vmware SOLUTION READINESS

Virtualizing SAP GUI with VMware ThinApp™

VMware, Inc
3401 Hillview Ave
Palo Alto, CA 94304
www.vmware.com

Contents

1. Introduction

This white paper provides a beginner's guide to virtualizing the SAP® desktop client SAP GUI on Windows using VMware ThinApp™. The target audience is administrators who are new to ThinApp and are responsible for deploying the SAP front end client. This document contains screenshots showing an example deployment of SAP GUI 7.20 with ThinApp 4.6, and covers some of the use cases of a virtualized SAP GUI such as administration of configuration files and integration with Microsoft Office.

SAP GUI is the GUI client in the SAP R/3 three-tier architecture of database, application server and client. It is software that runs on a Microsoft Windows, Apple Macintosh or Unix desktop, and allows a user to access SAP functionality in the SAP applications suite (for example, ERP, Business Warehouse, CRM, and so on). VMware ThinApp virtualizes client applications on the Windows desktop. Using ThinApp, applications are packaged into single executables that run completely isolated from each other and the operating system for conflict-free execution on end point devices.

SAP GUI can be pre-installed in a virtualized ThinApp container. The container is then deployed onto target desktops or can be executed from a file share. Unlike traditional mass deployments no installation of the SAP GUI is involved. Virtualizing SAP GUI with ThinApp follows the same best practices and guidelines discussed in existing VMware ThinApp documentation (see the Additional Resources section). These resources should be consulted for additional background on ThinApp concepts and terms covered in this paper.

The virtualization of SAP GUI with VMware ThinApp is supported by SAP, for details please see the following SAP note available at the SAP Service Marketplace: 66971 - Supported SAP GUI platforms.

2. Features and Benefits of Virtualizing SAP GUI

This section summarizes the main features and benefits of virtualizing SAP GUI with ThinApp and refers to SAP GUI and ThinApp terminology and concepts that are covered later in the document.

A ThinApp virtualized SAP GUI client is independent from the underlying desktop operating system so you can run a single packaged SAP GUI client across multiple Windows operating systems. This helps to streamline client application migration and ease the burden of cost and complexity for IT.

Deployment of the virtualized SAP GUI across many desktops is much simplified. SAP GUI is installed and captured once in a virtual package which can then be deployed to multiple desktops. Multiple installations on each desktop are not required and no reboot or end-user configuration is required on the local desktop. For example:

- The local desktop `services` file does not need to be maintained with the necessary message server ports for the different backend systems as the required entries are captured in the `services` file in the ThinApp package. The `services` file in the ThinApp package does not impact the native `services` file on the local desktop, which operates normally.
- No SAP GUI configuration files such as `saplogon.ini` or `sapmsg.ini` are required on the local desktop.

SAP logon configuration files like saplogon.ini and sapmsg.ini can be stored in the ThinApp package and rolled out to desktops with either `saplogon.exe` or `saplgpad.exe` to differentiate between users who are allowed to change or not change the SAP logon selection list. For those users who use `saplogon.exe` and change the default selection list, they have the option to remove the sandbox on their local desktop and revert back to the default `saplogon.ini` stored in the package. If a user needs to run multiple versions of SAP GUI (for example, to test and compare features before upgrading) this can easily be achieved with a virtual SAP GUI package. Multiple versions of SAP GUI in separate ThinApp packages can be deployed to the same desktop and each SAP GUI version can be executed in turn.

Though a virtualized SAP GUI package operates in its own container independently from the underlying desktop operating system you can still perform standard SAP GUI functions that require integration with the desktop such as:

- Integration with natively installed Microsoft Office applications is still supported.
- SAP GUI file upload and download functionality to the desktop directory structure still performs as in native environments.
- If required, it is possible to make a call out from the virtual SAP GUI package to a centralized `saplogon.ini` file stored on a central share using the `/ini_file` input parameter for `saplogon.exe` (as with normal SAP GUI functionality).

3. SAP GUI

SAP GUI is the SAP universal client for accessing SAP functionality in SAP applications. There are three forms of SAP GUI:

- SAP GUI for Windows

  SAP GUI for Windows is an implementation especially designed for the Windows operating system. It provides a Windows-like user experience and integration with other applications based on OLE interfaces or ActiveX controls.

- SAP GUI for the Java environment

  SAP GUI for the Java environment is a unified SAP front end for multiple platforms including Windows. It is based on a platform-independent architecture and Java implementation.

- SAP GUI for HTML

  SAP GUI for HTML automatically maps the screen elements in SAP transactions to HTML using HTML business functions available within the SAP Internet Transaction Server. A Web browser is sufficient to access almost all transactions. This option does not need SAP software to be deployed on the client.

The focus of this document is on SAP GUI for Windows. Appendix B provides additional background on SAP GUI and its configuration files.

3.1 SAP GUI for Windows

SAP GUI for Windows provides support for the most common Microsoft Windows platforms. Version 7.20 is supported on: Windows 2000; Windows XP; Windows 2003 Server; Windows 2009 Server; Windows Vista and Windows 7.

SAP GUI for Windows uses the Microsoft Controls technology, which means that all controls are registered locally during the installation. As a result, the system database contains an entry indicating where each control can be found. Because the system always registers the latest controls, only the most recently installed version of a control is available at any time.

On a single computer only one version of SAP GUI for Windows can be installed. In native environments, during installation of a new SAP GUI release, any older SAP GUI release present on the computer is uninstalled.

4. ThinApp Overview

Just as VMware virtual machine technology decouples the operating system from hardware, VMware ThinApp decouples the desktop application from the operating system to deliver the same benefits of flexibility, portability, and isolation. Administrators can package applications once and deliver to users across multiple environments consisting of physical desktop, virtual desktop and shared Terminal Services.

VMware ThinApp creates application containers by using a build process to package application files and registry into a single compressed executable file that can be run on a variety of operating systems without installation. The application container utilizes block-based streaming with transparent decompression into memory to execute all application functions. Applications can be executed from a user's desktop, a network path, or removable storage such as a USB flash drive or CD ROM. Applications run entirely in user mode and are visible to the operating system as standard windows processes to maintain the appropriate security context. VMware ThinApp presents operating system resources and functions to the virtualized application, allowing full functionality and a seamless user experience while still encapsulating the application's files, registry entries, COM/ActiveX controls, and services in a portable container for use across multiple operating systems.

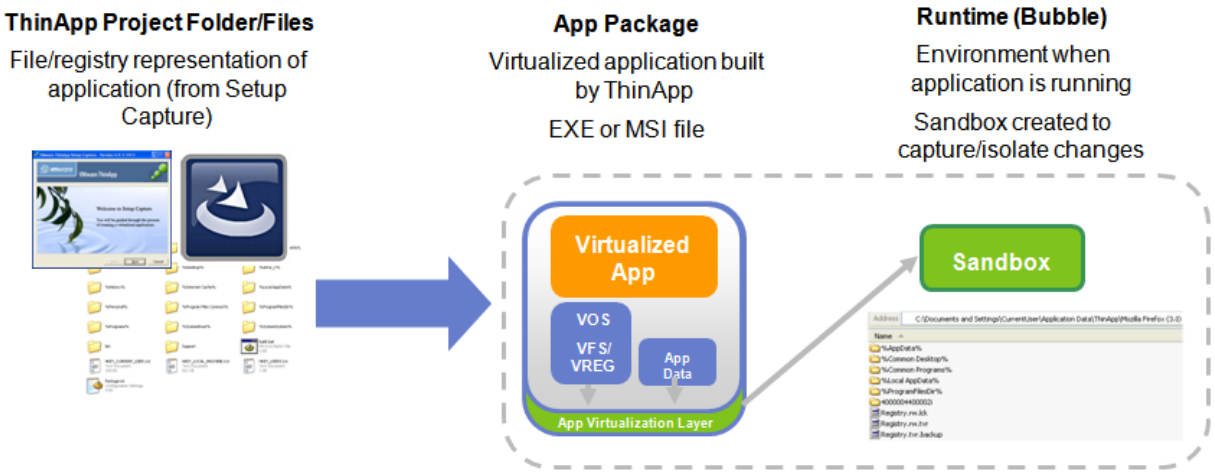
4.1 Virtualizing Client Applications

Virtualization of an application involves a setup capture and build process. The setup capture captures the application and creates a project to which further administrator configurations can be added. Setup capture involves: a pre-scan snapshot; application installation and configuration; and a post-scan snapshot. The difference between the pre- and post-scan snapshots represents the application, and is placed into the project directory. The output of the setup capture process is called a Project. The project contains everything needed to build and modify the virtualized application. This is followed by the build process by which the project directories and configuration settings are compressed and embedded into a package. The build process can be re-run at any time to incorporate different settings or application changes into a package. The package is deployed to target desktops.

VMware recommends running the capture on a clean operating system where only the basic components of the operating system are installed. However, an updated Windows Installer 3.1 will likely be required for the application install to commence. It is also recommended that you use the oldest version of the operating system to make sure that the application install has all required files. The reason to capture from a clean operating system is to make sure that all the files and components necessary for the application are detected by the Setup Capture process. If there is application install logic that looks for a certain version of a DLL file and the capture machine finds it in the local operating system, then that DLL file is not be included in the virtualized application package, and that may prevent the application from functioning when deployed to an operating system instance that has not been updated.

Figure 1 illustrates the process of virtualizing an application with ThinApp.

Figure 1. ThinApp Virtualization Overview



The ThinApp architecture includes:

- The Virtual Operating System (VOS)

  The VOS transparently merges the virtual system environment with the real system environment (it encapsulates the Windows OS environment) and is comprised of: virtual file system (VFS) and virtual registry (VREG). It performs process and DLL loading, and thread and process management.

- The Package

  The package is the result of the Setup Capture and Build process. It is an application container that encapsulates the application, the ThinApp run-time engine and all of the required configuration settings. A package is created in a ready to run, read-only `EXE` format. The `packaged .exe` is compressed and encrypted to protect the virtualized application from tampering or modification. Additionally, the package can be placed in an MSI wrapper for native integration with software deployment tools.

- Sandbox

  This is a directory where all application runtime changes are stored (for example, user configurable settings like Web browser home page), allowing persistence of user settings. The next time you start the application, the changes are incorporated from the sandbox. When you delete the sandbox directory, the application reverts to its captured state. The default location is `%AppData%\Thinstall\<`*appname*`>`, where *appname* is the name of the application. Folder macro names are used instead of hard-coded shell folder location paths. For example, `%AppData% = C:\Documents and Settings\username\Application Data`.

4.2 Deployment

The process for deploying ThinApp application containers is simple as there is no actual installation of the application and interoperation with the local operating system is minimal. You can deploy the captured ThinApp application via the methods described below (for more information see the *VMware ThinApp Deployment Guide*):

- Deploying ThinApp With Deployment Tools

  Medium and large enterprises often use major deployment tools, such as Symantec, BMC, and SMS tools. The package (containing the virtualized application) can be placed in an MSI wrapper that can then be used with all major deployment tools to deploy in the same way as native MSI files.

- Deploying ThinApp in the VMware View™ Environment

  The steps to work with VMware View are documented in the *VMware ThinApp Reference Architecture Guide*.

- Deploying ThinApp on Network Shares

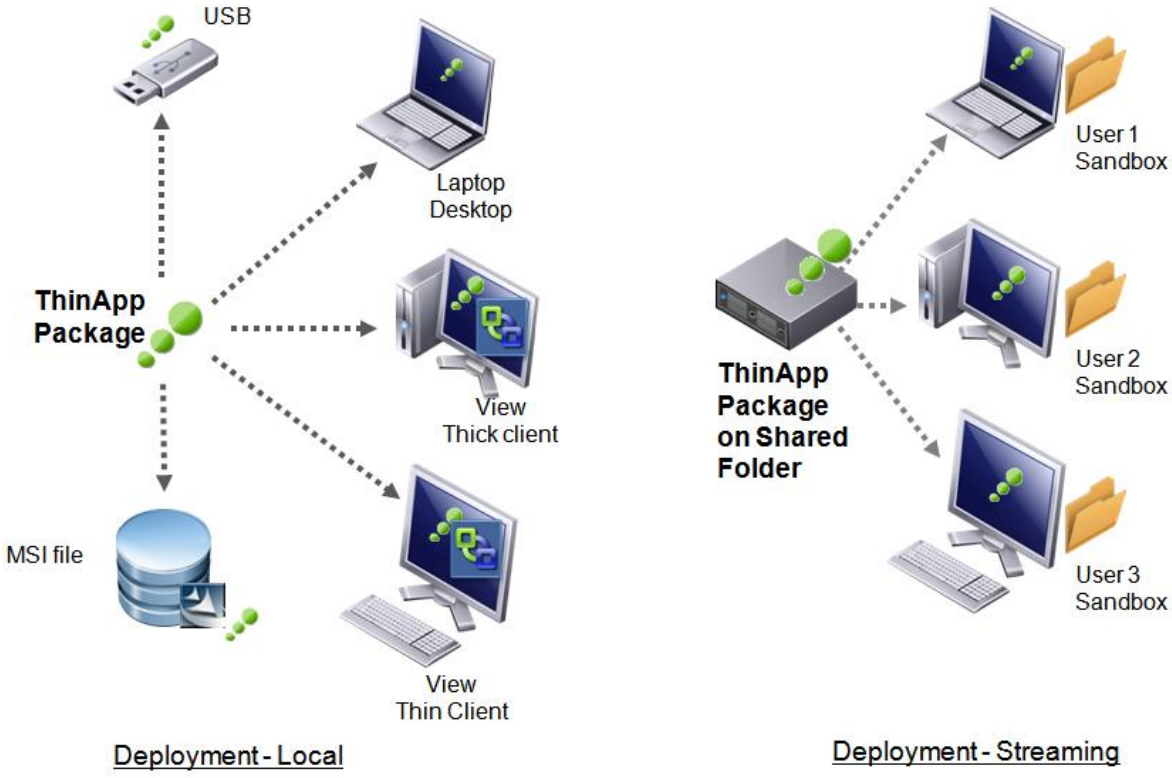
  You can create executable files for the captured application and store them on a network share.

