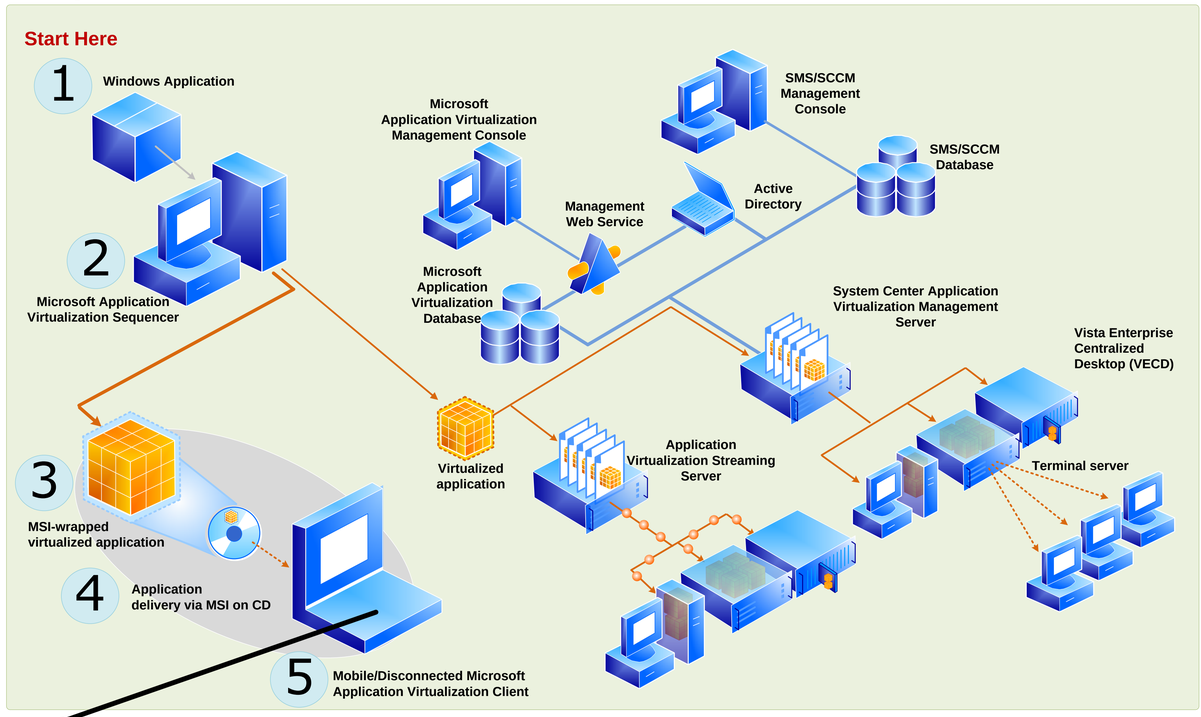
Application Virtualization for the Mobile User

Integrating the mobile (disconnected) user environment in an enterprise

The enterprise architecture, for delivering and maintaining application deployments, can be complex. The model to the right lays out a sample of what the full infrastructure could look like with a virtualized environment using sequencing. Microsoft Application Virtualization (App-V) in combination with a software distribution server, and management server can administer and report any configuration, including a sequenced application to a mobile user.

Mobile users whether connected or disconnected, have a solution within the enterprise virtualized environment as well.

For mobile users, using Microsoft Application Virtualization (App-V) to sequence the application then distribute the *.msi package via CD or USB drive. This enables the user to be disconnected from the network when accessing the virtualized application.



Sequencing

The process of creating an application package by using the App-V Sequencer. In this process, the application is monitored, its shortcuts are configured, and a sequenced application package is created that contains the .osd, .sft., .sprj, and .ico files.

Sequenced application

An application that has been transformed from a traditional installed application to one that runs in an App-V virtual environment.

Task List

Step 1:
Determine which application to virtualize

Step 2:
Install the application, run the application, customize the application.

Step 3:
Save the package

Step 4:
Deliver the msi-wrapped virtualized application via CD

Step 5:
Initiate installation and publishing of msi-wrapped virtualized application on the disconnected/mobile client

Assumptions

- Insufficient bandwidth
- Security standards
- Tracking roaming profiles and application licensing
- Backup and recovery
- Disconnected users and remote users

Benefits

- Removes incompatibilities and enables two applications installed on the same instance of the operating system.
- Changes are easier to implement.
- Application regression testing is made easier
- Simplified backup and recovery.
- Flexible configurations.

