

## How Unattended Installation Works

### In this section

- [Unattended Installation Architecture](#)
- [Unattended Installation Physical Structures](#)
- [Unattended Installation Processes and Interactions](#)
- [Related Information](#)

Unattended installation is an automated installation technology that you can use to install an operating system with minimal administrator or technician intervention. Unattended installation is typically used during large-scale rollouts when it would be too slow and costly to have administrators or technicians interactively install the operating system on individual computers. It can also be used to upgrade operating systems.

One of the essential components of an unattended installation is an answer file that you create. An answer file is a simple text file that instructs the Windows Setup program how to install and configure the operating system. To perform an unattended installation, you run Setup by using a special command-line parameter that instructs Setup to read the answer file, or you start the destination computer with the product CD and then insert a floppy disk that contains an answer file. Then, during the setup process, instead of prompting you for installation and configuration information interactively, Setup follows the instructions contained in the answer file.

The following sections provide an in-depth view of how unattended installation works in an optimal environment. An optimal environment for unattended installation contains the following elements:

- Plug and Play hardware that is compatible with Windows Server 2003.
- High-bandwidth network connectivity between the computer on which the unattended installation is being performed (destination computer) and any installation file that resides in a shared folder ("distribution share").
- Physical access to the destination computer.
- BIOS and hardware configured so that you can start the computer from the product CD.
- BIOS and hardware configured so that you can start the computer from the floppy disk.

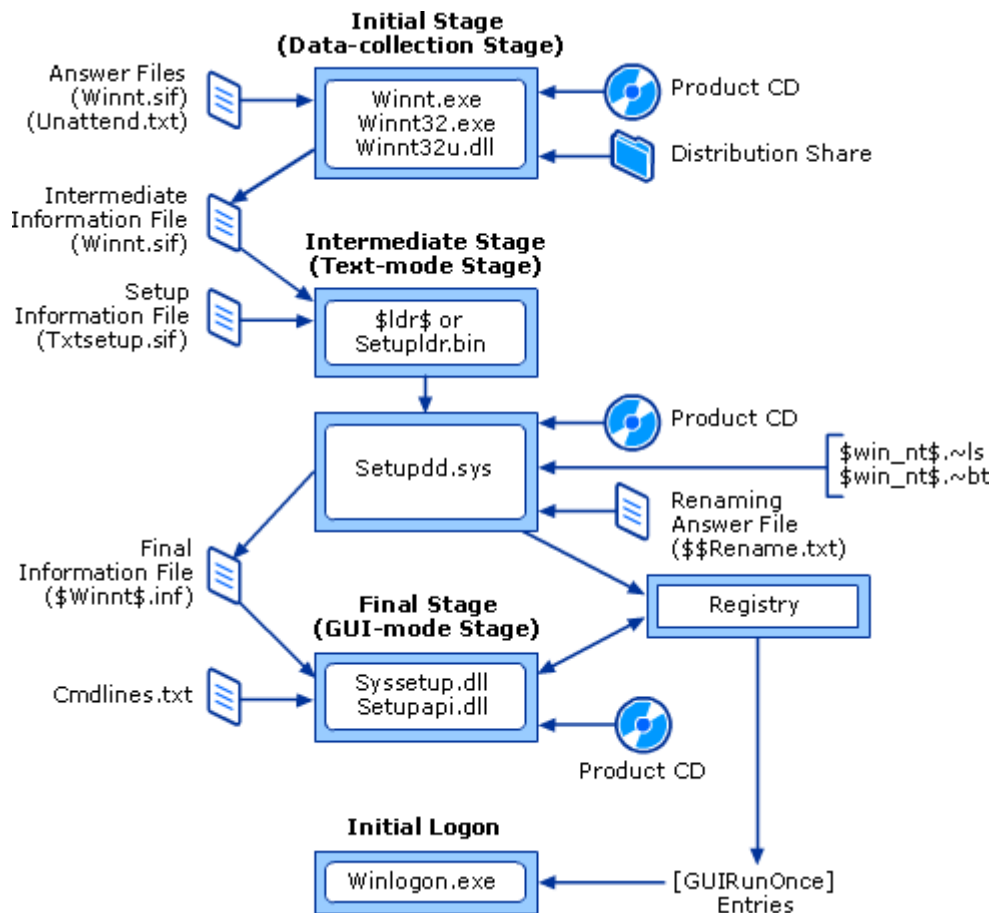
---

[Back to Top](#)

### Unattended Installation Architecture

Unattended installation relies on the same setup components as an interactive setup. These components are used in three distinct setup stages, including the initial stage (also known as data-collection stage), the intermediate stage (also known as text-mode stage), and the final stage (also known as GUI-mode stage). These three stages culminate with a final restart, after which a user logs onto the computer for the first time. In addition, unattended installation relies on several answer files and optional components, such as a distribution share, a product CD, and a renaming answer file to rename long file and folder names. How these components interact is illustrated in the following figure.

### Unattended Installation Architecture



The following are descriptions of the primary unattended installation components.

**\$ldr\$** The Setup loader, similar to `Ntldr`, which starts the destination computer in a special setup state during which the intermediate stage of unattended installation occurs. This file is named `Setupldr.bin` on the product CD. `Setupldr.bin` is renamed `$ldr$` and copied to the destination computer's hard disk when you do not start the destination computer from the product CD.

**\$win\_nt\$.~ls** Temporary installation folder created on the hard disk of the destination computer that contains the files and folders that are copied from the `I386` folder in the distribution share.

**\$win\_nt\$.~bt** Temporary installation folder created on the hard disk of the destination computer that contains temporary startup files. The startup files are used to restart the computer into a special setup state that allows the intermediate stage of unattended installation to occur.

**\$Winnt\$.inf** Final version of the unattended installation answer file that Setup creates and saves in the `systemroot\System32` folder on the destination computer. Setup makes this a hidden file and names it `$winnt$.inf`, regardless of what the name of the original answer file is.

**\$\$Rename.txt** A file that lists all files contained in a particular folder that need to be renamed from an 8.3 format name to a long file name during the setup process. All file names must be in 8.3 format if you run Setup by using `Winnt.exe`, or if you copy a set of files from one computer to another using MS-DOS.

**Cmdlines.txt** A configurable text file that you use to customize an unattended installation. `Cmdlines.txt` contains a list of commands that run synchronously after Setup finishes, but before a computer restarts.

**Destination computer** Computer on which you are installing an operating system. During a large-scale rollout, there can be typically hundreds or thousands of destination computers.

**Distribution share (I386\)** Hierarchical folder structure that is stored in a shared folder on a remote computer (usually a server). The distribution share contains all of the Windows Server 2003 installation files that are necessary to perform an unattended installation. You can use `Setupmgr.exe` to create a distribution folder, or you can create one manually by using Windows Explorer.

**[GUIRunOnce]** A section in an unattended installation answer file that contains a list of commands, programs, scripts, or batch files that run the first time a user logs on to the computer after the operating system is installed.

**Setupapi.dll** Dynamic-link library that is responsible for installing hardware.

**Setupdd.sys** System file responsible for performing tasks during the intermediate stage of unattended installation, such as creating or modifying the system folder (usually, C:\Windows or C:\Winnt), copying system files and folders from the temporary folder or the product CD to the system folder, and building the registry. It is also responsible for creating partitions and formatting disks.

