Table of Contents

Appendix 1

Appendix 2

Appendix 3

Appendix 4

Appendix 5

Appendix 6

Appendix 1.1 – Glossary of terms

Name	Description
AA Administration Agent	A program run by a computer running Windows that allows ADS to control the computer
ADS Automated Deployment Services	The Windows Server 200 3 services the website interacts with
Computer	Either a desktop computer or data centre server
Customer	A person who is paying for the use of the ILS labs
DA Deployment Agent	A program loaded in to a RAM drive when a computer PXE boots to allow ADS to control the computer
ILS Team Infrastructure and Laboratory Support team	The department at Microsoft UK the program in this project is primarily being produced for
IIS	Internet Information Services
MSDN	Microsoft Developer Network
NTFS	New Technology File System
OS Operating System	A large software program that helps a user to interface with a computer and allows the running of additional programs
PKI	Public Key Infrastructure
PXE Pre-Boot eXecutable Environment	Allows programs to be run prior to the computers operating system loading
RIS Remote Installation Services	A deployment technology designed for use with Windows Server 2000
Technician	A person who is responsible for the computers maintenance, including data centre workers and system administrators
User	Any person who uses the website
VLAN	Virtual Local Area Network
Windows	Refers to Windows XP, Windows Server 2000 and Windows Server 2003 unless otherwise stated
WMI Windows Management Instrumentation	Allows accesses to management information in an enterprise environment

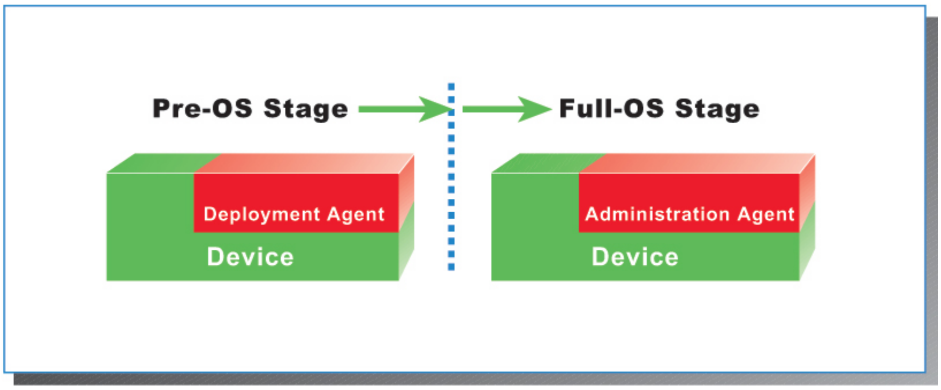
Appendix 1.2 – Further explanation of ADS

Microsoft Automated Deployment Services (ADS) is a Microsoft product that runs on Microsoft Windows Server 2003. ADS is described as a 'push' deployment system as the central server has control over all of the client computers, or nodes, entered into the ADS system.

When the nodes are in a Windows environment, the ADS server uses a program called the 'Administration Agent' (AA) to send commands and receive feedback. The ADS server uses a Public Key Infrastructure (PKI) generated certificate to prove its identity; this is used by the nodes to ensure they only execute commands from the company's ADS server and not an unauthorized server.

When nodes are not in the Windows environment they are controlled with a program called the 'Deployment Agent' (DA). This program is sent over the network and loaded into a RAM drive when the node enters the Pre-Boot eXecutable Environment (PXE). Like the Administration Agent, the Deployment Agent allows the ADS server to send command and receive feedback from the nodes. As the DA is loaded before an Operating System (OS), the ADS server has raw access to the nodes hard disk drives and can repartition, format and read/write data to the disk.

The two programs are represented graphically in Extract A1.1



Extract A1.1: ADS Agents (Microsoft, 2003)

A very basic flow of the process a node goes through when it boots is shown in Figure A1.1. Some steps are missing such as the DHCP process and other loading tasks as they are not fully appropriate to this document; the diagram only represents the paths a node could follow.

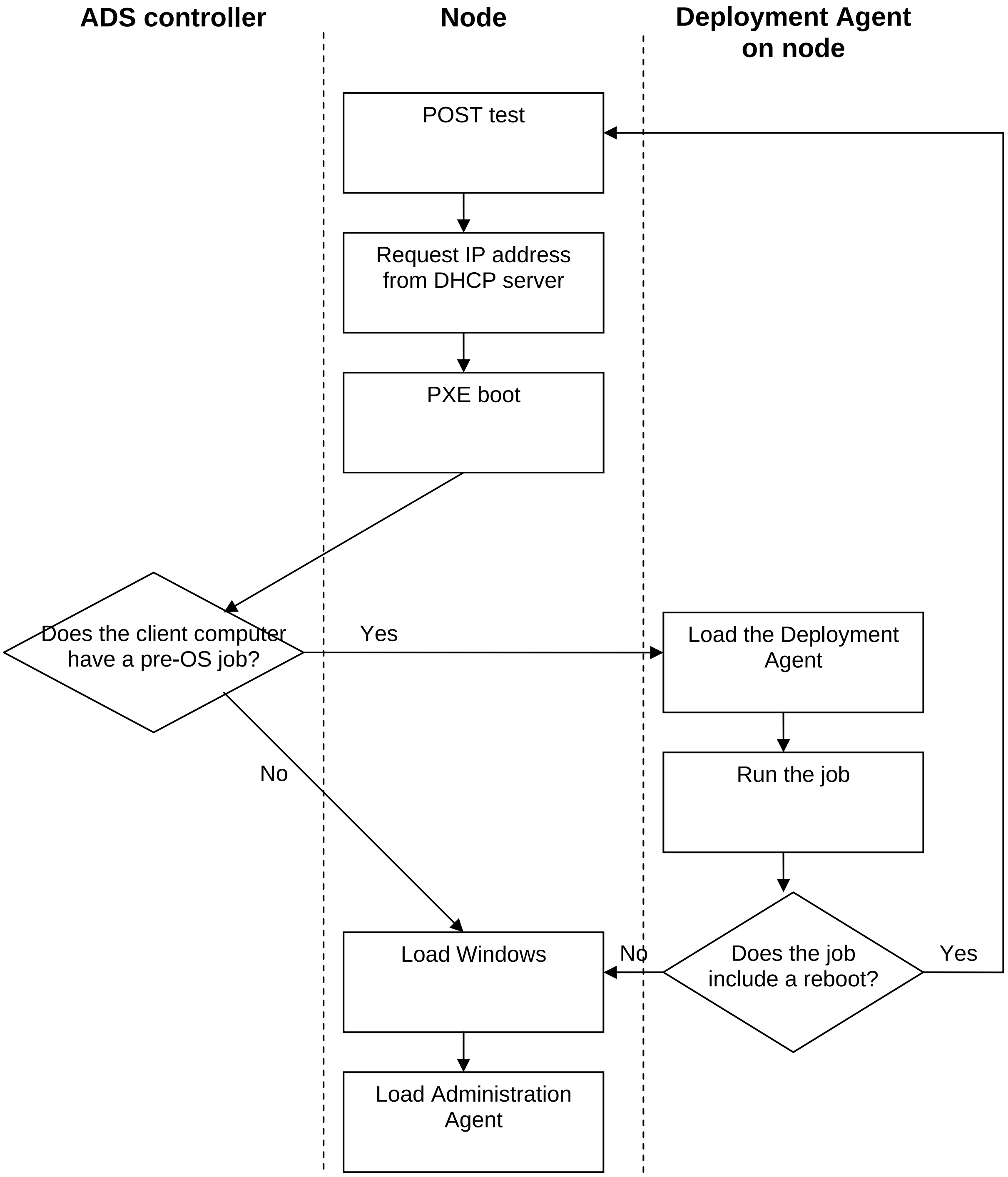

ADS controller
Node
Deployment Agent on node
POST test
Request IP address from DHCP server
PXE boot
Does the client computer have a pre-OS job?
Yes
No
Load the Deployment Agent
Run the job
Load Windows
No
Does the job include a reboot?
Yes
Load Administration Agent

Figure A1.1: Node boot process with Automated Deployment Services

Jobs can be set to run on a node via the command line, an MMC snap-in or by using WMI on the ADS server. A job can be set to run on single or multiple nodes simultaneously. Once the ADS server has been instructed to run a job it is placed in the 'Running Jobs' list. The ADS server polls the node(s) to see if it is in a suitable state to run the job, i.e. currently in the DA or connected via the AA, and sends commands to execute if it is. If the node(s) is currently rebooting, busy or otherwise unavailable, the ADS server will keep the job in the 'Running Jobs' list and wait for the node(s) to become available.

An ADS job is an XML file that describes a number of tasks, or steps, which should be taken by the node(s). These XML files will typically be stored on the ADS sever which will process them and send commands to the node(s), the XML files themselves are not sent. An example ADS job can be seen below in Extract A1.2.

```
<!-- 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>/imaging/imgbmdeploy.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>
</sequence>
```

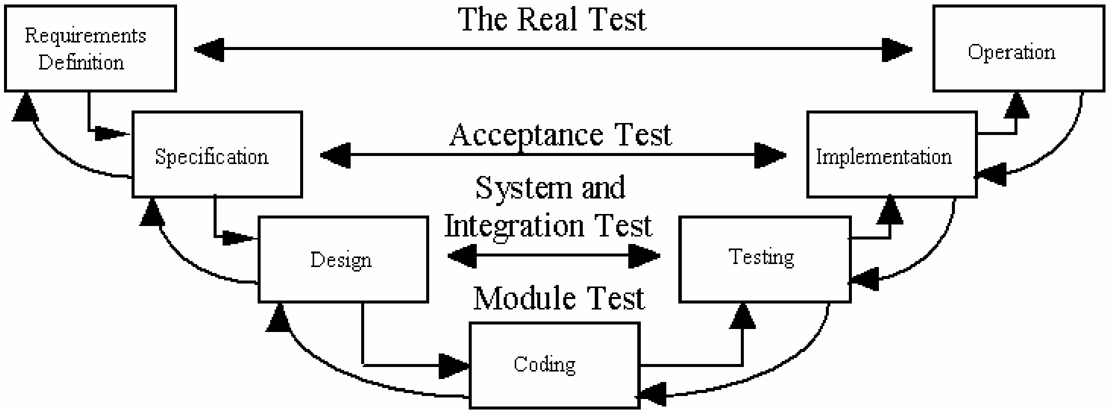
Extract A1.2: Example ADS Job (Microsoft, 2003)

The node(s) must be in the DA for this job to be completed. In Extract A1.2 it is assumed that the node(s) is already in the DA as the extract does not contain a task to do this.

The first task of this job is for the node(s) to repartition its hard disk drive. The ADS server will process the XML file and send the command
“/bmonitor/bmpart.exe \device\harddisk0 /C:5000 /a”
The program bmpart.exe will have been downloaded with the DA and placed in the RAM drive. The node(s) will run the command and report back to the ADS sever when the job has either succeeded or failed.

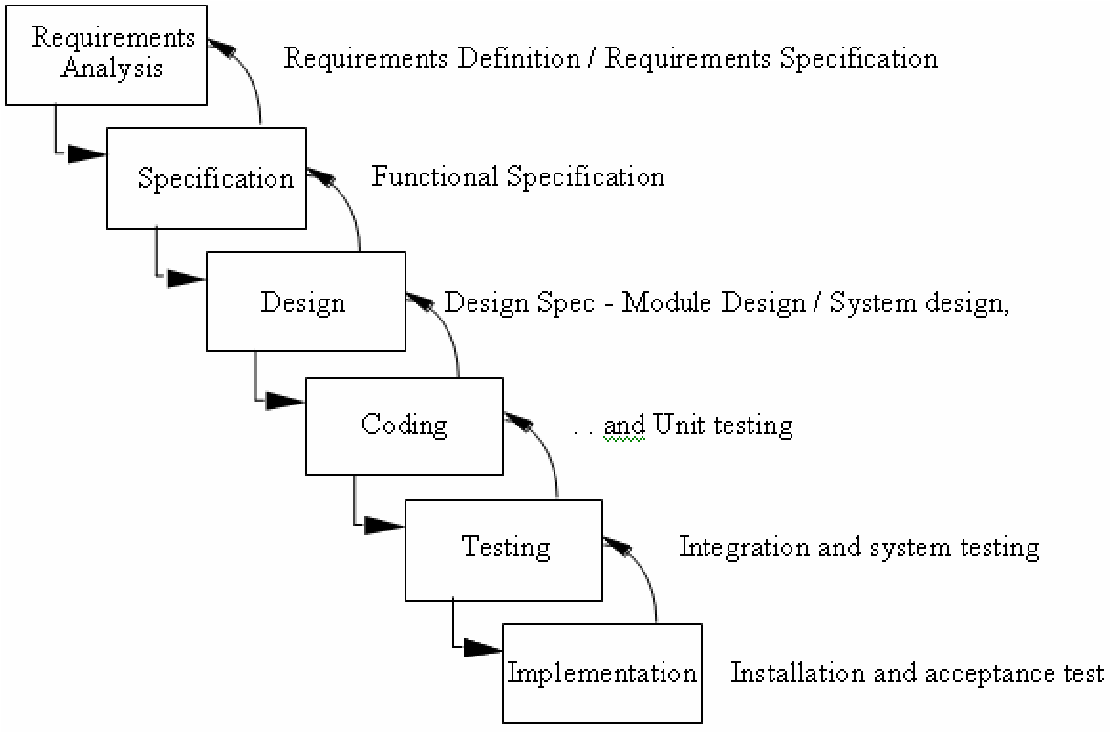
Once all of the tasks in a job have successfully been executed on a node, the ADS server will move the job from the ‘Running Jobs’ list to the ‘History’ list and mark the job as ‘Successful’. If a node reports that a task failed, the job will be moved from the ‘Running Jobs’ list to the ‘History’ list but will be marked as ‘Failed’. (Microsoft, 2003)

Appendix 1.3 – The V process model



(Chris. W. Dawson, 2004)

Appendix 1.4 – The Waterfall model


Requirements Analysis
Requirements Definition / Requirements Specification
Specification
Functional Specification
Design
Design Spec - Module Design / System design,
Coding
. . and Unit testing
Testing
Integration and system testing
Implementation
Installation and acceptance test

(Chris. W. Dawson, 2004)

Appendix 1.6 – Revised Gantt chart

	03/10/2004	10/10/2004	17/10/2004	24/10/2004	31/10/2004	07/11/2004	14/11/2004	21/11/2004	28/11/2004	05/12/2004	12/12/2004	19/12/2004	26/12/2004	02/01/2005	09/01/2005	16/01/2005	23/01/2005	30/01/2005	06/02/2005	13/02/2005	20/02/2005	27/02/2005	06/03/2005	13/03/2005	20/03/2005	27/03/2005	03/04/2005	10/04/2005	17/04/2005	24/04/2005
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	Week 24	Week 25	Week 26	Week 27	Week 28	Week 29	Week 30
Introduction		■	■																											
Requirements Definition				■	■		▨																							
Specification						■	■		▨																					
Research								■	■		▨																			
Design										■	■		▨			▨														
Coding												■	■	■	■	■	■		■		▨									
Testing																		■		▨										
Evaluation																								■	■					
User Manual																				▨	▨	■	■							
Finishing up																										■	■	■	■	■

■ Major work
▨ Minor work

Appendix 2.1 – Further explanation of Remote Installation Services

Microsoft Remote Installation Services (RIS) is a Microsoft product that runs on Window Server 2000 and 2003. RIS is described as a ‘pull’ deployment system as the installation process is user initiated.

When a client computer, or node, boots the following process is followed. Some steps are missing such as the full Dynamic Host Configuration Protocol (DHCP) process and other loading tasks as they are not fully appropriate to this document; the diagram only represents the paths a node could follow.

