ThinApp 4.5 Parameters

Parameters apply to Package.ini or ##Attributes.ini. Most commonly changed parameters are in bold.
AccessDeniedMsg error message to display to users who do not have permission to run a package.
AddPageExecutePermission for apps not working in Data Execution Prevention (DEP) environment.
AllowExternalKernelModeServices whether apps can create and run native kernel driver services.
AllowExternalProcessModifications whether captured apps can write to a native process.
AllowUnsupportedExternalChildProcesses whether 64bit child processes run in physical environment.
AnsiCodePage displays numerical value representing language of the OS on which you capture the app.
AppSyncClearSandboxOnUpdate whether to clear the sandbox after an update.
AppSyncExpireMessage message if app starts & webserver connection fails after expiration period ends
AppSyncExpirePeriod sets package expiration in minutes (m), hours (h), or days (d).
AppSyncUpdatedMessage message that appears when updated package first starts.
AppSyncUpdateFrequency how often ThinApp checks web server for application updates.
AppSyncURL web server URL or fileshare that stores updated version of app. HTTP, HTTPS or UNC.
AppSyncWarningFrequency how often a warning appears before the package expires.
AppSyncWarningMessage message that appears when the warning period starts.
AppSyncWarningPeriod sets the start of the warning period before a package expires.
AutoShutdownServices whether to shut down virtual services when last non-service process exits.
AutoStartServices whether to start virtual services when first virtual application starts.
BlockSize size of blocks when files compressed for build. Larger = higher compression, but slower.
CachePath deployment system path to a cache directory for font files and stub executable files.
CapturedUsingVersion ThinApp version for capture, decides macros that must expand. **Do not modify**.
ChildProcessEnvironmentDefault whether to run all child processes in the virtual environment.
ChildProcessEnvironmentExceptions exceptions to ChildProcessEnvironmentDefault.
CommandLine command-line arguments that start a shortcut executable file.
CompressionType compress files in package except Portable Executable files, doesn't affect MSI files.
DirectoryIsolationMode read/write access to physical file system - *Merged*, *WriteCopy* or *Full*.
Disabled whether app build target is just placeholder, preventing executable file being generated in /bin.
DisableTracing prevents .trace file generation when you run Log Monitor.
ExcludePattern excludes files or directories during application build process.
ExternalCOMObjects whether Windows creates/runs COM objects in physical environment.
ExternalDLLs can force Windows to load specific DLL files from the virtual file system.
FileTypes file extensions `thinreg.exe` associates with executable, can remove but not add value.
ForcedVirtualLoadPaths loads DLL files as virtual DLL files even if files reside outside the package.
Icon the icon file to associate with the generated executable file.
InventoryName string that inventory tracking utilities use for package identification.
IsolatedMemoryObjects shared memory objects to isolate from other apps or from system objects.
IsolatedSynchronizationObjects synchronization objects to isolate from other apps.
LocaleIdentifier numeric ID for locale that affects layout and formatting.
LocaleName name of locale when capturing application on Microsoft Vista.
LogPath location to store .trace files during logging activity.
MSIArpProductIcon icons to represent app in Windows Add/Remove Programs. **Do not modify**.
MSICompressionType whether to compress MSI files for package distribution, improves performance.
MSIDefaultInstallAllUsers installation mode of the MSI database.
MSIFilename triggers the generation of MSI database and specifies its filename.
MSIInstallDirectory relative path of MSI installation directory, relative to %ProgramFilesDir% for machine installs and %AppData% for user installs.
MSIManufacturer manufacturer/packaging company of MSI database, value in Add/Remove Programs.
MSIProductCode product code for the MSI database, makes upgrading easier. **Do not modify**.
MSIProductVersion product version number for MSI database to facilitate version control.
MSIRequireElevatedPrivileges (Vista) specifies elevated privilege requirements for MSI database.
MSIUpgradeCode code for MSI database, allowing updates. Don't modify unless value is valid GUID.
MSIUseCabs use of .cab files that can affect application performance.
NetRelaunch restart app from local disk when running from network share or removable disk.
NotificationDLLs makes calls to third-party DLLs to provide notification of events.
NotificationDLLSignature works with NotificationDLLs & verifies specified DLL has signature.
ObjectTypes virtual COM object types that are visible to other applications in physical environment.
OptimizeFor whether to compress executables or reduce memory consumption and pagefile usage.
OptionalAppLinks similar to RequiredAppLinks but ignores errors, starts app if import operation fails.
OutDir directory that stores the `build.bat` output. **Do not modify**.
PermittedGroups restricts package to set of AD users, can specify group names or SID strings.
Protocols like FileTypes but apps that handle protocols not file types, can remove but not add values.
QualityReportingEnabled whether VMware can collect anonymous data on package.
ReadOnlyData read-only virtual registry file created by build, primary data container. **Do not modify**.
RegistryIsolationMode isolation mode for registry keys in the package.
RemoveSandboxOnExit deletes the sandbox and resets application when the last child process exits.
RequiredAppLinks required external packages to import to base package at runtime.
ReserveExtraAddressSpace amount of extra address space to reserve for captured executable file.
RetainAllIcons keeps original icons of executable file listed in Source within application.
RuntimeEULA controls EULA display, addressing legacy EULA requirements. **Do not modify**.
SandboxCOMObjects if apps in physical environment can access COM objects that virtual app registers
SandboxName name of directory that stores the sandbox. Keep the same name for incremental updates.
SandboxNetworkDrives to write to network drive or sandbox, regardless of isolation mode settings.
SandboxPath path to sandbox, relative or absolute, can include folder macros or environment variables, can exist on network drive.
SandboxRemovableDisk whether app can write removable disk changes to disks or to sandbox.
Shortcut pointer to primary data container containing virtual file system & registry. **Do not modify**.
Shortcuts locations where the `thinreg.exe` utility creates a shortcut to a virtual application.
Source executable that loads when using shortcut executable file. **Do not modify**.
StripVersionInfo whether to remove version information from source executable during build.
UACRequestedPrivilegesLevel privileges for programs requiring User Account Control information.
UACRequestedPrivilegesUIAccess user interface access on Vista or later.
UpgradePath location of information and files for Application Sync and integer updates.
Version.XXXX overrides application version strings or adds new version strings in Version tab.
VirtualComputerName whether to virtualize hostname to avoid naming conflicts.
VirtualDrives extra drive letters available to app at runtime, doesn't override isolation mode settings.
VirtualizeExternalOutOfProcessCOM if out-of-process COM objects can run in virtual environment.
WorkingDirectory first location application looks for files and places files.
Wow64 simulates a 32bit environment.