**Setupldr.bin** The Setup loader, similar to Ntldr, which starts the destination computer in a special setup state during which the intermediate stage of unattended installation occurs. Setupldr.bin is renamed \$ldr\$ and copied to the destination computer's hard disk if you do not start the destination computer from the product CD.

**Syssetup.dll** Dynamic-link library that performs installation and configuration tasks during the final stage of unattended installation. This component might have to access the CD in order to install optional components.

**Technician computer** Computer used by an administrator or technician to use Setup Manager or Notepad to create and modify answer files, such as Unattend.txt or Cmdlines.txt.

**Txtsetup.sif** System information file that Setup uses during the setup process. Txtsetup.sif defines the names of installation files and folders, where files and folders are copied, and how to process existing files and folders (for example, delete or overwrite). Txtsetup.sif is saved in the I386 folder of the distribution share and resides in the I386 folder of every product CD.

**Unattend.txt** Default name for an unattended installation answer file. You can name an answer file anything you want, but Unattend.txt is the name that Setupmgr.exe uses when you first create an answer file.

**Winlogon.exe** A component of the Windows operating system that provides interactive logon support. Winlogon is designed around an interactive logon model that consists of three components: the Winlogon executable, a Graphical Identification and Authentication dynamic-link library (DLL) referred to as the GINA, and many network providers.

**Winnt32.exe** The 32-bit and 64-bit Setup program. You can run the 32-bit version of this program on Windows 95, Windows 98, Windows Millennium Edition, Windows NT, Windows 2000, Windows XP, or a product in the Windows Server 2003 family. The 32-bit version of this program can be found in the I386 folder on any 32-bit Windows operating system product CD. You can run the 64-bit version of this program on Windows XP 64-bit Edition, the 64-bit version of Windows Server 2003, Enterprise Edition, and the 64-bit version of Windows Server 2003, Datacenter Edition. The 64-bit version of this program can be found in the IA64 folder on any 64-bit operating system product CD.

**Winnt32u.dll** Dynamic-link libraries that perform installation tasks during the initial stage of unattended installation. These include assessing the current operating system that is installed on the destination computer, copying installation files and folders from the distribution share to the temporary folders, parsing the answer file, and creating intermediate information files that are used by the intermediate stage of unattended installation.

**Winnt.exe** The 16-bit Setup program. You can use this program to start an unattended installation from any 16-bit operating system, such as MS-DOS.

**Winnt.sif** Name you must use for the answer file if you are performing an unattended installation by starting a destination computer from the Windows Server 2003 product CD. Winnt.sif is also the name of an intermediate information file that Setup creates after the initial stage of unattended installation. This intermediate file contains all of the information that is in the answer file (regardless what the name of the answer file is), as well as information about the destination computer and the type of installation you are performing.

---

[Back to Top](#)

## Unattended Installation Physical Structures

Two physical structures are essential to an unattended installation: the distribution share and the answer file. You can create these structures by using Setup Manager (Setupmgr.exe) or you can create them manually by using Windows Explorer and a text editor, such as Notepad. These structures are illustrated in the following figure and are described in the sections that follow.

### Unattended Installation Physical Structures



automatically copy folders, standard 8.3 format files, and any tools needed for your automated installation process to the \$OEM\$ subfolder. However, the contents of this folder are copied only when you include the following setting in your answer file:

```
[Unattended]
OemPreInstall = Yes
```

If you want to create the \$OEM\$ subfolder in a different location than the distribution share, you can use the OemFilesPath parameter in the answer file to specify a different location.

The \$OEM\$ subfolder can contain the optional file Cmdlines.txt, which contains a list of commands to be run at the end of the final stage of unattended installation. These commands can be used to install additional services or applications that you want to include with your installation.

As long as Setup finds the \$OEM\$ subfolder in the root of the distribution share, Setup copies all the files found in this folder to a temporary folder that is created during the initial stage of unattended installation on the destination computer.

**\$OEM\$\Textmode** The \$OEM\$\Textmode subfolder contains new or updated files for installing mass storage device drivers and hardware abstraction layers (HALs). These files can include an OEM HAL, drivers for a small computer system interface (SCSI) device, and a Txtsetup.oem file, which directs the loading and installing of these components. Be sure to include the Txtsetup.oem file in this folder: most device drivers cannot be installed without a Txtsetup.oem file. All files placed in the \$OEM\$\Textmode subfolder (HALs, drivers, and Txtsetup.oem) must be listed in the [OEMBootFiles] section of the answer file.

**\$OEM\$\\$\$** The \$OEM\$\\$\$ subfolder is equivalent to the systemroot or windir environment variable. The subfolder contains additional files that need to be copied to the various subfolders of the Windows Server 2003 or Windows XP Professional installation directory. The structure of this subfolder must match the structure of a standard Windows Server 2003 or Windows XP Professional installation, where \$OEM\$\\$\$ matches systemroot or windir (for example, C:\Windows), \$OEM\$\\$\$\Fonts matches *systemroot*\Fonts, and so on. Each subfolder must contain the files that need to be copied to the corresponding system folder on the destination computer.

**\$OEM\$\\$\$\Help** A subfolder that contains the OEM Help files to be copied to the *systemroot*\Help folder during the setup process.

**\$OEM\$\\$\$\System32** A subfolder that contains files to be copied to the *systemroot*\System32 folder during the setup process.

**\$OEM\$\\$1** This folder points to the drive on which Windows Server 2003 or Windows XP Professional is installed; \$1 is equivalent to the systemdrive environment variable. For example, if you are installing Windows Server 2003 on drive D, \$OEM\$\\$1 points to drive D.

**\$OEM\$\\$1\Pnpdrvrs** You can use the \$OEM\$\\$1\Pnpdrvrs subfolder to place new or updated Plug and Play device drivers in your distribution shares. These folders are copied to the *systemdrive*\Pnpdrvrs location on the destination computer. Adding the OemPnPDriversPath parameter to your answer file directs Windows Server 2003 or Windows XP Professional to look (both during and after the setup process) for new or updated Plug and Play drivers in the folders that you created, in addition to those originally included with the system. Note that you can replace the file name Pnpdrvrs with any name that is eight or fewer characters long.

**\$OEM\$\\$1\Sysprep** The \$OEM\$\\$1\Sysprep subfolder is optional. This subfolder contains the files that you need to run the System Preparation (Sysprep.exe) tool.

**\$OEM\$\drive\_letter** During the intermediate stage of unattended installation, the structure and contents of each \$OEM\$\drive\_letter subfolder are copied to the root of the corresponding drive on the destination computer. For example, files that you place in the \$OEM\$\D subfolder are copied to the root of drive D. You can also create subfolders within these subfolders. For example, \$OEM\$\E\Misc causes Setup to create a subfolder called Misc on drive E.