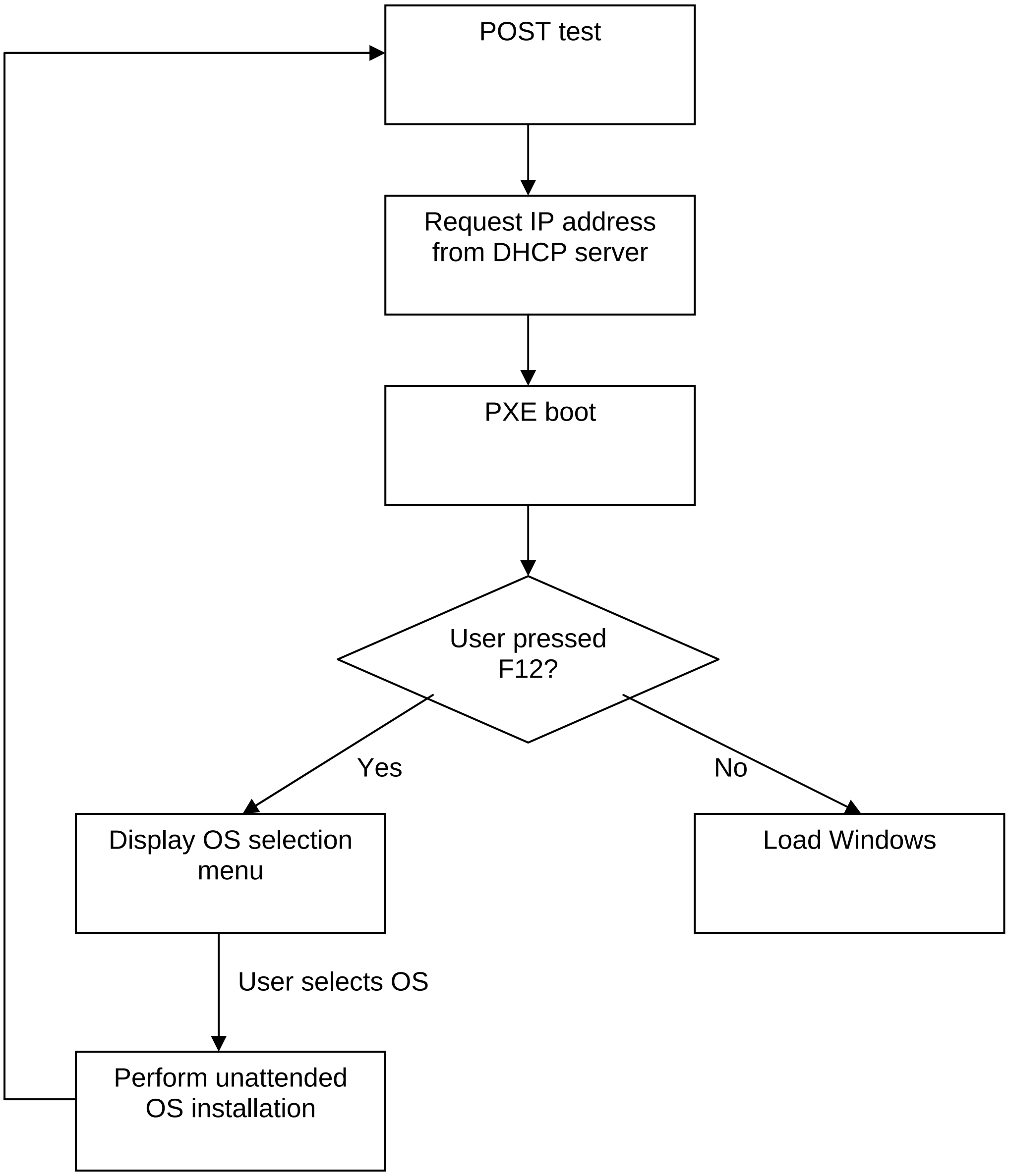

POST test
Request IP address from DHCP server
PXE boot
User pressed F12?
Yes
No
Display OS selection menu
Load Windows
User selects OS
Perform unattended OS installation

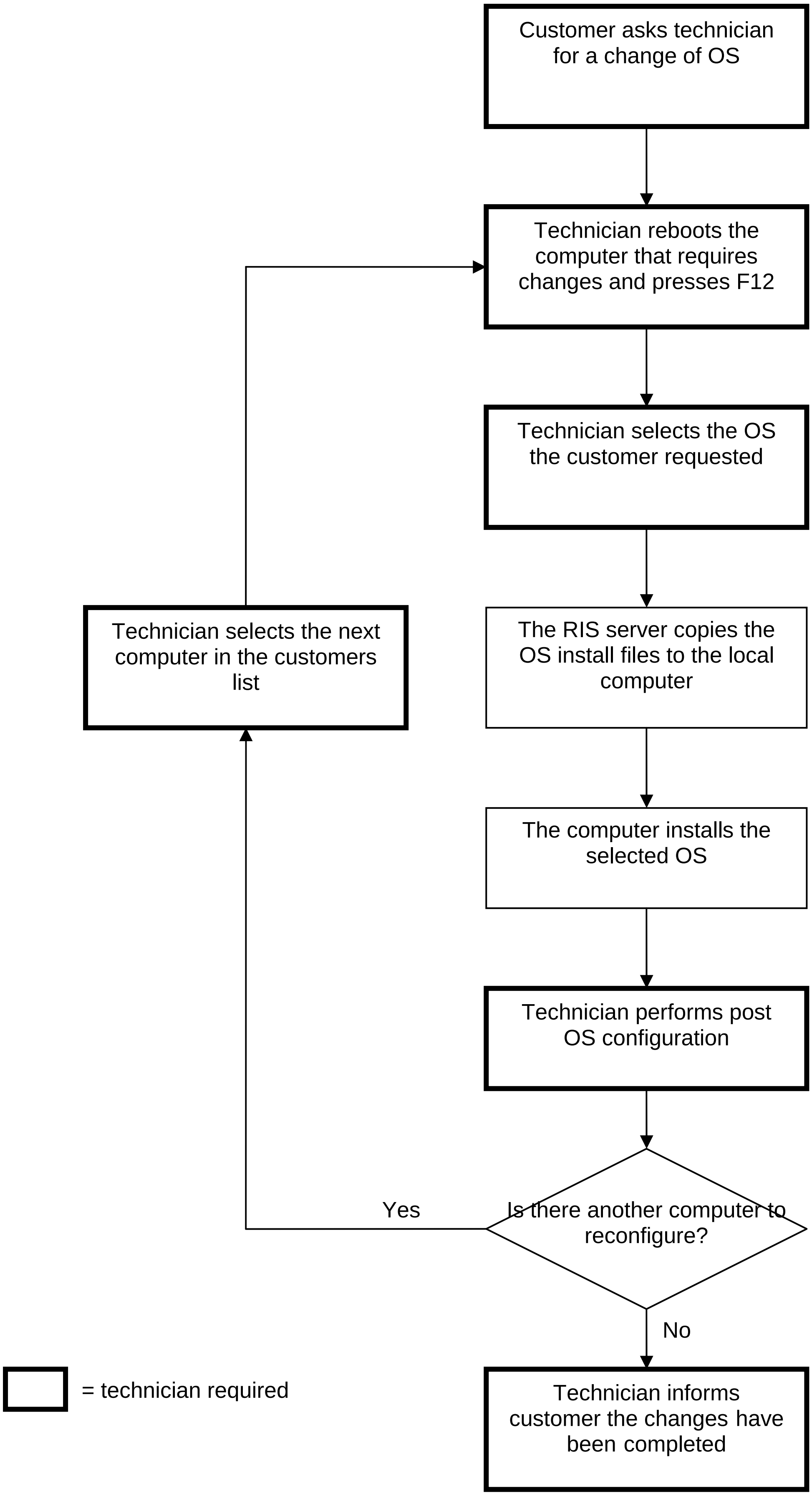
Figure A2.1.1: The process of installing an OS using RIS

When the node PXE boots the user has two options, to press F12 or not. If the user does not press F12 then the computer loads Windows as normal. If the user does press F12 they are presented with a list of OSs that can be installed. Once the user selects the OS they want to install from the list, the

RIS server copies the Windows source files for the selected OS to that node's hard disk drive. Once the files have finished copying, the RIS server starts the installation process for the OS on the node.

RIS has no way to control the nodes connected to the network. The only way a RIS sever can copy files to a node is for the user to initiate the process by pressing F12 when the node PXE boots. (Microsoft, 2003)

Appendix 2.2 – Current Infrastructure and Laboratory Support (ILS) work flow


Customer asks technician for a change of OS
Technician reboots the computer that requires changes and presses F12
Technician selects the OS the customer requested
The RIS server copies the OS install files to the local computer
The computer installs the selected OS
Technician performs post OS configuration
Is there another computer to reconfigure?
Yes
Technician selects the next computer in the customers list
No
Technician informs customer the changes have been completed
= technician required

Appendix 2.3 – Times for current Infrastructure and Laboratory Support (ILS) work process

The following data is the observed amount of time it took to complete each step of the current working process of the Microsoft ILS team, see Appendix 2.2. The OS that could have been chosen by the customer was any ‘flavour’ of Windows 2000, Windows XP or Windows 2003.

Reboot to RIS select screen (Average 128 seconds)
Table A2.3.1 shows the time in seconds it took a technician to reboot a computer to the RIS selection screen.

53s	152s	61s
258s	174s	154s
328s	48s	58s
164s	48s	44s
	Average	128 seconds

Table A2.3.1: Data collected on time taken for a technician to reboot a computer to RIS select screen

The amount of time it took a technician to reboot a computer fell in to three very distinct categories.

No problem
If the technician did not have any difficulties then the reboot took between 44-58 seconds.

Did not PXE boot
If the computer did not PXE boot when it was restarted then the technician had to go back to the computer, insert a PXE boot floppy disk and reset the computer. This would then force the computer to PXE boot. If this problem occurred the technician took between 152-164 seconds

Multiple problems
If the computer did not PXE boot multiple times, the technician could not find a PXE boot floppy disk or the computer was locked and they could not reboot it, then the technician took over 258 seconds to reboot the computer.

Select requested OS (Average 46 seconds)
Table A2.3.2 shows the average time for a technician to select the OS the customer chose was 46 seconds. There was very little variation in this as the selection process is identical for each OS.

43	42	45
46	52	49
44	44	47
46	51	43
	Average	46

Table A2.3.2: Data collected on time taken for a technician to select an OS

Copy OS files to the local computer (Average 327 seconds)
Table A2.3.3 shows the time it took each computer to copy the OS install files to the local computer.

346	312	376
308	332	296
343	302	294
376	312	322
	Average	327

Table A2.3.3: Data collected on time taken for a computer to copy the OS files

A number of things were identified that resulted in a variation in the time it took to copy the files to the local computer such a the network traffic, the number of other computers using the RIS server, the age of the computer (new computers have faster and better quality parts) and the operating system being copied (Windows 2003 has a slightly more files that need to be copied).

Install the selected OS (Average 1804 seconds)
Table A2.3.4 shows the amount of time it took for each OS to install on the computer.

1822	1687	1948
1922	1434	1512
1784	2004	1952
1889	1957	1743
	Average	1804

Table A2.3.4: Data collected on time taken for a computer to install an OS

Perform post OS configuration (Average 173 seconds)
Table A2.3.5 shows the amount of time it took for the technician to perform the standard post OS configuration of changing the background picture, changing the screen resolution, setting static IP addresses and showing the network icon in the task bar.

162	156	186
168	177	174
192	184	179
165	167	175
	Average	173

Table A2.3.5: Data collected on time taken for a technician to complete post OS configuration

Other steps
Other tasks in the process, such as looking at which computer needs to be reconfigured next and telling the customer the computer is ready, took a negligible amount of time, so are not included in this study of timings. The total average time to reconfigure a computer using the current ILS process was 2478 seconds, or 42.3 minutes.

Appendix 2.4 – Analysis of current Infrastructure and Laboratory Support (ILS) work process

Appendix 2.3 has shown us that on average a technician in the ILS team will spend 3 minutes of their time pre-installing a new OS and 3 minutes of their time doing post OS configuration. These values allow us to calculate the maximum number of computers a single technician can reconfigure per hour.

$$\frac{\text{6 minutes/computer}}{\text{60 minutes/hour}} \quad = \text{10 computers/hour}$$

Time	0-	3-	6-	9-	12-	15-	18-	21-	24-	27-	30-	33-	36-	39-	42-	45-	48-	51-	54-	57-
Computer 1	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥									
Computer 2		🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥								
Computer 3			🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥							
Computer 4				🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥						
Computer 5					🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥					
Computer 6						🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥				
Computer 7							🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥			
Computer 8								🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥		
Computer 9									🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	
Computer 10										🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥

🟥 = engineer required 1 member of staff
🟩 = unattended

Table A2.4.1: Theoretical maximum installations with one technician

Table A2.4.1 is only a theoretical maximum as the calculations have show that the unattended stage of the installation process actually takes 42 minutes, so only 5 of the computers would have finished installing the OS within one hour; this is shown in Table A2.4.2.

Time	0-	3-	6-	9-	12-	15-	18-	21-	24-	27-	30-	33-	36-	39-	42-	45-	48-	51-	54-	57-
Computer 1	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥				
Computer 2		🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥			
Computer 3			🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥		
Computer 4				🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	
Computer 5					🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥

🟥 = engineer required 1 member of staff
🟩 = unattended

Table A2.4.2: Actual maximum installations with one technician

If a situation arises where a customer needs 15 machines reconfigured then it would be sensible to assume that three members of staff could be used to install three lots of five machines in parallel, as shown in Table A2.4.3.

Time	0-	3-	6-	9-	12-	15-	18-	21-	24-	27-	30-	33-	36-	39-	42-	45-	48-	51-	54-	57-
Computer 1	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥				
Computer 2		🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥			
Computer 3			🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥		
Computer 4				🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	
Computer 5					🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥
Computer 6	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥				
Computer 7		🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥			
Computer 8			🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥		
Computer 9				🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	
Computer 10					🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥
Computer 11	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥				
Computer 12		🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥			
Computer 13			🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥		
Computer 14				🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	
Computer 15					🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥

🟥 = engineer required 3 members of staff
🟩 = unattended

Table A2.4.3: Showing three technicians reconfiguring 15 computers

Although the data in Table A2.4.3 is true from what we have seen so far, performing this many simultaneous OS installations and each installation still taking 42 minutes is not possible due to the limitations of the network card. The most common speed network is currently 10/100Mb Ethernet, although high-tech data centres are moving towards 10/100/1000Mb Ethernet. As 10/100Mb Ethernet is the most common network speed, and is also the speed of the network card in the ILS RIS server, 100Mb will be used as the maximum speed of the network in the following calculations.

The theoretical maximum speed to copy OS install files can be calculated as follows

$$\frac{\text{600MB (Windows CD)}}{\text{100Mbits/second}} = \frac{\text{4800Mbits}}{\text{100Mbits/second}} = \text{48 seconds}$$

The observed speed of copying OS install files (Average) can be calculated as follows

$$\frac{\text{4800Mbits}}{\text{X Mbits/second}} = \text{327 seconds}$$

$$\frac{\text{4800Mbits}}{\text{327 seconds}} = \text{X Mbits/second} = \text{14.7Mbits/second}$$

Since the observed download speed for an average computer was 14.7Mbits/second, the maximum number of computers that can simultaneously download files from a RIS server without impacting the time taken for another computer to download the install files can be calculated as follows.

$$\frac{100 \text{ Mbits/second}}{14.7 \text{ Mbits/second}} = 6.8 = 6 \text{ computers}$$

It can be seen that it is not possible to simply increase the number of technicians to perform the number of configuration changes required in an hour, as when there are six installation instances each additional instance will increase the amount of time taken for other computers to download the installation files.

It is possible to create a formula that will calculate the amount of time taken to configure X computers, given that we have an unlimited number of staff who can start the configuration process. We assume that hardware can run at its quoted top speed and we assume the observed values in Appendix 2.3 are correct.

$$\text{Time for X computers} = \frac{\text{technicians time + file copy time + install time}}{60}$$

$$= \frac{(173 + 46 + 128) + (4800/(100/X)) + (1804)}{60}$$

$$= \frac{2151 + 48X}{60}$$

$$= 35.85 + 0.8X \text{ minutes}$$

Another limitation that should be taken into account with this formula is that RIS only supports up to 75 deployments at a time (Microsoft, 2003). If 100 computers were required to be reconfigured then 25 of the computers would have to wait for the first 75 to finish copying before they could start copying the install files.

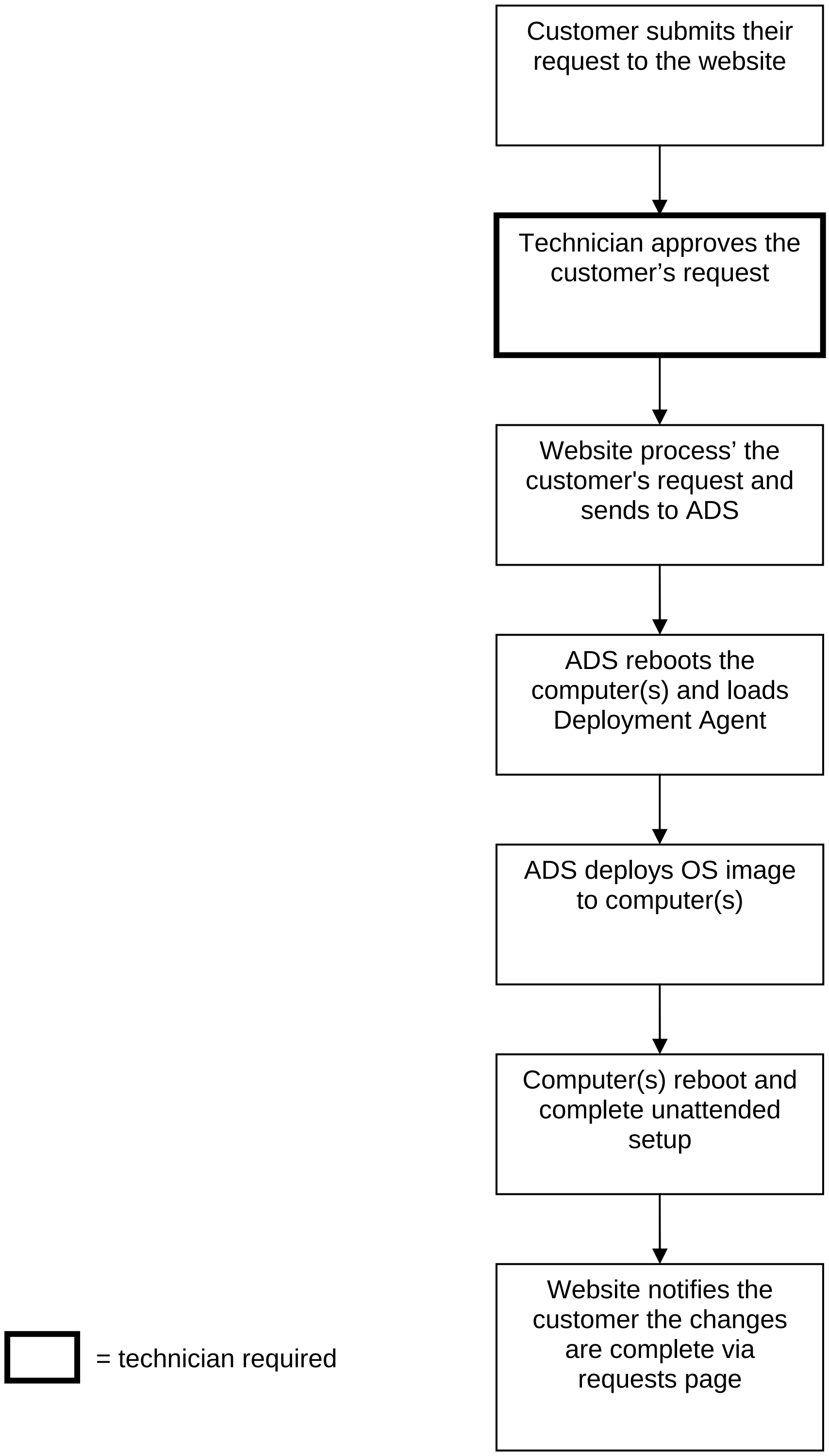
The maximum time it took to reconfigure a computer using the RIS system was

RIS(MAXIMUM TIME) = 328 + 376 + 2004 + 192
= 2900
= 48 minutes

The minimum time it took to reconfigure a computer using the RIS system was

RIS(MINIMUM TIME) = 44 + 294 +1434 + 156
= 1928
= 32 minutes

Appendix 2.5 – Requested Infrastructure and Laboratory Support (ILS) work flow


Customer submits their request to the website
Technician approves the customer's request
Website process' the customer's request and sends to ADS
ADS reboots the computer(s) and loads Deployment Agent
ADS deploys OS image to computer(s)
Computer(s) reboot and complete unattended setup
Website notifies the customer the changes are complete via requests page
= technician required

Appendix 2.6 – Times for requested Infrastructure and Laboratory Support (ILS) work process

The following data are the observed amount of time it takes an Automated Deployment Services (ADS) server to complete each of the steps in the new system, shown in Appendix 2.5.