ThinApp 4.5 Functions

Callback functions:
OnFirstParentExit when the first parent process exits.
OnFirstParentStart before running executable, even if sandbox is owned by other captured executable.
OnFirstSandboxOwner only when an application first locks the sandbox.
OnLastProcessExit when the last process owning the sandbox exits.
API functions:
AddForcedVirtualLoadPath(Path) loads all DLLs from path as virtual DLLs even if not 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 command outside virtual environment.
ExecuteVirtualProcess(CommandLine) runs command inside virtual environment.
GetBuildOption(OptionName) value set in [BuildOptions] section of Package.ini for capturing apps.
GetFileVersionValue(Filename, Value) returns version information value.
GetCommandLine command-line parameters passed to running program.
GetCurrentProcessName full virtual path name of current process.
GetOSVersion() information about current version of Windows.
GetEnvironmentVariable(Name) environment variable associated with Name variable.
RemoveSandboxOnExit(YesNo) whether to delete sandbox when last child process exits.
SetEnvironmentVariable(Name, Value) value of environment variable.
Setfile systemIsolation(Directory, IsolationMode) isolation mode of directory.
SetRegistryIsolation(RegistryKey, IsolationMode) isolation mode of registry key.
WaitForProcess(ProcessID, TimeOutInMilliSeconds) waits until process ID is finished running.

vReference.com
Virtual Desktops

by **Forbes Guthrie**

Version 1.0
released 6 May 2010

View 4.0 Desktops

Maximums: Hosts per custer = 8 (HA limit for dense hosts & View Composer limit)
RDP connections: 2000 per connection server, 5000 for 5 servers Tunneled RDP: 2000 for 3 servers
PCoIP connections: 2000 per connection server 100 PCs per server 200 TS sessions per server

