Teradici clients and physical machines that have Teradici-enabled host cards. PCoIP can compensate for an increase in latency or a reduction in bandwidth. It is optimized for delivery of images, audio and video content. Limitations:
  - Must be Windows XP Pro SP 2 or 3, or Windows Vista SP 1 or 2.
  - Cannot use the virtual printing feature.
  - Smart cards are not supported.
  - View Portal cannot use PCoIP.
  - Requires x86-based processor with SSE2 extensions.

	RDP	PCoIP	HP RGS
Monitors	Multiple monitors in span mode	Up to 4 at 1920x1200	Multiple monitors in span mode
Copy & paste between local & View desktops	Yes	Yes	
Configurable Adobe Flash bandwidth	Yes	Yes	Yes
Colour depth	32 bit	32 bit	
Encryption	128 bit	128 bit, AES	
VPN support		Yes	
Connection to Security server in DMZ	Yes		

- Administrators can configure the ability to use USB devices, such as thumb flash drives, VoIP (voice-over-IP) devices and printers, from a View desktop. This feature is called USB redirection.
- The virtual printing feature allows end users to use local or network printers from a View desktop without requiring that additional print drivers be installed in the View desktop. Not available for USB printers, you must install the required print drivers in the View desktop.
- Wyse MMR (multimedia redirection) enables full-fidelity playback when multimedia files are streamed to a View desktop. Supports:
  - AC3
  - MP3
  - MPEG-1, MPEG-2, MPEG-4-part2
  - WMA
  - WMV 7, 8, and 9
- The Wyse MMR port is 9427 by default.
- If you do not use the single-sign-on feature, end users must log in twice. They are first prompted to log in to View Connection Server and then are prompted log in to their View desktop. If smart cards are also used, end users must sign in three times because users must also log in when the smart card reader prompts them for a PIN.
- Regardless of the display protocol, you can use multiple monitors with a View desktop. If you use PCoIP, you can adjust the display resolution and rotation separately for each monitor. PCoIP allows a true multiple-monitor session rather than a span mode session.
- You can create a virtual desktop pool from one of the following sources:
  - A physical system such as a physical desktop PC or a Windows Terminal Services server.
  - A VM that is hosted on an ESX server and managed by vCenter Server.
  - A VM that runs on VMware Server or some other virtualization platform that supports View Agent.

  If you use a vCenter VM as a desktop source, you can automate the process of making as many identical virtual desktops as you need. Using pools to manage desktops allows you to apply settings to all virtual desktops in a pool.
- **Persistent pools** - each user is assigned a particular View desktop and returns to the same virtual desktop at each login.
- **Non-persistent pools** - virtual desktop is optionally deleted and re-created after each use, offering a highly controlled environment. Can be used like a computer lab, kiosk environment or a shifts of users.
- Using View Composer increases the storage savings because all desktops in a pool share a virtual disk with a base image.
- View Composer uses a base image, or parent VM, and creates a pool of up to 512 linked-clone VMs.
- When you create a linked-clone desktop pool, a full clone is first made from the parent VM. The full clone, or replica, and the clones linked to it are placed on the same datastore, or LUN (logical unit number). If necessary, you can use the rebalance feature to move the replica and linked clones from one LUN to another. When you create persistent desktop pools, View Composer also creates a separate user data disk for each virtual desktop. The end user's profile and application data are saved on the user data disk.
- The recompose feature allows you to make changes to the parent VM, take a snapshot of the new state, and push the new version of the image to all, or a subset of, users and desktops.
- You can use the refresh feature to bring the desktop back to its default values. This feature also reduces the size of linked clones, which tend to grow over time.
- Using existing processes for Application Provisioning - If you push applications out to large numbers of virtual desktops at exactly the same time, you might see significant spikes in CPU usage and storage I/O.
- Requirements depend largely on the type of worker who uses the virtual desktop and on the applications that must be installed.
  - **Task workers** - perform repetitive tasks within a small set of applications, usually at a stationary computer. Might all log in to their virtual desktop at the same time.
  - **Knowledge workers** - daily tasks include accessing the Internet, using email, and creating complex documents, presentations, and spreadsheets.
  - **Power users** - include application developers and people who use graphics intensive applications.
- If the RAM allocation is too low, storage I/O can be negatively affected because too much memory swapping occurs. If the RAM allocation is too high, storage capacity can be negatively affected because the paging file in the guest operating system and the swap and suspend files for each VM grow too large.
- Because virtual desktop performance is sensitive to response times, you must set nonzero values for RAM reservation settings. Guarantees that idle but in-use desktops are never completely swapped out to disk.
  - **Windows page file** - by default this file is sized at 150% of guest RAM. This file causes linked-clone VMs and thin-provisioned storage to grow.
  - **ESX swap file** - .vswp, the size of the swap file is equal to the unreserved portion of guest RAM.
  - **ESX suspend file** - .vmss, created if you set the desktop pool logoff policy so that the virtual desktop is suspended when the end user logs off. The size of this file is equal to the size of guest RAM.

- PCoIP RAM overheads:

Display	1 monitor	2 monitors	4 monitors
640x480	64MB	64MB	64MB
800x600	64MB	64MB	96MB
1280x720	64MB	96MB	128MB
1600x1200	96MB	128MB	256MB
1920x1080	96MB	128MB	256MB
1920x1200	96MB	128MB	256MB
2048x1536	128MB	128MB	384MB
2560x1600	128MB	256MB	512MB

- A good starting point is to allocate 1024MB for Windows XP desktops and 1536MB for Windows Vista desktops.
- When estimating CPU, calculate that another 10 to 25 percent of processing power is required for virtualization overhead and peak periods of usage.
- During your pilot phase use a performance monitoring tool, such as Perfmon, to understand both the average and peak CPU use levels for groups of workers.
- A good starting point is to pilot 8 VMs per core.
- Storage space required must take into account the following files for each virtual desktop:
  - ESX suspend file is equivalent to the amount of RAM allocated to the VM.
  - Windows page file is equivalent to 150 percent of RAM.
  - Log files take up approximately 100MB for each VM.
  - The vmdk file must accommodate the operating system, applications and future applications and software updates. The virtual disk must also accommodate local user data and user installed applications if they are located on the virtual desktop rather than on file shares.

  You can also add 15 percent to this estimate to be sure that users do not run out of disk space.
- Use the LSI Logic SCSI adapter for Windows XP desktops (not the default).
- View Composer can create and provision up to 512 desktops per pool. View Composer can also perform a recompose operation on up to 512 desktops at a time.
- Maximum number of simultaneous connections that a VMware View deployment can accommodate.

Type	Connection servers	Connections
RDP	1	2,000
RDP	5	5,000
RDP – tunneled	3	2,000
PCoIP	1	2,000
Unified to Physical PC	1	100
Unified to terminal server	1	200

- A node is a single VMware ESX server that hosts VM desktops in a VMware View deployment. A node can host 8 VMs per core and 64 VMs per LUN.
  - Cores per node - 8 for ESX 3.5, 16 for ESX 4.0 U1
  - NICs - 32 VMs per NIC
- Because each ESX server in a View cluster usually hosts more than 40 VMs, and because of View Composer limitations (EDIT – I think this is related to the HA limit, not Composer), the cluster must contain no more than 8 servers.
- Each View desktop pool must be associated with a vCenter resource pool.
- In very large VMware View deployments, vCenter performance and responsiveness can be improved by having only one cluster object per datacenter object.
- Storage IOPS has the most effect on how quickly desktops recover from a server failure.
- VMware View environments can occasionally experience I/O storm loads, during which all VMs undertake an activity at the same time. I/O storms can be triggered by guest-based agents such as antivirus software or software-update agents. I/O storms can also be triggered by human behavior, such as when all employees log in at nearly the same time in the morning.
- VMware recommends that you provide bandwidth of 1Gbps per 100 VMs, even though average bandwidth might be 10 times less than that.
- Display traffic - as a starting point for a pilot, plan for 150 to 200Kbps of capacity for a typical knowledge worker.
- For wide-area networks (WANs), if you use the RDP display protocol, you should have a WAN optimization product to accelerate applications for users in branch offices or small offices.
- RDP data is tunneled through HTTPS and is encrypted using SSL. A client can access multiple desktops over a single HTTPS connection. If a user temporarily loses a network connection, the HTTP connection is reestablished and the RDP connection automatically resumes.
- Clients that use the PCoIP or HP RGS display protocols do not use the tunnel connection.
- With direct client connections, an HTTPS connection is still made between the client and the View Connection Server host for users to authenticate and select View desktops, but the second HTTPS connection (the tunnel connection) is not used.
- For clients that use the Microsoft RDP display protocol, direct client connections are appropriate only if your deployment is inside a corporate network.

- Each View Connection Server instance is joined to an Active Directory domain. Users are authenticated against Active Directory for the joined domain and any additional user domains with which a trust agreement exists.
- Use the `vdmadmin` command to configure domain filtering.
- Administrators can enable individual View Connection Server instances for RSA SecurID authentication by installing the RSA SecurID software on the View Connection Server host and modifying View Connection Server settings.
- Enabling a View Connection Server instance to use smart card authentication typically involves adding your root certificate to a truststore file and then modifying View Connection Server settings.
- A security server is a special instance of View Connection Server that runs a subset of View Connection Server functions.
- You must implement a hardware or software load balancing solution if you install more than one security server.