- Deploying ThinApp locally on the desktop

  You can create executable files for the captured applications, copy them from a central repository, and run them locally on the desktop. Also the package can be installed on the local desktop using the `Thinreg.exe` utility—this registers application shortcuts on the desktop, creates file-type associations, and adds entries in the Add/Remove programs applet of the Control Panel. Alternatively, if the package is deployed using a MSI wrapper, it includes the Thinreg tool and automatically invokes it.

Deployment options are illustrated in Figure 2.

Figure 2. ThinApp Deployment Options

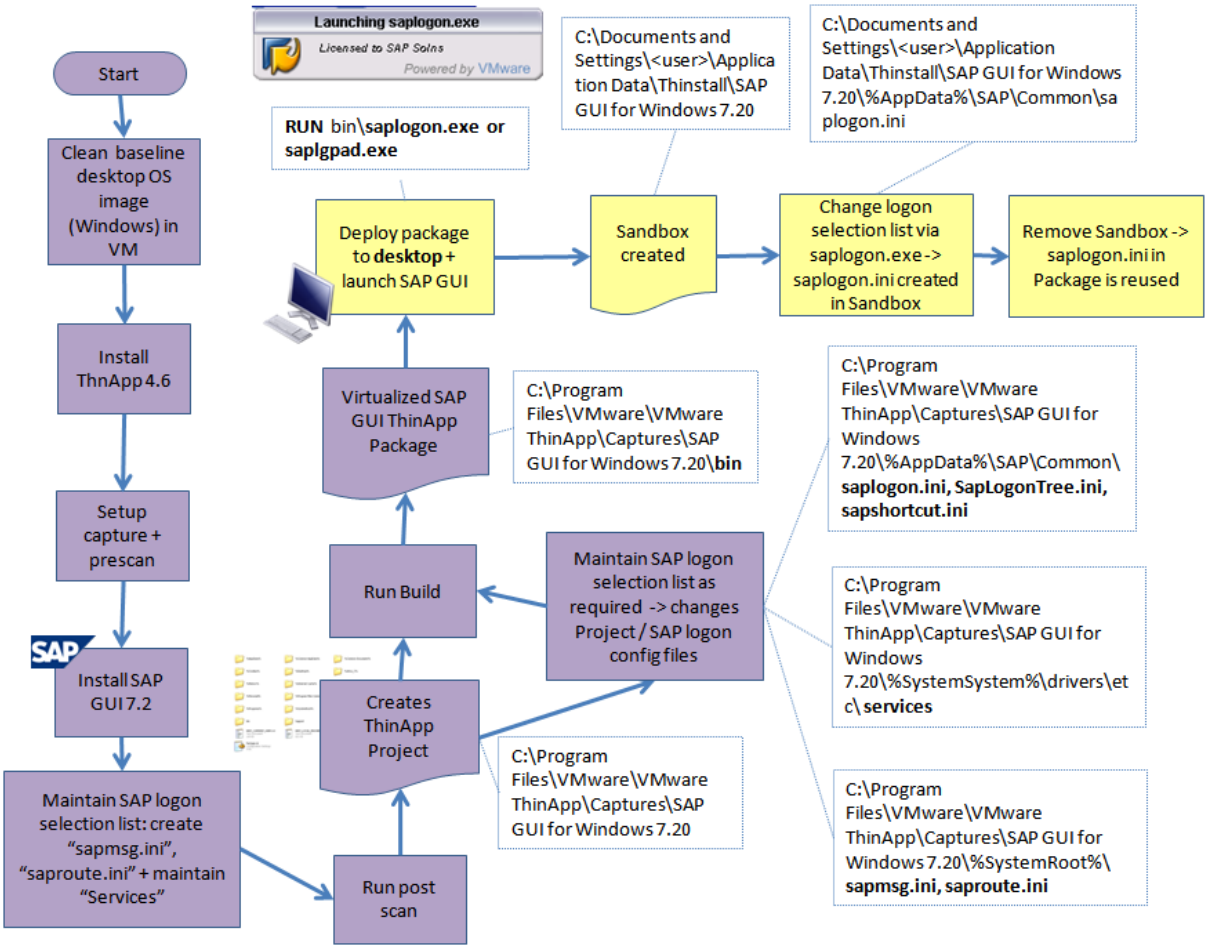


5. Deploying SAP GUI with ThinApp

5.1 Virtualizing SAP GUI

Appendix A documents in detail an example of virtualizing SAP GUI 7.2 with ThinApp. The process steps are shown at a high level in Figure 3.

Figure 3. Steps to Virtualize and Deploy SAP GUI with VMware ThinApp



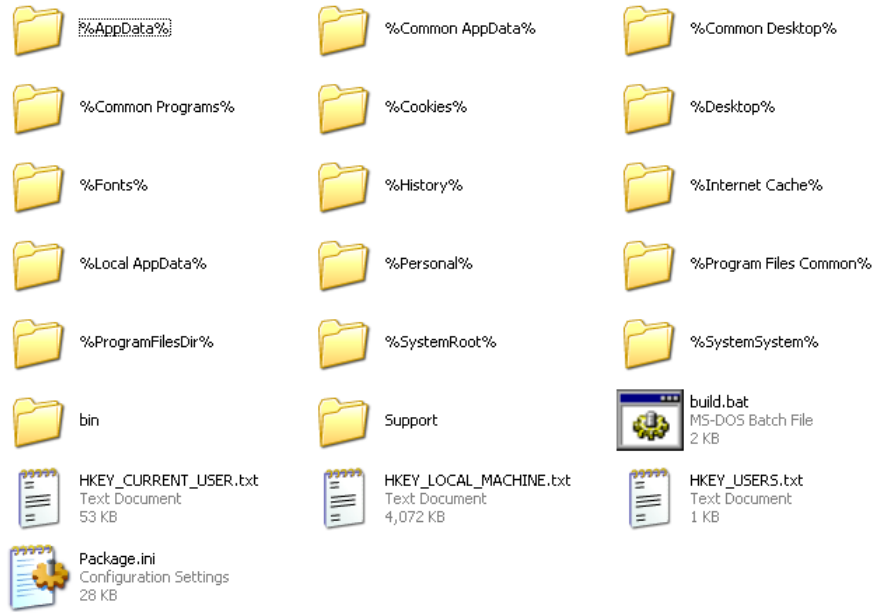
The process steps in the purple are conducted on a cleanly installed desktop image which is typically on a virtual machine. The SAP GUI client application is captured and packaged by ThinApp. The package is then deployed onto a target desktop and subsequent execution steps of SAP GUI are shown in yellow.

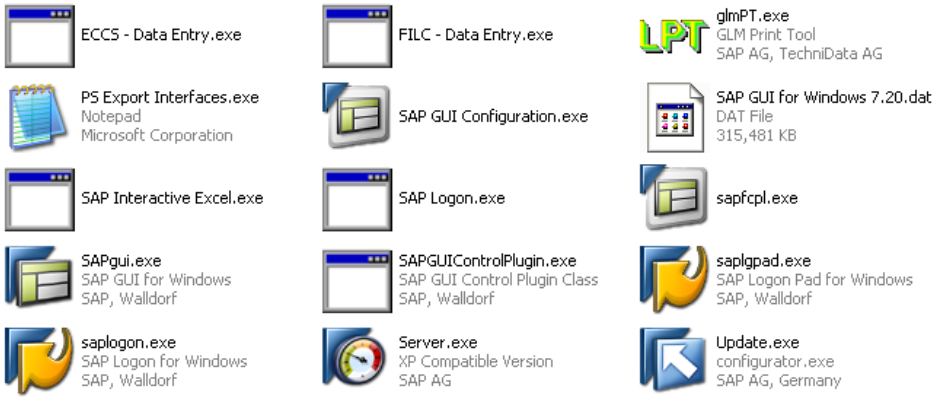
5.2 Project Folder and Files

The relevant directories of the virtualized SAP GUI are listed in Table 1. The virtualized package is deployed onto target desktops as illustrated in Figure 2.

Table 1. Project Files

Title	Directory

PROJECT	`C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20`
	 
	Notes:
	The SAP logon configuration files (`saplogon.ini`, `sapLogonTree.ini`, `sapshortcut.ini`) are located in `..\%AppData%\SAP\Common`
	`sapmsg.ini` and `saproute.ini` are located in `..\%SystemRoot%\`
	The `services` file is located in `..\%SystemSystem%\drivers\etc\`
	Changes to the SAP logon selection list are made by updating the above files in their respective directories. The project then needs to be rebuilt so the configuration changes are updated in the package.
	Run `Build.bat` to rebuild the package. The `Build.bat` file is a batch file that is used by the Setup Capture process, or can be run manually to create the ThinApp packaged executable. If required, see VMware KB article 1006159 – Configuring VMware ThinApp for information about running from a non-default location.
	The `Package.ini` file is contained in every `Project` folder and is a repository of all ThinApp configuration data for deployment, update, shortcuts, and entry points of a particular application package.

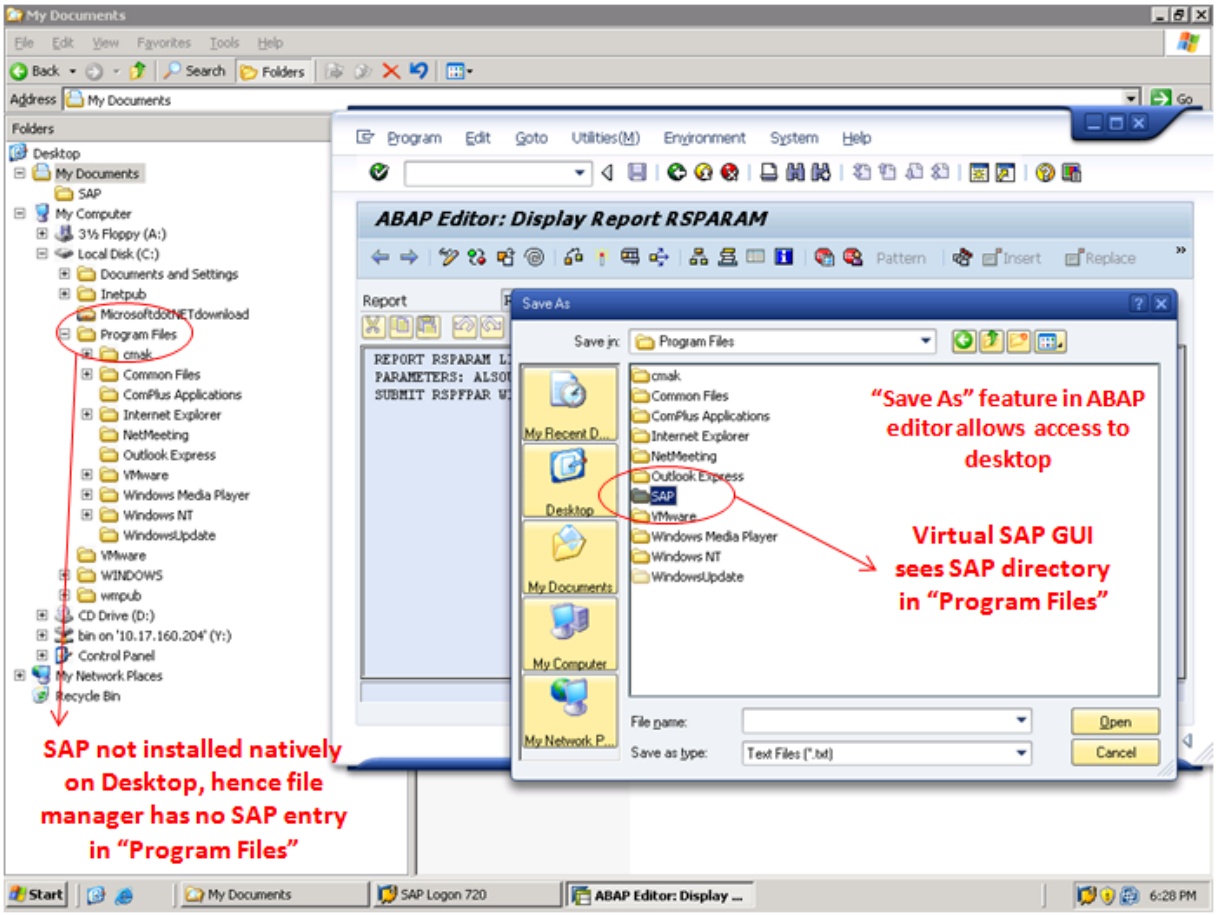
Virtualized SAP GUI Package/ Container	`C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20\bin`  Notes: The above contents represent the virtualized SAP GUI ThinApp package/container. It encapsulates the SAP GUI client, the ThinApp runtime engine, and all of the required configuration settings. It is this content that needs to be deployed on target desktops to run SAP GUI. The executables were recorded during the capture process and selected as entry points. To launch SAP GUI, only `saplogon.exe` or `saplgpad.exe` is required. The others can be discarded. To handle larger packages, VMware ThinApp creates a separate SAP GUI for the Windows `7.20.dat` data container (when the estimated size of a data container is over 200MB).