Requirements: ESX(i)4 U1, VI3.5 U3/U4 (U5 from 4.0.1), VI3.0.2, vCenter 4 (& VC2.5 U6 from 4.0.1). Connection Server 32bit Win 2003 SP2 Std/Ent (cannot have TS role), 2GB RAM (3GB recommended for > 50 Desktops). For SecureID - RSA Auth Mgr 6.1 or 7.1. Agent 32bit Win XP Pro SP2/SP3, Vista Ent/Bus/Ult SP1/SP2, 2003 Ent SP2/SP3, Win 7 (tech preview)

FW Port	Source	Destination	Protocol	Description
80 (in)	Connection/Security	Client/Portal	TCP	Web Access if SSL off
88 & 445	Connection Svr	AD DC	TCP/UDP	AD authentication
443 (in)	Connection Svr	Clients	TCP	Client & Portal access
443 (in)	Security Svr	Clients	TCP	Client & Portal access
443	Connection Svr	vCenter	TCP	vCenter commands
1024-65535	Connection Svr	Connection/Agent	TCP	Embedded LDAP
3268-3269	Connection Svr	AD DC	TCP	Global Catalog queries
3389	Security Svr	Agent (desktop)	TCP	Tunnelled RDP
4001 (out&in)	Connection Svr	Security/Agent	TCP	Java Messenger Service
4100	Connection Svr	Connection Svr	TCP	Java Messenger Service
8009 (out&in)	Connection Svr	Security Srv\\vr	TCP	Apache Jserv Protocol

Possible extras: 902 (offline desktop), 9427 (MMR), 32111 (USB redirection), 42966 (HP RGS), 50002 (PCoIP)

Licensing: View Enterprise basic package View Premier adds Composer, ThinApp, Offline Desktop.
View Administrator (runs on Connection Server): web site to configure Connection Server, deploy/manage desktops, control user authentication & troubleshoot.
Connection Server: client broker responsible for AD Authentication, authorization, directs to desktop, manages sessions, secures connections, single sign on, policies. Configuration data stored in embedded LDAP. **Types**: standard, replica (provides high-availability & load balancing) & security servers. Must be an AD domain member but cannot be a DC. Users authenticated against domain & any trusted domains (domain filtering with `vdmadmin`). External URL for standard/replica servers set in View Administrator, security server set in config.properties file.
RDP data is tunneled through HTTPS & encrypted using SSL. Direct RDP connections for internal LANs only. Clients using PCoIP or HP RGS do not use tunnel.
Smart Card authentication needs root certificate added to server truststore file & modified settings. Certificates only required for standard, replica or security servers that receive direct client connections.
Security Server: Connection Server running subset of functions, not required to be AD member, does not contain LDAP component, needs a hardware or software load balancing solution if more than one.
Agent: software installed in desktop so it can be managed by Connection Server. Configure subnet that Agent uses if desktops has multiple NICs: HKLM\Software\VMware Inc.\VMware VDM\Node Manager\subnet = n.n.n.n/m (REG_SZ) Vista - set Windows power settings to ensure VM is always on. Configuration file: C:\Program Files\VMware\VMware View\Server\sslgateway\conf\config.properties Cannot configure Restricted Entitlements, must be configured on Connection Server it is paired with.

Commands: `C:\Program Files\VMware\VMware View\Server\bin\vdmadmin.exe –D` assign default desktop to user, `-U` user information, `-L` change user assignments, `-R` first desktop user, `-S` permanently remove replica server, `-O` orphaned desktops, `-N` domain filters, `-A` override IP
Export configuration: `vdmexport.exe > config.ldf`
Import: `C:\windows\adam\LDIFDE -i -f config.ldf -s 127.0.0.1 –z`
Support script: `C:\Program Files\VMware\VMware View\component\DCT\cscript vdm-support.vbs` Log levels: normal, debug (default) or full. Creates vdm-sdct folder on desktop.
LDAP backup files: C:\Documents & Settings\All Users\Application Data\VMware\VDM\backups
Connection/Security logs: C:/Documents & Settings/All Users/Application Data/VMware/VDM/logs