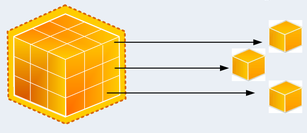
How App-V deployment works for the mobile user

Step 1: The Mobile user clicks *.msi file.

Step 2: The Windows Installer "installs" the virtual application. This is not an actual installation. Instead, Installer calls the client configuration: SFTMIME.EXE.

Step 3: The SFTMIME.EXE adds the application to the client, publishes shortcuts, adds the OSD, loads the SFT into the local file system cache: SFTFS.SFD. Switches the required registry setting from 1 to 0 for Requiring Authentication even if cached.

Windows Installer (*.msi) file



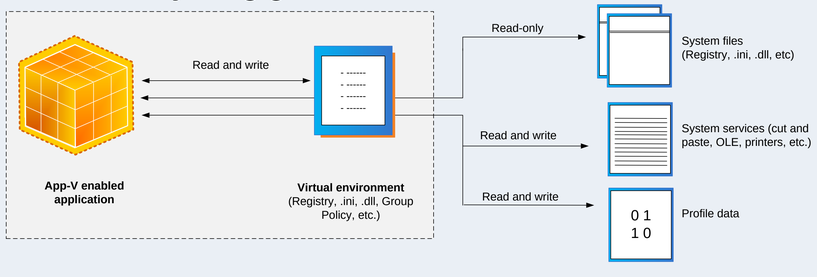
*.osd (Open Software Description file)

*.ico (Icon file)

All files pertaining to what needs to be done to install the application.

The *.msi file is the container that holds metadata so that a distribution system can recognize it and control the virtualization applications. It resembles a little database laid out in a structure storage file like an OLE document or compound document.

Where everything goes



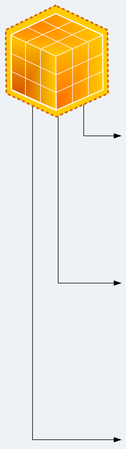
What happens after you initiate delivery to a mobile user

Files contained in sequenced images

SFT	.osd	.ico	.sprj	.sml Manifest	.msi
All application files. Binary file containing all assets and state organized into streamable feature blocks. Optionally a DSFT (Differential SFT).	Description of an application including environment, dependencies, package location, shell integration, and scripts.	Icons associated with each shortcut or FTA defined in an OSD or manifest. These are extracted from application resources.	Sequencing project with references to OSDs, default package settings, list of all parser items, classifications, and exclusions.	Publishing parameters for all applications in a package. Includes definitions of shell integration (FTAs, shortcuts, DDE, etc.). Typically used for patching or upgrades.	MSIInstaller to publish and load ("install") a virtual application package in stand-alone environments. Embeds all but SFT.

Dynamic Suite Composition

Dynamic Suite Composition Tool enables you to control which virtual applications will be combined.



Local User Cache Files:

- **AutoLoad** packages ensure that all assets of the application are available for mobile users, instead of only caching the primary feature block.

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SoftGrid\4.5\Configuration

- **ICO Cache** is stored in the following local locations:

  \AppData\Roaming\SoftGrid Client (for Windows Vista users) **\Application Data\SoftGrid Client** (for Windows XP users)

- **User Location**:
  The usrvol_sftfs_v1.pkg file contains user-specific files that are modified or new files that are created by any user process in the virtual environment.

SFTFS.FSD

The App-V client creates a cache file called **sftfs.fsd** that is used to store the application packages for use. At application runtime, the cache file contents are mounted to the application virtualization drive letter in the virtual environment. The mount gives access to the file system and files in the package. When packages are inserted into the cache via streaming or *.msi-based virtual application installations, the client stores them persistently in cache for subsequent launches. The **sftfs.fsd** file is located in the **Public** profile in Windows Vista and in the **All Users** profile in Windows XP.

Inside the .osd file: Controlling when, where, and how the script runs:

Open Software Description (.osd file) Specific information about the application and how to launch it is contained in a simple .xml text file.

```
<DEPENDENCY>
<SCRIPT TIMING="PRE|POST"
EVENT="STREAM|LAUNCH|SHUTDOWN"
PROTECT="TRUE|FALSE"
WAIT="TRUE|FALSE" TIMEOUT="n"
SUCCESSRESULT="n" ABORTRESULT="n">
<SCRIPTBODY>
    batch script commands
</SCRIPTBODY>
</SCRIPT>
</DEPENDENCY>
```

The following command shows how to install an MSI-based package for stand-alone users, when the SFT File is located on a file server.

Msiexec.exe /i \\PathToMsi\packagename.msi SFTPATH=\\server\share\package.sft /q

DRAFT

Microsoft