



Automated Deployment Services Technical Overview

Microsoft Corporation

Published: August 2003

Abstract

Administrators can now build and manage very large, scaled-out deployments of Windows servers with Microsoft® Windows Server™ 2003 Automated Deployment Services (ADS). ADS includes a new set of imaging tools developed by Microsoft and an infrastructure for rapidly deploying both Windows 2000 Server and Windows Server 2003 remotely onto bare metal servers. In addition, ADS offers improved communication security and a reliable script execution framework that enables administrators to perform script-based administration on 1,000 servers as easily as they once did on a single server.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred. <INCLUDE THIS DISCLAIMER ONLY WHEN APPLICABLE TO YOUR CONTENT>

© 2003 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, MS-DOS, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

Introduction	1
Architectural Overview	2
ADS Services and Tools	2
Controller Service.....	2
Network Boot Service.....	4
Image Distribution Service	5
Volume Imaging Tools.....	6
ADS Agents.....	7
ADS Task Sequences and Task Sequence Editor	8
ADS Administration Tools and Programmatic Interfaces	9
WMI Object Model.....	9
Administration Tools.....	10
ADS Network Environment	12
ADS Security Model.....	12
Role-based Job Execution	13
Using ADS: Examples	15
Image Deployment and Maintenance	15
Bare Metal Server Deployment and Configuration.....	15
Preparing and Capturing an Image	15
Hardware Configuration	16
Deploying and Personalizing Images.....	17
Post Operating System Deployment Configuration.....	18
Mass Server Administration	18
Supported Jobs and Scripts	18
Running Jobs and Scripts	19
Capturing and Logging Results	19
Example Task Sequences	21
PXE-Based Bare Metal Purposing.....	21
Script/Job Remote Execution.....	24
Summary	25

Related Links 26

Introduction

As both the rate of growth and the installed base of Microsoft® Windows Server™ operating systems increases, managing the deployment and administration of these systems becomes a significant driver of the overall cost of ownership.

Today automated operating system and application deployment technologies are typically script-based or rely on traditional imaging and deployment tools from third-party vendors. Script-based installation solutions provide flexibility across a wide range of hardware configurations but tend to be very slow, and few standard methodologies exist for them. Traditional imaging technologies, although much faster, are inflexible and require considerable effort to adapt and maintain an image collection over both hardware variations and time.

Script-based administration of a large number of Windows® servers traditionally has not been easy. Unlike in the UNIX environment, in which operators can use tools such as rsh, ssh, and rdist to perform remote administration on groups of servers, script-based administration in the Windows Server environment has required operators to deal with each server individually.

With Microsoft Windows Server 2003, Microsoft extends the platform to enable rapid, flexible deployment and seamless, script-based administration of a large number of Windows servers. Table 1 shows the key features and benefits of Automated Deployment Services (ADS).

Table 1. Key features and benefits of ADS

Feature	Benefit
Scalable, remote deployment architecture	An intelligent Preboot Execution Environment (PXE) server and dynamically built Deployment Agent enable remote server builds of PXE-compliant, bare-metal boxes, reducing the cost to deploy servers.
Powerful task-sequence-driven automation	Sample task sequences can be extended to automate hardware configuration, operating system deployment, and application installation, enabling you to encode your organization's operational practices and eliminate human error.
Flexible new imaging tools	New Microsoft-built tools use knowledge of the NTFS file system structure to create smaller images that can be updated and edited without first being deployed to a server. As a result, the speed of traditional imaging is combined with the flexibility of script-based installations.
Reliable remote execution framework	ADS enhances existing scripting investments and extends your ability to administer hundreds of servers.
Virtual Floppy	ADS incorporates standard server vendor MS-DOS tools into the deployment process to automate hardware configuration.
Diverse choice of user interfaces	ADS offers a simple-to-use graphical user interface (GUI), a set of command-line tools, and a Windows Management Instrumentation (WMI) program interface. ADS enables point-and-click operation or integration with other solutions in your deployment process.
Centralized data store	You can maintain a complete history of all administrative tasks carried out using the ADS infrastructure.

Architectural Overview

Figure 1 summarizes the ADS architecture.

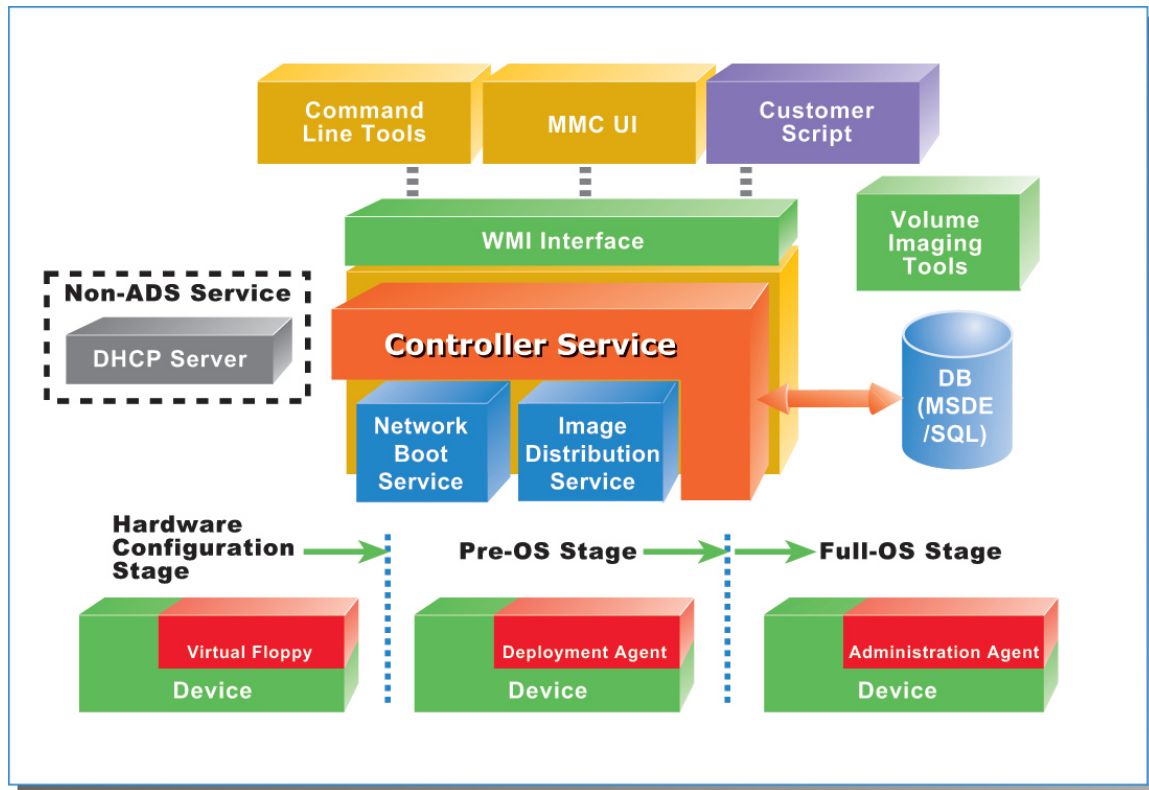


Figure 1: ADS architecture

ADS Services and Tools

ADS is composed of the following:

- An integrated set of services, including Controller Service, Network Boot Service (NBS), and Image Distribution Service.
- Volume Imaging Tools.
- A set of agents.

These features work together to enhance the ability of administrators to deploy and administer large numbers of Windows servers.