5.3 Running Virtualized SAP GUI on Desktop

This section describes how: the packaged SAP client runs on the desktop; call outs can be made out of the virtualized package to an external `saplogon.ini` file; the package integrates to Microsoft applications already installed on the desktop.

The figure below shows a screenshot of the SAP GUI package deployed on a target desktop that has no native install of the SAP client.

Figure 4. Screenshot of Virtualized SAP GUI Deployed on Desktop



The figure shows how the virtualized SAP GUI client behaves as if it is natively installed on a desktop. The client application sees the file system from within the ThinApp runtime environment and sees the SAP subdirectory in `Program Files` as expected. The actual physical file system does not have the SAP client installed and therefore there is no SAP entry in `Program Files`.

5.3.1 Maintaining a Centralized SAP Logon Configuration File saplogon.ini

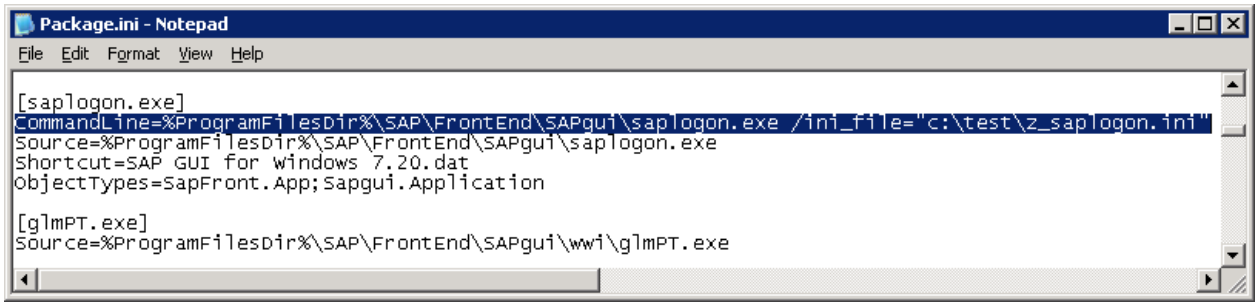
Run `saplogon.exe` or `saplgpad.exe` to launch the virtualized SAP GUI package—the executable reads the configuration file `saplogon.ini` that is in the package. If users are required to access an alternative `saplogon.ini file,` it is possible to pass an argument `/ini_file` to `saplogon.exe` to specify an alternative `ini` file (for example, `z_saplogon.ini` – see SAP note 38119 for more background). The latter can reside outside of the virtual bubble and can be accessed by the virtual package by maintaining an entry in the ThinApp `package.ini` file in the project (`C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20\Package.ini`).

For example, if `z saplogon.ini` is located in `C:\test`, the following line needs to be inserted in `Package.ini`:

CommandLine=%ProgramFilesDir%\SAP\FrontEnd\SAPgui\saplogon.exe /ini_file=C:\test\z_saplogon.ini

Position the line in the `Package.ini` file as shown in Figure 5.

Figure 5. CommandLine Entry in Package.ini



The Project then needs to be rebuilt by executing `Build.bat`.

5.3.2 Integration with Microsoft Office

The virtualized SAP GUI package can execute natively installed MS office applications from within the GUI as is normal SAP GUI behavior. For example, this integration can be tested using ABAP program SAPRDEMOOFFICEINTEGRATION. However, there are two workarounds for PowerPoint and Excel.

Microsoft PowerPoint

The following line needs to be added to the `Package.ini` file:

ForcedVirtualLoadPaths=%ProgramFilesDir%\Microsoft Office\Office12\PPCORE.DLL

The entry needs to be added twice in `Package.ini` to the `saplogon.exe` and `saplgpad.exe` sections. Examples follow.

Figure 6. ForcedVirtualLoadPathes Entry in saplogon.exe Section of Package.ini

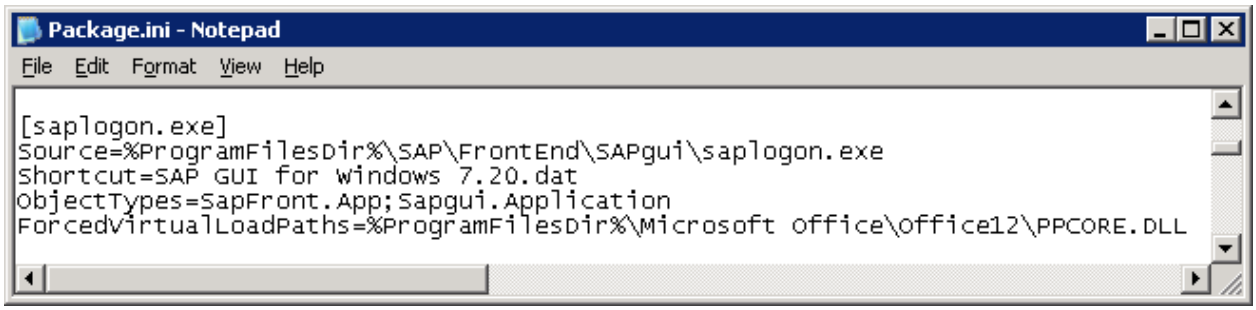
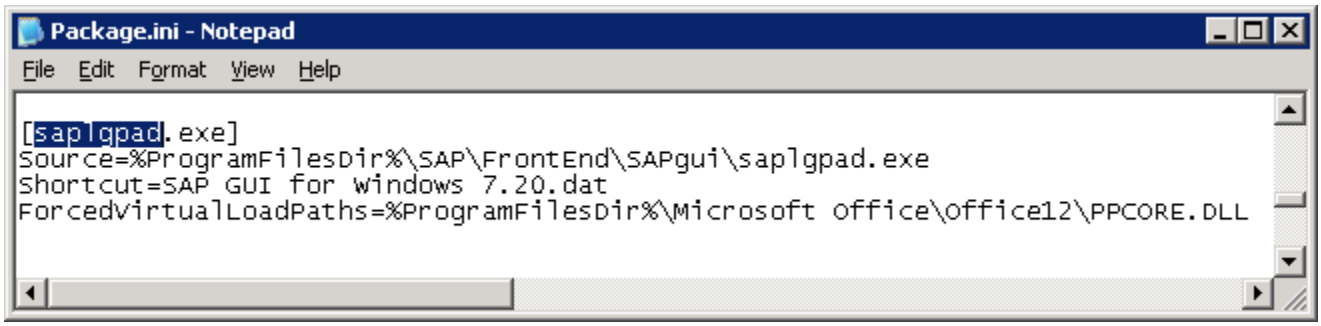


Figure 7. ForcedVirtualLoadPathes Entry in saplgpad.exe Section of Package.ini



Microsoft Excel

The following workaround in `Package.ini` is required to enable Excel charts to be visible within SAP GUI:

Uncomment "VirtualizeExternalOutOfProcessCOM=0" by removing the ";" at the beginning of the entry.

After updating `Package.ini`, rebuild the package using `build.bat`.

6. Conclusion

This white paper discusses how you can virtualize SAP GUI 7.2 using VMware ThinApp and how the SAP GUI configuration files such as `saplogon.ini` and `sapmsg.ini` operate in a VMware virtual environment on the desktop. Virtualization of the SAP GUI client with ThinApp provides the following benefits:

- Deployment of SAP GUI across many desktops is much simplified. SAP GUI is installed and captured once in a virtual package which can then be deployed to multiple desktops. Multiple installations on each desktop are not required and no reboot or end user configuration is required on the local desktop.
- SAP logon configuration files like `saplogon.ini` and `sapmsg.ini` can be stored in the package and rolled out to desktops using either `saplogon.exe` or `saplgpad.exe` to differentiate between users who are allowed to change or not change the SAP logon selection list.
- If a user may need to run multiple versions of SAP GUI (for example, to test and compare features before upgrading), this can easily be achieved with a virtual SAP GUI package. Multiple versions of SAP GUI in separate ThinApp packages can be deployed to the same desktop and each SAP GUI version can be executed in turn.
- Though a virtualized SAP GUI package operates in its own container independent from the underlying desktop operating system, you can still perform standard SAP GUI functions that require integration with the desktop such as uploading and downloading files and launching Microsoft Office applications.
- Integration with Microsoft Office applications installed on the desktop perform as they normally do with the virtual SAP GUI package, but integration for PowerPoint requires the workaround documented in this white paper.

About the Author

Vas Mitra is a SAP Solutions Architect in the VMware Technical Services organization. Vas has worked on SAP projects since 1993 as an ABAP programmer, Basis Administrator and technical SAP Solutions Architect.

Acknowledgements

The author would like to thank the following for their contributions and reviews: Aaron Black; Dean Flaming; Michael Hesse; Andre Kemp; Scott Salyer; Rupen Sheth; and Matthew Wood.

6.1 Additional Resources

6.1.1 VMware Resources

- VMware ThinApp Documentation:
  http://www.vmware.com/support/pubs/thinapp_pubs.html
  http://www.vmware.com/products/thinapp/
- *VMware View 4 and VMware ThinApp Integration Guide*
  http://vmware.com/files/pdf/VMware-IG-ViewThinApp-EN.pdf
- *VMware ThinApp Deployment Guide:*
  http://www.vmware.com/files/pdf/VMware_ThinApp_Deployment_Guide.pdf
- VMware ThinApp Reference Architecture, A Guide for Enterprise VMware ThinApp Deployments:
  http://www.vmware.com/files/pdf/thinapp-ref-arch.pdf
- VMware KB article 1006159 – *Configuring VMware ThinApp to run from a non-default location*
  http://kb.vmware.com/kb/1006159

6.1.2 SAP Resources

- SAP GUI Family (SAP Service Marketplace logon credentials required):
  http://service.sap.com/sapgui
- SAP Notes (SAP Service Marketplace logon credentials required)
  http://service.sap.com/support
  - Note 66971 - Supported SAP GUI platforms
  - Note 38119 – SAP Logon: Administration of functions
  - Note 1409494 – SAP Logon (Pad) 7.20: configuration (INI) files storage

Appendix A: SAP GUI ThinApp Setup Capture and Build

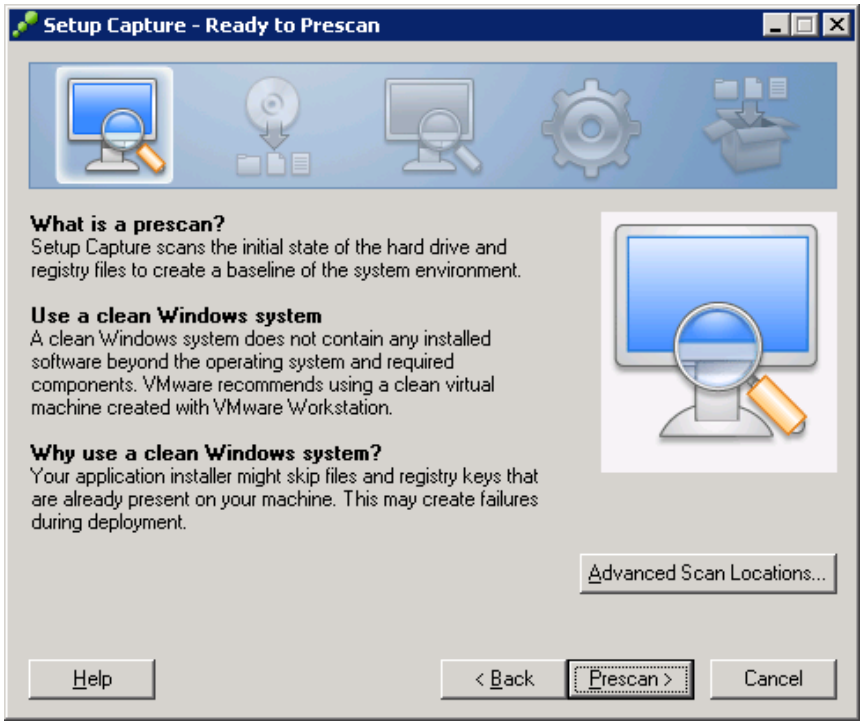
This appendix provides procedures for virtualizing SAP GUI 7.2 using ThinApp, starting with the ThinApp capture step. These procedures assume that ThinApp has been already been installed on a clean base desktop image. The overall process is illustrated in Figure 3.