Scaling: depends on workers & applications. This section contains scaling advice (not limits).
Typical types of workers: Task workers repetitive tasks, small set of applications, usually at stationary computer, may simultaneously log in. Knowledge workers internet, email & complex documents/ presentations/spreadsheets. Power users - application development & graphics intensive applications.
RAM: RAM too low – bad for storage I/O (memory swapping). RAM too high – bad for storage capacity (guest pagefile & swap/suspend files). Set RAM reservation to nonzero value, desktops are sensitive to response times. Guarantees idle desktops never completely swapped to disk. Initial recommendation: XP - 1024MB, Vista -1536MB. PCoIP uses extra 64MB to 512MB RAM for displays depending on number of screens & resolution.
Storage: IOPS effects how quickly desktops recover from server failure. I/O storms triggered by guest software (updates or antivirus) or user behavior (simultaneous log ins). Windows pagefile default 150% of guest RAM, causes linked-clones & thin-provisioned disks to grow. ESX swap file (.vswp) swap file equal to unreserved guest RAM. ESX suspend file (.vmss) equal to guest RAM, created if pool logoff policy is suspend when user logs off. Log files approx 100MB per VM. vmdk file OS, applications, updates (& future applications). Local user data & user installed applications if not on file shares. Add 15% to estimate to ensure users do not run out of disk. XP use LSI Logic SCSI adapter (not default
CPU: 10 to 25% extra required for virtualization overhead & peak usage periods. Use performance tools during pilot (e.g. Perfmon), to understand average & peak CPU for groups of workers.
Host: 8 VMs per core, 64 VMs per LUN, 32 VMs per VMNIC, 1Gbps per 100 VMs.
vCenter:associate pool with vCenter resource pool, best performance with single cluster per datacenter

View 4.0 Desktops (cont)

Pools: Persistent users assigned own desktop on first connection. Non-persistent desktop returned to pool after each use and optionally re-created. Desktop sources: VMs, physical PCs & TS servers. Desktop delivery models: Individual Desktop a pre-existing desktop. Entitled to many users or groups, only one active user at a time, not provisioned automatically. Manual Pool pool of desktops. Multiple users to multiple desktop mapping, only one active user per desktop at a time, not provisioned automatically. Both persistent & non-persistent access modes. Automated Pool pool of dynamically generated desktops from vCenter template. Multiple users to multiple desktop mapping, only one active user per desktop at a time, provisioned automatically. Both persistent & non-persistent access modes. Terminal Server Pool pool of TS desktops served by one or more TS servers. Least session count based load balancing. Uses roaming profile to propagate user settings, only RDP supported.
Unique ID: used to identify desktop pool & name that clients see when logging in.
Linked clones: rapidly deploy desktops from single Parent VM for automated desktop pools.

Restricted Entitlements: restrict desktop or pool access based on tags assigned to Connection Server (can also restrict on user-authentication method). Tag matching rules:

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

Cannot modify/remove tag from Connection Server if still assigned to desktop/pool or other Server has same tag. Restricted Entitlements take precedence over other desktop entitlements. Disable Connection Server to prevent clients from logging in, currently logged in users not affected.

Desktop policies: assigned via View Administrator or with GPOs to users or desktops.
Power Policy controls desktops only when not in use: until first log-in or once user disconnects/logs off
Power Policy definitions: • Do nothing (VM remains on) • Always on • Suspend VM • Power off VM
Power Policy applies to: • Individual Desktop (vCenter VM) • Persistent Automated/Manual Pool - only applies to unassigned desktops • Non-Persistent Automated/Manual Pool - *Power Off* session is discarded after disconnection. *Suspend* orphan could be created after disconnection (user may never return). Ensure *Automatic logoff after disconnect* set to *Immediately* • Physical & TS desktop - N/A
View Administrator policies control client components at global, pool or desktop user level. User policy inherits from Pool policy which inherits from Global policy. Pool policy can override Global policy (if more restrictive). User policies override Global & Pool policies. System policies override User policies.
GPO templates on Connection Server: C:\Program Files\VMware\VMware View\Server\Extras\GroupPolicyFiles\
• vdm_agent.adm • vdm_client.adm • vdm_server.adm • vdm_common.adm
User GPOs override Computer policies regardless of desktop they connect to. Agent GPOs *CommandsToRunOnConnect* & *CommandsToRunOnReconnect* can pass information to custom scripts.