Controller Service

The Controller Service is the operational heart of ADS. The Controller orchestrates all ADS activity by providing configuration information to the other ADS services and allowing administrator inputs through the WMI interface, ADS command-line tools, and an MMC snap-in component with a GUI. It maintains

the master records that detail each device known to ADS along with the actions associated with those devices.

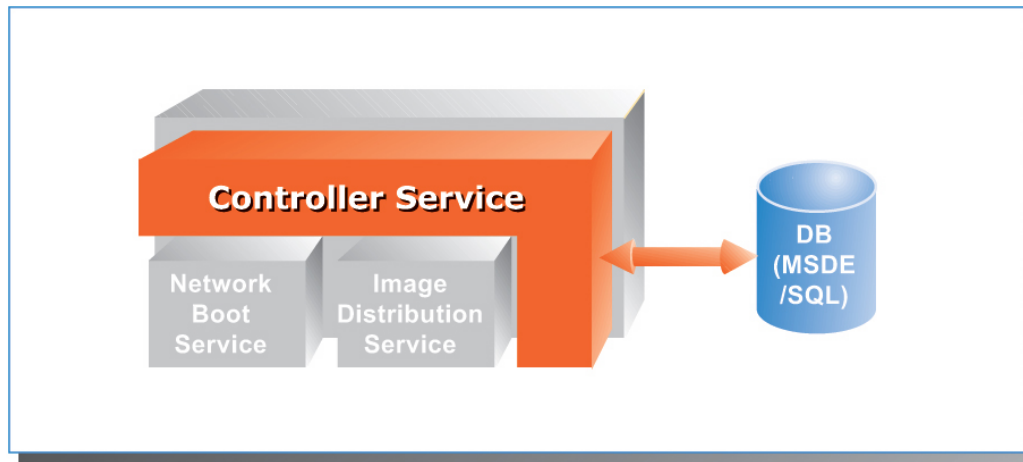


Figure 2: Controller Service

As Figure 2 illustrates, Controller Service functionality includes the following:

- Coordination and sequencing of tasks.** The Controller Service coordinates administrative activity throughout deployment and administration. For example, it provides the PXE¹ service within the NBS with the appropriate boot commands for each device. In the deployment phase, the Controller Service coordinates the task sequence designed to configure a server and install images locally on a system. When rebooting after an initial image deployment, devices check with the Controller Service for any waiting boot commands, a feature that enables you to completely repurpose a device on its next boot, as though it were a bare metal server.
- Device communication with security.** The Controller Service provides security in communicating with the ADS Deployment and Administration agents that reside on devices to run task sequences.
- Centralized recording of device data and administrative activity.** The Controller Service uses either Microsoft SQL Server™ 2000 Desktop Engine (MSDE)² or Microsoft SQL Server 2000 to store all device and configuration data and to log information for all the tasks performed on the devices. A complete audit trail for each task run at the device is available through the Controller Service.
- Logical grouping of devices assets.** The Controller Service can manage each device individually if needed, but in general you work with sets. Sets are groups of managed devices that can be addressed as a single entity. When deploying operating systems or performing post-deployment administration, you need to reference only the set name belonging to the group of devices you want to affect. A set can also contain references to other sets, allowing a hierarchical model in which everyday management commands can be executed on all devices with a single command and smaller groups can be used to

¹ The PXE (*Preboot Execution Environment*) standard was created by Intel and allows a network adapter to act before a server loads an operating system from a load hard disk.

² MSDE is a SQL 2000-compatible database engine to a local client system. For more information, see the [SQL Server Web](http://www.microsoft.com/sql) site at <http://www.microsoft.com/sql>.

provide more detailed control over your devices. With ADS, a single command from the command line or MMC snap-in can initiate actions to hundreds of devices at one time.

Network Boot Service

NBS works in conjunction with a network Dynamic Host Configuration Protocol (DHCP) server to give ADS boot command capability as Figure 3 shows. A PXE-enabled device discovers the PXE service using DHCP, and then begins a communication session with ADS by means of the NBS.

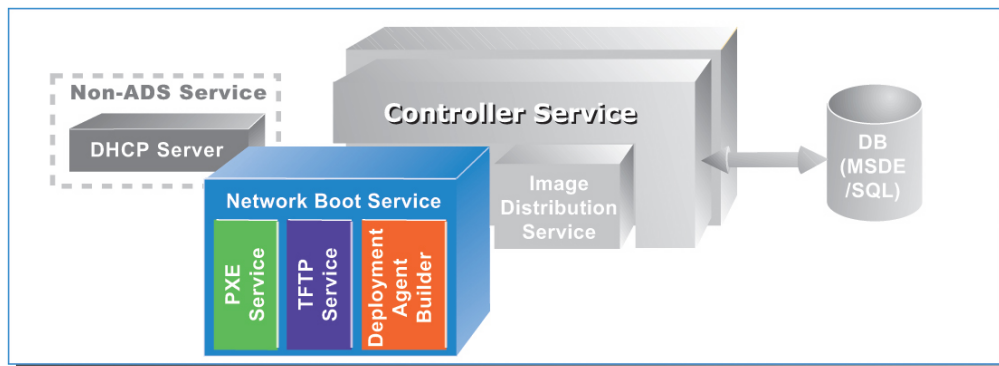


Figure 3: Network Boot Service

NBS includes two services:

- **ADS PXE Service.** This service sends PXE boot commands to the device. The PXE service can instruct the destination device to do the following:
 - Download and boot an ADS Deployment Agent that is created for it by the agent builder service.
 - Download and boot a Virtual Floppy image.
 - Ignore the PXE boot request.
 - Stop PXE operations and boot to hard disk.
- **ADS Deployment Agent Builder.** This service creates a device-specific agent at boot time. After the PXE boot occurs and the Deployment Agent option is chosen, the builder service does the following:
 - Performs a local hardware discovery on the device and sends that data to the Controller Service.
 - Builds a customized agent based on the device's hardware configuration.
 - Downloads that agent to the device on which it executes in local memory.

In addition to enabling Deployment Agent downloading, the NBS also provides a way to download and execute an MS-DOS-based Virtual Floppy image that also executes in memory. Utilities are usually provided by your server vendor and can perform hardware configuration tasks such as RAID drive configuration and BIOS flashing in the Virtual Floppy environment. The virtual floppy feature supports an optional status in the form of a return code and text reported to the job's standard output.

The NBS uses the Windows Server 2003 Trivial File Transfer Protocol (TFTP)³ service to download both deployment agents and Virtual Floppy images.

Either on the Controller or elsewhere in the network, ADS requires a DHCP⁴ server to be present to provide IP addresses to configurable devices.

Image Distribution Service

The Image Distribution Service (IDS) is a component of Controller Service as Figure 4 shows. IDS provides storage and communications capabilities for ADS to manage device images, and it enables administrators to deploy device operating system images that are created using the ADS imaging tools.