Start Capture

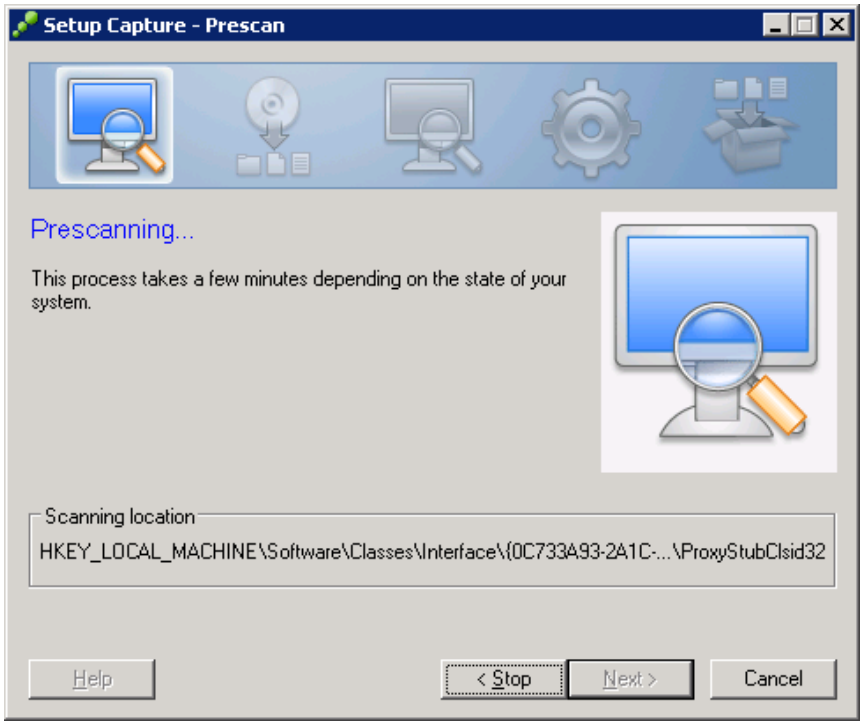
1. Select **Start** > **All Programs** > **VMware** > **ThinApp Setup Capture** to launch ThinApp.

2. Click **Next**.

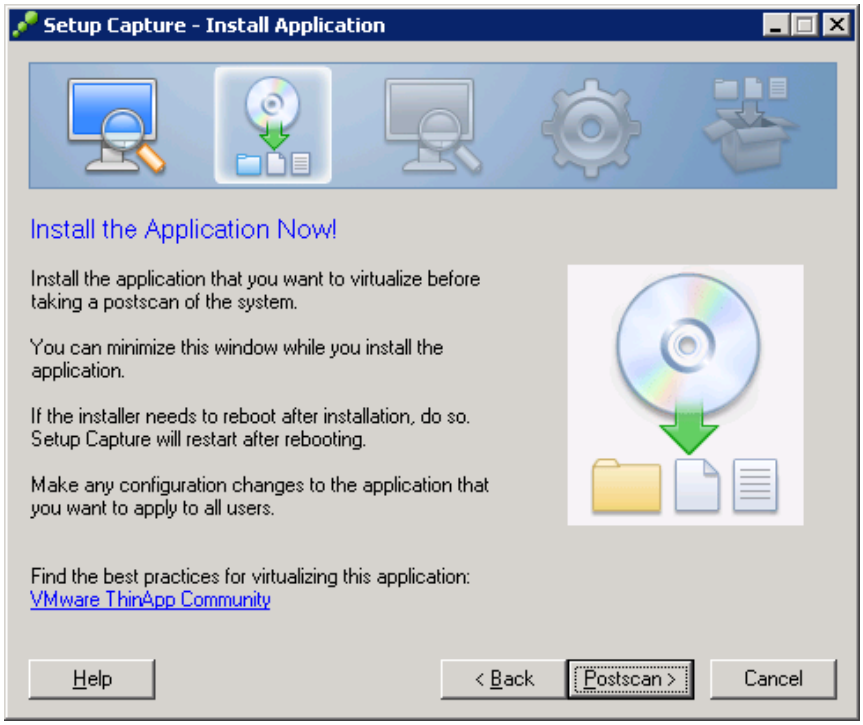
3. Click **Prescan**.



4. The Prescan is run.

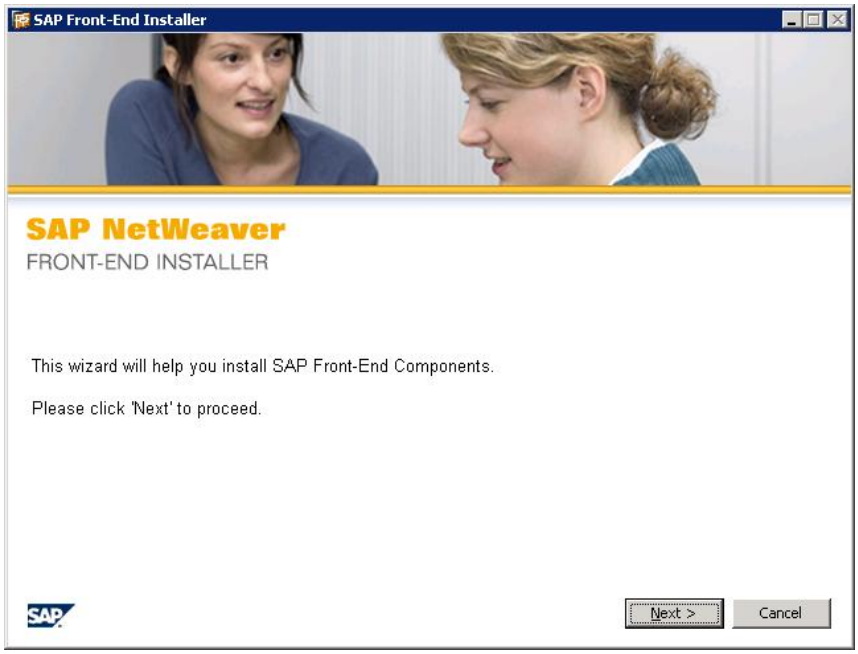


5. Minimize the ThinApp Capture Wizard and install the SAP GUI.

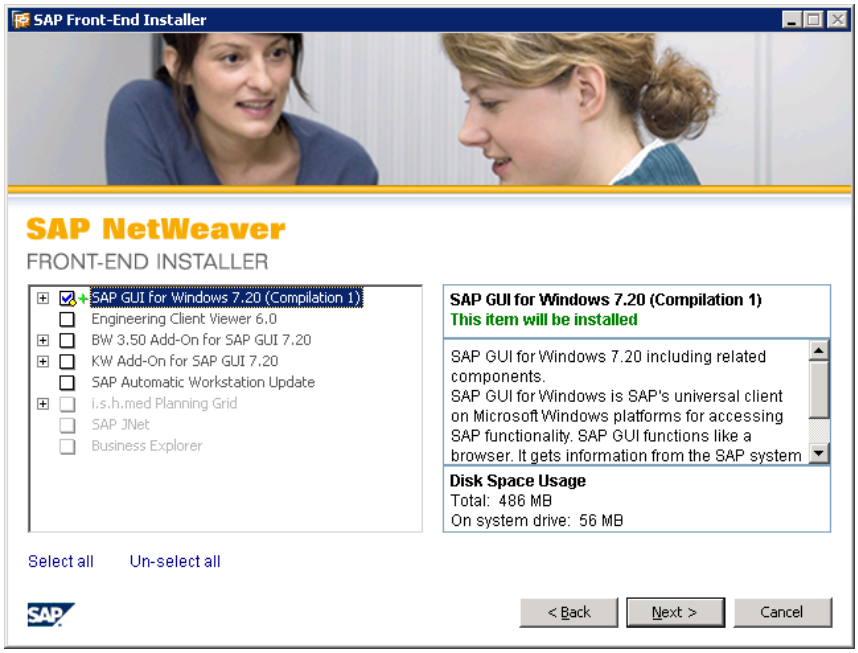


Install SAP GUI

1. Perform a standard install of the SAP GUI.



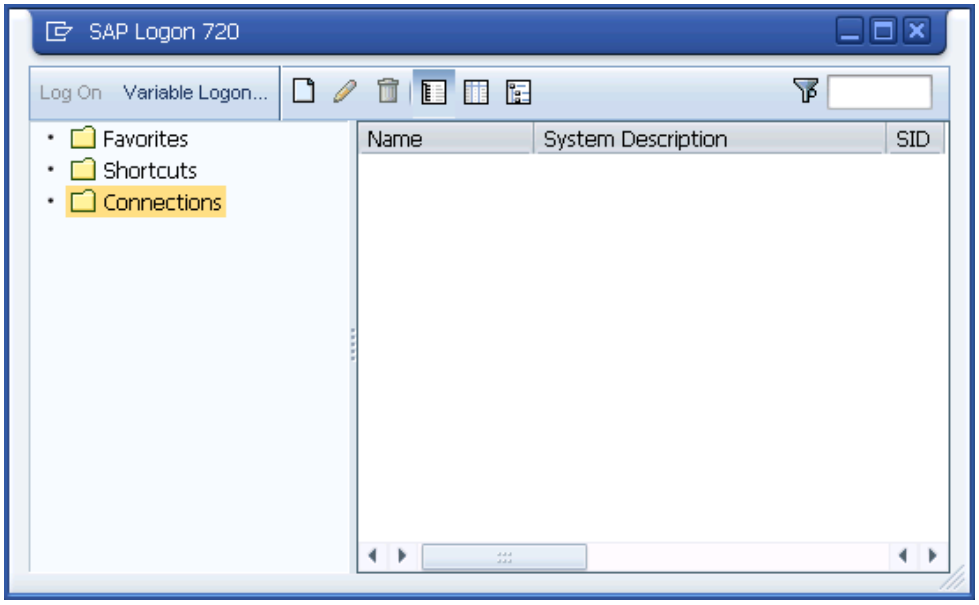
2. Select components and complete installation as per the standard SAP GUI install process.



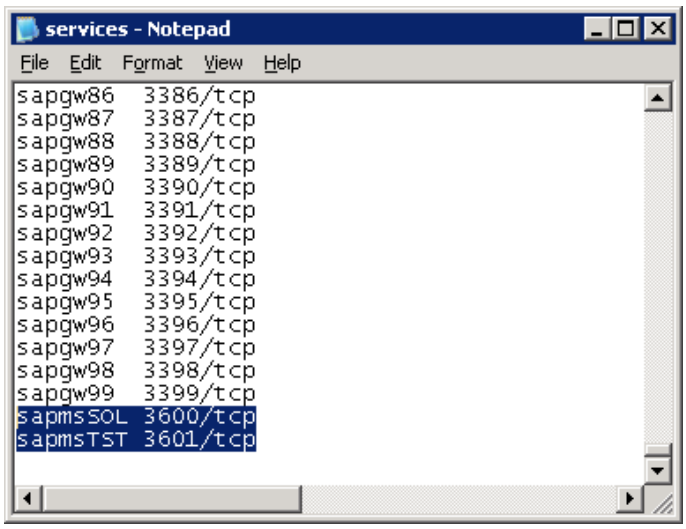
3. When the installation completes, click **Done**.

Configure SAP Logon Selection Screen

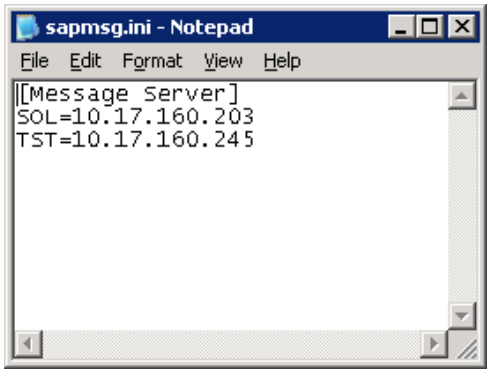
1. Double-click the SAP icon on the desktop “SAP Logon”.