FW Port	Source	Destination	Protocol	Description
80	Any	Security server	HTTP	External devices connect when SSL is disabled
443	Any	Security server	HTTPS	External devices connect when SSL is enabled (default)
3389	Security server	View Desktop	RDP	RDP traffic to desktops
4001	Security server	Connection server	JMS	JMS traffic
4100	Connection server	Connection server	JMS	JMS inter-router traffic
8009	Security server	Connection server	AJP13	AJP13 forwarded traffic
9427	Security server	View Desktop	TCP	Wyse MMR
32111	Security server	View Desktop	TCP	USB redirection

- TCP ports that are opened on the Windows firewall by the View Agent installation program. Ports are incoming TCP ports :
  - RDP 3389
  - USB redirection 32111
  - MMR 9427
  - PCoIP 50002 (TCP and UDP)
  - HP RGS 42966
- View Client with Offline Desktop data is downloaded and uploaded through port 902. It must be accessible to your ESX host.
- You can use the restricted entitlements feature to restrict View desktop access based on the View Connection Server instance that a user connects to. You assign one or more tags to a View Connection Server instance. Then, when configuring a desktop or desktop pool, you select the tags of the View Connection Server instances that you want to be able to access.
- You can also use restricted entitlements to control desktop access based on the user-authentication method.
- High-level tasks for creating a View deployment:
  1. Set up the required administrator users and groups in Active Directory.
  2. (Optional) Install and set up VMware ESX servers and vCenter Server
  3. (Optional) Install View Composer on vCenter Server.
  4. Install View Connection Server.
  5. Copy the Active Directory GPO templates from the View Connection Server machine to the Active Directory server and import them.
  6. Do an initial configuration of View Connection Server.
  7. Create one or more VMs that can be used as a template for full-clone desktop pools or as a parent for linked-clone desktop pools. Install the desired applications or VMware ThinApp applications.
  8. Install View Agent on the VMs and physical machines you want to use as desktop sources.
  9. Create an individual View desktop or a View Desktop Pool.
  10. Entitle users or user groups to desktops.
  11. Set desktop policies.
  12. Install View Client on end users' machines or direct them to use View Portal to install the required components.
  13. Have end users access their View desktops.
  14. Manage and monitor users and desktops.

VMware View Upgrade Guide

- VMware View 4.0 components are compatible with most VMware View 3.0.x and 3.1.x components.

View 4	Connection server 3.x	Agent 3.x	Client 3.x	Thin Client 3.x	Web Portal 3.x	Composer 1.0
Connection server 4.0		Yes	Yes	Yes		No
View Agent 4.0	Yes		Yes	Yes	Yes	Yes
View Client 4.0	Yes	Yes			Yes	Yes
Thin Client 4.0	Yes	Yes			Yes	Yes
Web Portal 4.0		Yes	Yes	Yes		Yes
View Composer 2.0	Yes	No				

- Until both View Connection Server and View Composer are upgraded, View Composer operations do not work. If you have View Connection Server 4.0 and View Composer 1.0, users can connect to their desktops, but no new linked-clone desktops can be created. Recompose, rebalance, and refresh operations will not work.
- You must have at least one ESX Server 3.5 Update 3 or 4 and one VirtualCenter Server 2.5 Update 3 or 4.

- You must install View Composer on the same system as vCenter Server or VirtualCenter Server.
- View Administrator 4.0 can be used with Internet Explorer 7 & 8 and Firefox 3.0 & 3.5
- Before you upgrade View Connection Server or before you upgrade any of the vSphere components that View Connection Server relies on:
  - Verify it meets the system requirements for View 4.
  - If View Connection Server is installed in a VM, take a snapshot of the VM.
  - Open View Administrator and document all the settings in the Desktop and Pools View and the Global Settings section of the Configuration View.
  - Use the vdmexport.exe utility to back up the View LDAP database.
  - Document the IP address and system name of the machine on which View Connection Server is installed.
  - Determine if your company has written any batch files or scripts that run against the View database on the View Connection Server instance, and if so, document their names and locations.
  - If you use load balancers for View Connection Server, document the configuration settings for the load balancers.
- If you are upgrading only View Composer and not upgrading VirtualCenter Server:
  1. If View Composer is installed in a VM, take a snapshot of the VM.
  2. Back up the VirtualCenter database and the View Composer database.
  3. Back up the SSL certificates, if applicable.
  4. Document the IP address and system name of the machine on which vCenter Server is installed.
  5. For all linked-clone desktop pools, disable provisioning of new VMs.
  6. If any desktop pools are set to refresh the OS disk on logoff, edit the *Desktop/Pools* settings for that pool and set *Refresh OS disk on logoff* to *Never*.
  7. If any desktop pools are scheduled to do a refresh or recompose operation, cancel these tasks.
- Components that you must upgrade include View Connection Server, View Client, and View Agent. You might also need to upgrade View Composer, vCenter Server, ESX hosts and the VMs on ESX hosts.
- When you upgrade vSphere components separately from View components:
  1. Back up the VirtualCenter database and the View Composer database.
  2. Back up the View LDAP database from a View Connection Server instance by using the `vdmexport.exe` utility.
  3. After you upgrade VMware Tools in VMs that are used as View desktops, reinstall View Agent.
- Although vCenter Server is supported on 64-bit operating systems, View Composer is not.

View Manager Administration Guide

- Major features of View Manager:
  - Connection brokering - manages the connections between users and their virtual desktops.
  - “Smart pooling” capabilities—range of persistent and non-persistent pooling.
  - Flexible deployment options.
  - High availability.
  - Integration with Microsoft Active Directory.
  - Integration with VMware vSphere.
  - Secure access.
  - Support for two-factor authentication—With RSA SecurID.
  - USB client device and Virtual Printing support.
  - Web-based management.
  - Support for non-vSphere systems—physical machines or terminal services.
  - Linked clone technology allows multiple desktops to be deployed from a single base image.
- View Manager components:
  - Connection Server - acts as a broker.
  - Agent - allows desktops to be managed by View Manager.
  - Client - a locally installed software application that communicates with View Connection Server.
  - Client with Offline Desktop (experimental).
  - Portal - Web-based version of View Client.
  - Administrator - Web application to configure View Connection Server, deploy and manage desktops, control user authentication, initiate and examine system events, and carry out analytical activities.
  - Composer - a software service on the vCenter Server to rapidly deploy multiple linked clone desktops.
- View Connection Server is not supported on servers that have the Windows Terminal Server role installed.
- 2GB RAM or higher—3GB RAM is recommended for deployments of 50 or more View Manager desktops, applies to any Connection Server instances for high availability or external access.
- View Connection Server can be installed on 32-bit Windows Server 2003 SP 2 - R2 or non-R2, Std or Ent. VMware vSphere 4 U1, VI 3.5 U3 or U4 is recommended. U5 not supported. VI 3.0.2 is supported.
- To use View Composer, VMware vSphere 4 Update 1 or VI 3.5 U3 or U4 is required.

- To use Offline Desktop experimental feature, VI 3.5 U3 or U4 is required. Offline Desktop is not supported with vSphere 4.
- Active Directory versions supported: 2000, 2003 & 2008.
- View Connection Server has been certified with version 6.1 and 7.1 of RSA Authentication Manager.
- OS support (32 bit):

OS	Agent - virtual	Agent - physical	Client	Offline Desktop
Windows 2000			Yes	
Windows XP Home SP2/SP3			Yes	
Windows XP Pro SP2/SP3	Yes	Yes	Yes	Yes
Windows XPe			Yes	
Windows Vista Home SP2			Yes	
Windows Vista Enterprise SP1	Yes	Yes		
Windows Vista Business/Ultimate SP1	Yes	Yes	Yes	
Windows Vista Enterprise/Business/Ultimate SP2	Yes	Yes	Yes	
Windows 7	Tech preview	Tech preview	Tech preview	
Windows 2003 Enterprise SP2/R2	Yes	Yes		

- Portal browser support:
  - Windows – IE 6 SP2 & 7
  - Linux – Firefox 2.0 & 3.0, Java JRE 1.5.0 & 1.6.0, rdesktop
  - Mac OS X – 10.4 & 1.5 Safari, , Java JRE 1.5.0, RDC 2.0
- You cannot use the View Composer feature of View Manager to deploy desktops that run Windows Vista Ultimate Edition or Windows XP Professional SP1.
- Smart cards and smart card readers that use a PKCS#11 or Microsoft CryptoAPI provider are supported.
- View Client, View Client with Offline Desktop, and View Agent cannot be installed on the same machine. You cannot install View Client with Offline Desktop on any system that has VMware ACE, Player, Server, or Workstation installed.
- Database Support and Requirements for View Composer:
  - Microsoft SQL Server 2000 Std/Ent SP4, 2005 Express, 2005 Std/Ent SP1/SP2. For Windows XP, apply MDAC 2.8 SP1 to the client. Use SQL Server driver for the client. Not compatible with vSphere 4 U1.
  - Microsoft SQL Server 2005 Std/Ent Edition 64bit SP2
  - Oracle 9i release 2 Std/Ent (9.2.0.8)
  - Oracle 10g Std/Ent R2 (10.2.0.1.0) - First apply patch 10.2.0.3.0 to the client and server, and then apply patch 5699495 to the client.
  - Oracle 10g Std/Ent R1 (10.1.0.3.0)
  - Oracle 10g Ent R2 (10.2.0.1.0) 64-bit - First apply patch 10.2.0.3.0 to the client, and then apply patch 5699495 to the client.
  - Oracle 10g Ent R2 (10.2.0.3.0) 64-bit
  - Oracle 11g Std/Ent - Not compatible with VirtualCenter Server 2.5.
- View Connection Server functions:
  - User authentication
  - User desktop entitlements with View LDAP
  - Virtual desktop session management
  - Coordination of the secure connection establishment, virtual desktop connection, and single sign-on
  - Administration server used by View Administrator Web client
  - Virtual desktop pool management
- The Connection Server must be joined to an Active Directory domain—but must not be a domain controller.
- The domain user account used to install View Connection Server must have administrator privileges on that server and administrative credentials for vCenter Server. The server can be installed as a either a standard, replica or security server.
- Security server instances do not contain the View LDAP component.
- Replica servers are additional View Connection Server instances that are installed in order to provide high-availability and load balancing.
- Assign the View Manager administrator the role of administrator for a datacenter or cluster where pools will be created so that they can make the required changes.
- View Administrator URL is: https://<server>/admin
- Do not rely on replica servers to act as your backup mechanism, as any data lost from one instance will be lost from all members.
- You can use the `vdmexport.exe` to manually export View LDAP data.
- `C:\Program Files\VMware\VMware View\Server\bin\vdmexport.exe`
  - Export View Manager configuration data: `vdmexport > vdmconfig.ldf`
  - Import View Manager configuration data: `C:\windows\adam\LDIFDE -i -f vdmconfig.ldf -s 127.0.0.1 -z`
- To support a large deployment of View Manager desktops, you can optimize the Windows Server computers on which you install View Connection Server.
  1. If your View Manager deployment is likely to use more than 800 concurrent client connections, you should increase the number of available ephemeral ports.
     - Number of ephemeral ports = ((5 * clients) / servers) + 10