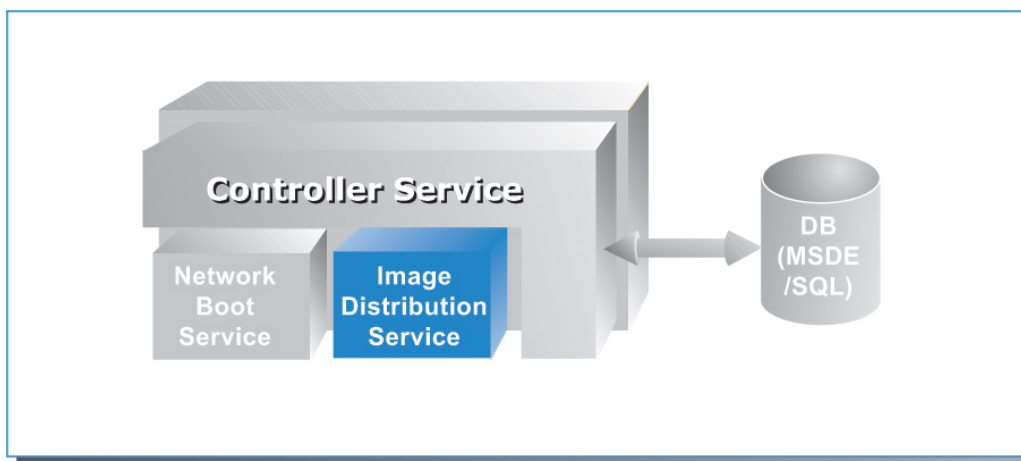


Figure 4: Image Distribution Service

To quickly and efficiently deploy an operating system and applications to devices, ADS uses new imaging technology from Microsoft. Imaging is a straightforward process that enables you to capture and rapidly deploy complete operating system and application bundles to one or many devices simultaneously.

After the images are captured (see “The Imaging Process” in the following section), they are added to the IDS image store, where they are kept on the file system of the IDS server. While the images are downloading to the device, the communications channel is encrypted to prevent eavesdropping on the image data. The Controller Service orchestrates the imaging operation, communicating with the device through a secure connection using a protocol based on Secure Sockets Layer (SSL). IDS can deploy images to multiple devices over unicast-enabled⁵ and multicast-enabled⁶ networks. In addition, the IDS

³ A TFTP server is a very weak file-transfer mechanism that is typically used to transfer small amounts of data to devices that either are not very powerful or are not meant to run extensive applications.

⁴ DHCP is primarily a way to dynamically assign IP addresses to machines that do not require fixed addresses.

⁵ Unicast transmission is a single data stream from the Controller Service to one device. When you deploy images to more than one device, each uses an individual data stream from the Controller, creating demand on network bandwidth. For more information, see “Differences Between Multicast and Unicast” at <http://support.microsoft.com/default.aspx?scid=kb;en-us;291786>.

provides users the ability to adjust the amount of network bandwidth that is used during the multicast imaging transmissions.

The Imaging Process

The imaging process is comprised of several discrete tasks:

1. **Install a reference device.** Using a hardware configuration similar to that of the devices you will deploy with this image, install the base operating system and any other applications and device drivers that this image will need in production. Also add the ADS Administration Agent, discussed later in this paper in the "[ADS Agents](#)" section
 2. **Prepare the reference device for imaging.** Use Sysprep to prepare the system for imaging. Sysprep removes unique system information and prepares the system to run a miniature version of Windows Setup the next time it boots. This step enables every device that is created from this image to generate the information that makes it unique to the Windows environment.
 3. **Capture the boot disk partition using image tools.** ADS includes tools to capture an image of a device manually by running a command-line tool on the reference system. Alternatively, you can use ADS to capture an image remotely when a system reboots.
 4. **Deploy the image to new or existing devices.** ADS deploys the operating system image to a new or existing device, which can then join the production environment. From a single image hundreds of devices can be deployed easily, quickly, and in a reproducible manner.
-

Volume Imaging Tools

Figure 5 summarizes the volume imaging tools in ADS.



Figure 5: Volume imaging tools

ADS has a powerful, flexible set of imaging tools. Any FAT or NTFS file system volume can be captured and deployed when used in conjunction with Sysprep, but enhanced capture and editing benefits are gained when you work with NTFS-based file systems. Table 2 describes the ADS imaging tools.

⁶ Multicast transmission employs a single data stream that is read by multiple servers. Network use is minimized when using multicast. For more information, see "What Is IP Multicast?" at <http://support.microsoft.com/default.aspx?scid=kb;en-us;165011>.

Table 2. ADS Imaging Tools

Imaging Tool	Purpose
Imgdeploy	The tool that performs the actual image capture and restoration. It features: <ul style="list-style-type: none"> ▪ Image compression to reduce image storage requirements. ▪ Image encryption for transport or alternative storage. ▪ Defragmented restoration to capture images in file order so that files are automatically defragmented when restored to a device.
Imgmount	This tool enables you to perform true editing of captured images. Images captured with Imgdeploy can be mounted and accessed like a file system. After an image is mounted in this fashion, it can be read from and written to by any standard Windows tool or application. The saving in administration time is considerable: The image can be maintained without modifying the reference system from which it was captured and then recapturing the image.
Adsimage	Accessed from the command line or the MMC UI, this tool lists images available for deployment and adds images, deletes images, or updates image properties,

ADS Agents

As shown in Figure 6, ADS provides two control agents that correspond to the pre-deployment and post-deployment stages:

- **Deployment Agent.** A highly optimized, reduced-functionality version of Windows Server 2003 that loads into a RAM disk⁷ on a device and handles deployment operations, such as disk partitioning and image downloading. This agent is available only for environments using PXE boot control and is the key facilitator of a pure network-based operating system deployment.
- **Administration Agent.** Used after a device has been deployed with an operating system for post-deployment task execution. Using Administration Agent, you can initiate operations on a device that can perform any scriptable command and run local applications or any application that can be reached on a network. With Administration Agent, you can also restart a deployment to repurpose a given device.

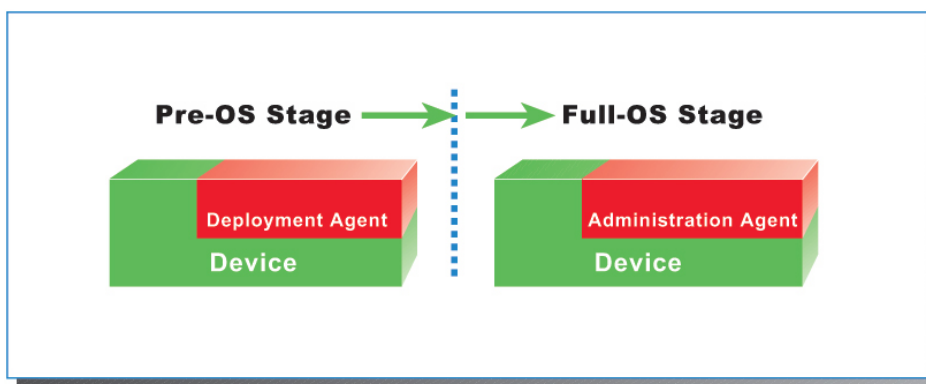


Figure 6: ADS agents

⁷ A RAM disk is an area of computer memory that appears to be a physical disk drive. It is accessed by a drive letter like any other system drive. RAM disks are used in situations in which an application needs maximum performance or when a system needs temporary storage that can be easily accessed. ADS uses a RAM disk in the second capacity.

ADS Task Sequences and Task Sequence Editor

At boot time or after deployment, the ADS Controller Service issues jobs to devices that consist of individual tasks sequenced together. These sequences can be run against one or more devices and are stored as Extensible Markup Language (XML) files on the controller. A typical task sequence used to deploy a new device with PXE is as follows:

1. Execute a Virtual Floppy to update system BIOS or configure a RAID controller.
2. Request and boot Deployment Agent.
3. Partition the hard disk.
4. Download an operating system image to the hard disk.
5. Modify the Sysprep.inf that was downloaded in the deployed image to give this device a unique host name, product identifier, and so on.
6. Configure the ADS Administration Agent in the downloaded image to communicate with the controller when this operating system first boots.
7. Reboot.
8. Instruct the device to boot to the local hard disk on the next boot.