Although singular computers were timed during these tests, this will not affect the results due to multicasting not decreasing in speed with an increased number of computers.

Reboot and boot to Deployment Agent (Average 60 seconds)
Table A2.6.1 shows the amount of time taken by computers to reboot and PXE boot to the Deployment Agent (DA).

62	53	61
57	72	74
57	60	64
51	59	55
	AVERAGE	60

Table A2.6.1: Data collected on time taken for a computer to reboot to the DA

Deploy OS image (Average 386 seconds)
Table A2.6.2 shows the amount of time it took the ADS server to deploy an image to the computers.

384	395	388
387	376	379
396	387	388
394	380	382
	AVERAGE	386

Table A2.6.2: Data collected on time taken for a computer to download the OS image

Reboot and unattended install (Average 752 seconds)
Table A2.6.3 shows the amount of time it took the computers to reboot and perform the unattended install of the OS.

775	746	731
768	733	787
754	724	772
769	748	732
	AVERAGE	752

Table A2.6.3: Data collected on time taken for a computer to perform OS installation

Although the time it would take a customer to make a request, a technician to approve a request and the website to notify the customer are not present due to the website not existing, these times should be somewhat negligible.

Appendix 2.7 – Analysis of requested Infrastructure and Laboratory Support (ILS) work process

The timings in Appendix 2.6 show the average amount of time measured for steps in the requested work process shown in Appendix 2.5. Using these values calculations can be made about deployment times using an Automated Deployment Services (ADS) system.

Using an ADS system, a computer will take on average 20 minutes to deploy an OS to a computer. As the ADS system uses multicast to deploy the OS the time taken for X computer to be deployed is

Time for X computers = 20 minutes

As this formula does not contain X on the calculation die of the equation, the time taken to deploy multiple computers is not larger than the time to deploy one computer. This formula will work for up to 128 computers. More than 128 computers will result in two 'Deploy OS image' steps due to the 128 computer limit on the multicasting.

Tables A2.7.1 and A2.7.2 show the amount of time taken for an ADS system to deploy to one computer and five computers respectively

Time	0-	3-	6-	9-	12-	15-	18-	21-
Computer 1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

🟩 = unattended

Table A2.7.1: Time taken to deploy one computer using ADS

Time	0-	3-	6-	9-	12-	15-	18-	21-
Computer 1	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Computer 2	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Computer 3	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Computer 4	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
Computer 5	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

🟩 = unattended

Table A2.7.2: Time taken to deploy five computers using ADS

The maximum time taken for a computer to receive a new OS using an ADS system can be calculated as

ADS(MAXIMUM TIME) = 72 + 396 + 787 seconds
= 1255 seconds
= 21 minutes

The minimum time taken for a computer to receive a new OS using an ADS system can be calculated as

ADS(MINIMUM TIME) = 51 + 376 + 724
= 1151
= 19 minutes

Appendix 2.8 – User requirements and features e-mail

Exhibit 1

Sorry for the delay in doing this, stuff I would like to see in your ADS project would be:

1) Ability to build lists of jobs to undertake in an order e.g. 1 Build a DC, 2 add all machines to the domain, 3 install software, 4 run an additional tasks e.g. AD scripts to create additional user accounts
2) Improved user interface i.e. one that actually refreshes and displays when jobs have been completed / failed

Those are the two main changes I would like to see. I think the most important aspect is what you discussed before in that there would be an improved interface that allows you to customise build orders and builds, so not only could you create a list of tasks and the order in which they should be executed. Also the ability to customise each of the steps so for a DC you could specify; the domain name, whether DNS is to be installed etc. The ability to save this config would also be good so that in future you could load the config and use it to build hardware in the same way as before. It would be quite cool if you could get it to send you an email when the build process had finished or failed so you could leave it and do something else and be notified when it finished.

Exhibit 2

Hello, this is what I want to be able to do
> Design the system on the screen so that I don’t have to keep running separate tasks. When you build a DC then install SQL server and you have to do it in two, it would be good if it would automatically run the SQL one after the DC.
> Kind of like the last point but to be able to chain any old jobs together. So if I want to format one machine, then install SQL on a different one, then reboot another etc etc. Even though the jobs might not have anything to do with each other.
> Would be good to have a backup job that we could set to run on the production machines so we don’t have to worry about it our selves.

Exhibit 3

I talked to [*my manager*] and he things it could be a neat little thing to impress customers. Obviously he said that you can’t try it in his data centre but if you do get something working then it might be handy to have behind the scenes for me to use. So [*my manager*] says it will be nice to have something that customers can report problems to that I can then check and fix, although I’d rather see something that will help me. It would be good if they could ‘design’ the computer as part of their request so basically all I have to do is approve it and the ADS server would go off and do it.
It would also be good if the website could produce a diagram of what the customers are using, just so I can print it off and give it to them instead of using visio.

Exhibit 4

It would be good if it was really easy to use. Someone asked me to redo a 2003 SQL server for them last week and it took me ages to work out.

Exhibit 5
Hey! I talked to [*my colleague 1*] about what he put so I want all that stuff and this stuff. So these are extras on top of the other stuff
1. A timer for tasks so I don't have to stay till 6pm on Fridays
2. Something to let me resume tasks when they fail
3. To be texted when to job finished (or failed) so I don't have to check my e-mails

Exhibit 6
The main thing I would like to use the website for is producing usage statistics for my reports. Currently we roughly keep track of what machines are in use, although I have to get [*a technician*] do this manually in a spreadsheet, so it would be nice if they could be produced automatically. I would also like to see what software is used most commonly by the customer, we know most of them use SQL and Exchange but it would be good to have actual fact and figures for all the software they use.

I mentioned it to [*my manager*], and as always he wants to know how it could be used for EMEA. He asked if it is possible for it to be used across multiple sites so the Germanys could use our images if they didn't have their own.

Exhibit 8
I talked to [my colleague] about what he put and it sounds good. I was just wondering what security you'll be putting in place? You know the setup so you know what kind of things we have in place, so username and passwords etc. You won't be able to use the domain accounts because we've had to physically disconnect the services lab from the cooperate network so there's not access anymore. I'm still working on that WUS server so it would be good to work with that some how or keep the firewall and DC up to date with the patches. I *think* I've fixed the multicasting issue across VLANs so that security should work now. You could go into security in your report, about the VLANs and network setup. Don't go into too much exact detail about how we do things though, we don't want everyone knowing exactly how we do things.

Exhibit 7
The main problem I remember is you have to go to a different screen to do anything, it would be good if you could improve on that.

Appendix 2.9 – Preliminary list of user requirements

Critical Features

- Ability to run ADS tasks remotely
- Ability to queue ADS tasks
- Ability to configure applications via the website
- Ability to design full, multiple computer setups
- Ability to reapply a saved computer setup
- Ability to schedule a job to run at a certain time
- The system must have some form of security

Additional features

- Better support for installing patches
- Ability to backup total systems for later redeployment
- Ability for customers to use the website
- Ability to resume tasks that terminated with errors
- Ability to produce statistics on computer and software usage
- To be alerted once the task was finished by e-mail or SMS
- To work over multiple geographical sites

Appendix 2.10 – Users' responses to preliminary requirements and features

Exhibit 1

Seems good to me. Let me know when you have something up and working so I can have a look. You might want to leave to configuring the application till last because I think that will be a bit of a problem

Exhibit 2

That all seems fine.

Exhibit 3

The main change I'd like to request is for the statistic to be put in as a critical feature. If you're looking for a feature to swap it with take out running a job at a certain time, I don't really want people leaving without making sure everything is ready for Monday. It would be good if the statistics could be shown in different ways such as pie charts or tables so I could use them for meeting, if not then I can make the graphs myself.

Exhibit 4

Yeah, that fine. One thing I would like to suggest though which isn't really a feature as such, is that if you make a customer facing part of the website make it look pretty. The backend bit doesn't matter so much but if the customer sees a nice looking page then they'll think the rest is all pretty too.

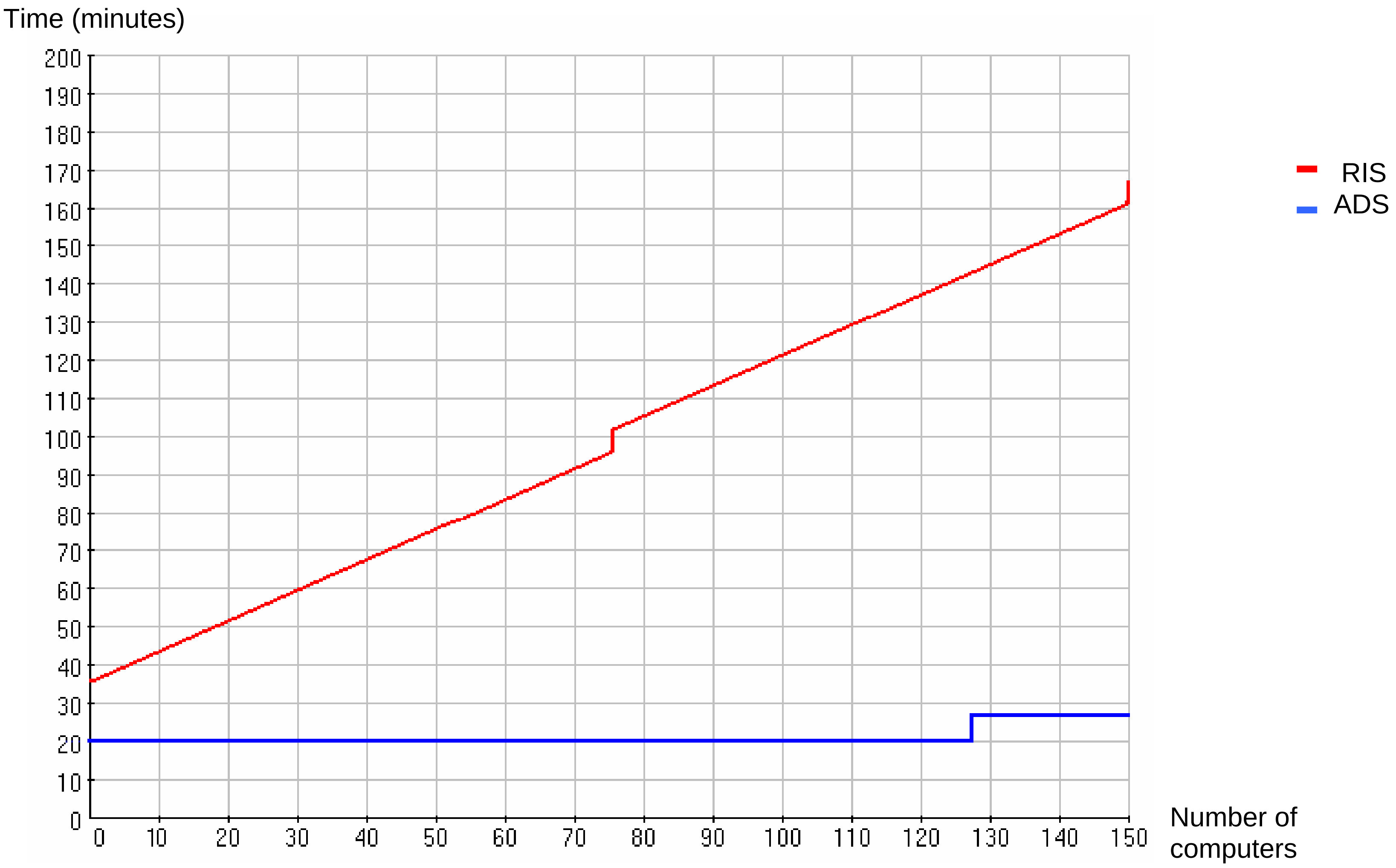
Exhibit 5

I don't really understand any of it. So it all sounds good to me.

Appendix 2.11 – Limiting factors explanation

An example of a technical limiting factor could be that the user of the car requests their 1.1 litre car should be able to go 200 mph with the same 1.1 litre engine installed. This feature could not be achieved due to a 1.1 litre car engine not being technologically advanced enough or powerful enough to reach such speeds. An example of a time limiting factor could be that the user wants their car to be able to go 200 mph by letting you change any components of the car, but they want it done with in one week. Although it is possible for a car to travel 200 mph, it may not be possible to modify a car within one week to reach such speeds.

Appendix 3.1 – Graph to show time taken by RIS and ADS to reconfigure a number of computers



Appendix 3.2 – Speed of deployment discussion

"In January 2003, the Slammer worm hit 55 million hosts within 11 minutes." (Phifer, 2004)

This was a massive disruption to companies, and took weeks for some to return their systems back to a fully operational state. Having a system in place that can quickly return computers back to their original state could have saved many companies' hours of work and disruption. Although a mass virus attack may not greatly affect the Microsoft ILS labs, deployment speed is still an important factor as customers want their requests fulfilled quickly and many computers may need to have software deployed to them in a short amount of time.

As shown in Appendix 2.4, with the use of RIS it takes 35.85 + 0.8x minutes (where x is the number of computers) to deploy up to 75 computers. Appendix 2.7 shows that with ADS it takes 20 minutes to deploy up to 128 computers.

If we assume 40 computers need to have new Operating System (OS) installed, we can calculate the total amount of time taken by RIS and ADS to deploy the new OSs. This is the typical number of computers a customer wanting to use the ILS labs will request. A graph of these times can be seen in Appendix 3.1.

Time for RIS = 35.85 + 0.8 * 40 = 68 minutes
Time for ADS = 20 minutes

These results show that on average an ADS system would be around 2.5 times faster at reconfiguring 40 computers, although in reality this is not the case. The time for RIS formula assumes that there are an unlimited number of technicians to rebuild the computers and using theoretical maximum top speeds, in the ILS lab there will be a maximum of 5 technicians available. The times also do not take in to account the extra software and configuration that may needed on each computer before it is finished.

Appendix 3.3 – Cost of deployment discussion

In the current RIS system 6 minutes of a technicians time is required for each computer that needs reconfiguring, shown in Appendix 2.4. In the new ADS system the time taken for one technician to build any number of machines will be negligible. A skilled technician at Microsoft earns around £15/hour.

$$\begin{aligned} \text{RIS time cost} &= \frac{\text{6 minutes * 40 computer}}{60} \text{ x £15/hour} \\ &= \text{4 hours x £15/hour} \\ &= \text{£60} \end{aligned}$$

ADS time cost = negligible

From these simple calculations it is possible to see that an ADS system can save money by reducing the amount of technician’s time required to reconfigure computers. Although these extremely simple calculations show it would take 4 hours of a technician’s time to rebuild 40 computers, in the ILS lab it typically takes one to two days of a technician’s time to fully configure a customer’s request. The advantage of ADS is not only that it takes less time to reconfigure computers, but also that it does not require any of the technicians time.

A better example of the savings that can be made with ADS and integration with other technologies is shown in the Advanced Internet Technologies (AIT) case study.

The following extracts show the reduction in time that can be achieved and the savings that can be made.

> “Using ADS, AIT reduced the time it takes to build a new server from more than 180 minutes to 16 minutes. Moreover, where AIT’s previously manual build process precluded a technician from building more than two or three servers simultaneously, the ADS console makes it possible for a single AIT technician to build hundreds of servers simultaneously.”

Extract A3.3.1: ADS time saving (Microsoft, 2003)

> “SMS reduced the time required to deploy patches to one minute per server, a 94 percent improvement that will save AIT at least U.S.$24,000 each year.”

Extract A3.3.2: SMS money saving (Microsoft, 2003)

Although Extract A3.3.2 shows the saving using Microsoft Systems Management (SMS) to deploy patches, this can be compared to using ADS to deploy software, as the technician’s involvement will be reduced and the total time taken to install the software will be decreased.

Additional monetary benefits of using ADS may be present but not directly visible. Decreasing the amount of time taken to configure computers for customers in the ILS lab could mean that more customers can use the lab simultaneously, as currently around 40% of the computers sit idle each week due to the long turn around time. Alternatively, with a fast turn around time 40% of the computers may no longer be needed and could be sold. Additionally, customer satisfaction may improve if the customers feel their requests are dealt with quickly and they are more likely to use the lab services again or recommend it to others.

Appendix 3.4 – Risk Assessment

As with any large project there are many unforeseen problems that can occur. To help deal with these problems a risk assessment can be completed to identify, minimise and plan for risks as much as possible.