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\MaxUserPort

2. Optimize the size of the TCB hash table on each View Connection Server instance.
   - For View Connection Servers: number of hash table rows on each View Connection Server instance = ((5 * clients) / servers) + desktops + 20
   - For Security Servers: number of hash table rows = ((5 * clients) / security servers) + 10

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\MaxHashTableSize
3. You can increase the size of the Java VM (JVM) heap memory configuration. The default heap size for the View Connection Server JVM is 512MB. This configuration can support approximately 750 concurrent View desktop sessions.
   Do not allocate a JVM heap size greater than 1.5GB. If you do, the View Connection Server service fails to start.
   HKEY_LOCAL_MACHINE\SOFTWARE\VMware, Inc.\VMware VDM\Plugins\wsnm\tunnelService\Params\JvmOptions
   Edit the -Xmx parameter to have the value -Xmx1024m.
4. Optimize the virtual memory by changing the system page-file settings.

- View Administrator:
  - **Desktops and Pools**- create, deploy, administer, and monitor your virtual desktops.
  - **Users and Groups** - monitor the desktop assignments and active sessions of entitled View Manager users.
  - **Configuration** - multiple sections that allow you to analyze desktop usage, configure licensing, connections, authentication criteria, and so forth.
    - Location of backup LDAP files. By default: C:\Documents and Settings\All Users\Application Data\VMware\VDM\backups
  - **Global settings** - information about global configuration parameters that apply to all areas of the product.
    - *Session timeout* - Determine how long (in minutes) users are allowed to keep sessions open after they log in - default is 600.
    - If a security server exists in, you must have an appropriately configured config.properties file resident on the security server.
  - **Events** - examine events generated by the actions taking place within the View Connection Server.
- Desktop sources:
  - View Manager Provisioned and vCenter Server Managed.
  - View Manager Non-Provisioned and vCenter Server Managed.
  - Unmanaged Desktop Sources - includes VMs running on VMware Server and on other virtualization platforms that support View Agent. Blade PCs, physical PCs, and Terminal Servers on which you can install View Agent.
- Desktop delivery models:
  - **Individual Desktop –** a single, pre-existing back-end source:
    - Entitled to many users or user groups; however, only one active user at a time.
    - Not provisioned automatically.
  - **Manual Pool –** a pool of desktop sources:
    - Multiple users to multiple desktop mapping; however, only one active user on a desktop at a time.
    - Not provisioned automatically.
    - Both persistent and non-persistent access modes.
  - **Automated Pool –** a pool that contains one or more dynamically generated desktops that are automatically created and customized by View Manager from a vCenter Server VM template:
    - Multiple users to multiple desktop mapping; however, only one active user on a desktop at a time.
    - Provisioned automatically.
    - Administrator specifies a template and a customization specification which is used to provision desktop sources.
    - Both persistent and non-persistent access modes.
  - **Terminal Server Pool:**
    - Pool of TS desktops served by a farm comprising of one or more terminal servers.
    - Least session count based load balancing.
    - Administrators should deploy a roaming profile solution to enable user settings and personalization to be propagated to the currently accessed desktop.
    - Only the RDP display protocol is supported for Terminal Server Desktop Pools.
- For guest systems with more than one virtual NIC, you must configure the subnet that the View Agent will use.
  HKLM\Software\VMware Inc.\VMware VDM\Node Manager\subnet = n.n.n.n/m (REG_SZ)
- The unique ID is used by View Manager to identify the desktop pool and is the name that clients see when logging in.
- If you are using a Windows Vista VM, you *must* set the power policy to *Ensure VM is always powered on*.
- Desktop pools:
  - **Persistent**—allocated statically in order to ensure that users connect to the same system each time they log in.
  - **Non-persistent**—allocated dynamically.
- Automated desktop pools can use the linked clone feature to rapidly deploy desktops from a single “Parent VM”.
- View Composer linked clone virtual desktops use a snapshot instead of a VM template as a base image.
- Customization specifications are optional, but they can greatly expedite automated desktop pool deployments.
- **VM naming pattern** - By default, a prefix is used to identify all desktops in a pool as part of the same group. The prefix can be up to 13 characters in length and a numeric suffix is appended to this entry in order to distinguish each desktop from others in the same pool. You can

override this behavior by entering a name that contains a token representing the pool number; the token can appear anywhere in the name. For example: amber-{n}-desktop After deployment{n} is replaced with the pool number of the desktop. Fixed length tokens can be entered using the n:fixed= construction. For example: amber-{n:fixed=3} After deployment{n:fixed=3} is replaced with a fixed-length pool number for each the desktop: amber-001, amber-002, amber-003 and so forth.A 15 character limit applies to names that contain a token, but only to the "replaced" form where the token length is fixed.

- The restricted entitlements feature uses tag matching to determine whether a View Connection Server instance can access a particular View desktop or desktop pool.
- Tag Matching Rules:

View Connection server	Desktop/Desktop Pool	Access Permitted
No tags	No tags	**Yes**
No tags	One or more tags	**No**
One or more tags	No tags	**Yes**
One or more tags	One or more tags	**Only when tags match**

- Desktops and desktop pools that do not have any tags can be accessed by any View Connection Server instance. View Connection Server instances that do not have any tags can only access desktops and desktop pools that also do not have any tags.
- If you use a security server, you must configure restricted entitlements on the View Connection Server instance the security server is paired with. You cannot configure restricted entitlements on a security server.
- You cannot modify or remove a tag from a View Connection Server instance if that tag is still assigned to a desktop or desktop pool and no other View Connection Server instances have a matching tag.
- Restricted entitlements take precedence over other desktop entitlements.
- Active sessions:
  - Disconnect Session - user is disconnected, but their session remains active.
  - Logoff Session - user is disconnected and their session is logged off.
  - Reset VM - desktop is shutdown and restarted without a graceful logoff and disconnection.
- If you want to prevent users from accessing their desktops you can disable the View Connection Server to prevent clients from logging in. Currently logged in users are not affected.
- *Delete a desktop pool from a View Connection Server*: You are given the option to remove the VMs from View Manager only, which means they are still visible in vCenter Server, or to delete them from disk.
- The functionality offered by View Client and View Portal is derived from a common set of locally installed base components. Users who have already installed View Client will be invited to install an additional ActiveX control on their browsers when they use View Portal for the first time.
- An quick way of installing the View Client application is to visit the View Portal page and allow the browser to automatically install the required components on the client system. However, if you do this:
  - View Portal does not provide Virtual Printing or USB support.
  - Windows Start Menu entries for View Portal are not created after installation.
- If you install View Client from the executable, Virtual Printing and USB support are offered within the application, and Start Menu entries are created. You can deselect the *USB Redirection* component. You can deselect the *Log in as current user* component.
- View Portal does not support USB redirection, regardless of installation path.
- Windows 2000, Windows XP Home, Windows Vista Home Basic, and Windows Vista Home Premium do not support the *Log in as current user* feature.
- Certain View Client features can be controlled through policies.
- The process of setting the external URL is not the same for all types of servers. For standard or replica servers you can set the URL from within View Administrator. For a security server you must create or edit a properties file that contains the inbound connection details and save it in a directory located under the security server installation path.
- Security server properties: C:\Program Files\VMware\VMware View\Server\sslgateway\conf\config.properties
  Restart the VMware View Security Server service for changes to take effect.
- Certificates are only required for standard, replica or security servers that receive direct connections from their clients.
- To create a self signed SSL certificate: `%ProgramFiles%\VMware\VMware View\Server\jre\bin\keytool -genkey -keyalg "RSA" -keystore keys.p12 -storetype pkcs12 -validity 360`
- To create a certificate signing request (CSR): `keytool -certreq -keyalg "RSA" -file certificate.csr -keystore keys.p12 -storetype pkcs12 -storepass <secret>`
- To import the certificate (request a certificate in PKCS#7 format): `keytool -import -keystore keys.p12 -storetype pkcs12 -storepass <secret> -keyalg "RSA" -trustcacerts -file certificate.p7`
- To configure the View Connection Server to use the new certificate:
  1. Place a new certificate file in the following location on a standard, replica, or security server instance of View Connection Server: `C:\Program Files\VMware\VMware View\Server\sslgateway\conf`
  2. Create or edit the following file on each server: `C:\Program Files\VMware\VMware View\Server\sslgateway\conf\locked.properties`
  3. Add the following properties:

keyfile=keys.p12
keypass=<secret>

4. Restart the View Connection Server service.

- Smart card authentication is only supported by View Client; it is not supported by View Administrator, View Portal, or by offline desktop instances accessed through View Client with Offline Desktop.
- Each client system using smart card authentication will require View Client and a Windows-compatible smart card reader to be installed.
- To add the third-party root CA to the NTAuth store in Active Directory: `certutil -dspublish -f <certificate> NTAuthCA`
- A truststore is a keystore that is used by View when making decisions about which clients to trust. In order for View Connection Server to authenticate smart card users and connect them to their desktops, the root certificate for all trusted users must first be added to the server truststore. `%ProgramFiles%\VMware\VMware View\Server\jre\bin\keytool -import -alias <alias> -file <certificate> -keystore <truststore_filename>`
- All types of View Connection Server support smart card authentication but it is recommended that only security servers are configured to allow smart card access.
- In environments where not all users will authenticate using a smart card it is also recommended that you configure a new (or an additional) security server specifically for the purpose of client smart card authentication.
  1. Copy the truststore file you created to the View Connection Server: C:\Program Files\VMware\VMware View\Server\sslgateway\conf\
  2. Create a text file called locked.properties that contains the following entries:
     trustKeyfile=<truststore filename>
     trustStoretype=JKS
     useCertAuth=true
  3. Restart the View Connection Server service
- Log files: <drive>:/Documents and Settings/All Users/Application Data/VMware/VDM/logs on the View Connection Server or Security Server.
- Smart card authentication replaces Windows password authentication only. If SecurID is enabled, users are required to authenticate using this mechanism also.
- The usual form of UPN is user@domain. For a user to connect using smart card authentication, their account in Active Directory must have a valid UPN associated with their userPrincipalName property. The UPN for each user who requires smart card authentication must be set to the subject alternative name (SAN) contained within the root certificate of the trusted CA. You need to provide this information only if the certificate was issued from a domain other than the one in which the user presently resides. The most straightforward way of adding this information to Active Directory is to use the ADSI Edit utility
- To launch View Client, with all the connection, user, and desktop criteria provided: `"C:\Program Files\VMware\VMware View\Client\bin\wswc" -serverURL <server> -userName <username> -password <password> -domainName <domain> -desktopName <desktop>`
- Command line properties override system policies, which in turn override user policies.
- ThinPrint Device print data compression levels: No images, Extreme, Maximum, Optimal and Normal.
- Adobe Flash bandwidth reduction is available for Internet Explorer sessions on Microsoft Windows only, and for Adobe Flash versions 9 and 10 only.
- Adobe Flash render-quality modes: Do not control, Low, Medium, High. The system defaults to a value of Low.
- You can reduce the frame rate and thereby reduce bandwidth. Throttling modes:
  - Disabled: No throttling is performed. The timer interval is not modified.
  - Conservative mode. The lowest number of dropped frames.
  - Moderate mode.
  - Aggressive mode. The highest number of dropped frames.
- Audio speed remains constant regardless of which throttling mode you select.
- Adobe Flash bandwidth reduction settings do not take effect until View Client reconnects with the desktop.
- Users can override Adobe Flash content display settings. Display quality is improved as long as the cursor remains inside the Adobe Flash content.
- You can configure the View Agent *CommandsToRunOnConnect* and *CommandsToRunOnReconnect* group policy settings to pass client computer information to View Agent when users connect and reconnect to View desktops. View Agent writes this information to the registry in the View desktop. With third-party tools, you can write custom scripts to use this information.
- HP RGS consists of a server-side component, called RGS Sender, and a client-side component, the RGS Receiver.
- The first time one or more linked clones are created, a uniquely identified copy of the Parent VM, called a replica, is also created. All the desktop clones are anchored directly to the replica and not to the Parent VM.
- Replica VMs can be identified within vCenter Server by their *replica-* prefix followed by a unique ID. In vSphere Client 4, replica VMs are only visible in the Hosts and Clusters Inventory view. In VMware Infrastructure Client 3.5, replica VMs are also visible in the VMs and Templates Inventory view.
- The Parent VM can be updated or replaced without directly affecting the linked clone desktops.
- Replicas are treated as protected entities within vCenter Server.

- View Manager administrators can simultaneously update (or change) the operating systems of all linked clone desktops, install or update client applications, or modify the desktop hardware settings by carrying out these activities on the Parent VM and then anchoring the linked clones to a new snapshot of this configuration. This is called desktop recomposition.
- Administrators can also return the operating system data of each linked clone desktop, which may have expanded through ongoing usage, to its original state (that of the Parent VM) by carrying out a desktop refresh. Desktop user data is configured to reside on a separate disk, so it remains unaffected by desktop recomposition or desktop refresh actions.
- Operating system data disks and user data disks use thin provisioning.
- The storage overcommit level calculates the amount of storage that is greater than the physical size of the datastore that the clones would use if each clone were a full desktop.
  - None Storage is not overcommitted.
  - Conservative 4 times the size of the datastore. This is the default level.
  - Moderate 7 times the size of the datastore.
  - Aggressive 15 times the size of the datastore.
- Storage overcommit applies only to delta disks. It does not apply to user disks or standard (non-linked) clones.
- With VMware Infrastructure 3.5, recomposition is expedited by an additional protected linked clone desktop in VirtualCenter Server, called a source VM. This source VM is created alongside the replica when a linked clone desktop pool is first deployed. With VMware Infrastructure 3.5, the source VM is located with the replica inside a folder called VMwareViewComposerReplicaFolder in VirtualCenter Server.
- VMware vSphere 4 optimizes cloning, so it does not have to create the source VM.
- A desktop refresh is similar to a desktop recomposition but without any change to the base image.
- A desktop refresh can be carried out either on demand, as a timed event, or when the operating system data reaches a specified size.
- Desktop rebalance - Rebalancing the LUNs evenly distributes any selected (or all) VMs between the available logical drives.
- In order to rebalance the desktops it is necessary for View Manager to automatically refresh their operating systems against their current base image and return the system data to its baseline state.
- In persistent configurations, dedicated disks can be used to keep the operating system and user data separate. In non-persistent configurations the user data is transient so user data is not protected if the system is recomposed or refreshed.
- Persistent desktops can be set to refresh automatically when the user logs off.
- Non-persistent pools can be set to delete after first use.
- QuickPrep is a system tool executed by View Composer during linked clone desktop deployment. QuickPrep personalizes each desktop created from the Parent VM. QuickPrep gives a new name, is joined to the appropriate domain and mounts the new volume that will contain the user profile information. It's restarted twice and a new computer account is created. These events also take place after a desktop refresh.
- After a desktop is created, refreshed or recomposed, a user-defined customization script can be run. Can also be run on desktops immediately before they are powered off. QuickPrep is responsible for ensuring that these scripts are executed.
- The View Composer service must be installed locally on the vCenter Server.
- You can deploy a linked-clone desktop pool on a cluster that contains ESX/ESXi 4 hosts mixed with ESX/ESXi 3.5 hosts, until you set *vSphere* mode.
- Create a user with the requisite level of authority to be used by the View Composer service to create linked clone desktops and add them to your domain.
- You must assign a role to the vCenter Server user entry and give it the appropriate level of authority over the objects it creates and manages.
- View Composer requires that the vCenter Server user is also a system administrator on the machine hosting the service (the vCenter Server).
- If you are using a Windows Vista VM, you must set the power policy to *Ensure VM is always powered on.*
- Only clusters of 8 hosts or fewer are supported for linked clones.
- You can recompose, refresh or rebalance only those linked clone desktops that are part of a persistent pool. If you want to change the Parent VM of a non-persistent linked clone desktop pool, you must modify the pool directly by using the pool deployment wizard.
- When selecting the ODBC data source during the installation of the View Composer service you can use an existing database that already contains View Composer data. You must first transfer the RSA key container created by the original View Composer service to the new host system.
- The SviConfig utility upgrades or restores the View Composer database: `C:\Program Files\VMware\VMware View Composer\sviconfig.exe`
- The View Client with Offline Desktop downloads a copy of their desktop VM and "locks" the online desktop VM. vCenter Server operations such as powering on the online desktop, taking snapshots and editing the VM settings are disabled.
- Once network access is restored the checked out VM can be:
  - Backed up—the online system is updated with all new data and configurations, but the offline desktop remains checked out on the local system and the online lock remains in place.
  - Rolled back—the offline desktop is discarded and the online lock is released.
  - Checked in—the offline desktop is uploaded to the online host and the online lock released.
- Once checked out, Offline Desktop uses thin provisioned virtual disks.
- Contact is attempted every 5 minutes.
- The data on each offline system is encrypted and has its lifetime controlled through policy.
- Offline Desktop supports tunneled or non-tunneled communications.

- When tunneling is enabled, all traffic is routed through the View Connection Server.
- Offline Desktop – supported Desktops:

Type	Persistence	Desktop Configuration	Offline Desktop
Individual desktop	Non-persistent	VM managed by vCenter	**Yes**
		Physical or unmanaged VM	No
Manual desktop pool	Persistent	VM managed by vCenter	**Yes**
		Physical or unmanaged VM	No
	Non-persistent	All	No
Automatic desktop pool	Persistent	Non-linked clone	**Yes**
		Linked clone	No
	Non-persistent	All	No
Terminal server desktop pool	n/a	n/a	No