When the device boots from a newly downloaded image for the first time, an abbreviated setup runs in an unattended mode to perform final configuration of the operating system.

Task sequences can consist of the following:

- PXE boot commands.
- Deployment agent internal commands.
- Any Windows application or script available to the destination device or on a network share accessible by the destination device.
- A script available on the Controller to be downloaded to the destination device and run there.
- A script available on the Controller to be run directly on the controller.

When using ADS tasks, the Controller logs all the commands that are run and their output to the Controller's database, providing a complete audit log as well as an error log for debugging commands. Some sample task sequences are provided later in this document.

These XML task sequences can be created with any XML authoring tool, but ADS also includes the simple Sequence Editor for administrators who prefer not to work directly with XML. It provides an easy-to-use, Windows-based user interface for creating ADS sequences. The Sequence Editor provides templates for the most common tasks that administrators want to carry out, including deployment steps, such as partitioning the disk, downloading an image, and personalizing the image. To create a new task sequence or to edit existing sequences, you open the Sequence Editor, manipulate parameters, and save the result as a new sequence.

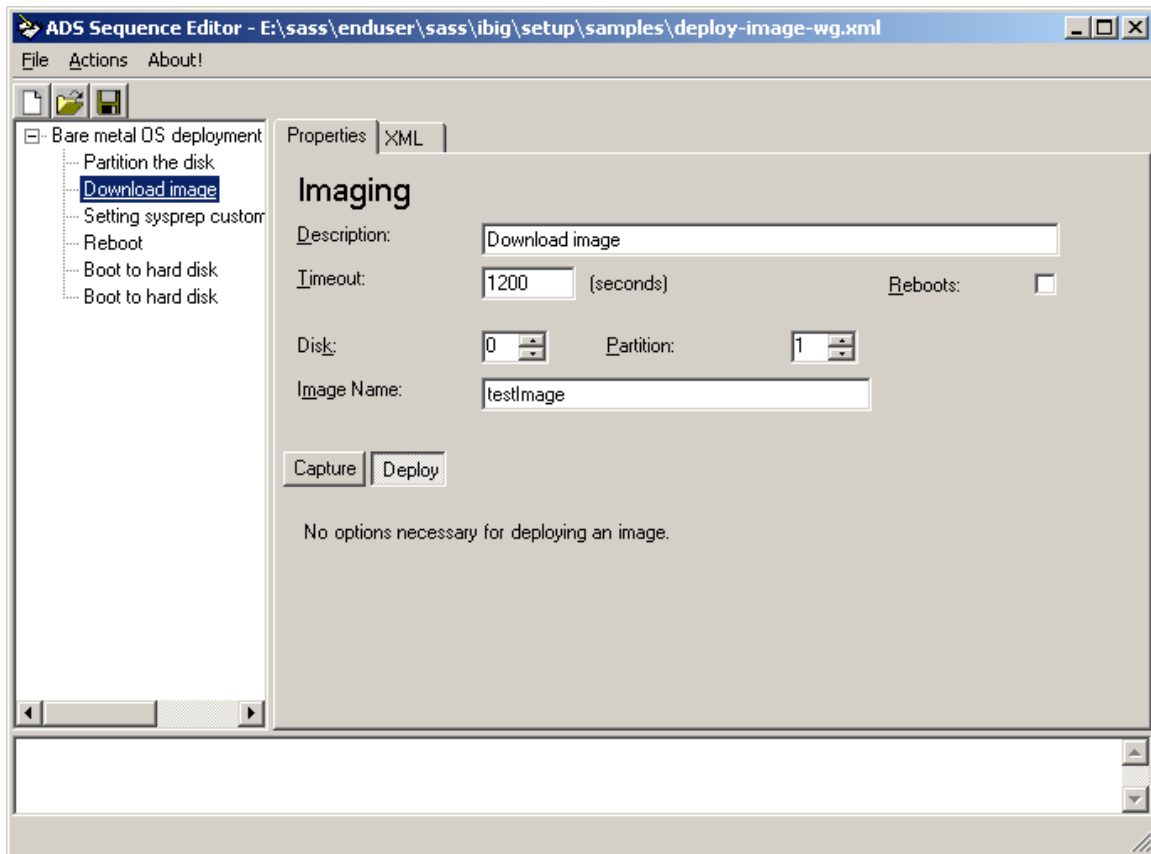


Figure 7: Task Sequence Editor

ADS Administration Tools and Programmatic Interfaces

ADS provides a series of command-line tools and an MMC snap-in to manage ADS services. However, a programmatic interface is also exposed through the WMI interface. All ADS features are available through the WMI interface. If your organization already has in-house talent to develop management applications, you can easily tailor and manage ADS in your environment.

WMI Object Model

Figure 8 illustrates the range of WMI object areas under ADS.



Figure 8: WMI Object Model

Using WMI, you can address all the individual aspects of ADS. Using Microsoft Component Object Model (COM) communication, all aspects of the ADS system are available for scripting or application

development. Table 3 identifies the primary object areas available for use by a scripting administrator or a developer as well as examples of the properties and methods available.

Table 3. Available Object Areas

Object	Description	WMI Classes
Devices	Devices represent physical devices within the data center. Several classes combine to represent the information available for one device.	Devices DeviceVariables DeviceHWAddr
Sets	Sets represent collections of devices. Each set has a unique name and can contain from zero to multiple devices or other sets. A given device may be in multiple sets.	Sets
Job templates	Job templates are job definitions available to be run. A job template may be a simple job (such as a script or program to run) or a task sequence. Each template has a unique name.	JobTemplates
Jobs	Jobs are representations of jobs in progress or jobs that have already completed. The Jobs object stores basic information, such as the description, job type, target, command, and parameters.	Jobs JobLogs
Images	Images represent captured volumes that are available to be deployed. They might be operating system volumes captured after Sysprep that can be deployed to multiple servers, operating system volumes captured without Sysprep that can be deployed to a single server, or data volumes.	Images ImageVariables
Services	Services represent each ADS service—Controller Service, NBS, and Image Distribution Service. These are used to configure the controller so that it knows which physical system hosts each of these services and to configure global properties for each service.	Services ServiceVariables

Administration Tools

ADS provides both a flexible, easy to use MMC user interface and a set of command-line tools for administration. Several command-line tools are available to manage and operate ADS. It is important to note that the tools that operate the Controller Service are built on the WMI interface. Other command-line tools that provide imaging or certificate management neither depend on nor require WMI functionality to operate.

Table 4. Administration Tools

Tool	Description
ADSArchive	Archives previous jobs by removing them from the database and storing them in an XML file.
ADSDevice	Manages device records and allows you to list, add, delete, control, or modify device records.
ADSDhcpConfig	Configures ADS and DHCP if they are located on the same server or on different servers.
ADSIImage	Lists images available for deployment and adds images, deletes images, or updates image properties.
ADSJob	Provides job management to run, stop, list, read logging information, or view the history of jobs run on the controller.
ADSJobTemplate	Lists the templates available to execute as jobs and installs and deletes templates.
ADSKey	Generates, signs, and registers controller certificates.
ADSRole	Displays and modifies permissions on ADS job templates.
ADSService	Manages the ADS services.
ADSSet	Provides set management to add devices to a set, remove devices from a set, create or delete sets, list set members, or modify set properties.
Dskimage	Creates a virtual floppy image from a floppy disk so that you can download BIOS configuration or other tools to the device during deployment.
Imgdeploy	Read or write file system images from or to disk. After running Sysprep on the system providing the deployment image, this tool creates the image file that is downloadable by the device.
Imgmount	Enables you to mount captured images to view, add, or remove files within the image.
Regcert	Registers agent certificates.