Security risks
As is becoming increasingly common with IT, security will be a major issue with the website. Due to the functionality of the website, anyone who managed to compromise the computer running the website (the web server) would have full access to all of the computers in the data centre. To help minimise this risk a number of steps can be taken.

Restrict physical access
The most dangerous situation would occur if an unauthorised person managed to gain physical access to the data centre. If this happened they would be able to circumnavigate all the other security defences put in place. To help counter this, Microsoft has a number of security systems including, but not limited to, swipe card access and CCTV.

Restrict network access
To help stop the website from being hacked access to the website can be restricted in the following ways.

No access – the simplest way of stopping unauthorised people accessing the web server is to give it no external access at all. Although this gives the maximum amount of security, it is not the most preferable solution as not providing any access would mean anyone using the website would have to use it locally and this would be contrary to the users' requirements.

No logical access – having the web server on a separate Virtual Local Area Network (VLAN) is advantageous as network access points not authorised to be on that VLAN will be unable to communicate with it. To use the website the computers in the data centre would have to be placed on the same VLAN, as would any other computers wishing to access the website.

Although having no logical access enables greater flexibility than having no access, it would still limit access to computers joined to an ILS lab switch. As one of the users' requirements was to be able to access the website from anywhere in the world, more flexibility is needed. By introducing a computer to bridge across VLANs, authorised users could access the website through this bridge, see Figure A3.5.1.

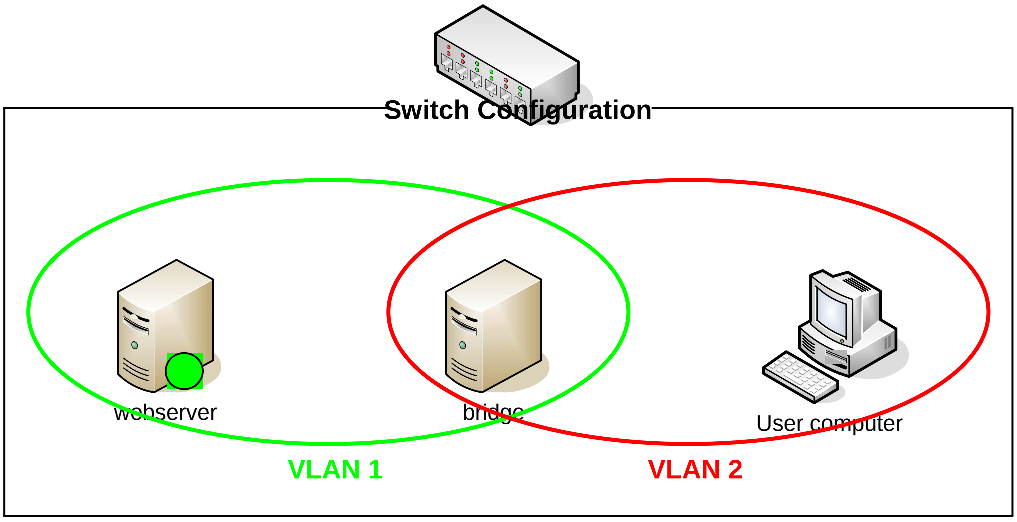


Figure A3.5.1: VLAN protection

The configuration in Figure A3.5.1 would mean that unauthorised users would first have to compromise the bridging computer or the network switch before they could try to compromise the web server. To help protect this bridge computer a Firewall can be put in place, see Figure A3.5.2.

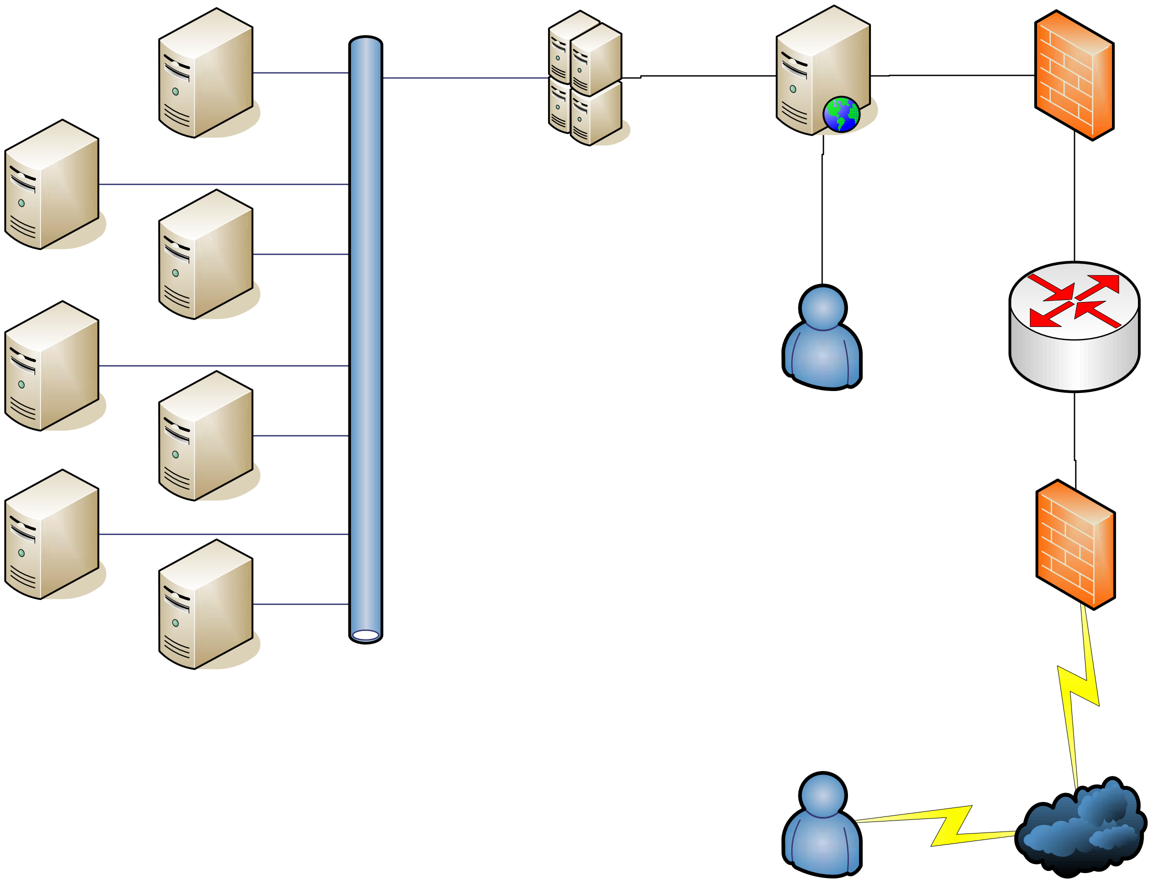
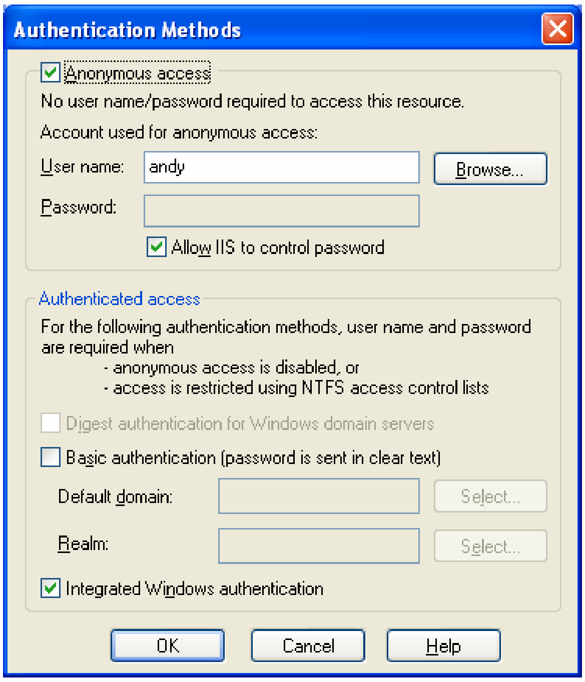


Figure A3.5.2: Potential network configuration

Although the network setup in Figure A3.5.2 is not the most secure of the possible network configurations, this trade-off must be made to meet the users' requirement of access the website remotely.

Website passwords
To restrict access to the website directly, and not just the web server, a number of password systems could be used. As the website will be running on an Internet Information Services (IIS) Server the website can be protected by Integrated Windows authentication or by using Domain authentication, See Figure A3.5.3. This could also allow the future possibility of using NTFS permissions to further secure the website.

Due to the time restrictions on the project, the website only implemented its own password security and left further security integration as a further development.



Authentication Methods

☑ Anonymous access

No user name/password required to access this resource.

Account used for anonymous access:

User name: andy

Password:

☑ Allow IIS to control password

Authenticated access

For the following authentication methods, user name and password are required when
- anonymous access is disabled, or
- access is restricted using NTFS access control lists

☐ Digest authentication for Windows domain servers

☐ Basic authentication (password is sent in clear text)

Default domain:

Realm:

☑ Integrated Windows authentication

Figure A3.5.3: IIS authentication

Website permissions
Finally, the user accounts for the website has a granular level of access. This restricts what users can do even if they have a valid user account and password to the website. For example, certain user accounts may only be able to view what jobs are running and not actually run any jobs themselves.

Technical risks
As the website is based on new technology there were many technological things that could go wrong. The main risk was that the system would not work at all. Although in theory the system should work, there are no similar websites that could be looked at to prove such a system will work in practice. The website must also be error free to be implemented in a production environment as the potential effects could be devastating, accidentally reformatting all the computers in a data centre would not be acceptable.

Project risks

Another sizeable risk to the success of the project was time. If the project ran too far behind plan then the project could have failed to produce anything of use. The risk of this was amplified due to the use of new technologies, which do not have much support or documentation relating to them, and the amount of knowledge that had to be learnt before the implementation was possible. The scope of the project is also a very important factor as producing 100 functions that do not work is no use, it is better to produce one function that does work. Due to these considerations, the scope of the project was limited to ensure that the core functionally was fully completed and tested.

Appendix 4.1 - Initial user data table

USER		
Attribute name	Data Type	Description
UserID	Integer	Unique auto-increment Primary Key
Username	Varchar(20)	Unique username
Password	Varchar (20)	Users password
CanView	Bit	Indicates if the user can view configurations
CanEdit	Bit	Indicates if the user can edit configurations
CanApply	Bit	Indicates if the user can apply configurations
CanActivate	Bit	Indicates if the user can activate configurations
CanDeactivate	Bit	Indicates if the user can deactivate configurations
CanSave	Bit	Indicates if the user can save configurations
IsAdmin	Bit	Indicates if the user is an Administrator
Active	Bit	Indicates if the user account is still active

Appendix 4.2 - Initial task data table

TASK		
Attribute name	Data Type	Description
TaskID	Integer	Unique auto-increment Primary Key
FullDescription	Varchar(500)	Long description of what the task does
ShortDescription	Varchar(30)	Short description on what the task does
XMLPath	Varchar(200)	File path to the XML file the task represents
WillRunOn2k	Bit	Indicates if the task will run on Windows 2000
WillRunOnXP	Bit	Indicates if the task will run on Windows XP
WillRunOn03	Bit	Indicates if the task will run on Windows 2003
WillRunOnLH	Bit	Indicates if the task will run on Windows Longhorn
IsXP	Bit	Indicates if the task installs Windows XP
Is2k	Bit	Indicates if the task installs Windows 2000
Is03	Bit	Indicates if the task installs Windows 2003
IsLH	Bit	Indicates if the task installs Windows Longhorn
CurrentCounter	Integer	Number of times the task has been run since the counter was last cleared
Counter	Integer	Total number of times that task has been run

Appendix 4.3 - Initial job data table

JOB		
Attribute name	Data Type	Description
JobID	Integer	Unique auto-increment Primary Key
Tasks	List	Ordered list of tasks in the job
CreatedBy	Integer	Foreign Key USER:UserID
CreatedDate	DateTime	The date and time the job was created
Active	Bit	Indicates if the job is still active

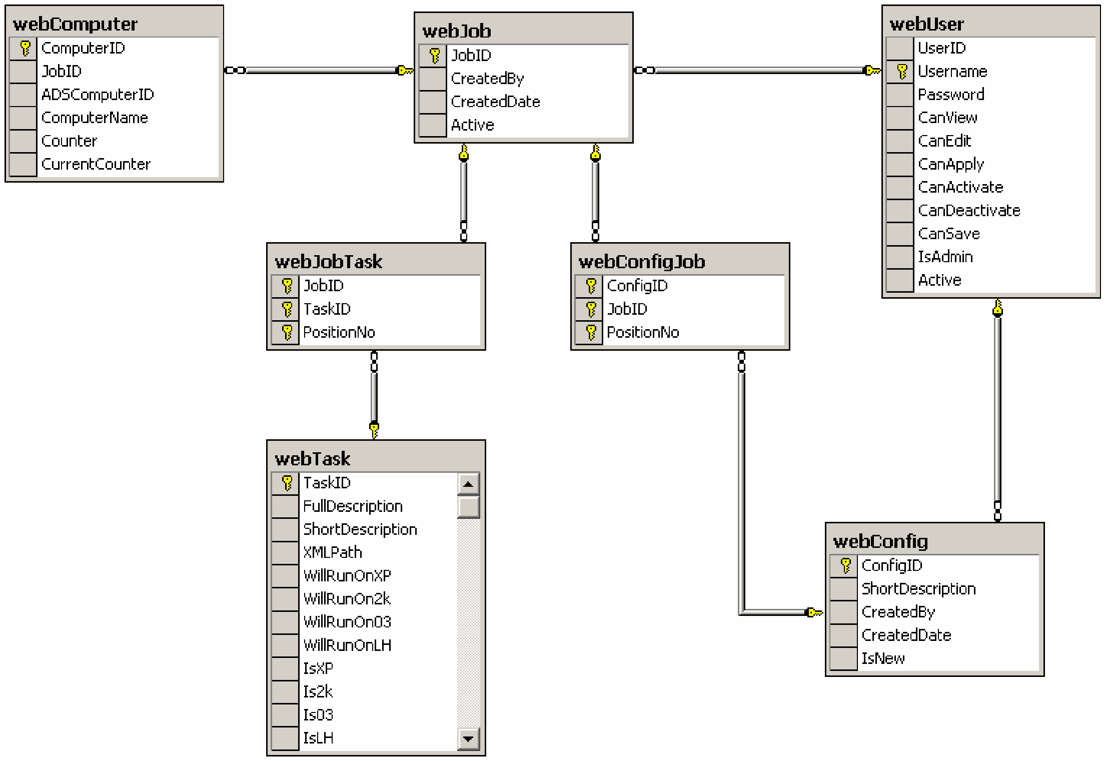
Appendix 4.4 - Initial configuration data table

CONFIGURATION		
Attribute name	Data Type	Description
<u>ConfigID</u>	Integer	Unique auto-increment Primary Key
Jobs	List	Ordered list of jobs in the configuration
Description	Varchar(30)	Description of what the configuration does
CreatedBy	Integer	Foreign Key USER:UserID
CreatedDate	DateTime	The date and time the job was created
IsNew	Bit	Indicates if the configuration has ever been applied

Appendix 4.5 - Initial computer data table

COMPUTER		
Attribute name	Data Type	Description
ComputerID	Integer	Unique auto-increment Primary Key
JobID	Integer	Foreign Key indicating which job is using the computer JOB:JobID
ADSComputerID	Integer	Foreign Key linking the ADS (Automated Deployment Services) database to the website database DEVICES:ID
CurrentCounter	Integer	Number of times the computer has been used since the counter was last cleared
Counter	Integer	Total number of times the computer has been used

Appendix 4.6 – Full website database with data tables



webComputer
- ComputerID
- JobID
- ADSComputerID
- ComputerName
- Counter
- CurrentCounter

webJob
- JobID
- CreatedBy
- CreatedDate
- Active

webUser
- UserID
- Username
- Password
- CanView
- CanEdit
- CanApply
- CanActivate
- CanDeactivate
- CanSave
- IsAdmin
- Active