Unified Access: provides users same interface to access desktops. Each type need different settings.

	Individual		Manual				TS pool
	managed	unmanaged	managed		unmanaged		
			persist	non-persist	persist	non-persist	
Desktop state	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VM power policy	Yes		Yes	Yes			
Auto logoff after disconnect	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Users can reset	Yes		Yes	Yes			
Multiple sessions per user				Yes		Yes	

Links: http://kb.vmware.com/kb/1015858 - Best practises for upgrading to View 4.0
http://kb.vmware.com/kb/1003658 - Troubleshooting pool issues

View Composer 2.0

Maximums: Desktops per pool (create/provision/recompose) = 512

vCenter Requirement: Composer installed on 32bit vCenter Server (vCenter can be on 64bit OS)
Hosts: ESX(i)4 U1 or ESX(i)3.5 U3/U4. **Database**: MS SQL 2000 SP4, 2005 Express, 2005 SP1/SP2, 64bit SP2, Oracle 9i R2 Std/Ent, 10g Std/Ent R1/R2, 10g Ent R2 64bit, 11g Std/Ent

Linked clone pools use snapshots (replicas) not templates as base image. Clones linked to replica not Parent VM. Parent VM can be updated/replaced without affecting linked clones. Replicas are identified by replica- prefix followed by unique ID. Replica & their linked clones placed on same datastore. Customization specifications expedite pool deployments. Linked-clone pool can be mixed ESX(i) 3.5 & 4 cluster until *vSphere mode* set. Composer cannot deploy Vista Ultimate or XP Pro SP1 desktops.
Recompose: change Parent VM & anchor linked clones to new snapshot. **Refresh**: returns linked clone to Parent VM state, reduces size of linked clones; on demand, timed event or when a specified size.
Rebalance: moves replica & linked clones to another LUN, but requires a refresh against Parent VM. Recomposition with ESX3.5 uses additional protected linked clone "Source VM", with replica in VMwareViewComposerReplicaFolder folder. ESX4 hosts do not need Source VM.
Persistent desktop pool: separate user disk with profile & application data for each desktop; can be refreshed after user logs off. **Non-persistent**: desktop can be deleted after use; user data is transient so not protected if recomposed/refreshed/rebalanced; use pool deployment wizard to update Parent VM.
QuickPrep: linked clone deployment tool to personalize desktop with new name, join domain & mount user profile volume. Restarted twice & computer account created. Also occurs after refresh. Can run customization script after desktop creation, refresh, recomposition, or before power off.
Thin provisioning: used for OS & user data disks. **Storage overcommit level**: storage required if each clone used all their space: • None - not overcommitted • Conservative - 4 times datastore size (default) • Moderate - 7 times • Aggressive - 15 times. Applies to delta disks, not user disks or non-linked clones.

Upgrade/restore DB: `C:\Program Files\VMware\VMware View Composer\sviconfig.exe`
Support script: `C:\Program Files\VMware\VMware View Composer\cscript.wsf svi-support.wsf`

View 4.0 Clients

Maximums: Simultaneous Offline Desktop checkouts: ESX = 2 ESXi = 5

Requirements: • Client x86 CPU with SSE2. Cannot install Client & Agent on same PC. • Offline Desktop (experimental) - VI3.5 U3/U4, not ESX4. Cannot install alongside VMware ACE, Player, Server or Workstation. • Portal: Windows - IE6 SP2/IE7. Apple Mac - MS RDC 2.0 Client, Java JRE 1.5/1.6, Safari. Linux - rdesktop, Java JRE 1.5/1.6, Firefox). • Thin clients use special client SW.

Client support (32bit only):	**RDP**	**PCoIP**	**HP RGS**	**USB access**	**MMR**	**Virt Print**	**Offline desktop**
Win 2000	Yes					Yes	
Win XP Home/Pro SP2/SP3	Yes	Yes	Yes (Pro)	Yes	Yes	Yes	Yes (Pro)
Vista Business SP1/SP2	Yes	SP2	Yes	Yes			
Vista Ultimate SP1/SP2	Yes	SP2	SP2	Yes	SP1	SP1	
Vista Enterprise SP2	Yes	Yes	Yes	Yes			

Portal only - RHEL 5.1, SLED 10, Ubuntu 8.04, Mac OS X 10.4, Mac OS X 10.5. Win 2000, XP Home, Vista Home Basic/Premium do not support *Log in as current user* feature.