GUI access to the ADS services is provided by an MMC snap-in, as shown in Figure 9. The tools provides administrators the flexibility of either scripting directly to the command-line interface or walking through tasks and procedures using a GUI.

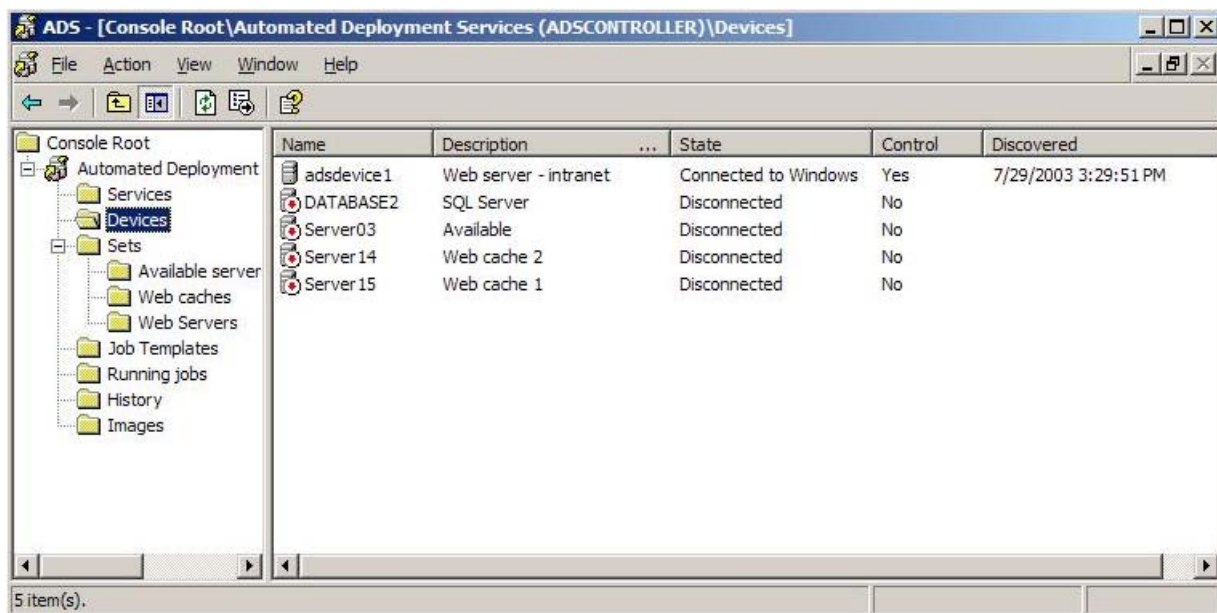


Figure 9: ADS devices

ADS Network Environment

ADS assumes a well-connected networking environment—10 MB or more connectivity among the controller, IDS, NBS, and managed devices. No provisions are made for general retries when tasking devices or downloading images; the operations either complete successfully or do not. That fact makes ADS impractical for use over low-speed wide-area network (WAN) links.

Deployment of ADS consists of the Controller Service, IDS, and NBS. In a subnetted or a virtual local area network (VLAN) environment, the DHCP broadcast scope is usually constrained to the IP subnet or VLAN of the attached network. This scope is the same for PXE broadcasts. In these environments, you must configure DHCP forwarding at your routing points to bring all the devices back to one DHCP and NBS server. In Cisco environments, you simply configuring an “IP helper” at the router or layer three switch. In environments that cannot provide DHCP forwarding in the routers, Microsoft provides a DHCP forwarding service in the server operating systems that perform this function.

For large device deployments, ADS provides multicast support, which can reduce the overall bandwidth used when deploying disk images. In a switched or routed environment, your equipment must be configured and capable of supporting this feature.

ADS Security Model

The ADS security model is divided into three distinct realms. The first security realm encompasses the initial stages of the bare metal deployment in which DHCP and PXE are being used initially. The DHCP and PXE phases are not authenticated because of the design of those protocols. Microsoft recommends that the deployment network interface reside on a dedicated management network.

The second security realm is the communication between the various ADS services and the device agents communicating with the ADS Controller. ADS uses a public key infrastructure (PKI) and SSL server authentication to create private, out-of-band management communication channels. In addition,

by default images transferred between the Image Distribution Service and the Deployment Agent during image capture and deployment are also encrypted.

The third security realm is the security context in which the ADS services and agents operate. The ADS Controller Service runs under the context of the system service, the equivalent of local administrator. This is necessary because of the configuration control delegated to the ADS Controller. On managed devices, however, the agent is installed by default using the local machine Administrator account. This account can be changed to any user that has the appropriate authority to run the applications and scripts that will be used as ADS jobs. Also, whatever account is chosen will require access to network resources if you execute scripts or applications from network share points.

The ADS security model is built on the assumption that the data center is physically secure and free from network intrusion. More specifically, the network between the servers running the ADS services and the devices should be secure. In general, access to the ADS Controller service is restricted to users who are members of the administrators group on the Controller. Users who are not members of the local Administrators group cannot run jobs on from Controller using ADS unless explicitly granted permission by an Administrator using the role-based security functionality. Administrators should limit physical access to the data center and to the Administrators group.

Besides access to the Controller Service, ADS services use additional data that must be secured. On the ADS Controller, XML task sequence files and script files that are referenced through job templates are stored on the file system and should be protected by file system access control lists (ACLs) to ensure that non-administrators do not possess modify rights to these files. The Image Distribution Service stores images ready for deployment, which are also protected by the file system. Administration Agent running on a device may also refer to scripts that should be protected by the file system.

Role-based Job Execution

The ADS security model supports multiple roles. The role of an administrator is to install and configure ADS into a data center. Configuration includes creating images, developing scripts, creating job templates, defining devices, and authorizing non-administrator roles. Role-based job execution is designed to be used by a data center administrator to grant non-administrators privileges to execute job templates or scripts from the ADS controller. When an administrator grants a non-administrator privilege to run a job template, he or she is granting the non-administrator the ability to run that job template on a specific device, a specific set, or any subset of devices from a specific set. Non-administrators can run only the job templates to which they have been assigned permission by an administrator.

Role-based job execution can be used both with and without Microsoft Active Directory® directory service. Administrators typically grant access to a named group or to individual accounts. In an Active Directory environment, for maintenance purposes these accounts should be domain accounts.

Job templates have an optional reference to a device or a set. When an administrator runs a job template, he or she may use this template as a default setting or reassign the job template to a different device. Non-administrators are granted execute privilege only on an assigned device or set of devices. If no device or set assignment exists, the job template cannot be used by a non-administrator.

In general, non-administrators who can log on to the Controller can read and view the job history of every job that has been run, regardless of the permissions assigned to the job template used to start that job. To avoid possible elevation of privilege on the Controller, non-administrators must not be given

modify privileges on image files stored in Image Distribution Service, job templates, XML sequence files, or scripts used by the Controller Service.

Using ADS: Examples

ADS provides a powerful tool set with which to manage deployment and administration of large numbers of Windows-based servers. The brief examples below illustrate the capability of ADS in a number of common procedures.

Image Deployment and Maintenance

Central to ADS functionality is the ability to deploy prepared images to managed devices. ADS can deploy Windows Server 2003 or Windows 2000 Server operating system images without your input.