- View Client with Offline Desktop does not support the use of smart cards.
- You cannot download a desktop if another user is currently logged in to that desktop.
- ESX supports two simultaneous desktop checkouts. ESXi supports five simultaneous desktop checkouts.
- When a desktop is checked out, NAT is used for network communications. The MAC address of the offline system remains the same as its online equivalent.
- You cannot install View Client with Offline Desktop on any system that has the following applications installed:VMware Workstation, ACE, Player, Server.
- Some component policies can be assigned through View Administrator, whereas others are contained within Group Policy Objects inside Active Directory and are applied to users or desktops at the Windows registry level.
- Power policy controls how desktops behave when they are not in use.
- A View Manager desktop is not in use before the user has logged in, or after the user has disconnected or logged off.
- Power Policy Definitions:
  - Do nothing (VM remains on)
  - Always on (ensure VM is always powered on)
  - Suspend VM
  - Power off VM
- Circumstances under which the power policy is applied:
  - Individual Desktop (vCenter Server Managed VM) - After user disconnection or logoff.
  - Persistent Automated Pool - When not in use or after user disconnection or logoff. This policy only applies to unassigned desktops.
  - Non-Persistent Automated Pool & Non-Persistent Manual Pool - When not in use or after user disconnection or logoff. If the *Power Off* policy is applied after a disconnection, the session is discarded. If the *Suspend* policy is applied after a disconnection, an orphaned session could be created (the desktop is non-persistent so there is no guarantee that the user will ever be able to return to it). Ensure that *Automatic logoff after disconnect* is set to *Immediately* in order to prevent either scenario.
  - Persistent Manual Pool (vCenter Server Managed VMs) - After user disconnection or logoff. This policy only applies to unassigned desktops.
  - Physical Systems / Terminal Services Desktop Pool - N/A
- The policies tab in View Administrator control client components at the global, desktop pool, or desktop user level. By default, the user-level policy inherits settings from pool-level policy that, inherits its setting from a global policy.
  - USB Access: default – allow
  - MMR: default – allow.
  - Offline desktop: default – allow.
  - User-initiated rollback: default – allow.
  - Max time without server contact: default - 7 days.
- Where the new pool-level policy is more restrictive, a pool-level policy can be configured to override the equivalent global policy. The reverse is not true.
- User-level policies override global-level or pool-level policies, they can be more or less restrictive than either.
- GPOs can be applied to View Manager components at a domain-wide level to provide granular control over various areas of the View Manager environment
- A number of component-specific GPO templates are provided with View Connection Server that can be imported into Active Directory.
  - vdm_agent.adm contains properties relating to the authentication and environmental components of a client desktop controlled by View Agent.
  - vdm_client.adm contains properties relating to the configuration parameters of View Client.
  - vdm_server.adm contains properties relating to View Connection Server.
  - vdm_common.adm contains properties relating to all components of View Manager.
- GPO template files: C:\Program Files\VMware\VMware View\Server\Extras\GroupPolicyFiles
- With the User Configuration GPO you can set policies that apply to users, regardless of which desktop they connect to. These policies override any equivalent Computer Configuration Policies that may have been applied to the target desktop.

- Unified Access enables View Manager to provide a unified interface through which users can access their desktops being delivered by multiple back ends.
- You must set desktop parameters when you are configuring managed and unmanaged individual desktops, desktop pools, and terminal servers. The desktop parameters differ for managed and unmanaged resources.

Types	Desktop state	VM power policy	Auto logoff after disconnect	Users can reset	Multiple sessions per user
Individual managed	Yes	Yes	Yes	Yes	
Individual unmanaged	Yes		Yes		
Manual managed pool: persistent	Yes	Yes	Yes	Yes	
Manual managed pool: non-persistent	Yes	Yes	Yes	Yes	Yes
Manual unmanaged pool: persistent	Yes		Yes		
Manual unmanaged pool: non-persistent	Yes		Yes		Yes
Terminal server pool	Yes		Yes		

- If multiple vCenters are running in your environment, make sure that another vCenter is not using the same unique ID.
- View Manager includes a script called `vdm-support` that collects information for use by VMware Technical Support.
- On the View Connection Server you can run the script manually or by using the support tool in the *Start* menu. For View Client or on View Manager desktops running View Agent, you must run the script manually.
- Log levels: normal, debug or full. Default is debug.
- The support tool creates a folder called vdm-sdct containing the generated log files, and places it on the desktop of View Connection Server.
  - View Connection Server: C:\Program Files\VMware\VMware View\Server\DCT
  - View Client or View Portal: C:\Program Files\VMware\VMware View\Client\DCT
  - View Manager desktops running View Agent: C:\Program Files\VMware\VMware View\Agent\DCT
- The support script is: `cscript vdm-support.vbs`
- The svi-support script provided with View Manager offers component-specific support for View Composer: `C:\Program Files\VMware\VMware View Composer\cscript.wsf svi-support.wsf`
- Troubleshooting end user connection issues: http://kb.vmware.com/kb/1003642
- Troubleshooting pooling issues: http://kb.vmware.com/kb/1003658
- Troubleshooting USB issues: http://kb.vmware.com/kb/1003706

Getting Started with VMware View

1. Prerequisites:
   1. Obtain a valid license key for View Manager
   2. Install and configure vSphere 4 U1, VI 3.5 (U3 or U4 recommended, U5 not supported) or VI 3.0.2.
   3. View Manager uses your existing Active Directory infrastructure. Requires 2000, 2003 or 2008.
2. (Optional) Install View Composer on the vCenter system.
3. Install View Connection Server on a dedicated server.
4. Perform initial configuration with View Administrator:
   - Log in to View Administrator
   - Install the license key
   - Add the vCenter
   - (Optional) Configure an External URL for View Connection server
5. Prepare the VM for Deployment.
   - Configure the guest OS
     Windows XP Deployment Guide:
     http://www.vmware.com/files/pdf/resources/vmware-view-xp-deployment-guide.pdf
   - Install View Agent on the Guest OS
   - (Optional) Create a VM customization specification
     Customization specifications can expedite automated desktop pool deployments by providing configuration information for general properties, such as licensing, domain attachment, and DHCP settings. View Composer linked clone virtual desktops do not use customization specifications.
   - Create a VM template for standard virtual desktops
     Use the template as a desktop source for automated desktop pools. View Composer linked clone virtual desktops use a snapshot instead of a VM template as a base image.
6. Deploy a virtual desktop or desktop pool.
   - Desktop delivery models:
     - Individual Desktop—a desktop that allows a single, preexisting back-end source.

    - Manual Pool—a pool of desktop sources that is not provisioned automatically. Multiple users are mapped to multiple desktops. However, only one user is active on a desktop at a time.
    - Automated Pool—a pool that contains one or more dynamically generated desktops that are automatically created and customized by View Manager from a vCenter VM template.
    - Terminal Server Pool—a pool of terminal server desktop sources served by one or more terminal servers. A terminal server desktop source can deliver multiple desktops.
  - Type of desktop pool:
    - Persistent—Desktops are allocated statically to make sure that users connect to the same system each time they log in. Desktop assignment takes place the first time each user connects.
    - Non-persistent—Desktops are allocated dynamically when the user logs in and are returned to the pool when the user disconnects.
7. (Optional) Deploy a view composer virtual desktop pool.
   - When you use View Composer, linked clones are created from a centralized base image, called a Parent VM. After you have created the guest system and installed View Agent, you must take a snapshot. The Parent VM must be completely shut down before you take the snapshot. This snapshot is used as the baseline configuration for the first set of linked clone desktops anchored to the Parent VM.
   - Make sure that *Use linked clone technology to create desktops in this pool* is selected.
   - The *Use vSphere mode for View Composer* check box becomes available if you select a cluster that contains ESX/ESXi 4 hosts only. The new vSphere mode is more reliable than in previous releases and supports up-to-date hardware configurations.
   - If you want user data to be preserved after a refresh or recomposition event, select *Redirect user profile to a separate disk* and specify the maximum size of the user data disk and associated drive. If you are using multiple datastores, you can select *Use different datastores for user data disks and OS disks.*
   - If you do not want user data to be preserved after a refresh or recomposition event, select *Store user profile on the same disk as the OS.*
   - The *Storage Overcommit* column entry determines how aggressively the system assigns new VMs to the free space available on a datastore. As the level increases, less space is reserved for individual VM growth but more VMs fit on the datastore.
   - To join linked clone desktops to a domain, View Manager requires domain administrator credentials for the target domain.
8. Entitle users and groups to desktops and desktop pools.
9. Install and run view client.
10. (Optional) Next steps.
    - You can use policies to configure View components by controlling the logging of information, managing client access, restricting device usage, establishing security parameters for client usage, and so forth.
    - You can use the Events view of View Administrator to examine events generated by the actions taking place in View Connection Server.

Command-line tool for View Manager

- You must run the tool locally on the View Manager server. The executable is: `C:\Program Files\VMware\VMware View\Server\bin\vdmadmin.exe`
- You can:
  1. Assign a default desktop to a user: `vdmadmin -D`
  2. List user information: `vdmadmin -U`
  3. Change user assignments: `vdmadmin -L`
  4. Show which user was first to access a desktop: `vdmadmin -R`
  5. Remove the View manager entry for a replica server which is removed permanently: `vdmadmin -S`
     Run this command on one of the remaining servers in the cluster and not on the server that is being uninstalled.
  6. List Orphaned desktops: `vdmadmin -O`
     Revoking the entitlement for a user assigned a persistent desktop or physical system does not revoke the associated desktop assignment. When a user leaves the organization the desktop is considered to be orphaned.
  7. Configure domain filters: `vdmadmin -N`
     View Manager determines which domains are accessible by traversing trust relationships. Any change that you make to the domain search configuration applies to all View Connection Server instances in a View Connection Server group. You cannot exclude the primary domain to which a View Connection Server instance or security server belongs.
  8. Override IP Addresses: `vdmadmin – A`
     A View Agent reports the discovered IP address of the machine on which it is running to the View Connection Server instance. In secure configurations you can override the value provided by the View Agent and specify the override IP address that the managed machine should be using.

Extending Virtual Machine deletion with scripts

- When you delete a VM from the VMware® View™ environment, you might want to run your own scripts to remove Active Directory or database entries that reference the deleted machine.
- To enable a deletion script, you add an entry for the script to the Windows Registry, configure the account of the user who has the appropriate privileges to run the script, and enable the VMware View Script Host service.

- You must configure all View Connection Server hosts that need to run deletion scripts. There is no mechanism to propagate registry changes, VMware View Script Host service configuration changes, and deletion scripts between View Connection Server hosts.
- Deletion scripts cannot run interactively.

View Manager 4.0.1 Release Notes

- View Manager 4.0.1 includes support for VirtualCenter 2.5 Update 6 and ESX 3.5 Update 5.
- PCoIP now supports the following features:
  - Virtual Printing, which allows end users to use local or network printers from a View desktop without requiring that additional print drivers be installed in the View desktop.
  - Single sign-on support for third party providers such as Sentillion and Imprivata.
  - View Client supports international keyboards when using PCoIP.
- If using vSphere 4, then View 4.0.1 requires vSphere 4 Update 1.