### Converting Short File Names to Long File Names by Using \$\$Rename.txt

If you are using Winnt.exe to perform an unattended installation, your distribution share can contain only files and folders that have short file names. This is because Winnt.exe runs only on MS-DOS, and MS-DOS can process only files and folders that use the 8.3 naming convention. The 8.3 naming convention allows only eight characters to the left of the decimal point and three characters to the right of the decimal point. Because of the short file name limitation, you need to shorten the names of files and folders so that you can put them into your distribution share. You can convert these short file and folder names back to long file and folder names during the setup process by using the \$\$Rename.txt file.

\$\$Rename.txt lists all of the files and folders that need to be renamed. You need only one \$\$Rename.txt file to rename files in an entire distribution share. If you are using Winn32.exe to perform an unattended installation, short file names are automatically converted to long file names during the setup process.

To use \$\$Rename.txt, put the file in the \$1 folder in your distribution share. Saving the \$\$Rename.txt file in this

location will ensure that the \$\$Rename.txt file is copied to the systemdrive on the destination computer. Setup automatically looks for \$\$Rename.txt; if it finds a \$\$Rename.txt file, Setup renames the files and folders that are in your distribution share. The syntax for \$\$Rename.txt is:

```
[section_name_1]
short_name_2 = "long_name_2"
short_name_1 = "long_name_1"
short_name_x = "long_name_x"
.
.
.
[section_name_2]
short_name_1 = "long_name_1"
short_name_2 = "long_name_2"
short_name_x = "long_name_x"
.
.
.
```

Where:

- *section\_name\_x* is the path to the folder that contains the files and folders. A section does not need to be named; it can have a backslash (\) as a name, which indicates that the section contains the names of the files or subfolders that are in the root of the drive.
- *short\_name\_x* is the name of the file or folder within this named folder that needs to be renamed. The name *must not* be enclosed in quotation marks.
- *long\_name\_x* is the new name of the file or folder. If the name contains spaces or commas, it *must* be enclosed in quotation marks.

### Answer File Structure

The answer file for unattended installations of Windows Server 2003 is usually named Unattend.txt, but you can name it anything you want. However, if you start a destination computer from a product CD and the answer file is on a floppy disk, you must name the answer file Winnt.sif. Setup will then detect the answer file on the disk without user input.

The answer file supplies Setup with answers to all the questions that an administrator or technician would be asked during a standard, interactive setup. The answer file also contains information about your installation and configuration requirements. In addition, an answer file tells Setup how to interact with the distribution shares and files that you have created (or the installation files on the product CD).

An answer file contains multiple sections — some mandatory and some optional — that you can modify. Section names are always enclosed in square brackets (for example, [Unattended]).

Every section has one or more entries that contain installation and configuration information. Entries are usually composed of an entry name, an equal sign, and a value (for example, **ComputerName = Computer1**). The entry name represents a specific computer setting or action; the value represents the unique way you want the setting configured or the action performed. In some cases, entries do not have an entry name. In these cases, a value or a list of values (usually commands) appear in the answer file below a section name. The following example shows a section that contains only values and does not have entry names.

```
[GUIRunOnce]
dcpromo /answer:a:\installdc.txt
%systemroot%\cluster\cluster.exe
```

Use the following guidelines when you create or modify an answer file:

- Sections do not have to be any particular order.
- Sections and entries are not case sensitive.
- Entries must have valid values. For example, "on" is not equivalent to "enable," and "off" is not equivalent to "disable." Invalid sections and entries can generate errors, or cause Setup to fail.
- Each entry must have a value.
- Each section and entry must appear on a separate line.
- Comments must begin with a semi-colon (;). Do not use "REM" to specify a comment.

### ◆ Important

- Do not modify undocumented sections or entries in an answer file; doing so can cause unpredictable results.

The following table lists all of the valid sections that can appear in an answer file for an unattended installation.

#### Answer File Sections for Unattended Installation

Section	Description
<i>[adapter_specific_protocol_section]</i>	Contains an entry for specifying a network component. Must be used in conjunction with the [Networking] section.
[Branding]	Contains entries for branding Microsoft Internet Explorer during an unattended installation. If you do not specify this section, Setup uses the default for Windows Server 2003.
[CertSrv_Client]	Contains entries for the Web client component of Certificate Services. You do not need to include this section in the answer file to install the Web client component of Certificate Services on the computer. Enabling Certificate Services in the [Components] section of the answer file installs the necessary files.
[CertSrv_Server]	Contains entries for specifying the server components of Certificate Services. The entries in this section are supported only for the Windows Server 2003.
[Components]	Contains entries for installing the optional components of Windows XP Professional and Windows Server 2003.
[Data]	Contains entries for starting directly from the Windows Server 2003 product CD when performing an unattended installation.
[DCInstall]	Contains entries for installing a domain controller after the initial setup of the operating system finishes.
[Display]	Contains entries for specifying display settings for graphics devices.
[FavoritesEx]	Contains the default Favorites settings for the Windows browser. If not present, Setup uses the default Uniform Resource Locator (URL) for Windows XP or Windows Server 2003.
[Fax]	Contains entries for specifying custom installation information for the Fax service.
[GuiRunOnce]	Contains commands that run the first time a user logs on to the computer after final (GUI-mode) stage of Setup completes.
[GuiUnattended]	Contains entries for preparing the graphical user interface (GUI) for unattended installation.
[Homenet]	Contains entries for installing Home Networking settings for network adapters, Internet Connection Sharing (ICS), and Internet Connection Firewall (ICF).
[Identification]	Contains entries for specifying the network identification of a computer. If these entries are not present, Setup adds the computer to the default workgroup. If there is not enough information for this entry, Setup prompts the user to provide this information.
[IEHardening]	Contains entries for configuring Web content zones.
[InternetServer]	Contains entries for configuring the installation folder for Internet Information Services (IIS) components.
[LicenseFilePrintData]	Contains entries for installing licensing information for Windows Server 2003.
[MassStorageDrivers]	Contains an entry for loading retail or original equipment manufacturer (OEM) small computer system interface (SCSI) drivers in the Setup Loader during the intermediate (text-mode) stage of Setup
<i>[MS_AppleTalk parameters]</i>	Contains entries for configuring AppleTalk.
<i>[MS_ATMArps parameters]</i>	Contains no entries.
<i>[MS_ATMLANE parameters]</i>	Contains no entries.
<i>[MS_ATMUni parameters]</i>	Contains no entries.