Bare Metal Server Deployment and Configuration

Initial device deployment showcases one of the two major features of ADS. The following example walks through the procedures needed to install a new device into your environment without help from a system administrator. The commands needed to carry out these tasks are provided, and the example indicates how a user can accomplish the same tasks using the MMC user interface.

Preparing and Capturing an Image

The first step in the process is to capture a working device operating system image that can be deployed to the new hardware. Microsoft provides preparation and imaging tools with which to complete this process.

To capture an image, prepare a working server system that can provide the “gold” image as follows:

1. Install the Windows Server operating system according to your site policy.
 - Install the ADS Administrative agent, which provides post-deployment control capability.
 - Install other Sysprep-compatible applications needed for the image.
 - Run the imaging tools from another bootable operating system partition or a Microsoft Windows Preinstallation Environment (WinPE)⁸ CD.
2. Run Sysprep on the partition to be imaged.

Below is an example Sysprep.inf file for Windows Server 2003. Notice the circumflex (^) characters in this script. These characters enable multiple images to be configured after deployment with individual settings. This file is available, along with some sample Sysprep files, after installing the ADS Controller service in the directory Program Files\Microsoft ADS\Samples\Sysprep.

```
; SetupMgrTag
[Unattended]
    OemSkipEula=Yes
    InstallFilesPath=C:\sysprep\i386
    TargetPath=\WINDOWS

[Gui Unattended]
    AdminPassword=^ADMINPASSWORD^
```

⁸ WinPE is a minimal operating system based on the Windows XP kernel. WinPE functionally replaces DOS and contains the minimum functionality needed to run Windows Setup, scripts, or custom installation and imaging applications.

```

EncryptedAdminPassword=NO
OEMSkipRegional=1
OEMDuplicatorstring=sysprep-i d
TimeZone=4
OemSkipWelcome=1

[UserData]
FullName="ADS"
OrgName="Microsoft"
ComputerName=^ADS_COMPUTER_NAME^
ProductKey=^ADS_WINDOWS_PRODUCT_KEY^

[LicenseFilePrintData]
AutoMode=PerSeat

[Identification]
JoinWorkgroup=^ADS_JOIN_WORKGROUP^

[Networking]
InstallDefaultComponents=Yes

[TapiLocation]
CountryCode=1
Dialing=Tone
AreaCode=425

[Branding]
BrandUsageUnattended=Yes

[Proxy]
Proxy_Enable=0
Use_Same_Proxy=0

[sysprepcleanup]

```

3. Use the ADS image capture utility to create the gold image:
`"imgdeploy /c <volume> <image file> <description>"`
4. Add the prepared image to the ADS Image Distribution Service to make it available for future deployment. From the server running the IDS add the image, type the following:
`"adsimage /add imagename /path imagepath "`

Hardware Configuration

As part of the deployment task sequence, specific hardware configuration tasks can be accomplished using a Virtual Floppy. Using the tools supplied with ADS, an administrator creates a floppy disk containing a subset of MS-DOS Version 6.22 and adds the MS-DOS utilities needed to configure the specific hardware items. The floppy contents are then captured into a single file—the virtual floppy image. That image is then used in the deployment task sequence, where it is downloaded to the device during the PXE boot sequence.

The MS-DOS utilities must be scriptable from the command line, they must automatically execute when the virtual floppy is booted into memory, and they must complete the cycle by rebooting the device so that the deployment can continue with the next task sequence. Using this method, an administrator can configure RAID controllers, flash BIOS items, or perform any other hardware-specific feature for which the manufacturer provides utilities that can be run from this Virtual Floppy.

Deploying and Personalizing Images

After preparing the gold image for deployment, you can prepare ADS to manage the new device as follows:

1. Prepare and install the job template that the Controller Service will use to deploy the image to the new device. Also prepare and install job templates to boot the device into the Deployment Agent and hard disk. For an example of a deployment sequence, see the [“PXE Based Bare Metal Purposing”](#) section later in this paper.
 - From the command line:


```
adsjobtemplate /create newtemplate /description "template desc." /sequence c:/path/filename.xml
```
 - From the MMC user interface, go to **Job Templates** off of the **Console Root**, right-click and choose **Add Job** for each of the three templates.
2. Add the new device to the ADS database, and mark it as a device to control from this controller.
 - From the command line:


```
adsdevice /add devicename /description "text" /mac macaddress /jobtemplate boot-to-da  
adsdevice /tc devicename
```
 - From the MMC user interface, go to **Devices** off of the **Console Root**, right-click and choose **Add Device**. After the device is added, right-click the device name and select **Control**.
3. Provide values for the Sysprep substitutions that are used during deployment.
 - From the command line, type the following:


```
adsdevice /edit devicename /setvar PASSWORD password \  
/setvar PID pid /setvar COMPUTERTNAME computername \  
/setvar WORKGROUP workgroup
```

Note The \ symbol indicates to place all these commands on a single line.

 - From the MMC user interface, right click the added device and choose **Properties**. Select the **User** tab, and then add the four variables and their values.
4. After ensuring that PXE is first in the BIOS boot order, boot the device or devices. This step causes the default job template (Boot-to-da) to be run, which makes the devices boot into the Deployment Agent.
5. If you are deploying to more than one device, create a set comprising the devices, and then run the deployment sequence using one of the following interfaces.
 - To download the image from the command line using multicast, type the following:


```
adsjob /run /set newset /sequence newtemplate
```

If you are deploying to only one device, run the deployment sequence with the following command:

```
adsjob /run /device devicename /sequence newtemplate
```

The image is downloaded using unicast.
 - From the MMC user interface, right-click **Sets**, and add all members to the set.
6. Modify the ADS device record so that subsequent system boots are to the internal hard disk. An

example sequence is listed here.

- From the command line, type the following:
`adsdevice /edit devicename /jobtemplate boot-to-hd`
- From the MMC user interface, right-click **Properties**, and select **boot-to-hd** from the Default Job Template drop-down menu.

Post Operating System Deployment Configuration

Simply getting an operational server image onto new devices is a great accomplishment, but with ADS, an administrator can remotely configure new or existing devices by using the ADS Administration Agent. Obviously, any device deployment fulfills a purpose within the production environment. After that device is given an image, the next task for the administrator is to configure the device for the specific tasks for which it is intended. Here, again, ADS provides a manageable, scalable solution to provide for those post-deployments tasks.

For example, to perform an unattended SQL Server 2000 installation on a newly deployed device:

1. Prepare and install the job template that the Controller Services will use to implement the post-deployment configuration. (For an example script, see the "[Script/Job Remote Execution](#)" section later in this paper.)
 - From the command line, type the following:
`adsjobtemplate /create newtemplate /description "template desc." /sequence c:\path\filename.xml`
 - From the MMC user interface, use the **Job Template Wizard** to help create your new job.
2. Start running the sequence as follows:
 - From the command line, type the following:
`adsjob /run /device devicename /template newtemplate`
 - From the MMC user interface, go to **Job Templates**, right-click and select **Run**.

Mass Server Administration

Sets and jobs are the true administration powerhouses in the ADS arsenal. Using sets, an administrator can issue the same job to one or to thousands of managed devices with the same commands. Today, with a standard Windows environment, an administrator generally provides device-by-device command execution through the console, Terminal Services, or a third-party remote control interface. These methods do not create a system-generated audit trail, and command consistency is not guaranteed.