ThinApp 4.5

ThinApp User's Guide

- Install ThinApp to isolate applications, simplify application customization, deploy applications to different operating systems and eliminate application conflict.
- ThinApp supports the following operating systems, applications and systems:
  - 32 bit platforms include Windows NT, Windows 2000, Windows XP, Windows XPE, Windows 2003 Server, Windows Vista, Windows Server 2008, Windows 7
  - 64 bit platforms include Windows XP 64 bit, Windows 2003 64 bit, Windows Vista 64 bit, Windows Server 2008 64 bit, Windows Server 2008 R2 64 bit, Windows 7 64 bit
  - 16 bit applications running on 32 bit Windows operating systems
  - 32 bit applications running on 32 bit and 64 bit Windows operating systems
  - Terminal Server and Citrix Xenapp
- Does not support:
  - 16 bit or non-x86 platforms such as Windows CE
  - 64 bit applications running on 32 bit or 64 bit Windows operating systems
  - 16 bit applications running on 64 bit Windows operating systems
- Cannot convert:
  - Applications requiring installation of kernel mode device drivers. ODBC drivers work because they are user mode drivers.
  - Antivirus and personal firewalls
  - Scanner drivers and printer drivers
  - Some VPN clients
- Some applications that provide shell integration have reduced functions.
- ThinApp isolates COM and DCOM services. They are accessible on the local computer only by other captured applications running in the same ThinApp sandbox.
- VMware recommends using a clean computer to install ThinApp because the environment affects the application capture process. Application installers skip files that already exist on the computer.
- A clean computer allows the capture process to scan the computer file system and registry quickly.
- Install ThinApp on the earliest version of the operating system you plan to support.
- Key files in the C:\Program Files\VMware\VMware ThinApp directory:
  - AppSync.exe Keeps captured applications up to date with the latest available version.
  - logging.dll Generates .trace files.
  - dll_dump.exe Lists all captured applications that are currently running on a system.
  - log_monitor.exe Displays the execution history and errors of an application.
  - relink.exe Updates existing packages to the latest ThinApp version installed on the system.
  - sbmerge.exe Merges runtime changes recorded in the application sandbox with the project and updates the captured application.
  - Setup Capture.exe Captures and configures applications through a wizard.
  - snapshot.exe Compares the pre-installation environment and post installation environment during the application capture process.
  - snapshot.ini Stores entries for the virtual registry and virtual file system that ThinApp ignores during the process of capturing an application. The snapshot.exe file references the snapshot.ini file.
  - template.msi Builds the MSI files.
  - thinreg.exe Registers captured applications on a computer. Includes setting up shortcuts, file type associations and Start menu items.
  - tlink.exe Links key modules during the build process of the captured application.
  - vftool.exe Compiles the virtual file system during the build process of the captured application.

    - vregtool.exe Compiles the virtual registry during the build process of the captured application.
- The Setup Capture wizard is the main method to capture applications and set initial application parameters.
- Advanced users who must capture applications from the command line can use the `snapshot.exe` utility.
- Capturing an application involves system scans, application configuration, package configuration and generation of the virtual application for distribution.
- ThinApp runs virtual applications according to the regional and language settings on the capture system rather than the settings on the system that runs the application. You can modify the default locale setting by commenting out the LocaleIdentifier parameter in the Package.ini file and rebuilding the application,
- Entry points are the executable files that act as shortcuts into the virtual environment and start the virtual application.
- During the build process, ThinApp generates one executable file for each selected entry point.
- Entry points start native executable files in a virtual context.
- ThinApp can use Active Directory groups to authorize access to the virtual application.
- Isolation modes determine the level of read and write access to the native file system outside of the virtual environment.
- The selection of isolation modes in the capture process determines the value of the DirectoryIsolationMode parameter in the Package.ini file. This parameter controls the default isolation mode for the files created by the virtual application except when you specify a different isolation mode in the ##Attributes.ini file for an individual directory.
- Directory isolation mode does not affect:
    - ThinApp treats write operations to network drives according to the SandboxNetworkDrives parameter in the Package.ini file.
    - If you save documents to the Desktop or My Documents folder, ThinApp saves the documents to the physical system.
- With *Merged* isolation mode, applications can read and modify elements on the physical file system outside of the virtual package.
- The advantage of using *Merged* mode is that documents that users save appear on the physical system in the location that users expect, instead of in the sandbox. The disadvantage is that this mode might clutter the system image.
- With *WriteCopy* isolation mode, ThinApp can intercept write operations and redirect them to the sandbox. You can use *WriteCopy* isolation mode for legacy or untrusted applications.
- The sandbox is the directory where all changes that the captured application makes are stored.
- When you delete the sandbox directory, the application reverts to its captured state.
- If you deploy the sandbox to a local machine, use the user's profile as the sandbox location.
- A network location is useful for backing up the sandbox and for users who log in to any computer and retain their application settings. Use the absolute path to the location.
- A portable device location is useful to keep the sandbox data on the device where the application resides.
- The primary data container is the main virtual application file that includes the ThinApp runtime and the read only virtual file system and virtual registry. The primary data container is specified in the ReadOnlyData parameter in the Package.ini file.
- MSI packages automate the process of registering file type associations, registering desktop and *Start* menu shortcuts and displaying control panel extensions. If you plan to deploy ThinApp executable files directly on each computer, you can accomplish the same registration by using the `thinreg.exe` utility.
- If the size of the primary container is smaller than 200MB, ThinApp creates an .exe file as the primary container.
- If the size of the primary container is larger than 200MB, ThinApp creates a separate .dat file as the primary container
- You can modify the Package.ini file to update the overall package. The file resides in the captured application folder. C:\Program Files\VMware\VMware ThinApp\Captures\
- Settings that you might modify:
    - DirectoryIsolationMode – Sets the isolation mode to *Merged*, *WriteCopy* or *Full*.
    - PermittedGroups – Restricts use of an application package to a specific set of Active Directory users.
    - SandboxName – Identifies the sandbox. You might keep the name for incremental application updates and change the name for major updates.
    - SandboxPath – Sets the sandbox location.
    - SandboxNetworkDrives – Specifies whether to direct write operations on the network share to the sandbox.
    - RequiredAppLinks – Specifies a list of external ThinApp packages to import to the current package at runtime.
    - OptionalAppLinks – Specifies a list of external ThinApp packages to import to the current package at runtime.
- Activate the parameter to edit by removing the semicolon at the beginning of the line.
- The ##Attributes.ini file exists in the folder macros of the project folder and applies configuration settings at the directory level. The Package.ini file applies settings at the overall application level. You can use the DirectoryIsolationMode, CompressionType and ExcludePattern parameters in an ##Attributes.ini file to override the Package.ini settings at the directory level.
- You can deploy captured applications with deployment tools, in a VMware View environment, on a network share, or as basic executable files.
- The `thinreg.exe` utility creates the *Start* menu and desktop shortcuts, sets up file type associations, adds uninstall information to the system control panel, and unregisters previously registered packages. The default location of the utility is C:\Program Files\VMware\VMware ThinApp.
- The Application Sync utility affects the `thinreg.exe` utility during the update process.
- If you add, modify or remove executable files, the `thinreg.exe` utility reregisters the file type associations, shortcuts and icons.

- The `thinreg.exe` utility monitors the PermittedGroups setting in the Package.ini file, using AD group membership.
- The `thinreg.exe` with the `/allusers` switch, creates all shortcuts and file type associations regardless of the PermittedGroups setting.
- `thinreg.exe` parameters:
  - `/a` registers a package for all users.
  - `/q` prevents the display of an error message
  - `/u` unregisters a package
  - `/r` re-registers a package
  - `/k` prevents the removal of registration information even if you are no longer authorized
  - `/noarp` prevents an entry in the Add/Removes programs
  - `/norelaunch` Starts the thinreg.exe without the elevated privileges
- Customising MSI files with package.ini parameters:
  - MSIInstallDirectory – sets the installation directory
  - MSIDefaultInstallAllUsers – sets the installation of the package for individual users.
  - MSIFileName – names the package. Must add this to generate the MSI files.
  - MSIRequireElevatedPrivileges – indicates whether an installer needs elevated privileges
  - MSIProductCode – makes it easier to install a new version. An MSI database contains a product code and an upgrade code.
- Application Streaming – the contents from the executable file stream to client computers in a block-based fashion. As an application requests specific parts of data files, ThinApp reads this information in the compressed format over the network using standard Windows file sharing protocol.
- After a client computer receives data, ThinApp decompresses the data directly to memory. Because ThinApp does not write data to the disk, the process is fast.
- VMware recommends that you use ThinApp streaming in a LAN-based environment.
- For WAN and Internet deployments that involve frequent or unexpected disconnections, either:
  - Use a URL to deploy the applications.
  - Use a desktop deployment solution to push the package to the background. Allow the application to run only after the entire package downloads.
- VMware recommends that you make the central shared directory for the package as read only.
- ThinApp stores application changes in the user sandbox. A common configuration is to place the user sandbox on another central storage device.
- You cannot use ThinApp to virtualize printer drivers.
- ThinApp provides the Application Sync and Application Link utilities to update applications with new versions or new components. The Application Sync utility updates an entire application package.
- The Application Link utility keeps shared components or dependent applications in separate packages.
- The Application Sync utility is useful for major configuration updates to the application.
- If the Application Sync utility attempts to update the application after an automatic application update, the version update stored in the sandbox take precedence over the files contained in the Application Sync version.
- The Application Sync utility updates entry point executable files.
- If you register virtual applications on the system using `thinreg.exe` and update applications with the Application Sync utility, you can update registrations by placing a copy of `thinreg.exe`, located in C:\Program Files\VMware\VMware ThinApp, alongside the updated package on the server.
- The Application Sync utility requires that the name of the primary data container, the file that stores virtual files and registry information, is the same for the old and new versions of an application.
- When a captured application creates child processes, ThinApp cannot complete the Application Sync process. To resolve the issue:
  - Log out and log in to the machine to stop the child processes.
  - Create a script to end the child processes.
  - Prevent the startup of the child process, e.g. ctfmon.exe associated with Microsoft Office and Internet Explorer applications.
- The Application Link utility connects dependent applications at runtime.
- ThinApp can link up to 250 packages at a time, which is useful for:
  - Large shared libraries and frameworks - Link runtime components, such as .NET, JRE, or ODBC drivers, with dependent applications.
  - Add-on components and plug-ins
  - Hot fixes and service packs
- ThinApp supports nested links with the Application Link utility.
- ThinApp loads an Application Link layer during application startup and merges registry entries and file system directories. If ThinApp finds a registry subkey or file system directory that did not previously exist in the main package or layer that is already merged, ThinApp uses the isolation mode specified in the layer being loaded. If the registry subkey or file system directory exists in the main package and a layer that is already merged, ThinApp uses the most restrictive isolation mode specified in any of the layers or main package. The order of most restrictive to least restrictive isolation modes is *Full*, *WriteCopy* and *Merged*.