FW Port	Source	Destination	Protocol	Description
80	Client/Portal	Connection svr	TCP	When SSL disabled
80	Client/Portal	Security server	TCP	Web Access proxy
443	Client/Portal	Connection svr	TCP	When SSL enabled
443	Client/Portal	Security server	TCP	When SSL enabled
443	Client	View manager	TCP	When SSL enabled
902	Client (offline desktop)	ESX/ESXi host	TCP	When offline VM synced
3389	Client	Agent (desktop)	TCP	Direct RDP connections
9427	Client	Agent (desktop)	TCP	MMR
42966	Client	ESX/ESXi host	TCP	HP RGS sender app
50002	Client	Agent (desktop)	TCP/UDP	PCoIP (AES 128bit)
50002	Agent (desktop)	Client	TCP/UDP	PCoIP (AES 128bit)

Client: local software connecting to Connection Server. Policies controll some features. Authorization via AD, UPN, Smart card PIN, RSA SecurID. GPO template for Client parameters: vdm_client.adm.
Portal: browser based client, no Virtual Printing or USB redirection. Win PCs use Client or Portal. Apple Macs & Linux only portal.

Smart card: authentication via Client; not Administrator, Portal or Offline Desktop. Servers receiving direct client connections (standard/replica/security) must have users Root Certificates added to their truststore. Replaces passwords only, users must authenticate using SecurID if enabled.
ThinPrint - printer data compression: No images, Extreme, Maximum, Optimal & Normal.
Adobe Flash bandwidth reduction: requires reconnection to get new settings, only IE & Flash 9/10. Render-quality: Do not control, Low (default), Medium, High. Frame throttling (audio constant): Disabled, Conservative, Moderate, Aggressive. Users override settings when cursor is over content.
Virtual Printing: no drivers installed for local/network printers (not USB printers).
Wyse MMR (multimedia redirection): enables full-fidelity playback for streamed multimedia files.

PCoIP: VMs, Teradici clients or PCs with card. No Virtual Printing (≥4.0.1 can), Smart Cards or Portal

Protocol feature	**PCoIP**	**RDP**	**HP RGS**
Monitors	up to 4 at 1920x1200	Span mode	Span mode
Copy/paste between local/view desktop	Yes	Yes	
Adobe Flash bandwidth reduction	Yes	Yes	Yes
Colour depth	32 bit	32 bit	
Encryption	128 bit, AES	128 bit	
VPN support	Yes		
Can connect via security servers		Yes	

Offline Desktop (experimental): downloads local copy of VM with thin provisioned disks & locks online VM. Disables vCenter ops - power, snapshots, editing settings. Offline VM encrypted & lifetime controlled by policy. Offline Desktop tunneled (routed via Connection Server) or non! tunneled. Smart cards not supported. NAT used, MAC address remains same. Cannot download desktop if other user is logged in. Tries to reconnect every 5 minutes. **Reconnection options**: • Backed up (online VM updated, offline VM remains checked out, online lock remains) • Rolled back (offline VM discarded & online lock released) • Checked in (offline VM uploaded & online lock released).
Desktops available for offline: • Individual desktop as non-persistent VM managed by vCenter • Manual Pool desktop as persistent VM • Automatic Pool desktop as persistent non-linked clone.

Links: http://kb.vmware.com/kb/1003642 - Troubleshooting end user connection issues
http://kb.vmware.com/kb/1003706 - Troubleshooting USB issues

ThinApp 4.5

Platform requirements: 32bit Win (NT, 2000, XP, XPE, 2003, Vista, 2008, 7), 64bit Win (XP, 2003, Vista, 2008, 2008 R2, 7), TS & Xenapp. Not supported: 16bit or non-x86 platforms.
Apps supported: 16bit apps on 32bit Win, 32bit apps on 32/64bit Win. Not supported: 64bit apps, 16bit apps on 64bit Win. Cannot capture apps requiring kernel mode device drivers, antivirus & personal firewalls, scanner & printer drivers and some VPN clients. Some apps with shell integration have reduced functions. COM and DCOM services are isolated to just other sandbox apps.