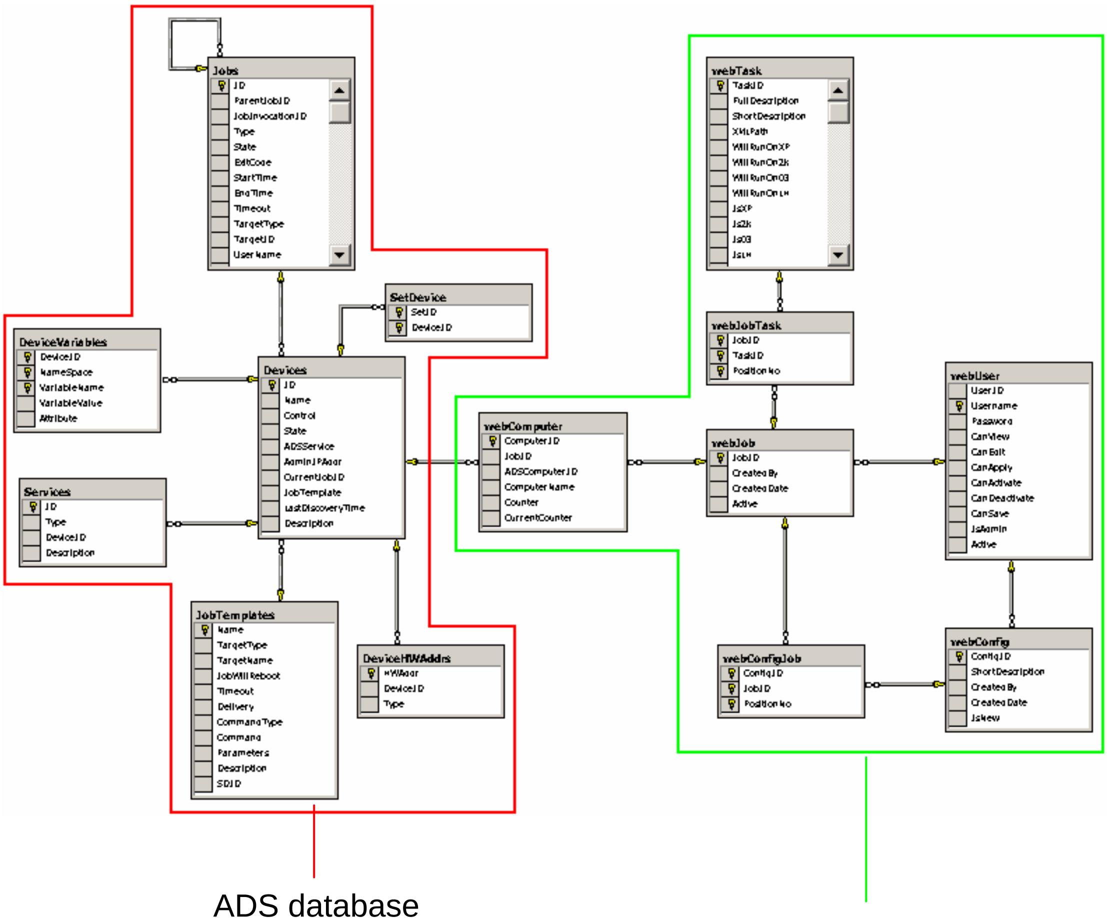
webJobTask
- JobID
- TaskID
- PositionNo

webConfigJob
- ConfigID
- JobID
- PositionNo

webTask
- TaskID
- FullDescription
- ShortDescription
- XMLPath
- WillRunOnXP
- WillRunOn2k
- WillRunOn03
- WillRunOnLH
- IsXP
- Is2k
- Is03
- IsLH

webConfig
- ConfigID
- ShortDescription
- CreatedBy
- CreatedDate
- IsNew

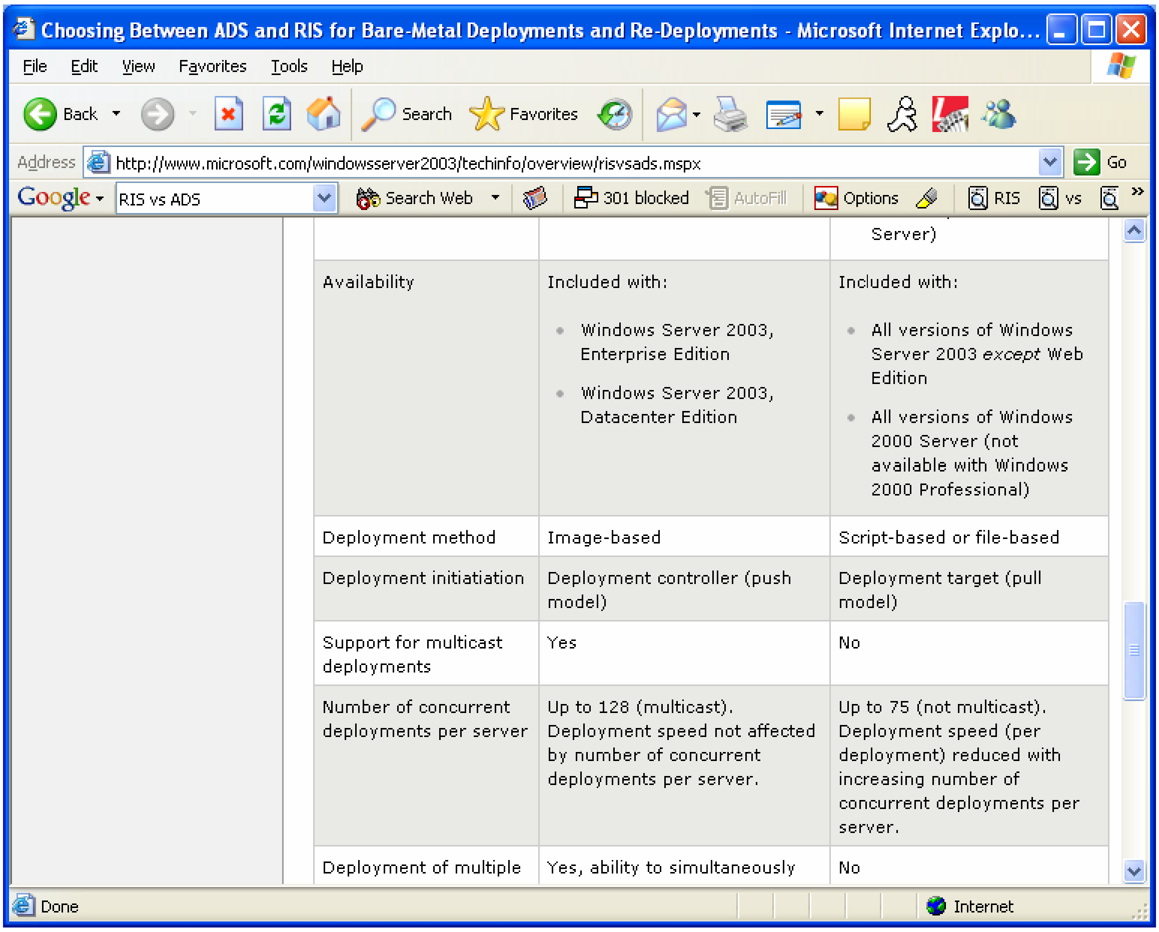
Appendix 4.7 – ADS (Automated Deployment Services) database linked with the website database



ADS database

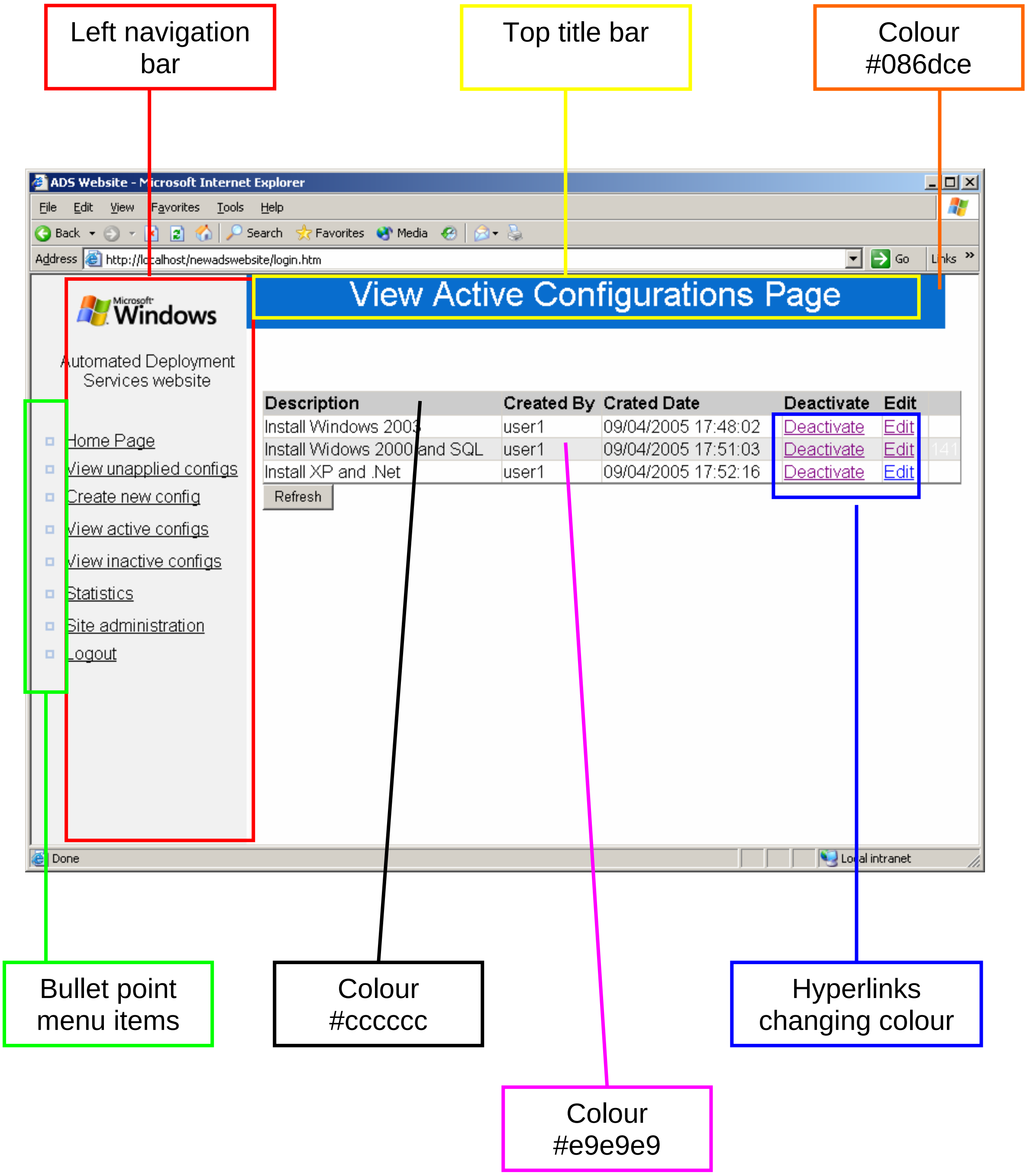
Website database

Appendix 4.8 – Screen shot of the Microsoft website showing colours used for the ADS website


Choosing Between ADS and RIS for Bare-Metal Deployments and Re-Deployments - Microsoft Internet Explo...
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http://www.microsoft.com/windowsserver2003/techinfo/overview/risvsads.mspx
Go
Google
RIS vs ADS
Search Web
301 blocked
AutoFill
Options
RIS
vs
Done
Internet

		Server)
Availability	Included with: • Windows Server 2003, Enterprise Edition • Windows Server 2003, Datacenter Edition	Included with: • All versions of Windows Server 2003 *except* Web Edition • All versions of Windows 2000 Server (not available with Windows 2000 Professional)
Deployment method	Image-based	Script-based or file-based
Deployment initiatiation	Deployment controller (push model)	Deployment target (pull model)
Support for multicast deployments	Yes	No
Number of concurrent deployments per server	Up to 128 (multicast). Deployment speed not affected by number of concurrent deployments per server.	Up to 75 (not multicast). Deployment speed (per deployment) reduced with increasing number of concurrent deployments per server.
Deployment of multiple	Yes, ability to simultaneously	No

Appendix 4.9 – Screen shot of the ‘view active configurations’ page with design points high lighted


Left navigation bar
Top title bar
Colour #086dce
Bullet point menu items
Colour #cccccc
Hyperlinks changing colour
Colour #e9e9e9
ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Done
Local intranet

View Active Configurations Page

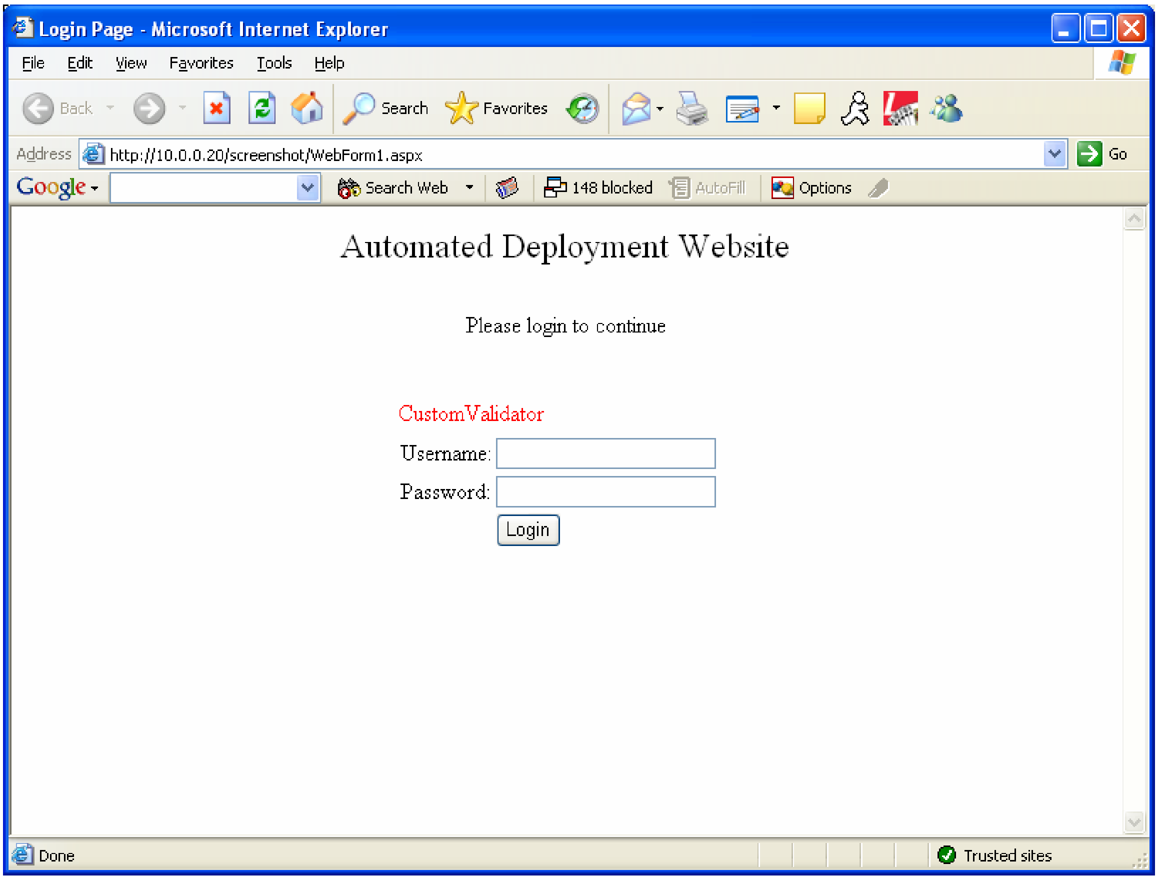
Automated Deployment Services website

- Home Page
- View unapplied configs
- Create new config
- View active configs
- View inactive configs
- Statistics
- Site administration
- Logout

Description	Created By	Crated Date	Deactivate	Edit
Install Windows 2003	user1	09/04/2005 17:48:02	Deactivate	Edit
Install Widows 2000 and SQL	user1	09/04/2005 17:51:03	Deactivate	Edit
Install XP and .Net	user1	09/04/2005 17:52:16	Deactivate	Edit

Refresh

Appendix 5.1 – Prototype Login page design



Login Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://10.0.0.20/screenshot/WebForm1.aspx Go
Google Search Web 148 blocked AutoFill Options
Done
Trusted sites

Automated Deployment Website

Please login to continue

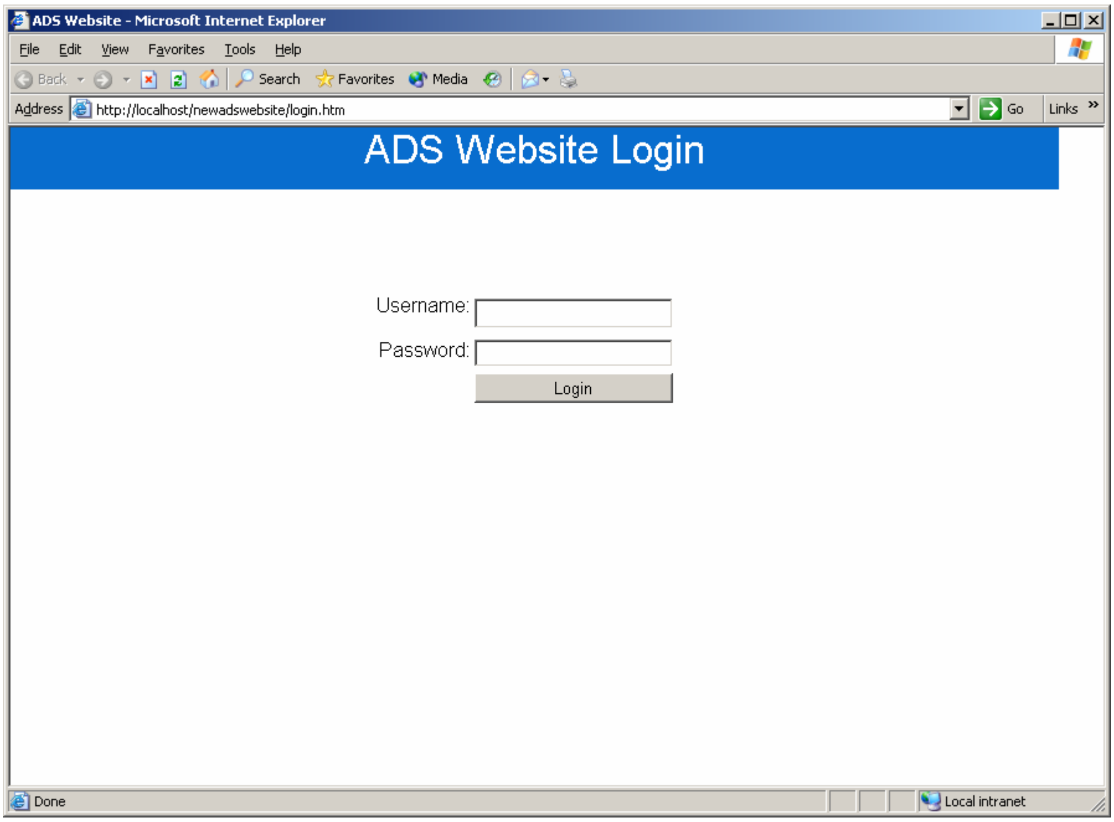
CustomValidator

Username:

Password:

Login

Appendix 5.2 – Final Login page screenshot



ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Done
Local intranet

ADS Website Login

Username:

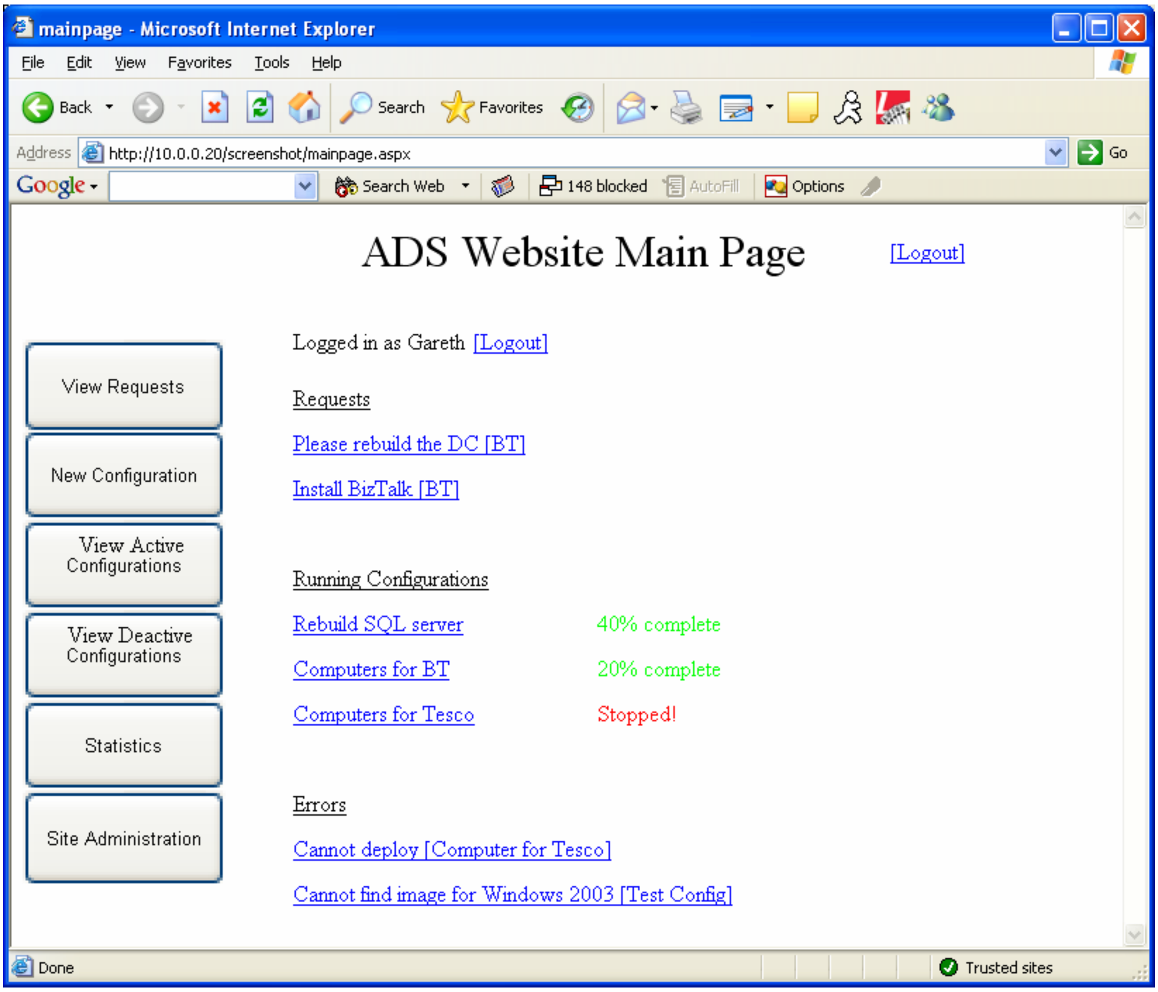
Password:

Login

Appendix 5.3 – Load running jobs function from the home page

```
private void LoadRunningJobs()
{
   try
   {
      //Load all the running tasks
      // Setup database connection
      SqlConnection conn = new SqlConnection(@"Data Source=(local);Initial
Catalog=adsdb;Integrated Security=true");

      // Create SQL command to find running jobs
      SqlCommand cmd = new SqlCommand("SELECT JobInvocations.Description AS
Description FROM Jobs INNER JOIN JobInvocations ON Jobs.ID=JobInvocations.ID WHERE
Jobs.State=3",conn);

      // Open database connection
      conn.Open();

      // Execute the SQL command
      readResult = cmd.ExecuteReader();

      // Bind the data to the Running Jobs table
      if(readResult.HasRows)
      {
         lblNoRunningConfigs.Visible=false;
         dtgRunningConfigs.Visible=true;
         dtgRunningConfigs.DataSource=readResult;
         dtgRunningConfigs.DataBind();
      }
      else
      {
         dtgRunningConfigs.Visible=false;
         lblNoRunningConfigs.Visible=true;
      }

      //Close database connection
      readResult.Close();
      conn.Close();
   }
   catch
   {
      lblError.Text="The running jobs could not be read from the database";
      lblError.ForeColor=Color.Red;
   }
}
```

Appendix 5.4 – Prototype Home page design



mainpage - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://10.0.0.20/screenshot/mainpage.aspx
Go
Google
Search Web
148 blocked
AutoFill
Options
View Requests
New Configuration
View Active Configurations
View Deactive Configurations
Statistics
Site Administration
Done
Trusted sites

ADS Website Main Page [Logout]

Logged in as Gareth [Logout]

Requests

Please rebuild the DC [BT]

Install BizTalk [BT]

Running Configurations

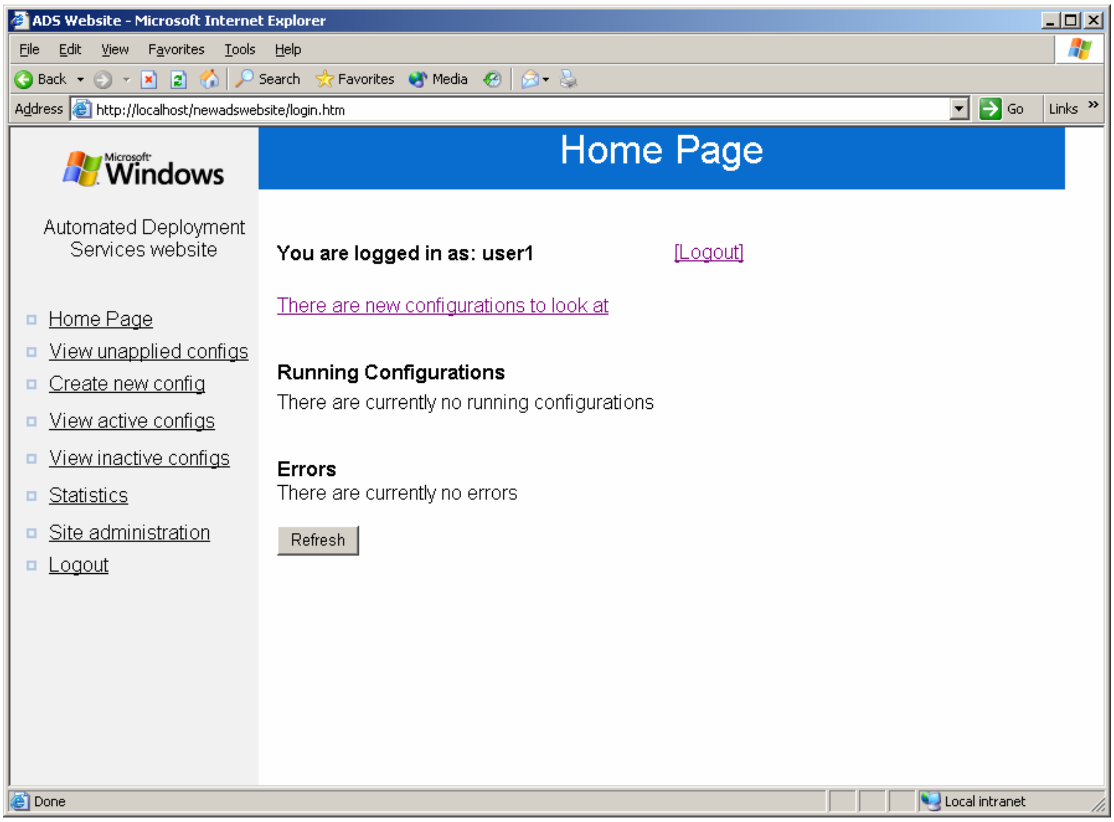
Rebuild SQL server	40% complete
Computers for BT	20% complete
Computers for Tesco	Stopped!

Errors

Cannot deploy [Computer for Tesco]

Cannot find image for Windows 2003 [Test Config]

Appendix 5.5 – Final Home page screen shot



ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Windows
Done
Local intranet

Home Page

Automated Deployment Services website

- Home Page
- View unapplied configs
- Create new config
- View active configs
- View inactive configs
- Statistics
- Site administration
- Logout

You are logged in as: user1 [Logout]

There are new configurations to look at

Running Configurations
There are currently no running configurations

Errors
There are currently no errors

Refresh

Appendix 5.6 – LoadComputers() Function

```
public void LoadComputers()
{
    try
    {
        // Setup database connection
        SqlConnection conn = new SqlConnection(@"Data Source=(local);Initial
Catalog=adsdb;Integrated Security=true");

        // If there is a config the find the jobs and get the computers that are in those jobs too
        string WhereCondition=" WHERE webComputer.JobID IS null";
        DataGrid JobIDs = new DataGrid();

        if(txtCurrentConfigID.Text!="0")
        {
            // SQL to get all the jobIDs for the current task
            SqlCommand tmpcmd = new SqlCommand("SELECT JobID, PositionNo FROM
webConfigJob WHERE ConfigID=@ConfigID ORDER BY PositionNo",conn);

            // Create cmd parameters for ConfigID
            tmpcmd.Parameters.Add("@ConfigID",SqlDbType.Int);
            // Set the parameter to the ConfigID
            tmpcmd.Parameters["@ConfigID"].Value= Convert.ToInt16(txtCurrentConfigID.Text);

            // Open database connection
            conn.Open();

            // Execute the SQL command
            readResult = tmpcmd.ExecuteReader();

            // Put the JobIDs in a datagrid
            JobIDs.DataSource=readResult;
            JobIDs.DataBind();

            // Close the reader and the database connection
            readResult.Close();
            conn.Close();

            for(int i=0; i<=JobIDs.Items.Count-1; i++)
            {
                WhereCondition+=" OR JobID=" + JobIDs.Items[i].Cells[0].Text;
            }
        }
        // Open database connection
        conn.Open();

        // Create SQL command to get computer info
       SqlCommand cmd = new SqlCommand("SELECT webComputer.ADSComputerID AS ID,
Devices.Name AS Name FROM webComputer INNER JOIN Devices ON
webComputer.ADSComputerID = Devices.ID" +WhereCondition+" ORDER BY ID",conn);

        // Create array to hold the OSTask Lists
        ListBox[] ListControlCopys = new ListBox[4];
        ListControlCopys[0]=listComputers1;
        ListControlCopys[1]=listComputers2;
        ListControlCopys[2]=listComputers3;
        ListControlCopys[3]=listComputers4;

        int ListBoxNumber=0;
```

```
      foreach(ListBox item in ListControlCopys)
      {
         // Execute the SQL command
         readResult = cmd.ExecuteReader();

         // Bind the data to the listbox so the user can choose
         item.DataSource=readResult;
         item.DataValueField="ID";
         item.DataTextField="Name";
         item.DataBind();

         // close the reader so its ready to use again
         readResult.Close();

         // loop through all the jobs and select the correct computers
         for(int i=0; i<=JobIDs.Items.Count-1;i++)
         {
            if(ListBoxNumber==Convert.ToInt16(JobIDs.Items[i].Cells[1].Text)-1)
            {
               // SQL to get all the ComputerIDs for the current Job
               SqlCommand tmpcmd = new SqlCommand("SELECT ADSComputerID FROM
webComputer WHERE JobID=@JobID",conn);

               // Create cmd parameters for JobID
               tmpcmd.Parameters.Add("@JobID",SqlDbType.Int);
               // Set the parameter to the JobID
               tmpcmd.Parameters["@JobID"].Value=
Convert.ToInt16(JobIDs.Items[i].Cells[0].Text);

               // Execute the SQL command
               readResult = tmpcmd.ExecuteReader();

               // Put ADSComputerIDs in a datagrid
               DataGrid ComputerIDs = new DataGrid();
               ComputerIDs.DataSource=readResult;
               ComputerIDs.DataBind();

               // Close the reader
               readResult.Close();

               // go through each computer and see if its ID matches one in the Job
               foreach(ListItem computer in ListControlCopys[ListBoxNumber].Items)
               {
                  for(int j=0;j<=ComputerIDs.Items.Count-1;j++)
                     if(computer.Value== ComputerIDs.Items[j].Cells[0].Text)
computer.Selected=true;
               }
            }
         }
         ListBoxNumber++;
      }
      // Close the database connection
      conn.Close();
   }
   catch
   {
      lblError.Text="The computers could not be loaded from the database";
      lblError.ForeColor=Color.Red;
   }
}
```

Appendix 5.7 – ComputerOnlySelectedOnce method

```
private bool ComputersOnlySelectedOnce()
{
    try
    {
        /* CHECK FOR COMPUTER BEING SELECTED MULTIPLE TIMES
             * return TRUE if computers are only selected once
             * return FALSE if computers selected multiple times
             */
        // Create an array for the computer lists
        ListBox[] ComputerCopies = new ListBox[4];

        // Create bool to indicate if there is an error
        bool OnlySelectedOnce = true;

        // Copy the computer lists in to the array
        ComputerCopies[0]=listComputers1;
        ComputerCopies[1]=listComputers2;
        ComputerCopies[2]=listComputers3;
        ComputerCopies[3]=listComputers4;

        // Outer loop for each computer list
        for(int i=0;i<=3;i++)
        {
            //for each computer in the computer list box see if it is also selected somewhere else
            foreach(ListItem item in ComputerCopies[i].Items)
            {
                if(item.Selected)
                {
                    for(int j=i+1;j<=3;j++)
                    {
                        // loop through the secondary box and see if there is a computer selectd with
the same ID
                        foreach(ListItem item2 in ComputerCopies[j].Items)
                        {
                            if(item2.Selected)
                            {
                                if(item.Value==item2.Value)
                                {
                                    // Set the computer list name to a user friendly one
                                    string FirstJob="";
                                    string SecondJob="";
                                    switch(ComputerCopies[i].ID)
                                    {
                                        case "listComputers1": FirstJob="Job 1";break;
                                        case "listComputers2": FirstJob="Job 2";break;
                                        case "listComputers3": FirstJob="Job 3";break;
                                        case "listComputers4": FirstJob="Job 4";break;
                                    }
                                    switch(ComputerCopies[j].ID)
                                    {
                                        case "listComputers1": SecondJob="Job 1";break;
                                        case "listComputers2": SecondJob="Job 2";break;
                                        case "listComputers3": SecondJob="Job 3";break;
                                        case "listComputers4": SecondJob="Job 4";break;
                                    }
                                   lblError.Text+=string.Format("{0} has been selected in {1} and {2}<br>",
item.Text,FirstJob,SecondJob);
                                    OnlySelectedOnce = false;
                        }
                    }
                }
            }
        }
    }
        }
        return OnlySelectedOnce;
        /* END CHECK FOR COMPUTER BEING SELECTED MULTIPLE TIMES */
    }
    catch
    {
        lblError.Text="The page could not determine if each computer has only been selected
once";
        lblError.ForeColor=Color.Red;
        return false;
    }
}
```