[MS_L2TP parameters]	Contains entries for the Layer 2 Tunneling Protocol (L2TP).
[MS_MSClient parameters]	Contains entries for Client for Microsoft Networks.
[MS_NetMon parameters]	Contains no entries.
[MS_NWClient parameters]	Contains entries for Client Service for NetWare.
[MS_NWIPX parameters]	Contains entries for the Internetwork Packet Exchange (IPX) protocol. Its parameters are network adapter-independent.
[MS_NwSapAgent parameters]	Contains no entries.
[MS_PPTP parameters]	Contains entries for the Point-to-Point Tunneling Protocol (PPTP).
[MS_Psched parameters]	Contains no entries.
[MS_RAS parameters]	Contains entries for the Routing and Remote Access service.
[MS_RasSrv parameters]	Contains entries for the Routing and Remote Access service for remote access servers.
[MS_Server parameters]	Contains entries for the File and Print Service.
[MS_TCPIP parameters]	Contains entries for configuring the TCP/IP protocol.
[MS_WLBS parameters]	Contains entries for configuring Network Load Balancing (formerly, Windows Load Balancing Service, or WLBS) and binding Network Load Balancing to multiple network adapters.
[NetAdapters]	Contains entries for network adapters that you want to install.
[NetBindings]	Contains entries for disabling, enabling, and demoting binding rules for network components such as services, protocols, and adapters.
[NetClients]	Contains an entry for specifying the network clients to install.
[NetOptionalComponents]	Contains entries for installing optional network components.
[NetProtocols]	Contains an entry for installing the network protocols.
[NetServices]	Contains entries for installing network services.
[Networking]	Contains no entries. However, to configure network protocols in unattended installation, you must include the [Networking] section name in your answer file.
[OEMBootFiles]	Contains entries for specifying OEM-supplied boot files.
[params.adapter_instance]	Contains entries for specifying an adapter's instance on a computer by using either PCI location information or a Plug and Play device ID.
[PCHHealth]	Contains entries for configuring error reporting, Remote Assistance, and Help and Support Center.
[protocol_parameters_section]	Contains an entry for specifying network protocol parameters that are also specified in the [NetProtocols] section.
[Proxy]	Contains entries for specifying proxy server settings for the Windows XP or Windows Server 2003 browser.
[RegionalSettings]	Contains entries for installing regional options during the setup process.
[SetupMgr]	Contains entries for specifying the server and folder in which the distribution share resides and the name of the distribution share. This section is used only by Setup Manager when an existing distribution share is edited. It is not used by Setup during an unattended installation.
[SetupParams]	Contains an entry for running an additional post-Setup command.
[Shell]	Contains entries for specifying the visual style of the user interface.
[SNMP]	Contains entries for specifying the Simple Network Management Protocol (SNMP) service.
[SystemFileProtection]	Contains entries for specifying Windows File Protection (WFP) settings.
[SystemRestore]	Contains entries for specifying the System Restore feature.
[TapiLocation]	Contains entries for specifying the telephony API (TAPI). It is valid only if a modem is present on the computer.
[TerminalServices]	Contains entries for configuring Terminal Services features (Terminal



	Server and Remote Desktop).
[Unattended]	Contains entries for running Setup.
[Uninstall]	Contains entries for allowing users who upgrade to Microsoft Windows XP to remove the operating system if they encounter problems.
[URL]	Contains entries for specifying default Uniform Resource Locator (URL) settings for the browser available in Windows XP or Windows Server 2003.
[UserData]	Contains entries for specifying user settings during the setup process.
[WindowsFirewall]	This section must be included in your answer file to configure firewall sections and entries in unattended installations. To configure Windows Firewall settings, you must include the [WindowsFirewall] section along with one or more of the other Windows Firewall settings sections.
[WindowsFirewall. <i>profile_name</i> ]	Contains entries for specifying a domain type of profile, a standard type, or a profile that applies to both domain and standard.
[WindowsFirewall. <i>program_name</i> ]	Contains entries for adding programs to the Windows Firewall Exceptions list.
[WindowsFirewall. <i>service_name</i> ]	Contains entries for adding services to the Windows Firewall Exceptions list.
[WindowsFirewall. <i>portopening_name</i> ]	Contains entries for adding static ports to the Windows Firewall Exceptions list.
[WindowsFirewall. <i>icmpsetting_name</i> ]	The default configuration for Windows Firewall blocks all Internet Control Message Protocol (ICMP) message types, the entries in this section allows you to modify this behavior.
[Win9xUppg]	Contains entries for specifying an unattended upgrade from Windows 98, Windows 98 Second Edition, or Windows Millennium Edition to either Windows XP Home Edition or Windows XP Professional.

You do not need to add every possible section and entry to an answer file. Most sections and entries are optional, and Setup ignores optional sections and entries that are not present in an answer file.

### Entries for Fully-Automated Unattended Installation

If you want to perform a fully-automated unattended installation, you must configure the following sections and entries in your answer file. If you do not provide values for these entries in your answer file, Setup will display a prompt asking for the values.

**[GuiUnattended]** You must specify values for the AdminPassword and TimeZone entries. The value for AdminPassword cannot begin with an asterisk (\*). Using a password that begins with an asterisk can cause the password to be set to a null value, which can be a security risk.

**[Identification]** You must specify values for the JoinDomain, DomainAdmin, and DomainAdminPassword entries. However, if you are concerned about security you might want to leave the DomainAdminPassword blank because this entry cannot be encrypted in the answer file. If you leave the DomainAdminPassword entry blank, you will be prompted for the password during the unattended installation.

**[LicenseFilePrintData]** You must specify values for the AutoMode and AutoUsers entries if you are installing Windows Server 2003.

**[Networking]** If your destination computer requires network connectivity, you must specify values for various network protocol entries.

**[Unattended]** You must specify values for the UnattendMode and TargetPath entries.

**[UserData]** You must specify values for the FullName and ComputerName entries.

### Entries for Unattended Upgrade

If you are performing an unattended upgrade, you can use only the following answer file sections and entries. All other sections and entries are ignored during an unattended upgrade.

**[Unattended]** You must specify values for either the NtUpgrade or the Win9xUpgrade entry. You must use the entry OEMPreinstall = No, which will prevent the files and folders in the \$OEM\$ folder in your distribution share from being copied to the destination computer. You can also set values for the AutoActivate, DUDisable, DUShare, and DUStopOnError entries.

**[UserData]** You can set a value for the ProductKey entry. If you do not, the administrator or technician will be prompted for a product key.

In addition, your answer file must reside on a floppy disk and you must use the product CD instead of a distribution share.

For more information about answer file sections, including descriptions and examples of entries and values, see "[Unattended Installation Tools and Settings](#)."

[Back to Top](#)

## Unattended Installation Processes and Interactions

Because there are numerous different ways to perform an unattended installation, it is impractical to describe all of the processes and interactions that occur for each method and variation. Therefore, this section describes the processes and interactions that occur during the two most commonly used methods of performing an unattended installation: a fully-automated unattended installation when you start the destination computer from a 32-bit operating system, and a fully-automated unattended installation when you start the destination computer from the product CD.

In addition, this section describes log file interactions that occur during an unattended installation as well as the processes that occur when you perform post-installation tasks by using a Cmdlines.txt file or the [GUIRunOnce] section of an answer file.