Install: Use clean computer • installers skip files that already exist • capture scans the file system and registry quickly. Install on the earliest version of Windows to support.
Key files: C:\Program Files\VMware\VMware ThinApp

`AppSync.exe`	Updates captured apps.
logging.dll	Generates .trace files.
`dll_dump.exe`	Lists all captured apps.
`log_monitor.exe`	Displays history and errors.
`relink.exe`	Updates packages to latest ThinApp version.
`sbmerge.exe`	Merges runtime changes in sandbox with the project.
`Setup Capture.exe`	Captures & configures apps through wizard.
`snapshot.exe`	Compares installation environment during capture. References snapshot.ini file.
snapshot.ini	Entries for virtual registry & virtual file system that are ignored during capture.
`template.msi`	Builds MSI files.
`thinreg.exe`	Registers apps, shortcuts, file type associations & Start menu items. MSI alternative.
`tlink.exe`	Links key modules during build.
`vftool.exe`	Compiles virtual file system during build.
`vregtool.exe`	Compiles virtual registry during build.

Capture: system scans, app config, package config, generation of virtual app. Setup Capture wizard - main method to capture apps & set initial parameters. snapshot.exe - capture from command line. Modify Package.ini to update package. Virtual apps are set to region/language on capture system.

ThinApp 4.5 (cont)

Entry points: native executables that act as shortcuts to virtual environment and start apps. One executable per entry point created during the build. Can use AD groups to authorize access to app.
Parameters: C:\Program Files\VMware\VMware ThinApp\Captures\Package.ini applies settings at app level, ##Attributes.ini in macros folder of project, applies settings at directory level. CompressionType, DirectoryIsolationMode & ExcludePattern in ##Attributes.ini override Package.ini settings.
Package.ini headings: [BuildOptions] - applies to all apps unless app specific entries override them, [<application>.exe], [FileList], [Compression], [Isolation]. The [FileList], [Compression] & [Isolation] parameters act as [BuildOptions] parameters but are grouped separately for backward compatibility.
Isolation mode: determines read/write access to native file system outside of virtual environment.
DirectoryIsolationMode controls default isolation mode. Does not affect: • writes to network drives (set via SandboxNetworkDrives) • saving to Desktop or My Documents, instead saves to physical system.
DirectoryIsolationMode: *Merged* & *WriteCopy* available during capture, *Full* used outside wizard.
Merged - apps can read & modify elements on physical file system outside virtual package. Advantage is documents appear on physical system not in sandbox. Disadvantage is it might clutter system image.
WriteCopy - writes are redirected to sandbox, for legacy or untrusted apps.
Full - blocks visibility to system elements outside virtual app package. Prevents conflict between apps on virtual & physical system. Only use *Full* as override in ##Attributes.ini files.
RegistryIsolationMode: not set during capture, but configured only in Package.ini. Default is *WriteCopy*. *Full* only set as an override. Exceptions set in registry key text files in the project directory.
Sandbox: directory storing captured application changes. Deleting sandbox reverts app to captured state
Local machine: store sandbox in user's profile. Network location: to back up sandbox & allow settings across multiple computers. Use absolute path. Portable device: keep sandbox on same device as app. Only one computer can access a shared sandbox. If already being used, a new sandbox is created until the previous copy closes. Use SandboxPath to store sandbox on mapped drive or portable device.
Sandbox files: Registry.rw.tvr – registry modifications, Registry.rw.lck – controls locking on network share, Registry.tvr.backup – backup of .tvr file, *%AppData%*, *%ProgramFilesDir%* & *%SystemRoot%*.
Primary data container: main virtual app file, includes ThinApp runtime & read only file system & registry. Specified in ReadOnlyData in Package.ini. If < 200MB it creates .exe, > 200MB creates .dat.
thinreg.exe: creates Start menu items, desktop shortcuts, file type associations, adds uninstall information to control panel & unregisters previously registered packages. After adding, modifying or removing executables, `thinreg.exe` reregisters the file type associations, shortcuts and icons.
`thinreg.exe` monitors PermittedGroups in Package.ini, utilising AD groups for authorisation.
Switches: `/a` (`/allusers`) Registers package for all users regardless of PermittedGroups setting.