Appendix 5.8 – XML task file to install .NET framework

```
<?xml version="1.0" encoding="utf-16"?>
<sequence version="1" description="Install DonNet v1.1" command=""
xmlns="http://schemas.microsoft.com/ads/2003/sequence">
  <!-- STEP 14 update device record to always boot to hard disk in the future -->
  <!-- ADS Install folder needs to be updated, if you have installed ADS to a folder -->
  <!-- other than "C:\program files\Microsoft ads" -->
  <task description="Set default job template as boot to hard disk" doesReboot="false">
    <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>
  <task description="Download Install File" doesReboot="false">
    <command>/BMONITOR/BmFileXfer.exe</command>
    <parameters>
      <parameter>-d</parameter>
      <parameter>"C:\Inetpub\wwwroot\ADSwebsite\config\Install DotNet
1.1\InstallDotNet.bat"</parameter>
      <parameter>"\device\harddisk0\partition1\InstallDotNet.bat"</parameter>
    </parameters>
  </task>
  <task description="Download Wait For" doesReboot="false">
    <command>/BMONITOR/BmFileXfer.exe</command>
    <parameters>
      <parameter>-d</parameter>
      <parameter>"C:\Inetpub\wwwroot\ADSwebsite\config\wait_for.vbs"</parameter>
      <parameter>"\device\harddisk0\partition1\wait_for.vbs"</parameter>
    </parameters>
  </task>
  <!-- STEP 6 set BMCP port number -->
  <task description="Set Install file to run" doesReboot="false">
    <command>/BMONITOR/bmsetreg.exe</command>
    <parameters>
      <parameter>"-
h:\device\harddisk0\partition1\windows\system32\config\software"</parameter>
      <parameter>"Microsoft\Windows\CurrentVersion\RunOnce"</parameter>
      <parameter>"InstallDotNet"</parameter>
      <parameter>REG_SZ</parameter>
      <parameter>"c:\InstallDotNet.bat"</parameter>
    </parameters>
  </task>
  <!-- STEP 6 set BMCP port number -->
  <task description="Set Autologon" doesReboot="false">
    <command>/BMONITOR/bmsetreg.exe</command>
    <parameters>
      <parameter>"-
h:\device\harddisk0\partition1\windows\system32\config\software"</parameter>
      <parameter>"Microsoft\Windows NT\CurrentVersion\Winlogon"</parameter>
      <parameter>"AutoAdminLogon"</parameter>
      <parameter>REG_SZ</parameter>
      <parameter>"1"</parameter>
    </parameters>
  </task>
  <!-- STEP 6 set BMCP port number -->
  <task description="Set Default Username" doesReboot="false">
    <command>/BMONITOR/bmsetreg.exe</command>
    <parameters>
      <parameter>"-
h:\device\harddisk0\partition1\windows\system32\config\software"</parameter>
      <parameter>"Microsoft\Windows NT\CurrentVersion\Winlogon"</parameter>
      <parameter>"DefaultUserName"</parameter>
      <parameter>REG_SZ</parameter>
      <parameter>"netlab"</parameter>
    </parameters>
  </task>
  <!-- STEP 6 set BMCP port number -->
  <task description="Set Default Password" doesReboot="false">
    <command>/BMONITOR/bmsetreg.exe</command>
    <parameters>
      <parameter>"-
h:\device\harddisk0\partition1\windows\system32\config\software"</parameter>
      <parameter>"Microsoft\Windows NT\CurrentVersion\Winlogon"</parameter>
      <parameter>"DefaultPassword"</parameter>
      <parameter>REG_SZ</parameter>
      <parameter>"TinM1ne"</parameter>
    </parameters>
  </task>
  <!-- STEP 6 set BMCP port number -->
  <task description="Set Default Domain" doesReboot="false">
    <command>/BMONITOR/bmsetreg.exe</command>
    <parameters>
      <parameter>"-
h:\device\harddisk0\partition1\windows\system32\config\software"</parameter>
      <parameter>"Microsoft\Windows NT\CurrentVersion\Winlogon"</parameter>
      <parameter>"DefaultDomainName"</parameter>
      <parameter>REG_SZ</parameter>
      <parameter>"%computername%"</parameter>
    </parameters>
  </task>
  <task description="Reboot Computer" doesReboot="true">
    <command>/BMONITOR/reboot</command>
  </task>
  <task description="Boot Harddisk" doesReboot="true">
    <command>/PXE/boot-hd</command>
  </task>
</sequence>
```

Appendix 5.9 – Sample XML wrapper file

```
<?xml version="1.0" standalone="no"?>
<sequence>
  <groupRef href="C:\Inetpub\wwwroot\newadswebsite\Apps\ boot_to_da.xml " />
  <groupRef href="C:\Inetpub\wwwroot\newadswebsite\Apps\ install_dotnet_11.XML " />
  <groupRef href="C:\Inetpub\wwwroot\newadswebsite\Apps\ boot_to_da.xml " />
  <groupRef href="C:\Inetpub\wwwroot\newadswebsite\Apps\ install_FPSE_2002.XML" />
</sequence>
```

Appendix 5.10 – XSLT document

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
     xmlns:ts="http://schemas.microsoft.com/ads/2003/sequence"
     xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

   <xsl:template match="/">
      <sequence command="" description="Wrapper Install"
xmlns="http://schemas.microsoft.com/ads/2003/sequence" version="1">
         <xsl:apply-templates select="//groupRef"/>
      </sequence>
   </xsl:template>

   <xsl:template match="groupRef">
      <xsl:copy-of select="document(@href)//ts:task" />
   </xsl:template>

</xsl:stylesheet>
```

Appendix 5.11 – Code to create the wrapper document and merge the XML files

```
// Go through each Job and write the tasks to an XML file
for(int i=0;i<=dtgJobPositions.Items.Count-1;i++)
{
   // Open the XML file and format it
   string XmlWrapperPath = "C:\\Inetpub\\wwwroot\\newadswebsite\\wrappers\\Config"+
txtCurrentConfigID.Text +"pt"+i.ToString()+"Wrapper.xml";
   string XmlOutputPath = "C:\\Inetpub\\wwwroot\\newadswebsite\\configjobs\\Config"+
txtCurrentConfigID.Text +"pt"+i.ToString()+".xml";
   string XslPath="C:\\Inetpub\\wwwroot\\newadswebsite\\transform.xsl";
   XmlTextWriter WrapperFile = new XmlTextWriter (XmlWrapperPath, null);
   WrapperFile.Formatting = Formatting.Indented;
   WrapperFile.WriteStartDocument(false);

   //Start sequence element
   WrapperFile.WriteStartElement("sequence");

   for(int j=0;j<=JobArray[i].Items.Count-1;j++)
   {
      // Update the task stats to show it has been used
      // Create SQL command to get the tasks counter stats
      cmd = new SqlCommand("SELECT Counter, CurrentCounter FROM webTask WHERE
TaskID=@TaskID",conn);

      // Create cmd parameters for the TaskID
      cmd.Parameters.Add("@TaskID",SqlDbType.Int);

      // Set the parameter to the TaskID
      cmd.Parameters["@TaskID"].Value=Convert.ToInt16(JobArray[i].Items[j].Cells[0].Text);

      // Execute the SQL command
      readResult = cmd.ExecuteReader();

      //Create a grid to hold the Counter values
      DataGrid TaskStats = new DataGrid();
      TaskStats.DataSource=readResult;
      TaskStats.DataBind();

      //Close the reader
      readResult.Close();

      // Update the counter stats for that task
      // Create SQL command update the task counter stats
      cmd = new SqlCommand("UPDATE webTask SET Counter=@counter,
CurrentCounter=@CurrentCounter WHERE TaskID=@TaskID",conn);

      // Create cmd parameters for the counter stats and TaskID
      cmd.Parameters.Add("@Counter",SqlDbType.Int);
      cmd.Parameters.Add("@CurrentCounter",SqlDbType.Int);
      cmd.Parameters.Add("@TaskID",SqlDbType.Int);

      // Set the parameter to add one to stats

   cmd.Parameters["@Counter"].Value=Convert.ToInt16(TaskStats.Items[0].Cells[0].Text)+1;
      cmd.Parameters["@CurrentCounter"].Value=
Convert.ToInt16(TaskStats.Items[0].Cells[1].Text)+1;
      cmd.Parameters["@TaskID"].Value=Convert.ToInt16(JobArray[i].Items[j].Cells[0].Text);
```

```
        // Execute the SQL command
        readResult = cmd.ExecuteReader();

        // Close the reader
        readResult.Close();

        // Read the path for the task from the database
        // Create SQL command to get the XML path
        cmd = new SqlCommand("SELECT XMLPath FROM webTask WHERE
TaskID=@TaskID",conn);

        // Create cmd parameters for the TaskID
        cmd.Parameters.Add("@TaskID",SqlDbType.Int);

        // Set the parameter to the TaskID
        cmd.Parameters["@TaskID"].Value=Convert.ToInt16(JobArray[i].Items[j].Cells[0].Text);

        // Execute the SQL command
        readResult = cmd.ExecuteReader();

        // Create a DataGrid to hold the XML file path
        DataGrid XMLFilePath = new DataGrid();
        XMLFilePath.DataSource=readResult;
        XMLFilePath.DataBind();

        // Close the reader
        readResult.Close();

        //Write the groupRef element
        WrapperFile.WriteStartElement("groupRef", null);
        WrapperFile.WriteAttributeString("href",XMLFilePath.Items[0].Cells[0].Text);
        WrapperFile.WriteEndElement();
    }
    //End sequence element
    WrapperFile.WriteEndElement();

    //Write the XML file and close
    WrapperFile.Flush();
    WrapperFile.Close();

    // Merge the XML files by apply the XSLT document
    // Load the Xml doc
    XPathDocument XmlWrapperPathDoc = new XPathDocument(XmlWrapperPath) ;

    XslTransform XslTrans = new XslTransform() ;

    // Load the Xsl
    XslTrans.Load(XslPath) ;

    // Create the output stream
    XmlTextWriter XmlOutputFile = new XmlTextWriter(XmlOutputPath, null);

    // Do the actual transform of Xml
    XslTrans.Transform(XmlWrapperPathDoc,null, XmlOutputFile);
    XmlOutputFile.Close();
}
```

Appendix 5.12 – Code to create ADS computer sets, add computers to the computer sets and update the computer statistics

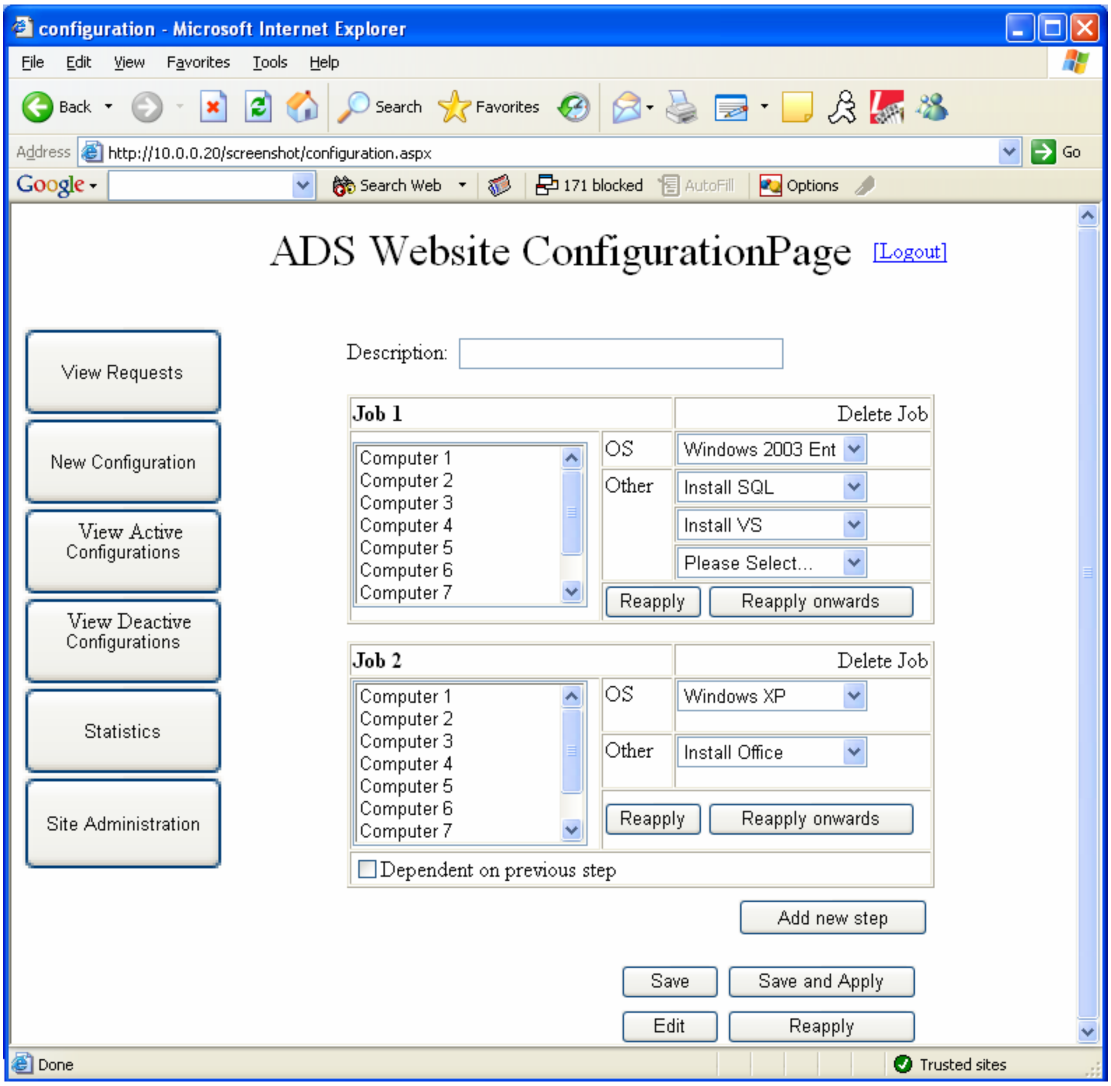
```
// Go through each job and add the computer from the relevant list to an ADS set
for(int i=0;i<=dtgJobPositions.Items.Count-1;i++)
{
   ListBox tempbox = new ListBox();

   switch(dtgJobPositions.Items[i].Cells[1].Text)
   {
      case "1":tempbox = listComputers1; tempbox.ID="list1"; break;
      case "2":tempbox = listComputers2; tempbox.ID="list2"; break;
      case "3":tempbox = listComputers3; tempbox.ID="list3"; break;
      case "4":tempbox = listComputers4; tempbox.ID="list4"; break;
   }

   // Create a new object to interface with ADS
   ManagementObject ADSSetToDelete = new
ManagementObject("\\\\.\\root\\microsoftADS:Sets.Name=\""+tempbox.ID+"\"");

   // Delete the set that already exists
   ADSSetToDelete.Delete();

   // Create a new class and an instance of a set
  ManagementClass ADSSetClass = new ManagementClass("\\\\.\\root\\microsoftADS:Sets");
   ManagementObject NewSet = ADSSetClass.CreateInstance();

   // Set the name and description for the set
   NewSet["Name"] = tempbox.ID;
   NewSet["Description"] = "A list for computers in " + tempbox.ID;

   // Add the set to the set list
   NewSet.Put();

   // Go through each computer and see if it is selected
   foreach(ListItem item in tempbox.Items)
   {
      if(item.Selected)
      {
         // Add the computer to the set
         object[] DeviceName = {"Devices.Name=\"" + item.Text + "\""};
         NewSet.InvokeMethod("AddDevice", DeviceName);

         // Update the computer stats to show it has been used
         // Create SQL command to get the computer counter stats
         cmd = new SqlCommand("SELECT Counter, CurrentCounter FROM webComputer
WHERE ADSComputerID=@ADSComputerID",conn);

         // Create cmd parameters for ADSComputerID
         cmd.Parameters.Add("@ADSComputerID",SqlDbType.Int);

         // Set the parameter to the computer id
         cmd.Parameters["@ADSComputerID"].Value=item.Value;

         // Execute the SQL command
         readResult = cmd.ExecuteReader();

         // Create a grid to hold the Counter values
```

```
            DataGrid CounterStats = new DataGrid();
            CounterStats.DataSource=readResult;
            CounterStats.DataBind();

            // Close the  reader
            readResult.Close();

            // Update the counter stats for that computer and job ID
            // Create SQL command to update the counter stats and the JobID
            cmd = new SqlCommand("UPDATE webComputer SET JobID=@JobID,
Counter=@counter, CurrentCounter=@CurrentCounter WHERE
ADSComputerID=@ADSComputerID",conn);

            // Create cmd parameters for counters, JobID and computerID
            cmd.Parameters.Add("@ADSComputerID",SqlDbType.Int);
            cmd.Parameters.Add("@Counter",SqlDbType.Int);
            cmd.Parameters.Add("@CurrentCounter",SqlDbType.Int);
            cmd.Parameters.Add("@JobID",SqlDbType.Int);

            // Set the parameter to add one to the stats and the new JobID for the selected
computer
            cmd.Parameters["@ADSComputerID"].Value=item.Value;
            cmd.Parameters["@Counter"].Value=
Convert.ToInt16(CounterStats.Items[0].Cells[0].Text)+1;
            cmd.Parameters["@CurrentCounter"].Value=
Convert.ToInt16(CounterStats.Items[0].Cells[1].Text)+1;
            cmd.Parameters["@JobID"].Value=dtgJobPositions.Items[i].Cells[0].Text;

            // Execute the SQL command
            readResult = cmd.ExecuteReader();

            // Close the reader
            readResult.Close();
        }
    }
}
```

Appendix 5.13 – Code to run the XML job on the ADS sets created

```
// Run the jobs on the sets
for(int i=0;i<=dtgJobPositions.Items.Count-1;i++)
{
   bool OkToRun=false;
   // Check to see if we should run the current job
   // if we are running all jobs then it is ok to run
   if(RunOnly==0&&RunFromOnwards==0) OkToRun=true;
   // if we are running 1 job and this is it then it is ok to run
   else if(RunFromOnwards==0 &&
RunOnly==Convert.ToInt16(dtgJobPositions.Items[i].Cells[1].Text)) OkToRun=true;
   // we we are running jobs higher than a certain jobs and this job is then its ok to run
   else if(RunOnly==0 &&
RunFromOnwards<=Convert.ToInt16(dtgJobPositions.Items[i].Cells[1].Text)) OkToRun=true;
   // else it is not ok to run the job
   else OkToRun=false;

   if(OkToRun)
   {
      string SetName="";
      string JobToRun = "C:\\Inetpub\\wwwroot\\newadswebsite\\configjobs\\Config"+
txtCurrentConfigID.Text +"pt"+i.ToString()+".xml";

      switch(dtgJobPositions.Items[i].Cells[1].Text)
      {
         case "1": SetName="list1"; break;
         case "2": SetName="list2"; break;
         case "3": SetName="list3"; break;
         case "4": SetName="list4"; break;
      }

      // Create a new scope
      ManagementScope NewScope = new ManagementScope("\\\\.\\root\\microsoftADS");
      // Build the object path for the set
      ManagementPath NewPath = new ManagementPath("Sets.Name=\"" + SetName + "\"");
      // Get Target Object
      ManagementObject SetToRunJobOn = new ManagementObject(NewScope, NewPath,
null);

      // Set Method Parameters
      ManagementBaseObject JobParameters =
SetToRunJobOn.GetMethodParameters("Execute");

      // Set the Command to run the job
      JobParameters["Command"] = JobToRun;
      JobParameters["Description"] = txtDescription.Text;