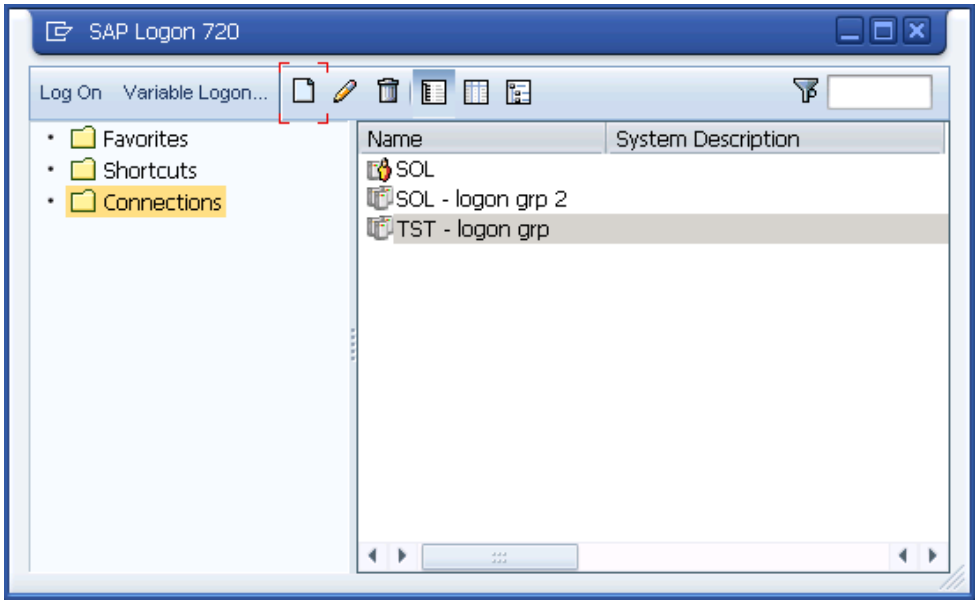
2. Add message server entries in the `services` file



3. Create `C:\WINDOWS\sapmsg.ini`.
4. In `sapmsg.ini`, create entries for backend SAP systems and test connection.

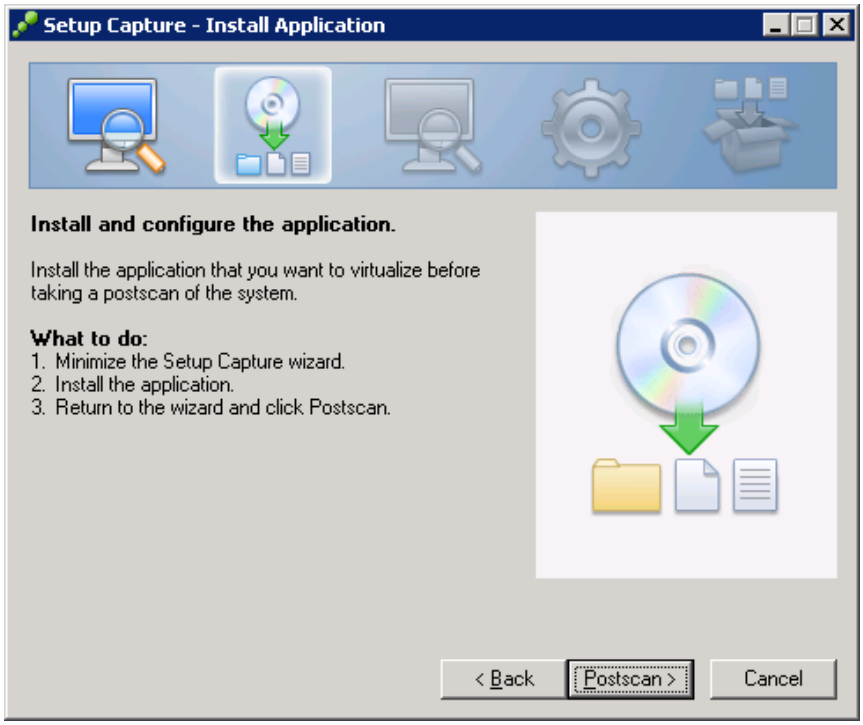


5. Verify that the entries were successfully created.

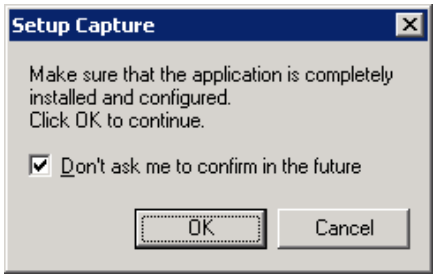


Run ThinApp Postscan

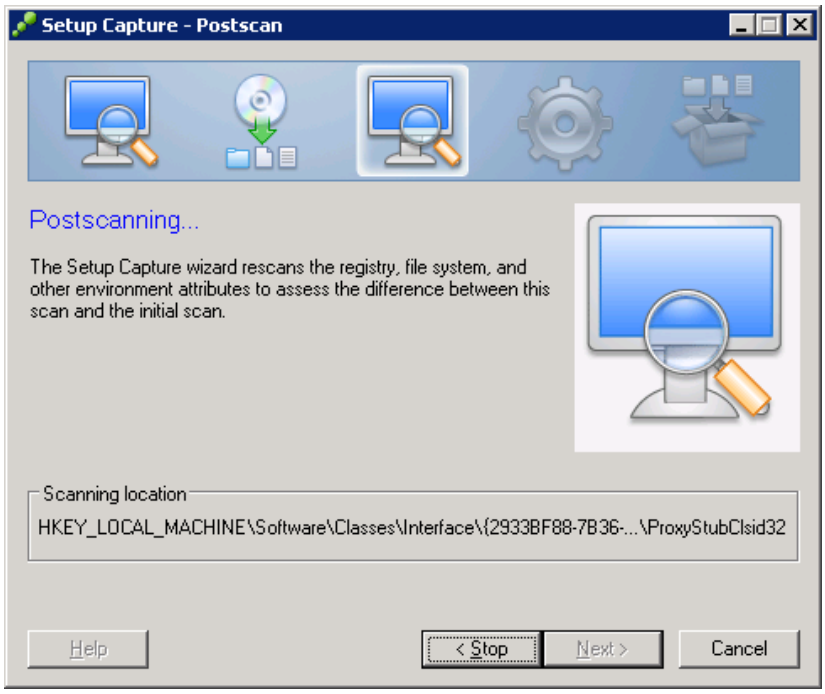
1. Access the ThinApp Setup Capture Wizard.



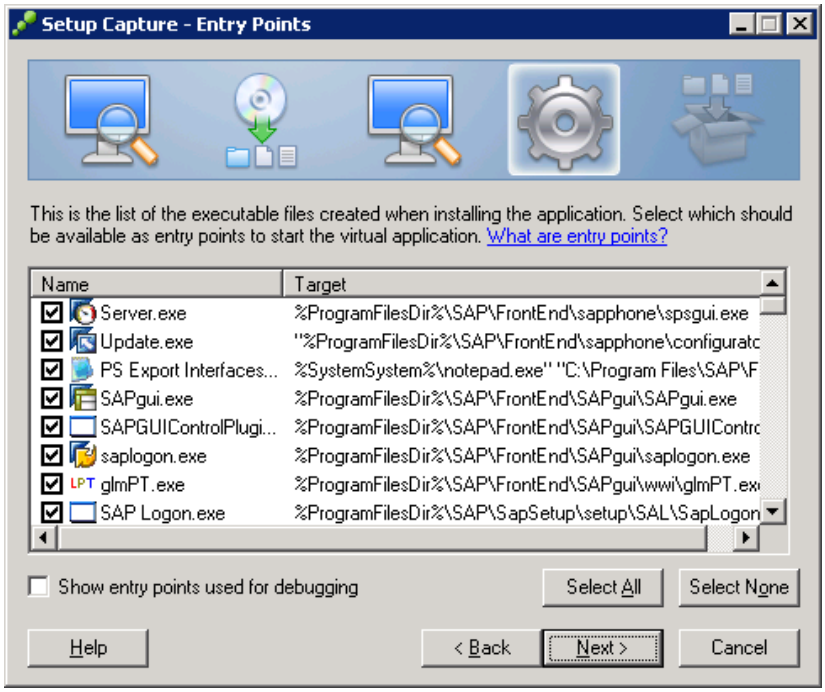
2. Click **Postscan**.



3. Click **OK**.
4. The Postscan is run.

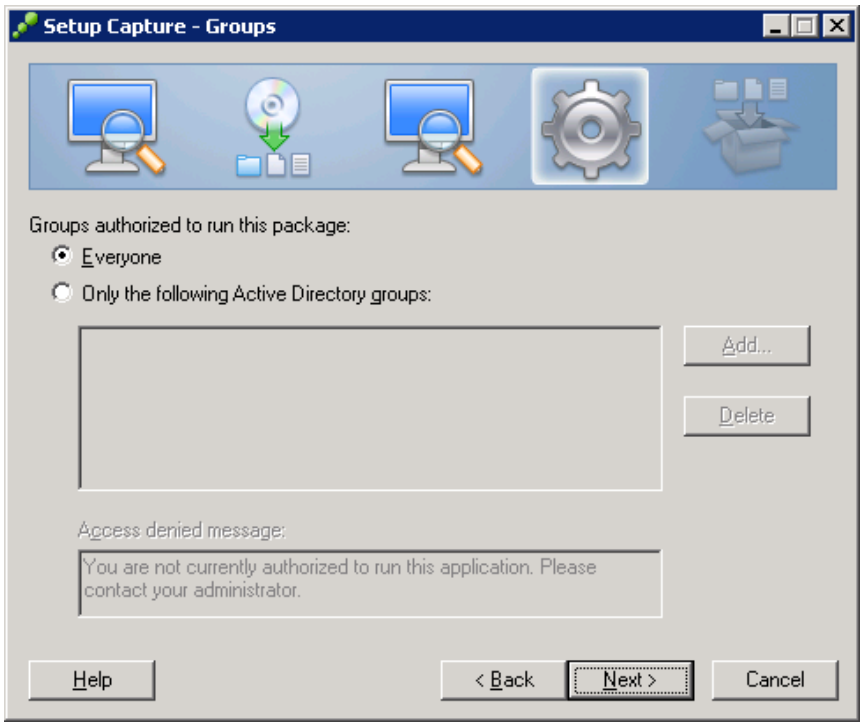


5. Specify the entry points. Entry points are the executable files that start the virtual application. The entry points depend on the executables that the SAP GUI-captured application created during installation.
   - Deselect entries that begin with `_sap.ex`.
   - Select entry `saplgpad.exe`.

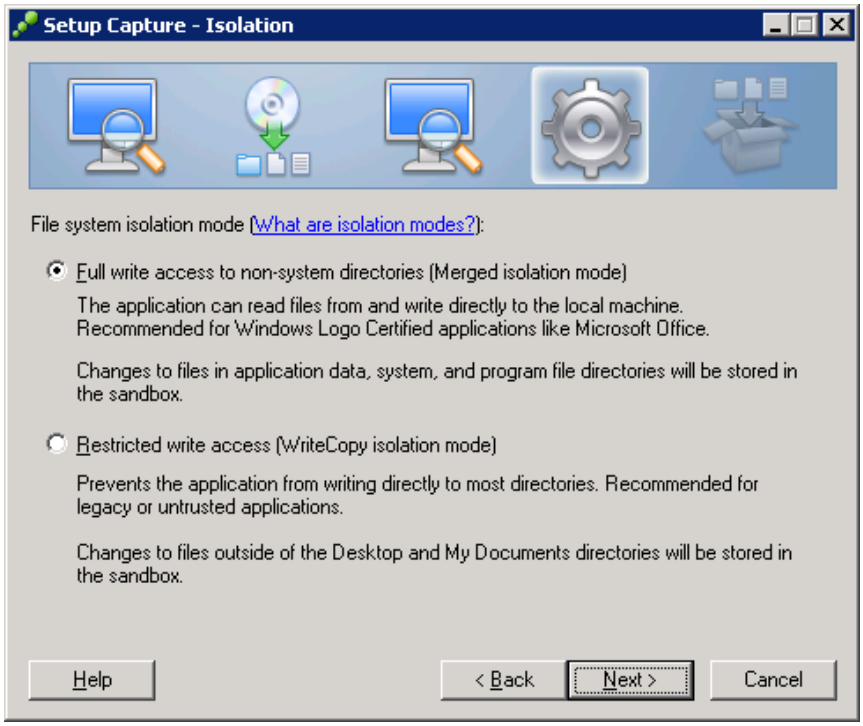


6. Click **Next**.

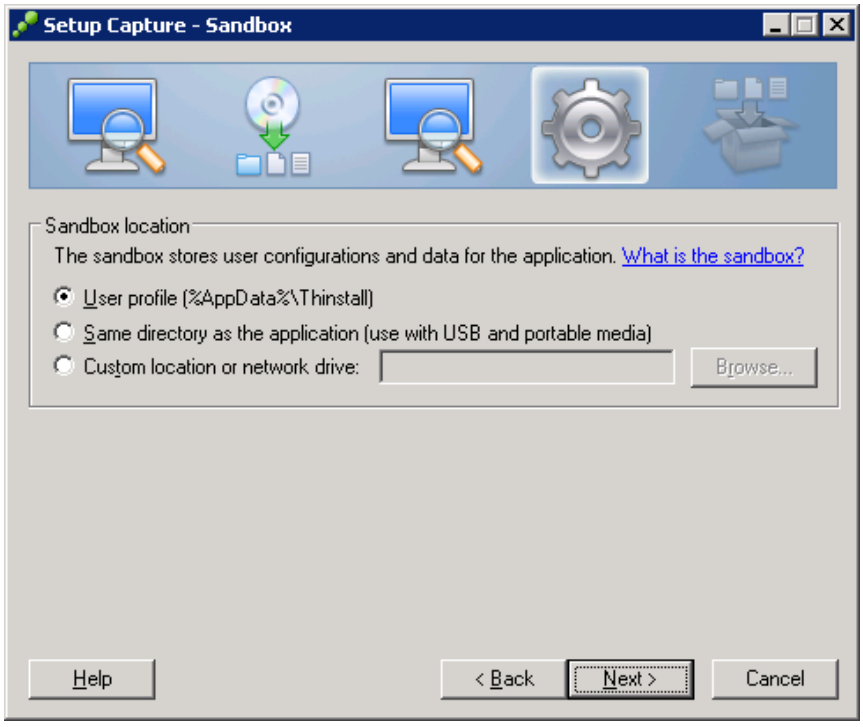
7. Select **Everyone**, or select **Only the following Active Directory groups** and add the authorized groups.



8. Click **Next**.
9. Select one of the isolation mode options:
   - **Full write access to non-system directories** (default) – allows SAP GUI to read resources on and write to all locations on the local desktop except for application data, system and program file directories.
   - **Restricted write access** – allows SAP GUI to read/write only to the `Desktop` and `My Documents` on the local desktop.