- If you link two applications and you specify a value for the PermittedGroups parameter, the user account used for starting the application must be a member of at least one of the Active Directory groups for this parameter in the Package.ini files of both applications.
- Sandbox changes from linked packages are not visible to the base executable file.
- ThinApp imports linked applications according to the order of applications in the RequiredAppLinks or OptionalAppLinks parameter. If either parameter specifies a wildcard character that triggers the import of more than one file, alphabetical order determines which package is imported first.
- If the base application and a dependent package linked to the base application contain file or registry entries at the same location, a collision occurs. When this happens, the order of import operations determines which package has priority. The last package imported has priority in such cases and the file or registry contents from that package are visible to the running applications.
- The `AppSync.exe` utility forces an Application Sync update on a client machine.
- The `sbmerge.exe` utility make incremental updates to applications.
- You cannot use `AppSync.exe` to update packages stored in a location where standard users do not have write access.
- The `sbmerge.exe` utility merges runtime changes recorded in the application sandbox back into a ThinApp project.
- In Terminal Server environments, you can have multiple users executing different versions at the same time during the transition period.
- Starting an application locks the executable file package.
- If you store an application in a central location for many users, this file lock prevents administrators from replacing a packaged executable file with a new version until all users exit the application and release their locks.
- When you upgrade an application, you can control whether users continue to use their previous settings by keeping the sandbox name consistent in the Package.ini file. You can prevent users from using an older sandbox with an upgraded application by packaging the upgraded application with a new name for the sandbox. Starting the upgraded application the first time creates the sandbox with the new name.
- You can use the `relink.exe` utility to update an existing package or tree of packages to the latest version of ThinApp. Although you can install the latest version of ThinApp and run the `build.bat` utility to rebuild each target package with the latest ThinApp version, the `relink.exe` utility is a faster method to upgrade the ThinApp version of existing packages.
- Advanced users can customize the parameters of the virtual application outside of the capture process.
- The Setup Capture wizard sets the initial values of some of the Package.ini parameters.
- The Package.ini file contains the following headings:
  - [BuildOptions] - applies to all applications unless the entries specific to applications override these settings.
  - [<application>.exe]
  - [FileList]
  - [Compression]
  - [Isolation]
- The [FileList], [Compression] and [Isolation] parameters act as [BuildOptions] parameters but are grouped separately for backward compatibility reasons.
- You can use the DirectoryIsolationMode, CompressionType, and ExcludePattern parameters in an ##Attributes.ini file. The ##Attributes.ini file exists in the folder macros of the project folder.
- Parameter that apply to Package.ini or ##Attributes.ini files:
  - NetRelaunch – determines whether to restart an application from the local disk when you run the application from a network share or removable disk to address the slow startup of applications.
  - RuntimeEULA – controls the End EULA display for the package, addressing legacy EULA requirements. Do not modify.
  - VirtualComputerName – determines whether to virtualize the computer name to avoid naming conflicts between the deployment system and the capture system.
  - Wow64 – simulates a 32 bit environment
  - QualityReportingEnabled – specifies whether VMware can collect anonymous data on a package to improve ThinApp application support.
  - DirectoryIsolationMode – specifies the level of read and write access for directories to the physical file system.
    - ThinApp provides only the *Merged* and *WriteCopy* isolation mode options in the capture process. You can use the *Full* isolation mode outside the wizard to secure the virtual environment.
    - With *Merged* isolation mode, applications can read and modify elements on the physical file system outside of the virtual package. The advantage of using *Merged* mode is that documents that users save appear on the physical system in the location that users expect, instead of in the sandbox. The disadvantage is that this mode might clutter the system image.
    - With *WriteCopy* isolation mode, ThinApp can intercept write operations and redirect them to the sandbox.
    - You can use *WriteCopy* isolation mode for legacy or untrusted applications. Difficult to locate user data files is useful for locked down desktops.
    - *Full* isolation mode, ThinApp blocks visibility to system elements outside the virtual application package. Prevents application conflict between the virtual application and applications installed on the physical system. Do not use the *Full* isolation mode in the Package.ini file because that mode blocks the ability to detect and load system DLLs. You can use *Full* isolation mode as an override mechanism in the ##Attributes.ini files.
  - RegistryIsolationMode – controls the isolation mode for registry keys in the package.

  - The capture process does not set the value of this parameter. You can configure the registry isolation mode only in the Package.ini file. ThinApp sets the initial registry isolation mode to *WriteCopy*.
  - You can use *Full* isolation mode as an override mechanism. You can place exceptions to the configured RegistryIsolationMode parameter in the registry key text files in the project directory.
- FileTypes – file extensions that the `thinreg.exe` utility associates with an executable file. The capture process generates initial values that you cannot add to. You can remove extensions that you do not want to associate with the virtual package.
- Protocols – is similar to the FileTypes parameter but deals with applications that handle protocols rather than file types. The capture process generates initial values that you cannot add to. You can remove entries for browsers or other applications.
- ExcludePattern – excludes files or directories during the application build process.
- Icon – specifies the icon file to associate with the generated executable file.
- OutDir – specifies the directory that stores the `build.bat` output. Do not modify.
- RetainAllIcons – keeps all of the original icons of the executable file listed in the Source parameter in the application.
- AccessDeniedMsg – contains an error message to display to users who do not have permission to run a package.
- AddPageExecutePermission – supports applications that do not work in a Data Execution Prevention (DEP) environment.
- PermittedGroups – restricts a package to a specific set of Active Directory users. You can specify group names or SID strings,
- AccessDeniedMsg – to instruct the user.
- UACRequestedPrivilegesLevel – specifies privileges for programs requiring User Account Control (UAC) information.
- UACRequestedPrivilegesUIAccess – specifies user interface access on Windows Vista or later.
- ExternalCOMObjects – determines whether Windows creates and runs COM objects in the physical environment rather than the virtual environment to facilitate application compatibility with ThinApp.
- ExternalDLLs – can force Windows to load specific DLL files from the virtual file system.
- ForcedVirtualLoadPaths – instructs ThinApp to load DLL files as virtual DLL files even if the files reside outside the package.
- IsolatedMemoryObjects – lists the shared memory objects to isolate from other applications or from system objects.
- IsolatedSynchronizationObjects – lists the synchronization objects to isolate from other applications.
- NotificationDLLs – makes calls to third-party DLL files to provide notification of events, such as application startup or shutdown.
- NotificationDLLSignature – works with the NotificationDLLs parameter and verifies that a specified DLL file has a signature.
- ObjectTypes – specifies a list of virtual COM object types that are visible to other applications in the physical environment.
- SandboxCOMObjects – indicates whether applications in the physical environment can access COM objects that the virtual application registers at runtime.
- VirtualizeExternalOutOfProcessCOM – controls whether out-of-process COM objects can run in the virtual environment.
- CachePath – sets the deployment system path to a cache directory for font files and stub executable files.
- UpgradePath – specifies the location of information and files for Application Sync and integer updates.
- VirtualDrives – specifies additional drive letters that are available to the application at runtime. Does not override isolation mode settings.
- AllowExternalKernelModeServices – controls whether applications can create and run native kernel driver services.
- AllowExternalProcessModifications – determines whether captured applications can write to a native process. Some virtualized applications require a method to interact with native applications.
- AllowUnsupportedExternalChildProcesses – specifies whether to run 64 bit child processes in the physical environment.
- AutoShutdownServices – controls whether to shut down virtual services when the last non-service process exits.
- AutoStartServices – controls whether to start the virtual services when the first virtual application starts.
- ChildProcessEnvironmentDefault – determines whether ThinApp runs all child processes in the virtual environment.
- ChildProcessEnvironmentExceptions – notes exceptions to the
- ChildProcessEnvironmentDefault – when you want to specify child processes.
- BlockSize – controls the size of compression blocks only when ThinApp compresses files for a build. A larger block size can achieve higher compression. Larger block sizes might slow the performance.
- CompressionType – can compress all files in a package except for Portable Executable files. Compression has some performance consequences, does not affect MSI files
- MSICompressionType – determines whether to compress MSI files for package distribution. Compression improves performance when opening MSI files and using the ThinApp SDK.
- OptimizeFor – controls whether to compress executable files or to reduce memory consumption and page file usage on the hard drive to improve startup performance.
- DisableTracing – prevents .trace file generation when you run Log Monitor for security and resource purposes.
- LogPath – sets the location to store .trace files during logging activity.
- CapturedUsingVersion – displays the version of ThinApp for the capture process and determines the file system macros that ThinApp must expand. Do not modify.
- StripVersionInfo – determines whether to remove all version information from the source executable file when ThinApp builds the application.
- Version.XXXX – overrides application version strings or adds new version strings in the *Version* tab of Windows properties.
- AnsiCodePage – displays a numerical value that represents the language of the operating system on which you capture the application.