`/q` (`/quiet`)	Prevents display of an error message.
`/u` (`/unregister`)	Unregisters package.
`/r` (`/reregister`)	Re-registers package.
`/k` (`/keep`)	Prevents removal of registration information even if no longer authorized.
`/noarp`	Prevents an entry in Add/Removes programs.
`/norelaunch`	Starts `thinreg.exe` without elevated privileges.

App streaming: contents from executable streams to client as block-based transfer. As app requests parts, ThinApp reads compressed parts over network. Data is decompressed directly to memory so process is fast. Make the package share read only. Recommended only for LAN environment.
App Sync: updates an entire app package, useful for major configuration updates. Updates are downloaded in background. Old version used until download complete, download pauses when app quits & resumes when app starts. When download finishes, new version activates next time app starts. Sandbox versions take precedence over App Sync versions. App Sync updates entry point executables. Primary data container name must be same for old & new versions. Cannot complete when app creates child processes. `AppSync.exe` forces an update on a client, users must have write access to packages. `sbmerge.exe` makes incremental updates by merging runtime changes in sandbox back into project.
App Link: keeps shared components or dependent apps in separate packages, reconnecting at runtime. Can link up to 250 packages, useful for large shared libraries, frameworks, plugins, hotfixes & service packs. App Link supports nested links. Sandbox changes from linked packages are not visible to base executable. App Link loads during app startup and merges registry entries and file system directories using isolation mode specified in layer being loaded. If it exists in the main package and a layer that is already merged, then most restrictive isolation mode is used. The order is *Full*, *WriteCopy* then *Merged*. Linked apps are imported in the order in RequiredAppLinks or OptionalAppLinks. Order of import determines package priority. Last package imported has priority. Linking two apps and specifying PermittedGroups, the user account must be member of AD group in Package.ini of both apps. Collision occurs if base app & dependent package contains file or registry entries at same location.
Versioning: App locks the executable when started. File locks created when multiple users access centralized apps, prevent upgrading packaged executables until all users exit app. Terminal Server environments can have multiple users executing different versions concurrently during transition period. User settings kept during upgrades if sandbox name is same in Package.ini. Change sandbox name to force a new sandbox during an app upgrade.
`relink.exe` updates packages to latest ThinApp, without rebuilding each package with `build.bat`
Snapshots: `snapshot.exe` snapshots file system & registry and creates ThinApp project. Started by Setup Capture wizard, copies: • file information for local drives • HKEY_LOCAL_MACHINE & HKEY_USERS registry trees. snapshot.ini specifies directories & subkeys to exclude.
Virtual file system formats: Build setup capture generated from physical file system. Embedded `build.bat` triggers build that embeds read-only file system in executables, providing block-based streaming to clients. Sandbox captured app generates the read-write sandbox directory.
Virtual registry formats: Build setup captures each registry hive in unicode .txt file. Embedded build process converts data into embedded format, storing registry data inside Primary Data Container. Sandbox (.tvr) - as app performs registry writes, differences from embedded are stored in sandbox. Some file system information is stored in virtual registry to improve runtime performance.
`vregtool.exe` compiles virtual registry during build process & can manipulate .tvr files. Imports registry files as regedit 4.0 (ansi text) or 5.0 (unicode text) format and exports as 5.0 format.
Macros: represent Windows shell folder locations, allows apps to migrate to different Win versions.
Scripts: modify behavior of apps dynamically. Create scripts to authenticate users & load configuration files from physical to virtual environment. Create .vbs ANSI file in root app project directory.
Callback functions: run code during specific events. If apps create child processes, callback functions run code only in main parent process. **API functions**: functions to interact with ThinApp runtime.
Log Monitor: captures executable's activity when app starts.

Links: http://kb.vmware.com/kb/1006306 - Troubleshooting VMware ThinApp installation issues
http://kb.vmware.com/kb/1006308 - Troubleshooting the build process in VMware ThinApp
http://kb.vmware.com/kb/1006317 - Troubleshooting virtualized apps that do not run in ThinApp
http://kb.vmware.com/kb/1017265 - Configuring isolation modes for File System & Registry
http://kb.vmware.com/kb/1019489 - Using ThinApp Capture Wizard to Virtualize an App (incl video)