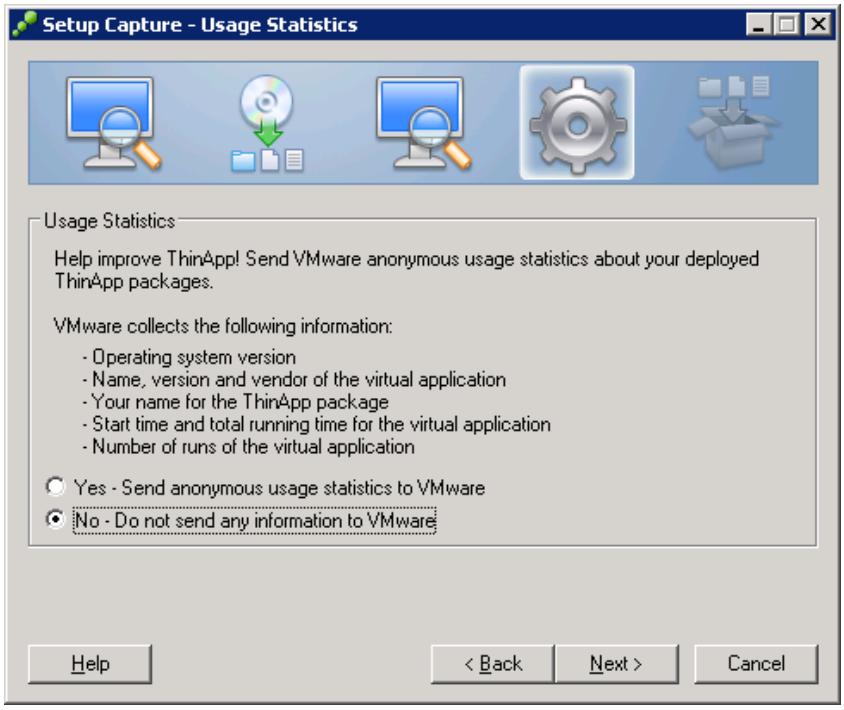


10. Select **User Profile** (default) for the sandbox location.

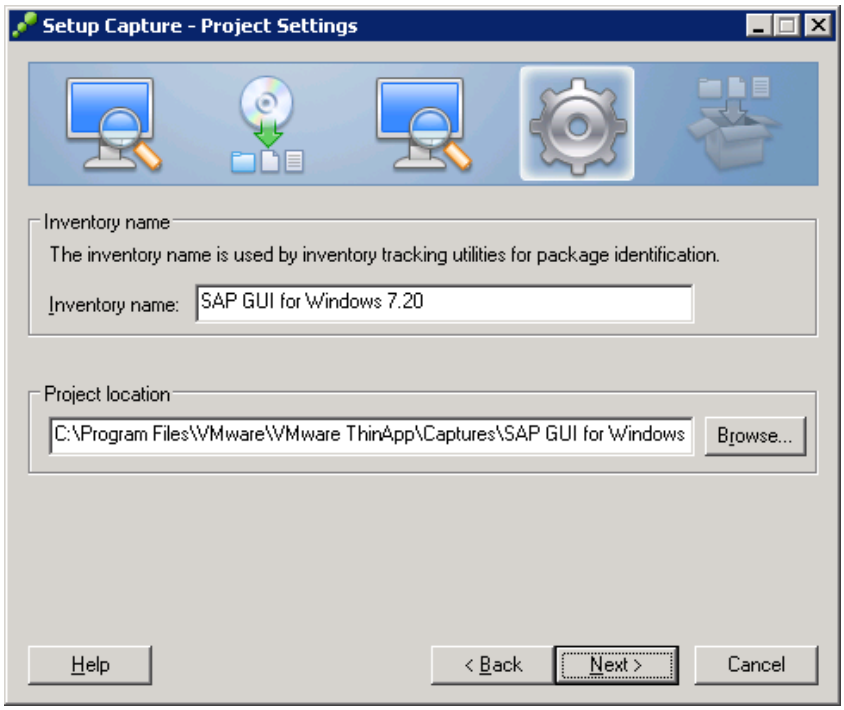


11. •Click **Next**.

12. Select **No – Do not send any information to VMware**.



13. Click **Next**.
14. On the Project Settings screen, accept the defaults.

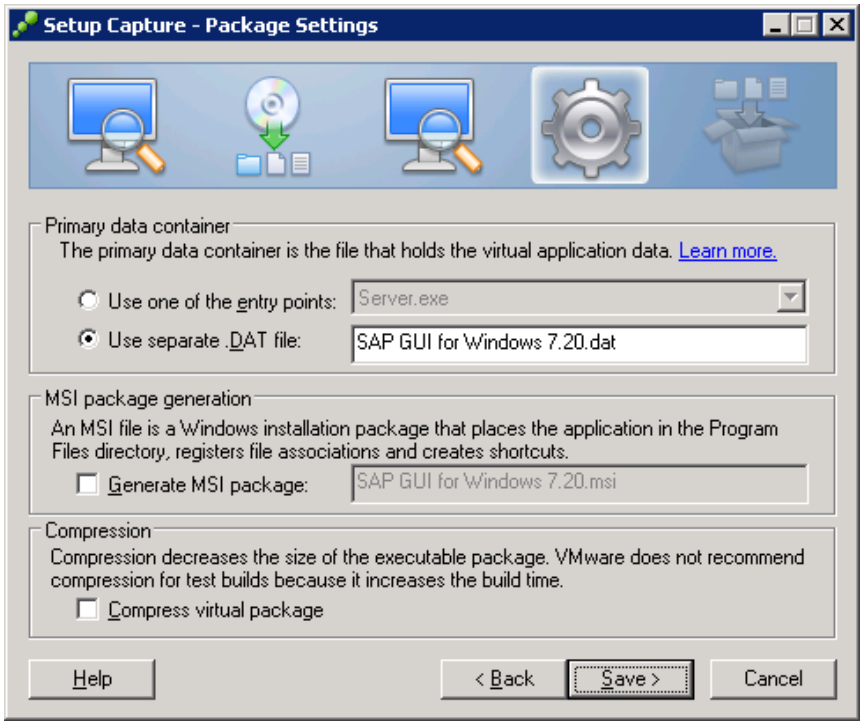


15. Click **Next**.

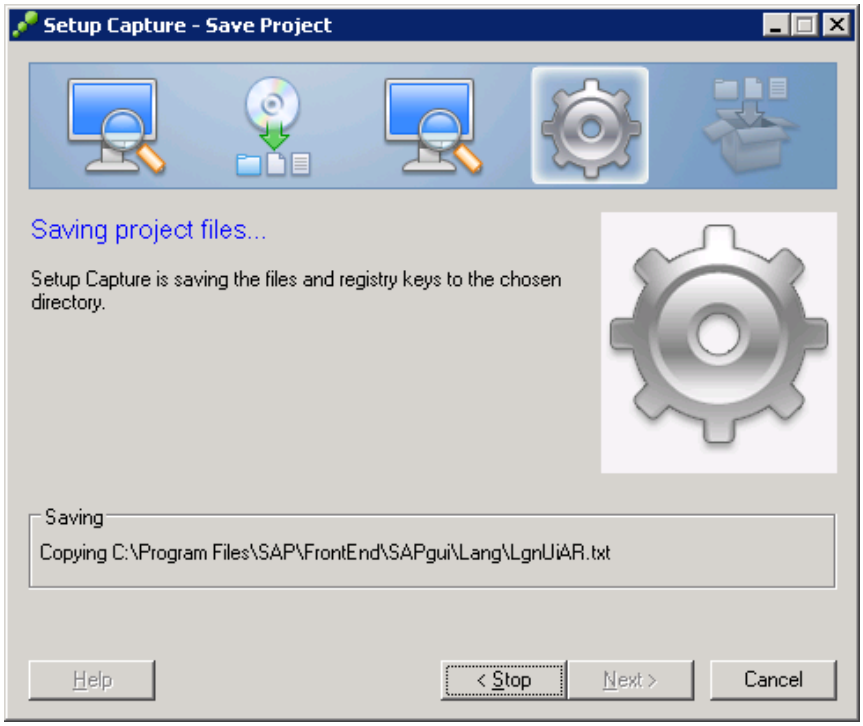
16. Select the primary data container from the list based on the selected entry points. This is the file that stores the virtual files and registry information,

    - If the size of the container is smaller than 200MB, ThinApp creates an `EXE` file as the primary container. For a small application such as Firefox, any `EXE` file can serve as the main data container.
    - If the size of the container is larger than 200MB (typical for a SAP GUI install), ThinApp creates a separate `DAT` file as the primary container because Microsoft Windows does not show shortcut icons stored in large `EXE` files. ThinApp generates small `EXE` files along with the `DAT` file to store the icons for Windows to display.

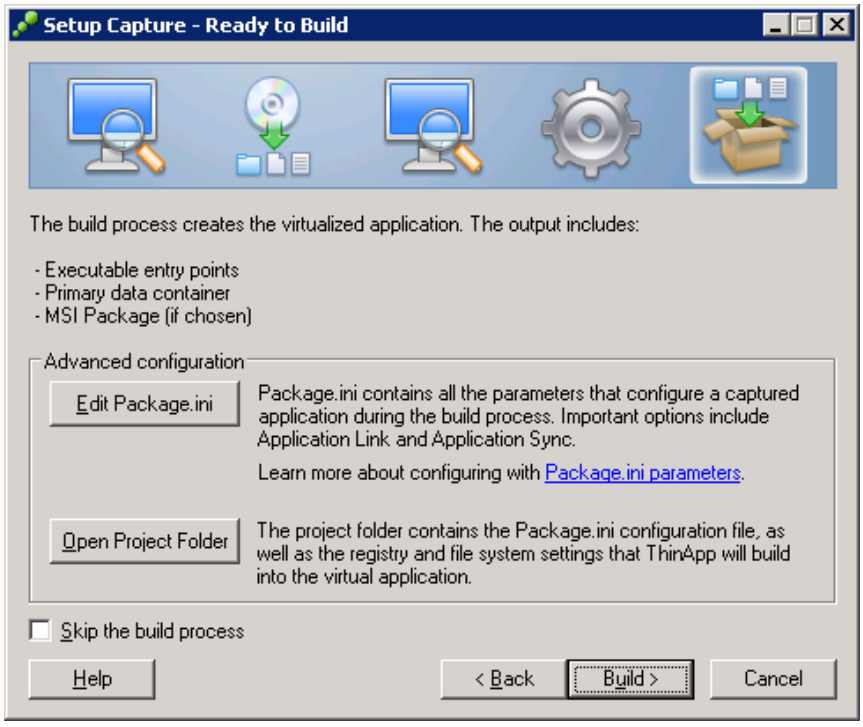
    In this example the default data container name is selected.



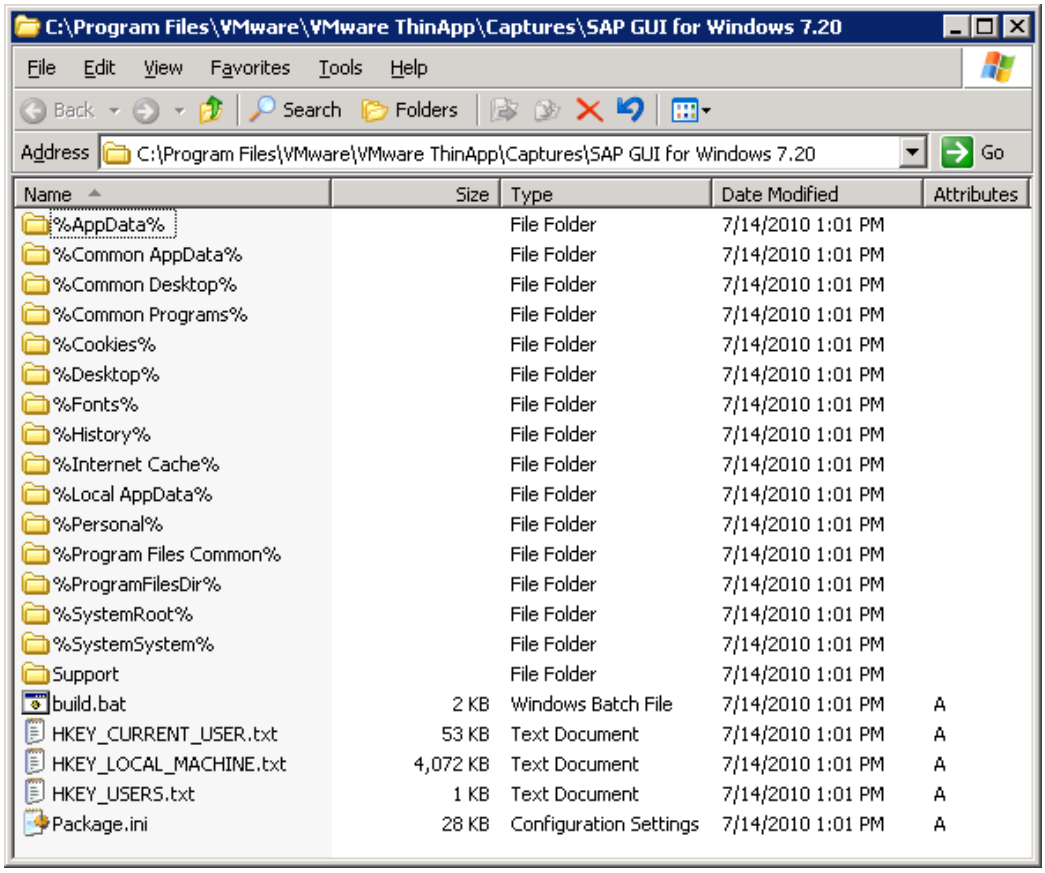
17. Click **Save**.