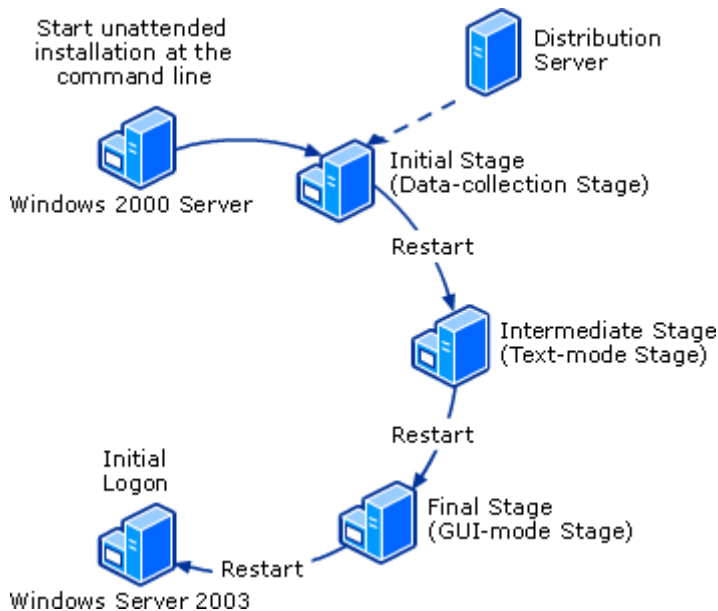
### Unattended Installation When You Start the Computer from a 32-bit Operating System

The following figure shows the processes and interactions that occur during a fully-automated unattended installation of Windows Server 2003 when you start the destination computer from a 32-bit operating system and you copy the installation files from a distribution share. The 32-bit operating system can be any of the following: Windows XP, Windows NT, Windows 2000, and Windows Server 2003. The distribution share can be saved on any accessible server that you can secure with file and folder permissions.

#### Note

- You can also copy the installation files from the product CD; however, it is more common to use a distribution share when you start the destination computer from a 32-bit operating system.

### Unattended Installation from a 32-bit Operating System Using a Distribution Share



This figure is based on the following requirements:

- The destination computer is running Windows 2000 Server (or an equivalent 32-bit Windows operating system).
- The distribution share is named Win2003\_Std and is saved in the OS\_Shares folder on the distribution server.

- The answer file is named Unattend.txt and is saved in the distribution share.
- All of the installation files, including Winnt32.exe, are saved in the I386 folder in the distribution share.
- The unattended installation is a clean installation of the operating system (not an upgrade).
- The unattended installation is started from the existing Windows 2000 Server operating system by running Winnt32.exe at the command line.

In addition, the following answer file is used to perform this installation:

```
[Unattended]
  UnattendMode=FullUnattended
  OemSkipEula=Yes
  OemPreinstall=Yes
  FileSystem = LeaveAlone
  TargetPath= *

[GuiUnattended]
  AdminPassword=ca8d43d0eae56d1019ac6cc7a3608820ae698f3d8ce553674c94fb185
  EncryptedAdminPassword=Yes
  OEMSkipRegional=1
  TimeZone=004
  OemSkipWelcome=1

[UserData]
  ProductKey=#####-#####-#####-#####-#####
  FullName="My Name"
  OrgName="Microsoft Corporation"
  ComputerName=Server2

[Display]
  BitsPerPel=24
  Xresolution=800
  YResolution=600
  Vrefresh=70

[LicenseFilePrintData]
  AutoMode=PerServer
  AutoUsers=5

[TapiLocation]
  CountryCode=1
  Dialing=Tone
  AreaCode=425
  LongDistanceAccess="9"

[SetupMgr]
  DistFolder=\\Server1\OS_Shares
  DistShare=Win2003_Std

[Identification]
  JoinDomain=MyDomain
  DomainAdmin=MyAdmin
  DomainAdminPassword=MyPassword

[Networking]
  InstallDefaultComponents=Yes
```

### Starting the Unattended Installation

The processes and interactions described in this section are valid only when you start the installation by mapping a network drive to the distribution share from the destination computer, navigating to the distribution share, and running Winnt32.exe with the `/unattend` parameter. You need to map a network drive to the distribution share because all of the installation files, including Winnt32.exe, and the answer file, Unattend.txt, are saved in the distribution share.

## Initial Stage

After you start an unattended installation from a 32-bit operating system, the initial stage (also known as data-collection stage) of unattended installation begins. The following actions occur during the initial stage of an unattended installation:

**Assess current operating system** The current operating system configuration is assessed, including operating system version, available disk space, hardware, and installed components.

**Process answer file** The answer file is parsed and the results are saved in a new file named Winnt.sif. This file contains all of the settings that are in the answer file as well as other information that was gathered during the system assessment.

**Begin Welcome to Windows Setup** The Welcome to Windows Setup dialog boxes appear briefly and then close. During an interactive setup this sequence of dialog boxes prompts you for several configuration settings, including installation type (upgrade or clean installation), product key, acceptance of the End-User License Agreement (EULA), installation folder, location of installation files, language options, accessibility options, and dynamic update options. Because this information is specified in the answer file, the introductory dialog box stays open for only a few seconds and then closes; other dialog boxes are not displayed.

**Copy files to temporary folders** During the initial stage, Setup creates two temporary folders on the destination computer's hard disk — \$win\_nt\$.~ls and \$win\_nt\$.~bt. The \$win\_nt\$.~ls folder (known as the local source folder) contains the installation files, which are copied from the \$OEM\$ folder in the distribution share.

The \$win\_nt\$.~bt folder (known as the boot folder) contains all of the files that are necessary for starting the computer and initiating the setup process after all of the installation files are copied to the destination computer's hard disk. These files are copied from the distribution share and include basic device drivers, such as mouse, keyboard, mass storage, and video drivers. The \$win\_nt\$.~bt folder also contains Txtsetup.sif and Winnt.sif. Txtsetup.sif contains definitions and processing instructions for the files and folders that are installed during the setup process. Winnt.sif contains all of the answer file settings as well as other information that was gathered during system assessment.

A file known as \$ldr\$ is also created during the initial stage. This file is similar to Ntldr and is used to start the computer in a special setup state during which the intermediate stage of unattended installation occurs (this stage is also known as the text-mode stage).

**Load boot sector and change boot.ini** The partition boot sector is changed so that it contains the master boot code for starting the computer with \$ldr\$. The Boot.ini file is also modified so that the computer uses the startup files in the \$Win\_nt\$.~bt folder, which start the computer in a special setup state during which the intermediate stage of Setup occurs.

At the end of initial stage, when all of the installation files have been copied to the destination computer's hard disk, the destination computer restarts and the intermediate stage of the unattended installation begins.

## Intermediate Stage

The following actions occur during the intermediate stage (also known as the text-mode stage) of an unattended installation from a 32-bit operating system:

**Start the Computer in a Special Setup State** The computer starts by running \$ldr\$ and the files that are in \$Win\_nt\$.~bt. During this stage of an interactive setup you would be prompted for information about partitions, file system formats, and system folder name and location. Because it is not possible to configure partitions or file formats when you start an installation from an existing operating system, and all other configuration information is contained in the answer file, these prompts do not appear.

**Build directory structure** Setup creates the system folder structure and copies files from the temporary installation folder (\$win\_nt\$.~ls) to the system folders. For example, for a default installation, Setup creates the Windows folder and all subfolders (Inf, Help, System, System32, and so forth). The Txtsetup.sif file defines which folders are created and which files are copied to which folders. In addition, Winnt.sif, the file that contains the original answer file settings and other operating system information, is renamed \$winnt\$.inf and saved in the *systemroot*\System32 folder. Finally, the default registry hive files are created in the *systemroot*\System32\Config folder.

### Note

- Not all files are copied from the temporary installation folders to the system folders during the intermediate stage of the installation. Many of the files for optional components are installed during the final stage of Setup instead. This makes the installation process more efficient.