Supported Jobs and Scripts

An ADS job can be either an internal command or external programs available to a managed device. Internal commands are those that ADS provides during task sequence execution. Operations can be performed using internal commands in the Administration Agent, including the following:

- Reboot the device.
- Shutdown the device.

In the Deployment Agent, operations that can be performed using internal commands include:

- Partition the disk or list partition information.
- Set a registry key on a newly downloaded image.
- Transfer a file to a newly downloaded image.
- Replace strings in a file on the device.

A job can also be any Windows application or script that is valid for the target device. The job could be a script sent from the Controller, a script or application that is already present on the device, or a script or application that is available to the ADS Administration Agent via a network share. In the post-deployment example, a script was loaded from an admin share.

Running Jobs and Scripts

Below is a typical administrative task that is performed as a job. In this example, our device executes the **ipconfig** command and returns the results:

```
C:\>adsjob /run /device ADS-Device /devicepath
c:\windows\system32\ipconfig.exe /all
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights
reserved.
Job started with job id 14
```

This command sequence was issued to one device, the job requiring it to execute the **ipconfig** command. By modifying one item on the command line, we could easily have hundreds or thousands of devices all execute the same command .

```
C:\>adsjob /run /set DATACENTER /devicepath c:\windows\system32\ipconfig.exe /all
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
Job started with job id 15
```

With the preceding command, each device in the set executes the same task sequence and reports its results to the ADS Controller, which logs every action into the database. ADS provides a complete audit trail for every action that is performed on each device. Because ADS also provides SSL security, every command issued and the results returned are also encrypted between the managed device and Controller Service.

Job commands may also use device variables that are stored in the ADS Controller Service. When running the same job against many devices, you often need a replaceable parameter to individualize each device correctly. You can create job variables initially with the device record or later using the command-line tools, GUI, or WMI interface.

Capturing and Logging Results

As each job is sent to the Controller Service for execution, it is issued a job ID that uniquely identifies it. When a job is sent to a set rather than to an individual device, a hierarchical structure is created. The

initial job ID returned when the command is run is referred to as the *parent* job ID. At the Controller, each device within the set has a separate job created for it, and a distinct job ID is assigned for each device. The administrator can track both the initial set job ID and each job ID. The **adsjob** command-line utility and the GUI and WMI interface provide job status and results. The results from the preceding examples are as follows.

```
C:\>adsjob /result 14 /full /verbose
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
Device: ADS-Device
Status: Succeeded
Output:
Windows IP Configuration
    Host Name . . . . . : ads-device
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Unknown
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix . :
    Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Physical Address. . . . . : 00-50-56-7F-99-1B
    DHCP Enabled. . . . . : No
    IP Address. . . . . : 192.168.100.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

The results from the job run against the set are as follows.

```
C:\>adsjob /result 15 /full
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
Device      Status      Output      Error      Exit Code
-----
ADS-Device  Succeeded  \n\nWindows IP      0
ADS-Device1 Succeeded  \n\nWindows IP      0
```

Example Task Sequences

The examples below are described earlier in this article. After installing ADS, you can find these and additional sample task sequences in the Program Files\Microsoft ADS\Samples directory.

PXE-Based Bare Metal Purposing

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!--
```

```
This file is part of the Microsoft ADS Samples
```

```
Copyright (c) Microsoft Corporation. All rights reserved.
```

```
Sample sequence to purpose a device that is currently running the
Deployment Agent and configure it to connect to a workgroup.
```

```
The image to be provisioned must have been captured with a Sysprep.inf
containing the strings ^ADS_WINDOWS_PRODUCT_KEY^ ^ADMINPASSWORD^
etc. as in the personalize step below.
```

```
Device variables must have been created for the devices as in the
personalize step below.
```

```
To use: Boot a target into the Deployment Agent, then run this sequence
against that device.
```

```
Assumes that the target will have ADS Administration Agent running
once in full-os.
```

```
If the Windows directory in your image is not \Windows,
you must update the relevant paths below. For Windows 2000
installations, the default Windows directory path is \WinNT.
```

```
-->
```

```
<!-- start sequence -->
```

```
<sequence command="da-deploy-image-wg.xml" description="Deploy an image and configure
the machine to join workgroup"
  xmlns="http://schemas.microsoft.com/ads/2003/sequence" version="1">
```

```
<!-- STEP 1 Create a single 5000 MB partition on the disk -->
```

```
<task description="Partition the disk">
```

```
  <command>/bmonitor/bmpart.exe</command>
```

```
  <parameters>
```

```
    <parameter>\device\harddisk0</parameter>
```

```
<!-- selects harddisk0 -->
```

```
    <parameter>/init</parameter>
```

```
<!-- erases all partitions
on harddisk0 -->
```

```
    <parameter>/C: 5000</parameter>
```

```
<!-- creates new partition
(#1) of size 5000 MB -->
```

```
    <parameter>/A</parameter>
```

```
<!-- activate the newly
created partition (#1) -->
```

```
  </parameters>
```

```
</task>
```

```
<!-- STEP 2 Download images -->
```

```
<task description="Download image">
```

```
  <command>/imagng/iigbmdeploy.exe</command>
```

```
  <parameters>
```

```
    <parameter>imagename</parameter>
```

```
<!-- name of the image to be
deployed-->
```

```
    <parameter>\device\harddisk0\partition1</parameter>
```

```
<!-- deploy the image to
partition1 -->
```

```
    <parameter>-r</parameter>
```

```
<!-- specifies deploy
mode-->
```

```

    <parameter>-client</parameter>                                <!-- required parameter -->
  </parameters>
</task>

<!-- STEP 3 Personalize the Sysprep.inf file -->
<task description="Set Sysprep custom info in the Sysprep.inf file">
  <command>/bmonitor/bmstrrep.exe</command>
  <parameters>
    <parameter>\device\harddisk0\partition1\sysprep\sysprep.inf</parameter>
    <parameter>^ADS_WINDOWS_PRODUCT_KEY^</parameter> <!-- key to be searched
                                                         in Sysprep.inf file -->
    <parameter>$ProductKey$</parameter>                <!-- update value -->
    <parameter>^ADMINPASSWORD^</parameter>
    <parameter>$adminpassword$</parameter>
    <parameter>^ADS_COMPUTER_NAME^</parameter>
    <parameter>$machinename$</parameter>
    <parameter>^ADS_JOIN_WORKGROUP^</parameter>
    <parameter>WORKGROUP</parameter>
  </parameters>
</task>

  <!-- STEP 4 Set Controller IP -->
  <task description="Set controller IP address">
    <command>/bmonitor/bmsetreg.exe</command>
    <parameters>
      <parameter>-
h: \device\harddisk0\partition1\windows\system32\config\system</parameter> <!-- NOTE:
                                                         for Windows 2000, change "windows" to "winnt" -->
      <parameter>control set001\control\bmss</parameter> <!-- registry path -->
      <parameter>controlleripaddress</parameter>        <!-- registry key
                                                         to update -->
      <parameter>reg_multi_sz</parameter>
      <parameter>$controller.system.adminipaddr$</parameter> <!-- registry
                                                         value -->
    </parameters>
  </task>

  <!-- STEP 5 Set BMDP port number -->
  <task description="Set BMDP port number">
    <command>/bmonitor/bmsetreg.exe</command>
    <parameters>
      <parameter>-
h: \device\harddisk0\partition1\windows\system32\config\system</parameter> <!-- NOTE:
                                                         for Windows 2000, change "windows" to "winnt" -->
      <parameter>control set001\control\bmss</parameter> <!-- registry path -->
      <parameter>bmdpport</parameter>                   <!-- registry key
                                                         to update -->
      <parameter>reg_dword</parameter>
      <parameter>$controller.system.bmdpport$</parameter> <!-- registry value -->
    </parameters>
  </task>

  <!-- STEP 6 Set BMCP port number -->
  <task description="Set BMCP port number">
    <command>/bmonitor/bmsetreg.exe</command>
    <parameters>
      <parameter>-
h: \device\harddisk0\partition1\windows\system32\config\system</parameter> <!-- NOTE:
                                                         for Windows 2000, change "windows" to "winnt" -->
      <parameter>control set001\control\bmss</parameter> <!-- registry path -->
      <parameter>bmcpport</parameter>                   <!-- registry key
                                                         to update -->
      <parameter>reg_dword</parameter>
      <parameter>$controller.system.bmcpport$</parameter> <!-- registry value -->
    </parameters>
  </task>

  <!-- STEP 7 Set Device AdminMAC -->
  <task description="Set device Admin MAC Address">
    <command>/bmonitor/bmsetreg.exe</command>
    <parameters>
      <parameter>-