18. Click **Open project folder** to display the ThinApp project files in Windows Explorer.

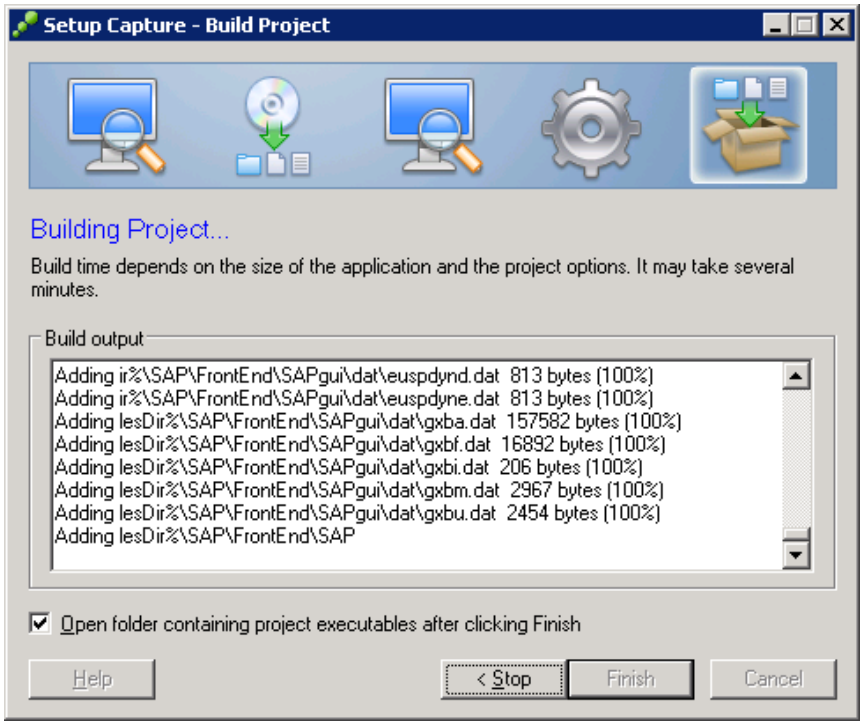


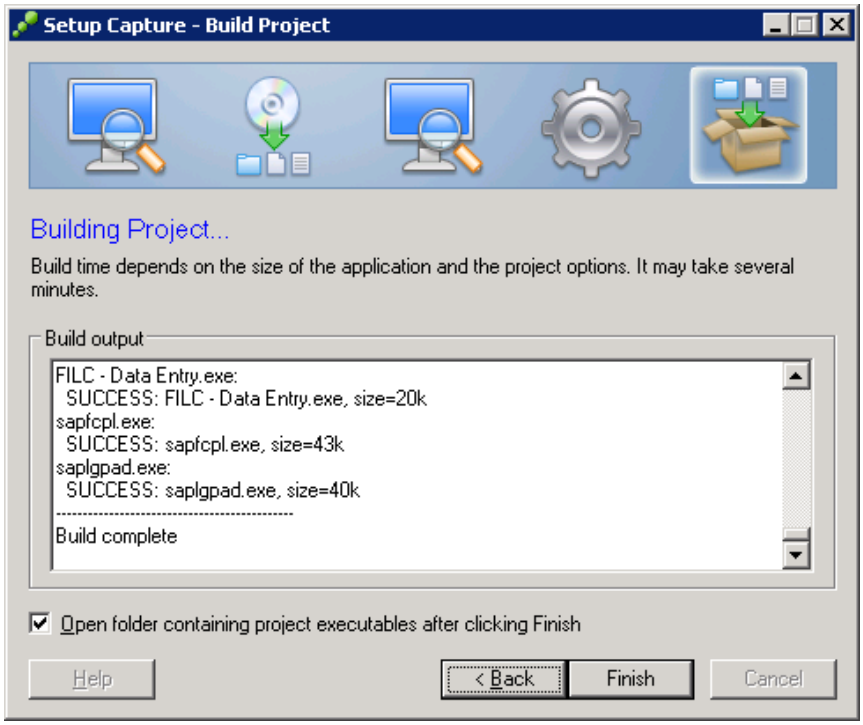
19. The project files are displayed.



Build ThinApp Package

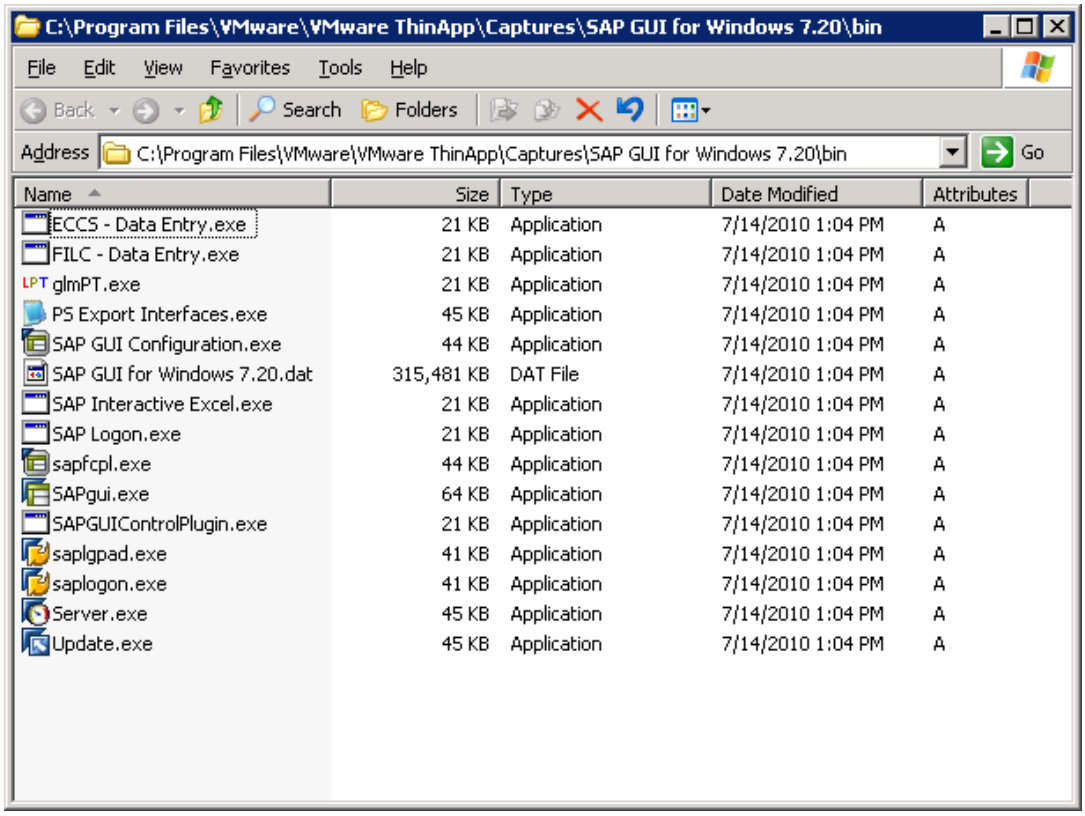
1. Click **Build**.





2. Click **Finish**.

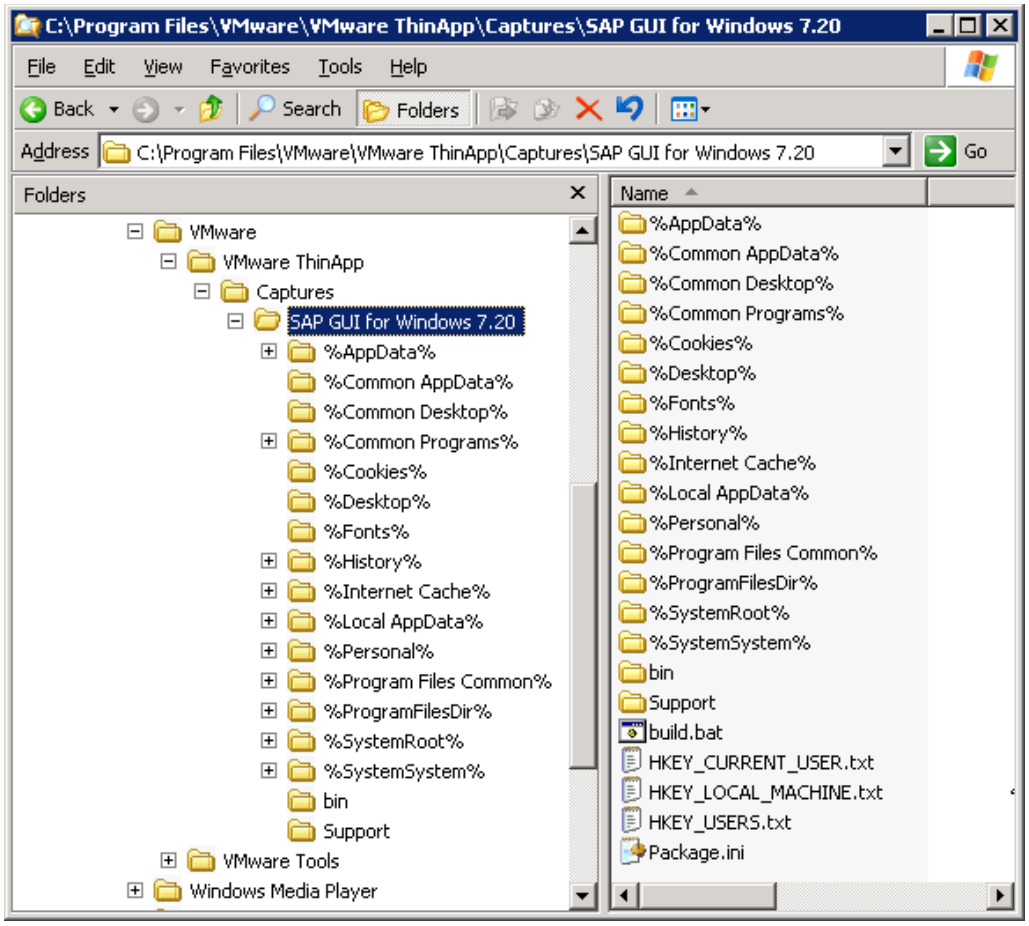
3. The SAP GUI ThinApp container/executable is displayed in Windows Explorer.



Summary of File Directory of Virtualized SAP GUI

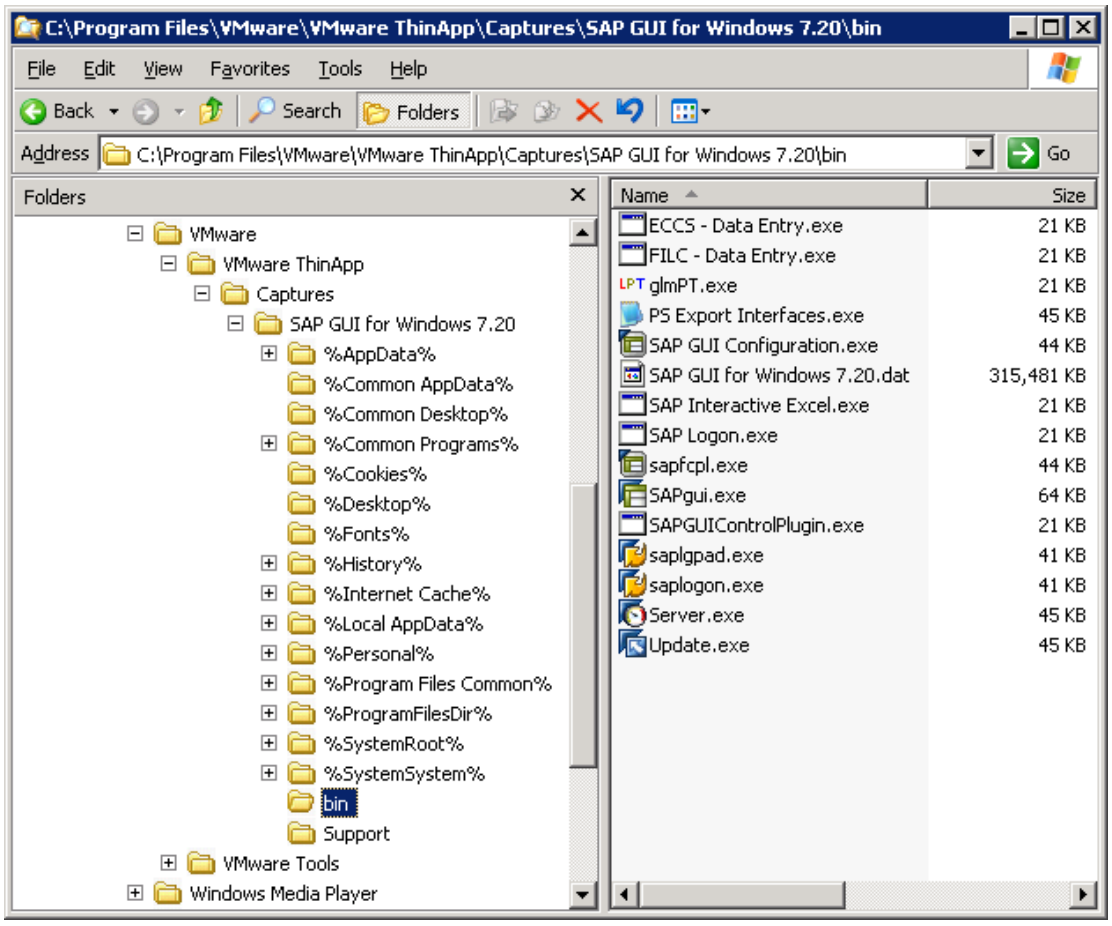
Project Folder

The project is located in `C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20`.