**Load boot sector and change boot.ini** The partition boot sector is changed again so that it contains the master boot code for starting the computer with Ntldr instead of \$ldr\$. The Boot.ini file is also modified so that the computer uses the Windows Server 2003 startup files instead of the startup files in the \$Win\_nt\$.~bt folder.

At the end of the intermediate stage, when all of the installation files have been copied from the temporary folders to the system folders, the destination computer restarts and the final stage of the unattended installation begins.

### Final Stage

The following actions occur during the final stage (also known as the GUI-mode stage) of an unattended installation from a 32-bit operating system:

**Start Windows Server 2003** The destination computer starts by using the Windows Server 2003 startup files and device drivers instead of the temporary startup files in \$Win\_nt\$.~bt. For example, the computer uses Ntldr instead of \$ldr\$, and the computer uses the startup files and device drivers in the systemroot folder instead of the startup files and device drivers in the \$Win\_nt\$.~bt folder. As with an interactive setup, the hardware detection and installation process also occurs.

**Install and configure optional components and settings** Optional components are installed (or not installed) and optional settings are configured based on the settings in the answer file. Examples of optional components and settings include regional and language options, text services and input languages, user name, company name, licensing mode, computer name, administrator account password, time zone settings, network adapter settings, protocol settings, and file and printer sharing. During an interactive setup you would be prompted for the installation information and the configuration settings, but these dialog boxes do not appear during a fully-automated unattended installation.

**Copy files from temporary folders** Based on the answer file settings for the final stage of the installation process, additional files might need to be copied from the temporary installation folders to the system folders. As with an interactive setup, a dialog box shows the progress of this file-copy process; however, no administrator or technician interaction is necessary or possible.

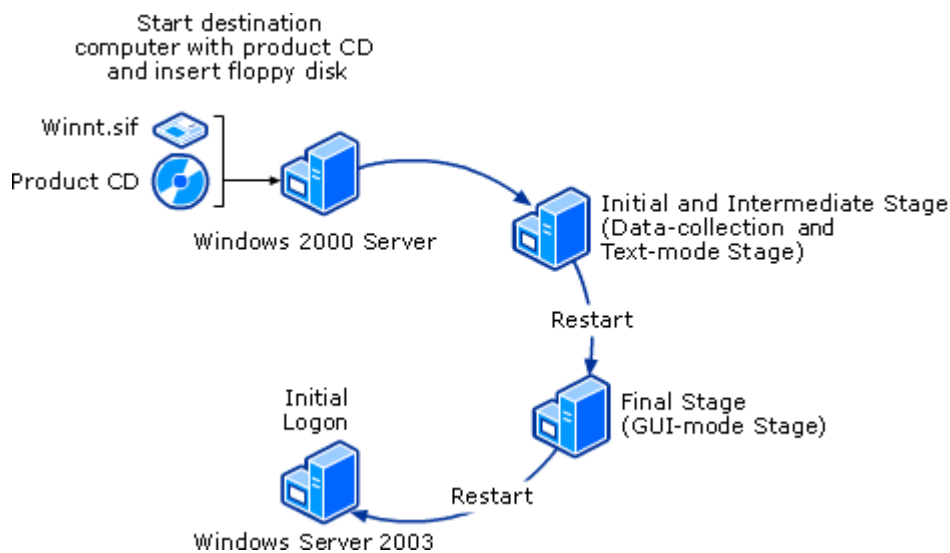
**Complete the installation** Windows performs the final configuration tasks. This includes configuring start menu items, registering components, deleting temporary installation folders, and running any commands specified in the Cmdlines.txt file.

At the end of the final stage, all of the required installation tasks have been completed and the destination computer is ready to run Windows Server 2003. If the answer file contains any instructions for performing post-installation tasks, Windows then restarts the destination computer, logs on with the Administrator account, and performs the post-installation tasks that are specified in the answer file.

### Unattended Installation When You Start the Computer from a Product CD

The following figure shows the processes and interactions that occur during a fully-automated unattended installation when you start the destination computer from a product CD.

#### Unattended Installation from a Product CD



This figure is based on the following requirements:

- The destination computer is running Windows 2000 Server.
- The answer file is named Winnt.sif and is saved on a floppy disk.
- The unattended installation is a clean installation of the operating system (not an upgrade).
- The primary partition on the hard disk is reformatted during the unattended installation.

- The unattended installation is started by starting the destination computer from the product CD.
- The unattended installation is fully automated (no administrator or technician interaction required).

In addition, the following answer file is used to perform this installation:

```
[Data]
  AutoPartition=1
  MsDosInitiated="0"
  UnattendedInstall="Yes"

[Unattended]
  UnattendMode=FullUnattended
  OemSkipEula=Yes
  OemPreinstall=No
  TargetPath=\WINDOWS
  Repartition = Yes

[GuiUnattended]
  AdminPassword=ca8d43d0eae56d10197a3608820ae698fa3254ef3d8ce553674c94fb185
  EncryptedAdminPassword=Yes
  OEMSkipRegional=1
  TimeZone=4
  OemSkipWelcome=1

[UserData]
  ProductKey=#####-#####-#####-#####-#####
  FullName="MyName"
  OrgName="MyCompany"
  ComputerName=MyComputer

[Display]
  BitsPerPel=24
  Xresolution=800
  YResolution=600
  Vrefresh=70

[LicenseFilePrintData]
  AutoMode=PerServer
  AutoUsers=5

[TapiLocation]
  CountryCode=1
  Dialing=Tone
  AreaCode=425
  LongDistanceAccess="9"

[Identification]
  JoinDomain=MyDomain
  DomainAdmin=MyAdmin
  DomainAdminPassword=MyPassword

[Networking]
  InstallDefaultComponents=Yes
```

### Starting the Unattended Installation from the Product CD

The processes and interactions described in this section are valid only when you start the destination computer from the product CD and quickly insert the floppy disk on which you saved your answer file into the floppy disk drive. You must insert the floppy disk immediately after the destination computer starts from the product CD because the Setup program immediately searches the floppy disk drive for a Winnt.sif file. If the Setup program does not find a floppy disk in the floppy disk drive, it assumes you are performing an interactive setup. When Setup finds a Winnt.sif file, Setup reads the Winnt.sif and initiates an unattended installation.

### Initial and Intermediate Stages

After you start an unattended installation from the product CD, the initial and intermediate stages of unattended

installation begin (these stages are also known as the data-collection and text-mode stages). Unlike in most installations, the initial and intermediate stages are not separate processes during an unattended installation from the product CD. Instead, these stages are combined into a single process and the destination computer does not restart between them.

The following actions occur during the initial and intermediate stages of an unattended installation from a product CD:

**Begin the setup process** Setup loads the files that are necessary to start the computer from the product CD. During this process a text-based Setup dialog box provides information about the Setup files that are loading. This is the same dialog box that appears at the beginning of an interactive setup. Usually, during an interactive setup from a product CD, several text-based dialog boxes appear after this initial Setup dialog box. For example, these dialog boxes ask you whether you are repairing an existing installation, whether you want to overwrite an existing operating system, and whether you want to use an existing partition or create a new partition. Because all of this information is contained in the answer file, these dialog boxes do not appear during an unattended installation from the product CD.

#### Note

- The Dynamic Update feature cannot be used during an unattended installation from the product CD because there is no network connectivity when you start a computer from the product CD.