- LocaleIdentifier – displays a numeric ID for the locale that affects layout and formatting.
- LocaleName – displays the name of the locale when you capture an application on Microsoft Vista.
- CommandLine – specifies the command-line arguments that start a shortcut executable file.
- Disabled – determines whether the application build target is just a placeholder and prevents ThinApp from generating the executable file in the /bin directory.
- ReadOnlyData – specifies the name of the read-only virtual registry file created during the application build and designates the primary data container for an application. Do not modify.
- ReserveExtraAddressSpace – indicates the amount of extra address space to reserve for the captured executable file.
- Shortcut – points a shortcut executable file to the primary data container that contains the virtual file system and virtual registry. Do not modify.
- Shortcuts – lists the locations where the `thinreg.exe` utility creates a shortcut to a virtual application.
- Source – specifies the executable file that ThinApp loads when you use a shortcut executable file. The parameter provides the path to the executable file in the virtual or physical file system. Do not modify.
- WorkingDirectory – determines the first location in which an application looks for files and places files.
- RequiredAppLinks – specifies a list of required packages to import to the base package at runtime.
- OptionalAppLinks – is similar to the RequireAppLinks parameter but ignores errors and starts the main application even when an import operation fails.
- AppSyncClearSandboxOnUpdate – determines whether to clear the sandbox after an update.
- AppSyncExpireMessage – sets the message that appears when the connection to the Web server fails after the expiration period ends and a virtual application starts.
- AppSyncExpirePeriod – sets the expiration of the package in minutes (m), hours (h), or days (d). If ThinApp cannot reach the Web server to check for updates, the package continues to work until the expiration period ends and the user closes it. Even after the expiration period ends, ThinApp tries to reach the Web server at each subsequent startup attempt.
- AppSyncURL – sets the Web server URL or fileshare location that stores the updated version of an application. ThinApp checks this location and downloads the updated package. HTTP, HTTPS and File protocols (UNC).
- AppSyncUpdateFrequency – specifies how often ThinApp checks the Web server for application updates.
- AppSyncUpdatedMessage – sets the message that appears when an updated package first starts.
- AppSyncWarningFrequency – specifies how often a warning appears before the package expires.
- AppSyncWarningMessage – sets the message that appears when the warning period starts
- AppSyncWarningPeriod – sets the start of the warning period before a package expires.
- MSIArpProductIcon – specifies the icons to represent the application in the Windows *Add or Remove Programs* dialog box. Do not modify.
- MSIDefaultInstallAllUsers – sets the installation mode of the MSI database.
- MSIFilename – triggers the generation of an MSI database and specifies its filename.
- MSIInstallDirectory – specifies the relative path of the MSI installation directory. The path is relative to *%ProgramFilesDir%* for installations on each machine and relative to *%AppData%* for installations for each user.
- MSIManufacturer – specifies the manufacturer or packaging company of the MSI database and displays the value in the Windows *Add or Remove Programs* dialog box.
- MSIProductCode – specifies a product code for the MSI database. Do not modify.
- MSIProductVersion – specifies a product version number for the MSI database to facilitate version control.
- MSIRequireElevatedPrivileges – applies to Windows Vista and specifies elevated privilege requirements for the MSI database.
- MSIUpgradeCode – specifies a code for the MSI database that facilitates updates. Do not modify unless the new value is a valid GUID.
- MSIUseCabs – determines the use of .cab files that can affect application performance.
- InventoryName – is a string that inventory tracking utilities use for package identification.
- RemoveSandboxOnExit – deletes the sandbox and resets the application when the last child process exits.
- SandboxName – specifies the name of the directory that stores the sandbox.
- SandboxNetworkDrives – determines whether ThinApp directs write operations to a network drive or to the sandbox, regardless of isolation mode settings.
- SandboxPath – determines the path to the sandbox. Can be relative or absolute, can include folder macros or environment variables, and can exist on a network drive.
- SandboxRemovableDisk – determines whether the application can write removable disk changes to the disks or to the sandbox.

- Application Sync can query a Web server to determine if an updated version of the package is available. If an update is available, ThinApp downloads the differences between the existing package and the new package and constructs an updated version of the package.
- The Application Sync utility downloads updates in the background. You can continue to use an old version of the application. If the user quits the application before the download is complete, the download resumes when the virtual application starts again. When the download is finished, ThinApp activates the new version the next time the application starts.
- The sandbox is the directory where all changes that the captured application makes are stored. The next time you start the application; those changes are incorporated from the sandbox. When you delete the sandbox directory, the application reverts to its captured state.
- ThinApp uses the first sandbox it detects. The search order is:

    - *<sandbox_name>*
    - *<sandbox_path>*
    - *<exe_directory>*
    - *<computer_name>*
    - *%AppData%*
- Only one computer at a time can use a shared sandbox. If a computer is already using a sandbox, ThinApp creates a new sandbox to allow you to continue working until the previous copy of the sandbox closes.
- You can use the SandboxPath to store the sandbox on a mapped drive or to set a portable device location for the sandbox.
- The sandbox contains the following registry files:
    - Registry.rw.tvr – Contains all registry modifications that the application makes.
    - Registry.rw.lck – Prevents other computers from simultaneously using a registry located on a network share.
    - Registry.tvr.backup – Contains a backup of the .tvr file
    - Also contains directories that include *%AppData%*, *%ProgramFilesDir%,* and *%SystemRoot%*.
- ThinApp stores file system information in the virtual registry. This ability increases the ThinApp runtime performance.
- The `vregtool` utility to view modified virtual registry changes. C:\Program Files\VMware\VMware ThinApp.
- The `snapshot.exe` utility creates a snapshot of a computer file system and registry and creates a ThinApp project from two previously captured snapshots. The Setup Capture wizard starts it and copies of the following data:
    - File information for all local drives
    - HKEY_LOCAL_MACHINE and HKEY_USERS registry trees
- The snapshot.ini configuration file specifies what directories and subkeys to exclude.
- You can use the `snapshot.exe` utility to create snapshot files of machine states, create the template file for the Package.ini file, create a ThinApp project and display the contents of a snapshot file.
- ThinApp stores the differences between snapshots during the setup capture process in a virtual file system and virtual registry. The virtual file system uses folder macros to represent Windows shell folder locations.
- ThinApp generates the following virtual file system formats:
    - Build – the setup capture process generates this format from files found directly on the physical file system.
    - Embedded – the `build.bat` file triggers a build process that embeds a read-only file system in executable files. It provides block-based streaming to client computers.
    - Sandbox – running the captured application generates the read-write directory structure that holds file data that the application modifies.
- The use of macros allows shared application profile information to instantly migrate to different operating systems.
- Scripts modify the behavior of virtual applications dynamically. You can create custom code before starting an application packaged with ThinApp or after an application exits. You can use scripts to authenticate users and load configuration files from a physical to virtual environment.
- Callback functions run code during specific events. If applications create child processes, use callback functions to run code only in the main parent process.
- API functions run ThinApp functions and interact with the ThinApp runtime. API functions can authenticate users and prevent the start of applications for unauthorized users.
- Adding scripts to your application involves creating an ANSI text file with the .vbs file extension in the root application project directory.
- Callback functions:
    - OnFirstSandboxOwner – only when an application first locks the sandbox.
    - OnFirstParentStart – before running a ThinApp executable file regardless of whether the sandbox is simultaneously owned by another captured executable file.
    - OnFirstParentExit – when the first parent process exits.
    - OnLastProcessExit – when the last process owning the sandbox exits.
- API functions:
    - AddForcedVirtualLoadPath(Path) – instructs ThinApp to load all DLLs from the specified path as virtual DLLs even if they are not located in the package.
    - ExitProcessExitCode – quits the current process and sets the specified error code.
    - ExpandPath(InputPath) – converts a path from macro format to system format.
    - ExecuteExternalProcess(CommandLine) – runs a command outside of the virtual environment, to make physical system changes.
    - ExecuteVirtualProcess(CommandLine) – runs a command inside of the virtual environment, to make changes to the virtual environment.
    - GetBuildOption(OptionName) – returns the value of a setting specified in the [BuildOptions] section of the Package.ini file used for capturing applications.
    - GetFileVersionValue(Filename, Value) – returns version information value
    - GetCommandLine – accesses the command-line parameters passed to the running program.v
    - GetCurrentProcessName – accesses the full virtual path name of the current process.
    - GetOSVersion() – returns information about the current version of Windows.
    - GetEnvironmentVariable(Name) – returns the environment variable associated with the Name variable.

  - RemoveSandboxOnExit(YesNo) – set toggles that determine whether to delete the sandbox when the last child process exits.
  - SetEnvironmentVariable(Name, Value) – set the value of an environment variable.
  - Setfile systemIsolation(Directory, IsolationMode) – sets the isolation mode of a directory.
  - SetRegistryIsolation(RegistryKey, IsolationMode) – sets the isolation mode of a registry key.
  - WaitForProcess(ProcessID, TimeOutInMilliSeconds) – waits until the process ID is finished running.
- Log Monitor captures detailed chronological activity for executable files that the captured application starts.

ThinApp Virtual Registry

- The virtual registry uses the following formats:
  - Build - The setup capture process generates this format, capturing each registry hive in its own Unicode file with a .txt extension.
  - Embedded - The build process converts the build format data into the embedded format data. ThinApp stores registry data inside of the primary data container.
  - Sandbox (.tvr extension) - As the application performs registry write operations, ThinApp stores the differences from the embedded format in sandbox format. When you start the application, ThinApp overlays the sandbox format data on the embedded format data to restore application settings from the last shutdown.
- The `vregtool.exe` utility compiles the virtual registry during the build process. You can use this utility to manipulate the .tvr files of the virtual registry.
- You can use `vregtool.exe` to perform operations such as exporting registry data to regedit format, listing diagnostic information about a .tvr file, and deleting a registry subkey.
- Imported registry files can be regedit 4.0 (ansi text) or 5.0 (unicode text) format.
- Exported registry files are regedit 5.0 format (unicode text).