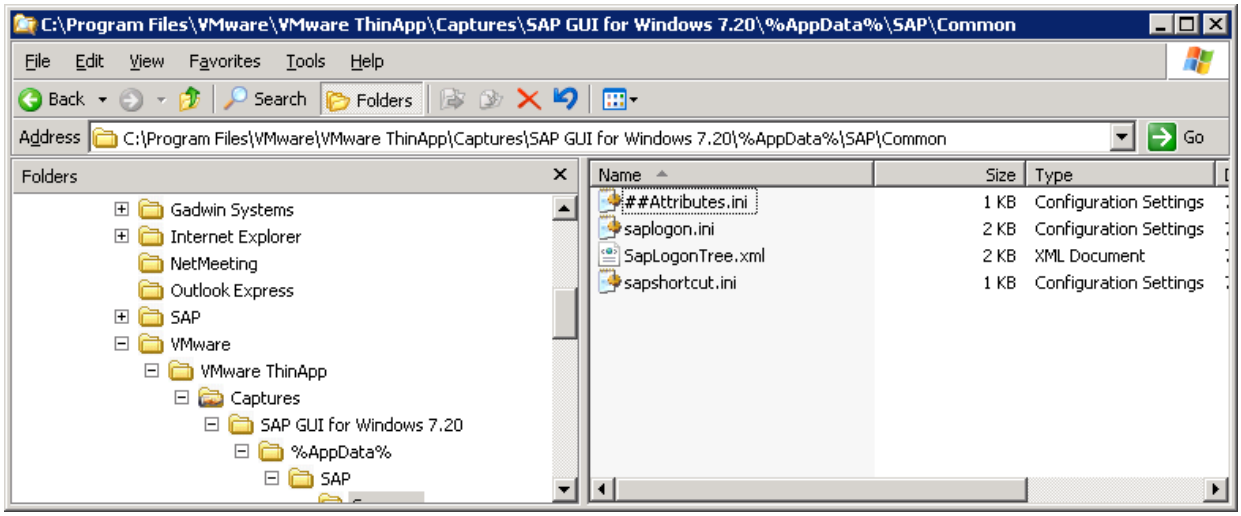
SAP GUI Application

The virtualized SAP GUI application is located in `C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20\bin`.



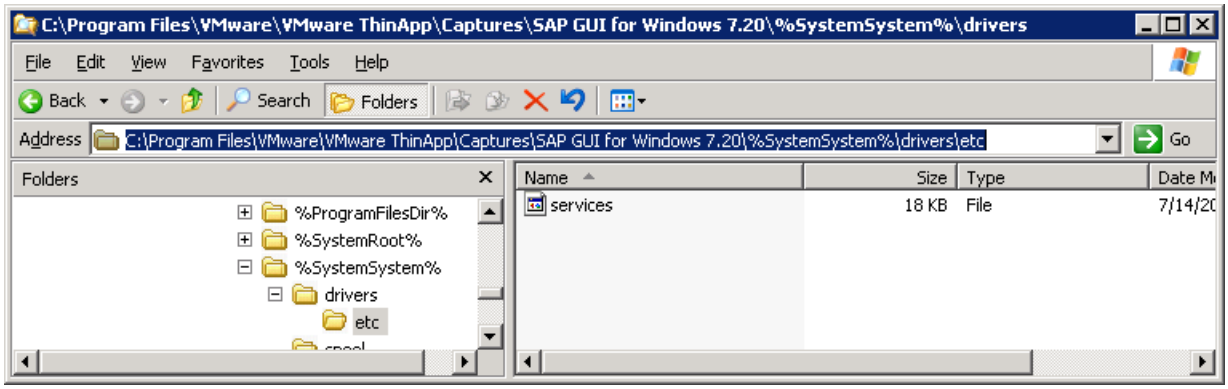
SAP Logon Configuration Files

Configuration files are located in `C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20\%AppData%\SAP\Common`.

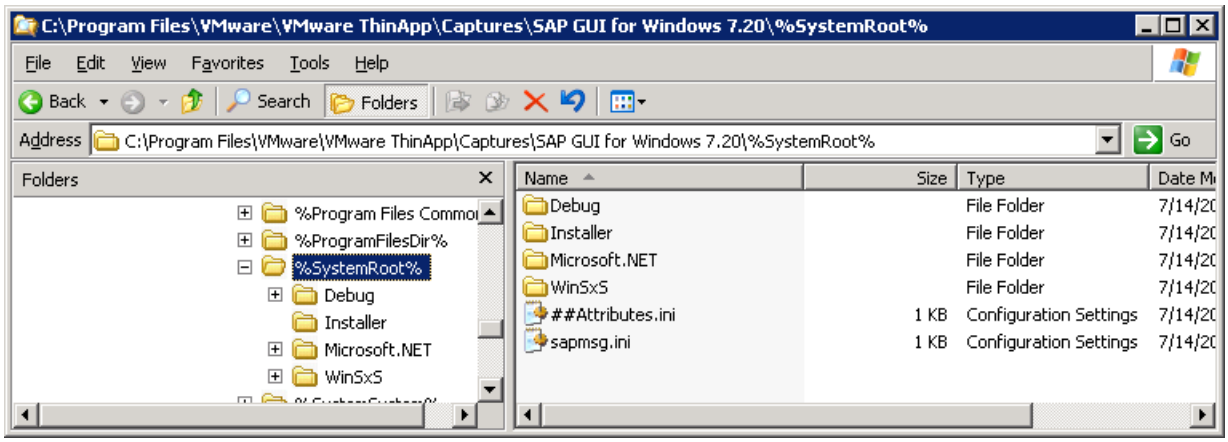


Services and sapmsg.ini

The `services` file is in `C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20\%SystemSystem%\drivers\etc`.



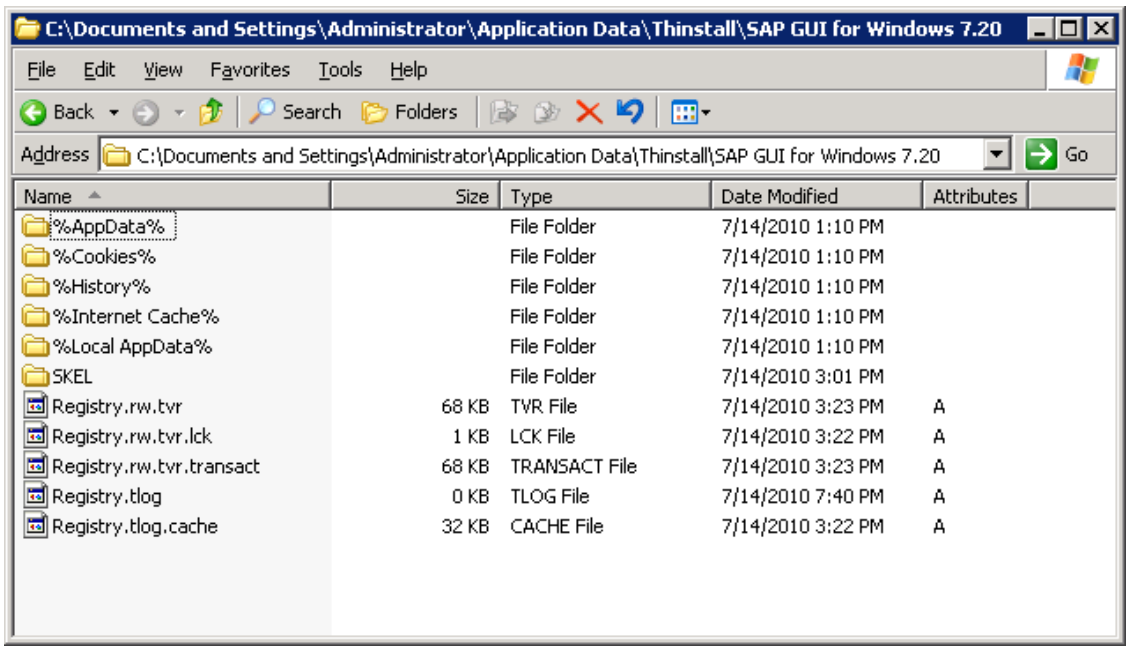
The file, `sapmsg.ini`, is located in `C:\Program Files\VMware\VMware ThinApp\Captures\SAP GUI for Windows 7.20\%SystemRoot%`.



In this example SAProuter information was not created, so the file `saproute.ini` is not displayed.

Sandbox

After the virtualized SAP GUI package has been deployed and executed on a target desktop, the sandbox is created. It is located at `C:\Documents and Settings\Administrator\Application Data\Thinstall\SAP GUI for Windows 7.20`. The sandbox directory structure is similar to the project.



Subsequent changes to SAP logon configuration files via `saplogonpad.exe` are captured in the sandbox at `C:\Documents and Settings\Administrator\Application Data\Thinstall\SAP GUI for Windows 7.20\%AppData%\SAP\Common`.

Appendix B: SAP GUI 7.2 Background

This appendix provides some background on SAP GUI 7.2. For more information, see the SAP notes identified in the Resources section.

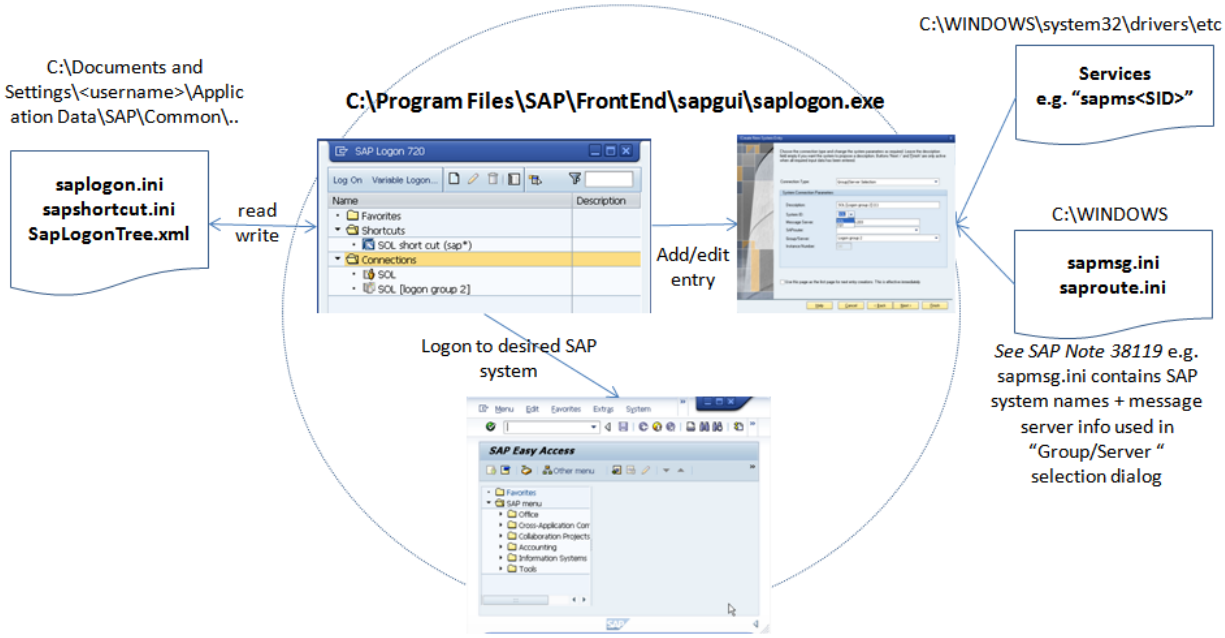
SAP logon involves running either `saplogon.exe` or `saplgpad.exe`. Both require the same set of configuration files. Execution generates a selection window that displays the different backend SAP systems—this is referred to as the *SAP logon selection list*. Double-clicking an entry opens the logon screen to the SAP system. The difference between the two executables is as follows:

- `saplogon.exe` – reads the configuration files, displays the logon selection list window and allows entries to be created, deleted and edited.
- `saplgpad.exe` – reads the configuration files and displays the same logon selection list, but does not allow any of the entries to be changed.

Desktop administrators roll out the appropriate executable to users depending on whether a user needs to edit the selection list. For example, Basis administrators typically need to edit the selection list, whereas business users are presented with a predefined selection list and do not need any edit capability.

The logon process and associated configuration files are illustrated in the next two sections.

Users Allowed to Edit Selection List – saplogon.exe



Users Not Allowed to Edit Selection List – saplgpad.exe