**Process answer file** After the Setup files are loaded, the answer file is parsed and the results are saved in a new file named Winnt.sif. This file contains all of the settings that are in the answer file. A dialog box notifies you when the answer file is being processed, but it requires no administrator or technician interaction because you are performing a fully-automated unattended installation.

**Format and repartition** The primary partition is deleted and a new partition is created and formatted based on the information in the answer file. A dialog box notifies you when the partition is being created and the status of the formatting process, but the dialog box requires no administrator or technician interaction because you are performing a fully-automated unattended installation.

**Load boot sector** The partition boot sector is changed so that it contains the master boot code for Windows Server 2003. This small executable code initiates the startup process and instructs the computer where to find Ntldr.

**Build directory structure** Setup creates the system folder structure on the hard disk. For example, for a default installation, Setup creates the Windows folder and all subfolders (Inf, Help, System, System32, and so on). The Txtsetup.sif file defines which folders are created.

**Copy files to system folders** During the file-copy process Setup copies files from the I386 folder on the product CD to the system folders on the hard disk. The Txtsetup.sif file contains definitions and processing instructions for the files and folders that are copied. In addition, Winnt.sif, the file that contains the original answer file settings, is renamed \$winnt\$.inf and saved in the *systemroot*\System32 folder.

At the end of initial and intermediate stage, when all of the installation files have been copied to the destination computer's hard disk, the destination computer restarts and the final stage of the unattended installation begins.

## Final Stage

The following actions occur during the final stage (also known as the GUI-mode stage) of an unattended installation from a product CD:

**Start Windows Server 2003** Windows Server 2003 starts by using the Ntldr file (not Setupldr.bin) and all of the necessary startup files and device drivers that are in the system folder. As with an interactive setup, the hardware detection and installation process occurs.

**Install and configure optional components and settings** Optional components are installed (or not installed) and optional settings are configured based on the settings in the answer file. Examples of optional components and settings include regional and language options, text services and input languages, user name, company name, licensing mode, computer name, administrator account password, time zone settings, network adapter settings, protocol settings, and file and printer sharing. During an interactive setup you would be prompted for the installation information and the configuration settings, but these dialog boxes do not appear during a fully-automated unattended installation.

**Copy files from the product CD** Based on the answer file settings for the final stage of the installation process, additional files might need to be copied from the product CD to the system folders. As with an interactive setup, a dialog box shows the progress of this file-copy process; however, no administrator or technician interaction is necessary or possible.

**Complete the installation** Setup performs the final configuration tasks. This includes configuring start menu items, registering components, deleting temporary files, and running any commands specified in the Cmdlines.txt file.

At the end of the final stage, all of the required installation tasks have been completed and the destination computer is ready to run Windows Server 2003. If the answer file contains any instructions for performing post-installation tasks, Setup then restarts the destination computer, logs on with the Administrator account, and performs the post-installation tasks that are specified in the answer file.

### Post Installation Processes

Several unattended installation processes can occur after an operating system is installed on a destination computer. These processes occur when you automate post-installation configuration tasks. Examples of post-installation configuration tasks include running command-line programs to configure system services, running scripts to configure Group Policy settings, and installing software programs. The most commonly used mechanisms for automating post-installation tasks is the `Cmdlines.txt` file, the `winnnt32` command with the `/cmd` parameter, and the `[GUIRunOnce]` and `[SetupParams]` sections of the unattended installation answer file.

#### Note

- Several answer file sections contain entries that are not configured during the intermediate or final stages of an unattended installation. These settings are configured after the operating system is installed on the destination computer and the destination computer has restarted. The answer file sections include `[DCInstall]`, `[CertSrv_Client]`, and `[CertSrv_Server]`.

### Cmdlines.txt Processes and Interactions

A `Cmdlines.txt` file contains a list of commands, programs, scripts, or batch files that run just after the final stage of Setup finishes but before the destination computer restarts. By default, `Cmdlines.txt` does not exist, so you must manually create the file with a text editor, such as Notepad.

The syntax for `Cmdlines.txt` is as follows:

```
[Commands]
"command_1"
"command_2"
.
.
"command_x"
```

Where `command_1`, `command_2`, and `command_x` refer to the commands, programs, scripts, or batch files that you want to run when final stage of Setup is complete. Note that all commands must be enclosed in quotation marks. Also, if you are using the command window (`Cmd.exe`) to run commands, programs, scripts, or batch files, then you need to use the `/c` parameter with the `cmd` command. For example, to create a new folder named `Test` on drive `C`, you type:

```
"cmd /c mkdir c:\Test"
```

In addition, to use a `Cmdlines.txt` file, you need to:

- Add the following section, entry, and value to your answer file:

```
[Unattended]
OemPreinstall = Yes
```

- Copy all of the programs, scripts, and batch files that are listed in your `Cmdlines.txt` file to the `$OEM$` folder in your distribution share.
- Save the `Cmdlines.txt` file in the `$OEM$` folder in your distribution share.

When you use the entry `OemPreinstall = Yes` in the `[Unattended]` section of your answer file, Setup searches the `$OEM$` folder in your distribution share for a `Cmdlines.txt` file. If Setup finds a `Cmdlines.txt` file, then the current path is changed to `c:\$win_nt$.~\ls\$OEM$`. This is why you need to put all programs, scripts, or batch files that are listed in `Cmdlines.txt` into the `$OEM$` folder in your distribution share.

You can only have one `Cmdlines.txt` file, but you can use it to run multiple commands, programs, scripts, or batch files. When these commands, programs, scripts, or batch files run, the following processes and interactions occur:

- The commands, programs, scripts, and batch files run synchronously, which means each command, program, script, or batch file runs to completion before the next one starts running.
- All commands, programs, scripts, and batch files specified in a `Cmdlines.txt` run under the Local System security account.
- Network connectivity is not available when the commands, programs, scripts, and batch files listed in a



Cmdlines.txt file run.

- Windows Installer packages (.msi files) cannot be run from a Cmdlines.txt file.

#### ◆ Important

- You cannot use Cmdlines.txt if you are using a product CD and a Winnt.sif file to perform an unattended installation. You can use Cmdlines.txt only if you are installing from a distribution share.

### [GuiRunOnce] Processes and Interactions

The [GuiRunOnce] section of an answer file contains a list of commands, programs, scripts, or batch files that run the first time an administrator or technician logs on to the computer after the operating system is installed. Because the [GuiRunOnce] section does not exist in a default answer file built by Setup Manager, you must add it manually by editing the answer file with a text editor, such as Notepad.

The syntax for [GuiRunOnce] is:

```
[GuiRunOnce]
"command_1"
"command_2"
.
.
"command_x"
```

Where *command\_1*, *command\_2*, and *command\_x* refer to the commands, programs, scripts, or batch files that you want to run after the operating system is installed and the administrator or technician logs on. Note that all commands must be enclosed in quotation marks.