```

```

h: \device\harddisk0\partition1\windows\system32\config\system</parameter> <!-- NOTE:
                                for Windows 2000, change "windows" to "winnt" -->
    <parameter>control set001\control\bmss</parameter> <!-- registry path -->
    <parameter>bindexcept</parameter> <!-- registry key
                                        to update -->
    <parameter>reg_multi_sz</parameter>
    <parameter>mac=$device.system.adminmac$</parameter> <!-- registry value -->
  </parameters>
</task>

<!-- STEP 8 set BMSS bind policy -->
<task description="Set BMSS bind policy">
  <command>/bmonitor/bmsetreg.exe</command>
  <parameters>
    <parameter>-
h: \device\harddisk0\partition1\windows\system32\config\system</parameter>
    <parameter>control set001\control\bmss</parameter>
    <parameter>bindpolicy</parameter>
    <parameter>reg_dword</parameter>
    <parameter>0</parameter>
  </parameters>
</task>

<!-- STEP 9 Copy public key to target -->
<task description="Copy public key certificate file to target">
  <command>/bmonitor/bmfilexfer.exe</command>
  <parameters>
    <parameter>-d</parameter> <!-- Download the file from
                                Controller to target -->
    <parameter>"c:\program files\Microsoft ADS\certificate\adsroot.cer"</parameter>
  <!-- source file -->
    <parameter>\device\harddisk0\partition1\windows\temp\adsroot.cer</parameter>
  <!-- destination file NOTE: for Windows 2000, change "windows" to "winnt" -->
  </parameters>
</task>

<!-- STEP 10 Set BMSS public key certificate -->
<task description="Set the BMSS public key certificate">
  <command>/bmonitor/bmsetreg.exe</command>
  <parameters>
    <parameter>-
h: \device\harddisk0\partition1\windows\system32\config\system</parameter> <!-- NOTE:
                                for Windows 2000, change "windows" to "winnt" -->
    <parameter>-f</parameter> <!-- read registry value string
                                from file -->
    <parameter>-r</parameter> <!-- remove file after
                                inserting into registry -->
    <parameter>control set001\control\bmss\bmcpcertificates</parameter> <!--
                                registry path -->
    <parameter>AgentCert</parameter> <!-- registry key to update -->
    <parameter>reg_binary</parameter>
    <parameter>\device\harddisk0\partition1\windows\temp\adsroot.cer</parameter>
                                <!-- registry value read from file
                                NOTE: for Windows 2000, change "windows" to "winnt" -->
  </parameters>
</task>

<!-- STEP 11 Reboot the machine -->
<task doesReboot="true" description="Reboot">
  <command>/bmonitor/reboot</command>
</task>

<!-- STEP 12 When PXE request comes in, boot to hard disk (starts mini-setup) -->
<task doesReboot="true" description="Boot to hard disk">
  <command>/pxe/boot-hd</command>
</task>

<!-- STEP 13 At end of mini-setup, machine reboots, so when PXE request comes in,
                                boot to the hard disk (start the newly installed operating system) -->
<task description="Boot to hard disk">
  <command>/pxe/boot-hd</command>
</task>

```

```

<!-- STEP 14 Update device record to always boot to hard disk in the future -->
<!-- ADS Install folder must be updated if you installed ADS to a folder -->
<!-- other than "C:\program files\Microsoft ads" -->
<task description="Set default job template as boot to hard disk">
  <command target="controller">C:\Program Files\Microsoft
ADS\tools\adsdevice.wsf</command>
  <parameters>
    <parameter>-e</parameter>
    <parameter>$Device.System.Name$</parameter>
    <parameter>-jobtemplate</parameter>
    <parameter>boot-to-hd</parameter>
  </parameters>
</task>

</sequence>

```

Script/Job Remote Execution

```

<?xml version="1.0" encoding="utf-8" ?>
<!--
  ADS Sample code

  Copyright (c) 2003 Microsoft Corporation

  Download and run script
-->
<sequence command="sample1.xml" version="1"
  xmlns="http://schemas.microsoft.com/ads/2003/sequence">
  <task timeout="600" description="Install SQL Server 2000">
  <command delivery="bmcps">\\AdminServer\MS-SQL2000-CD\Sqlins.bat</command>
</task>
</sequence>

```

Summary

Enterprise data center administrators face significant challenges as they scale their infrastructures and deploy larger numbers of Windows servers in their environments. Automated Deployment Services is a new component of Windows Server 2003 that helps customers address those challenges using the following features:

- The ADS Controller and NBS deployment automation supports operating system and application installation to dozens or even hundreds of bare metal systems in a manner that supports security and auditing without administrator input.
- Task sequence jobs issued to individual devices or sets of devices provide a level of automation previously unattainable. Organizations can now embody their operational best practices in discrete software modules and reliably issue the same commands repeatedly with consistent execution.
- New imaging tools delivered with ADS can quickly create gold-standard system image libraries and enable individual images to be updated without having to be recaptured on the target system.
- Through command-line operations, MMC user interface snap-ins, or a complete WMI interface, administrators can use familiar and comfortable management styles to control vast numbers of servers securely and with a complete audit trail.

ADS enables you to build a rock-solid foundation for deploying and administering Windows-based applications and services.

Related Links

See the following resources for further information:

- Developer downloads, including WMI resources, at <http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/msdn-files/027/001/566/msdncompositedoc.xml>
- Security Services in Windows Server 2003 at <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnolog/windowsserver2003/technologies/security/default.asp>
- IP multicast at <http://support.microsoft.com/default.aspx?scid=kb;en-us;165011>
- Differences between multicast and unicast at <http://support.microsoft.com/default.aspx?scid=kb;en-us;291786>
- Microsoft XML resources at <http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000438>
- Windows scripting resources at <http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28001169>
- Cloning Windows 2000 and Using Sysprep at <http://support.microsoft.com/?kbid=325554>
- Microsoft X.509 certificate resources on the Digital Identity, Authorization, and Authentication page at <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/certauth/default.asp>
- SQL Server 2000 Databases on the desktop at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_sa2_9gz4.asp?frame=true
- SQL Server 2000 unattended installations at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/instdsql/in_runsetup_8ege.asp
- Windows WinPE at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xpehelp/html/xetbswindowspreinstallationenvironment.asp>

For the latest information about Windows Server 2003, see the [Windows Server 2003 Web site](http://www.microsoft.com/windowsserver2003) at <http://www.microsoft.com/windowsserver2003>.