The commands that are listed in the [GuiRunOnce] section are stored in the following registry location:

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Runonce

When an administrator or technician logs onto the destination computer for the first time after the unattended installation is complete, Windows Server 2003 reads and then runs the commands that are listed in this registry location. When these commands, programs, scripts, or batch files run, the following processes and interactions occur:

- The commands, programs, scripts, and batch files run synchronously, which means each command, program, script, or batch file runs to completion before the next one starts running.
- The commands, programs, scripts, and batch files listed in the [GuiRunOnce] section run under the security account of the user who logged onto the destination computer (usually an administrator or a technician).
- Network connectivity is available when the commands, programs, scripts, and batch files run.
- You can use Windows Installer packages (.msi files) to install applications (unlike Cmdlines.txt).
- You can run commands, programs, scripts, and batch files from hard disk drives, CD drives, shared folders on the network, or other storage devices (unlike Cmdlines.txt, which runs everything from the \$OEM\$ folder in your distribution share).

In addition, there are some interactions and processes that you might need to control if you run certain commands, programs, scripts, and batch files with the [GuiRunOnce] section.

#### Prevent restarts

You can run commands, programs, scripts, and batch files only if you can prevent the command, program, script, or batch file from restarting the computer. When a computer restarts, all remaining entries in the [GuiRunOnce] section are lost because the Runonce registry entry is deleted. If the system restarts before completing entries listed in the [GuiRunOnce] section, the remaining items will not run. Therefore, you need to suppress restarts. If you cannot suppress a restart within a command, program, script, or batch file, you can try to repackage the command, program, script, or batch file into a Windows Installer package.

#### Do not rely on the Windows Explorer shell

The Windows Explorer shell is not loaded when the operating system starts running commands, programs, scripts, or batch files that are listed in the [GuiRunOnce] section of an answer file. In some cases, you can get an updated installer program from the application vendor that does not rely on the Windows Explorer shell. If you cannot, you might be able to repackage the command, program, script, or batch file as a Windows Installer package (.msi file).

#### Control the processes by using the /wait parameter

Commands, programs, scripts, and batch files often start and stop several different processes. In some cases,

when you are running multiple programs — especially installation programs — this can inadvertently cause the next program listed in the [GuiRunOnce] section to start before the previous program is finished running. When this occurs, the second program usually fails. To prevent this, you can run the programs from batch files by using the **start** command with the **/wait** parameter. This forces each program to run to completion before the next command listed in [GuiRunOnce] runs.

### Other Post-Installation Processes and Interactions

In addition to the Cmdlines.txt file and the [GuiRunOnce] section of an answer file, two other mechanisms can create post-installation processes and interactions during an unattended installation. These mechanisms are the [SetupParams] section of an answer file, and the **winnt32** command with the **/cmd** parameter.

**[SetupParams]** The [SetupParams] section of an answer file contains a single command, program, script, or batch file that runs immediately after the commands, programs, scripts, or batch files listed in the Cmdlines.txt file run. You typically use this answer file section and entry to perform a post-installation task on only a few computers without changing the Cmdlines.txt file. Because the [SetupParams] section does not exist in a default answer file built by Setup Manager, you must add it manually by editing the answer file with a text editor, such as Notepad.

The syntax for the [SetupParams] section is:

```
[SetupParams]
UserExecute = path_and_file_name_or_command
```

Where *path\_and\_file\_name\_or\_command* refers to the command, program, script, or batch file that you want to run. Note that the *path\_and\_file\_name\_or\_command* must be enclosed in quotation marks. Also, you can omit the path portion of *path\_and\_file\_name\_or\_command* only if the application is in the systemroot or systemroot\System32 folder or search path.

**Winnt32 /cmd** The **winnt32** command with the **/cmd** parameter specifies a single command, program, script, or batch file that runs immediately after the commands, programs, scripts, or batch files listed in the Cmdlines.txt file run. You typically use this answer file section and entry to perform a post-installation task on only a few computers without changing the Cmdlines.txt file.

The syntax for the **winnt32** command with the **/cmd** parameter is:

```
Winnt32 /cmd:path_and_file_name_or_command
```

Where *path\_and\_file\_name\_or\_command* refers to the command, program, script, or batch file that you want to run.

### Log File Interactions During an Unattended Installation

Several log files are created during an unattended installation and saved in the systemroot folder. The following table provides the names, locations, and descriptions of the Setup log files. The descriptions provide a summary of the contents of each log file when the logging level is set to its default state. You can increase the logging level (increase log file details) for the Winnt32.log and Setupact.log files by using the **winnt32** command with the **/debug** parameter. By default, the logging level is set to 2, which means severe errors, common errors, and warnings are included in the log files. By increasing the logging level you can include basic setup information and detailed debugging information in the Winnt32.log and Setupact.log log files.

#### Setup Log Files for an Unattended Installation

Log File Name	Location	Description
Setupact.log	Systemroot	Log file for the initial (data-collection) and final ( GUI-mode) stage of unattended installation. Contains information about detected hardware, installation requirements, and a list of device driver files that are copied from the product CD or \$Win_nt\$.~ls to various system folders.
Setupapi.log	Systemroot	Log file for Setupapi.dll, which processes .inf files during hardware installation and registration.
Setuperr.log	Systemroot	Log file for errors that occur during the initial (data -collection) and final (GUI-mode) stages of unattended installation. This log file consolidates Setup errors; the errors that appear in this log file also appear in other log files.
Setuplog.txt	Systemroot	Log file for the initial (data-collection) and final (GUI-mode) stages of unattended installation, as well as some post-installation tasks that occur after Setup completes and the destination computer restarts. Contains information about the files that are copied and deleted, the settings that are configured, and the security

settings that are applied to the destination computer.

**Winnt32.log**    **Systemroot**    Log file for the initial (data-collection) stage of the setup process. Contains information about the existing operating system, hard disk size, Boot.ini configuration, and a list of the files that are copied to the temporary installation folders \$Win\_nt\$.~bt and \$Win\_nt\$.~ls.

Setupapi.log and Setuplog.txt are not overwritten if you restart the unattended installation process: that is, information is appended to the log file each time you prematurely interrupt and restart the unattended installation process. All of the other log files are overwritten every time the unattended installation process is prematurely interrupted and restarted.

#### **Note**

- No log files are generated during the intermediate (text-mode) stage of Setup.

In addition to these log files, three other log files contain relevant information about the unattended installation process. The following table provides the names, locations, and descriptions of these log files.

#### **Supplemental Log Files for an Unattended Installation**

<b>Log File Name</b>	<b>Location</b>	<b>Description</b>
Wsdu.log	Systemroot	Log file for Dynamic Update. This is generated only if you use Dynamic Update during an unattended installation, which you can only do if you are performing an unattended installation from an existing 32-bit operating system such as Windows NT, Windows 2000, or Windows Server 2003.
Dberr.txt	systemroot\System32\CatRoot2	Log file for catalog file registrations. Contains information about catalog files that are not registered, and exists only if any nonregistered catalogs are identified during unattended installation. If a catalog is not registered, then the files in the catalog file are considered unsigned and will generate errors in Setuperr.log.
Scesetup.log	systemroot\security\logs	Log file for security settings.

---

[Back to Top](#)

#### **Related Information**

The following resources contain additional information that is relevant to this section.

- [Setup Technical Reference](#)
- [Sysprep Technical Reference](#)
- [Remote Installation Services Technical Reference](#)