      // Run the job
      ManagementBaseObject objReturn = SetToRunJobOn.InvokeMethod("Execute",
JobParameters, null);
   }
   else
   {
      lblError.Text="The ApplyConfig function takes inputs between zero and four only";
      lblError.ForeColor=Color.Red;
   }
}
```

Appendix 5.14 – Prototype Configuration page design



http://10.0.0.20/screenshot/configuration.aspx

ADS Website ConfigurationPage [Logout]

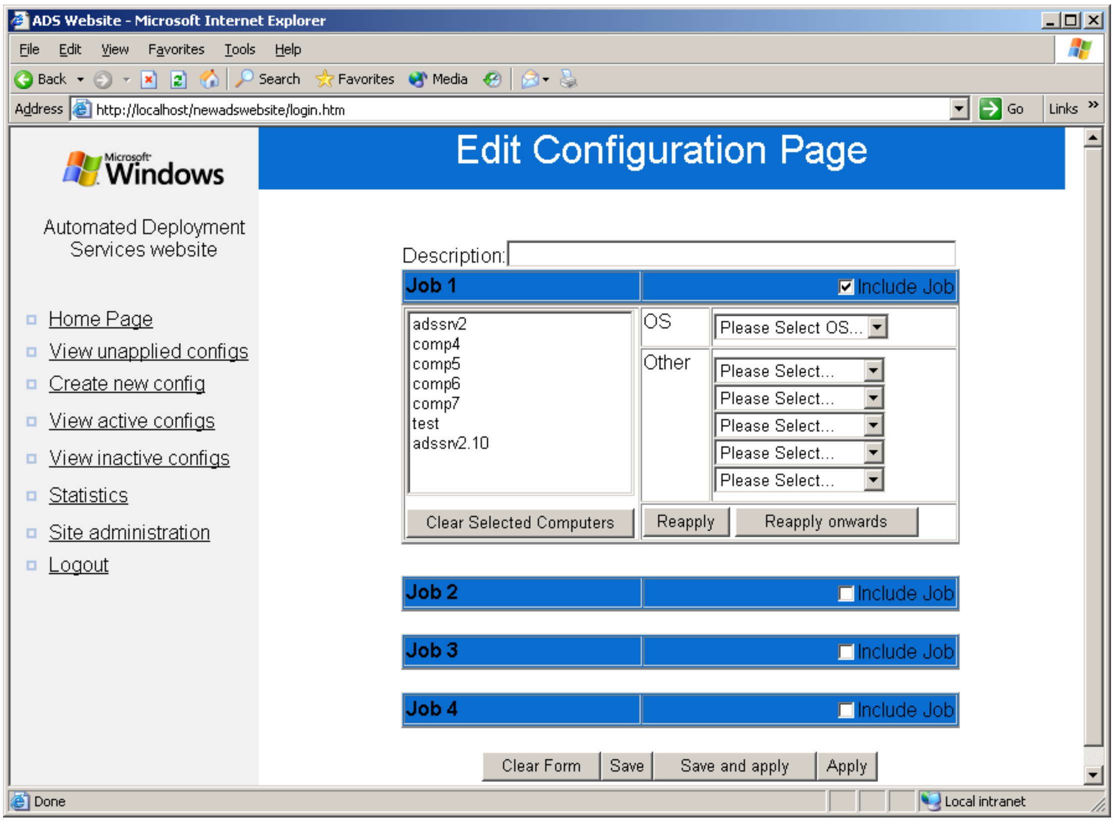
Description:

Job 1		Delete Job
Computer 1 Computer 2 Computer 3 Computer 4 Computer 5 Computer 6 Computer 7	OS	Windows 2003 Ent
	Other	Install SQL Install VS Please Select...

Job 2		Delete Job
Computer 1 Computer 2 Computer 3 Computer 4 Computer 5 Computer 6 Computer 7	OS	Windows XP
	Other	Install Office

☐ Dependent on previous step

Appendix 5.15 – Final Configuration page screenshot



Edit Configuration Page

Description:

Job 1 ☑ Include Job

adssrv2
comp4
comp5
comp6
comp7
test
adssrv2.10

OS: Please Select OS...

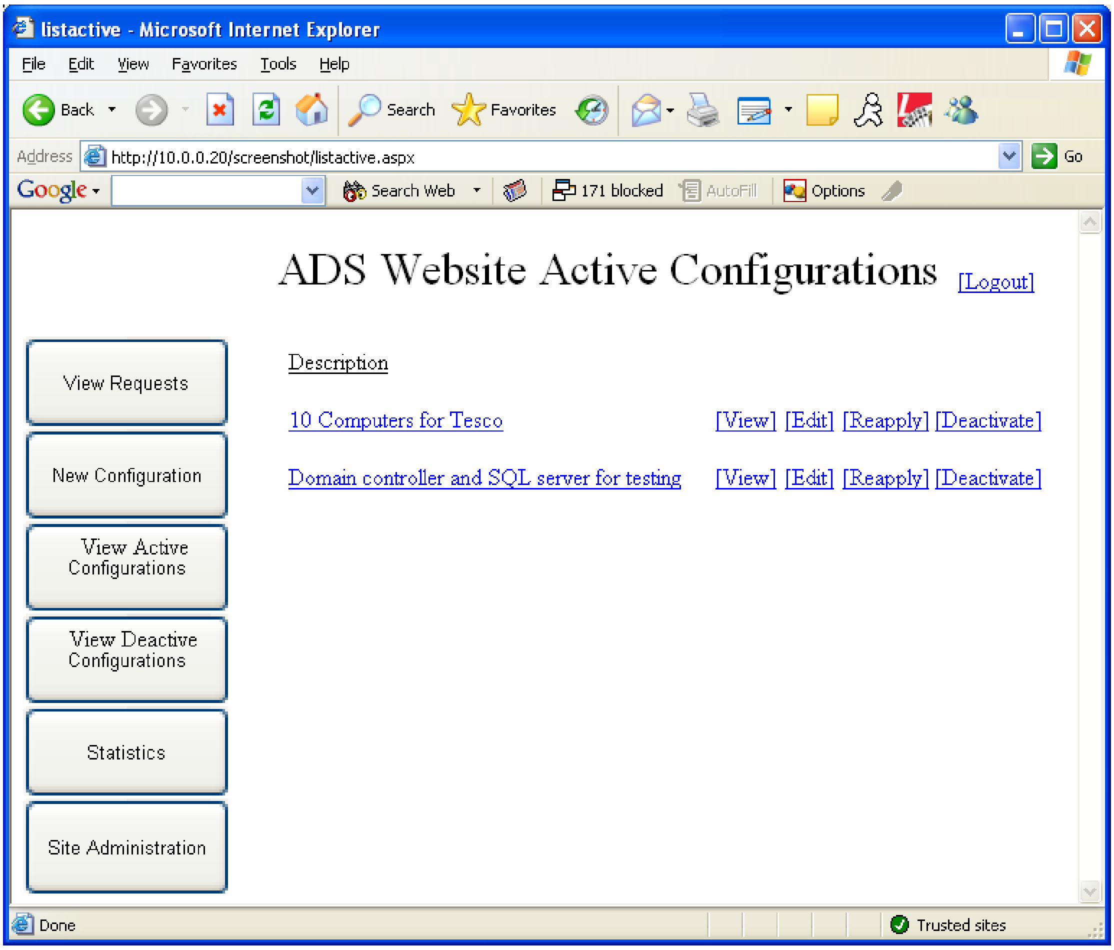
Other: Please Select... / Please Select... / Please Select... / Please Select... / Please Select...

Job 2 □ Include Job

Job 3 □ Include Job

Job 4 □ Include Job

Appendix 5.16 – Prototype view active configurations page



listactive - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://10.0.0.20/screenshot/listactive.aspx
Go
Google
Search Web
171 blocked
AutoFill
Options
Done
Trusted sites

ADS Website Active Configurations [Logout]

View Requests

New Configuration

View Active Configurations

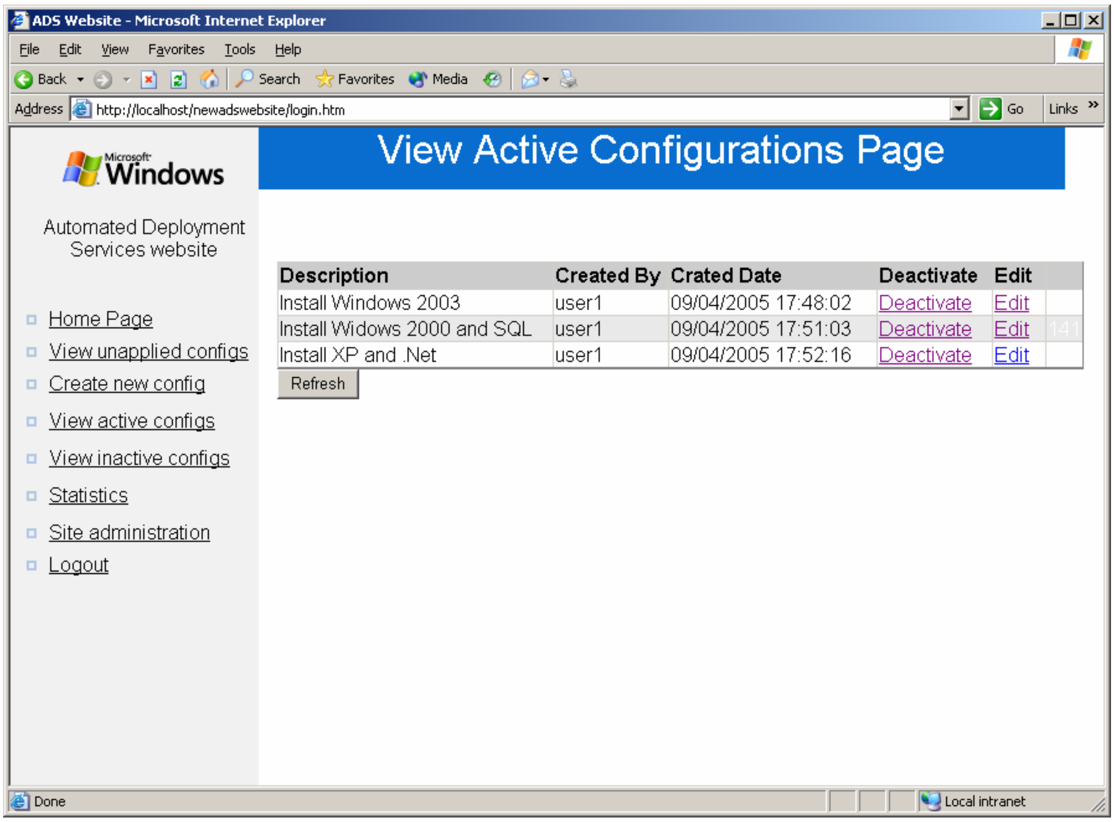
View Deactive Configurations

Statistics

Site Administration

Description	
10 Computers for Tesco	[View] [Edit] [Reapply] [Deactivate]
Domain controller and SQL server for testing	[View] [Edit] [Reapply] [Deactivate]

Appendix 5.17 – Final view active configurations page



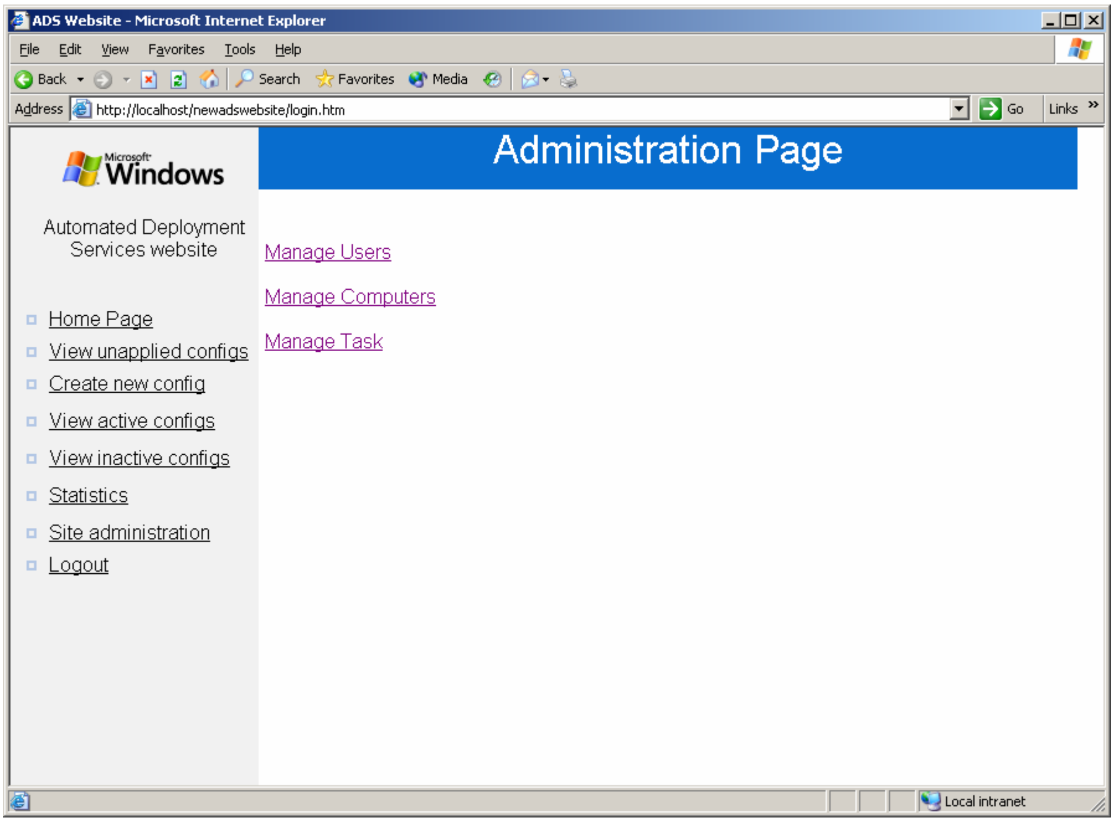
ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Automated Deployment Services website
Home Page
View unapplied configs
Create new config
View active configs
View inactive configs
Statistics
Site administration
Logout
Done
Local intranet

View Active Configurations Page

Description	Created By	Crated Date	Deactivate	Edit
Install Windows 2003	user1	09/04/2005 17:48:02	Deactivate	Edit
Install Widows 2000 and SQL	user1	09/04/2005 17:51:03	Deactivate	Edit
Install XP and .Net	user1	09/04/2005 17:52:16	Deactivate	Edit

Refresh

Appendix 5.18 – Final Administration page screenshot


ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Local intranet

Administration Page

Automated Deployment Services website

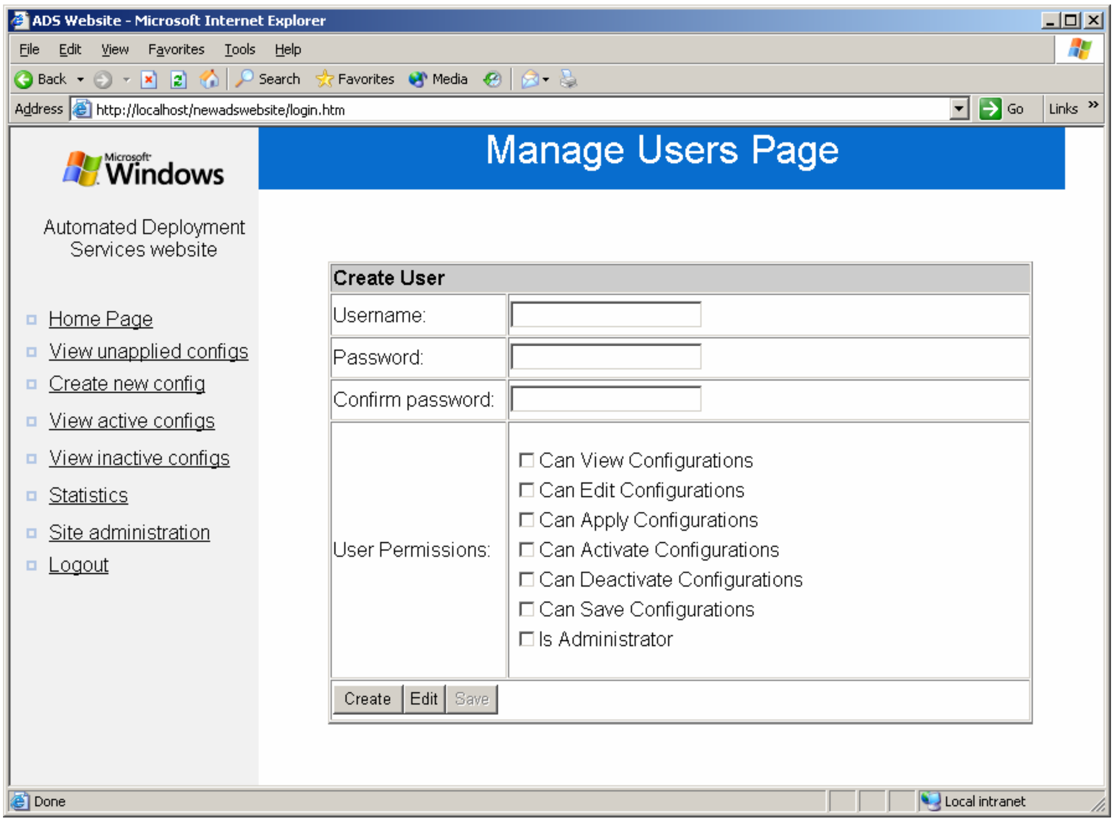
- Home Page
- View unapplied configs
- Create new config
- View active configs
- View inactive configs
- Statistics
- Site administration
- Logout

Manage Users

Manage Computers

Manage Task

Appendix 5.19 – Final Manage users page in create mode screenshot


ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Done
Local intranet

Manage Users Page

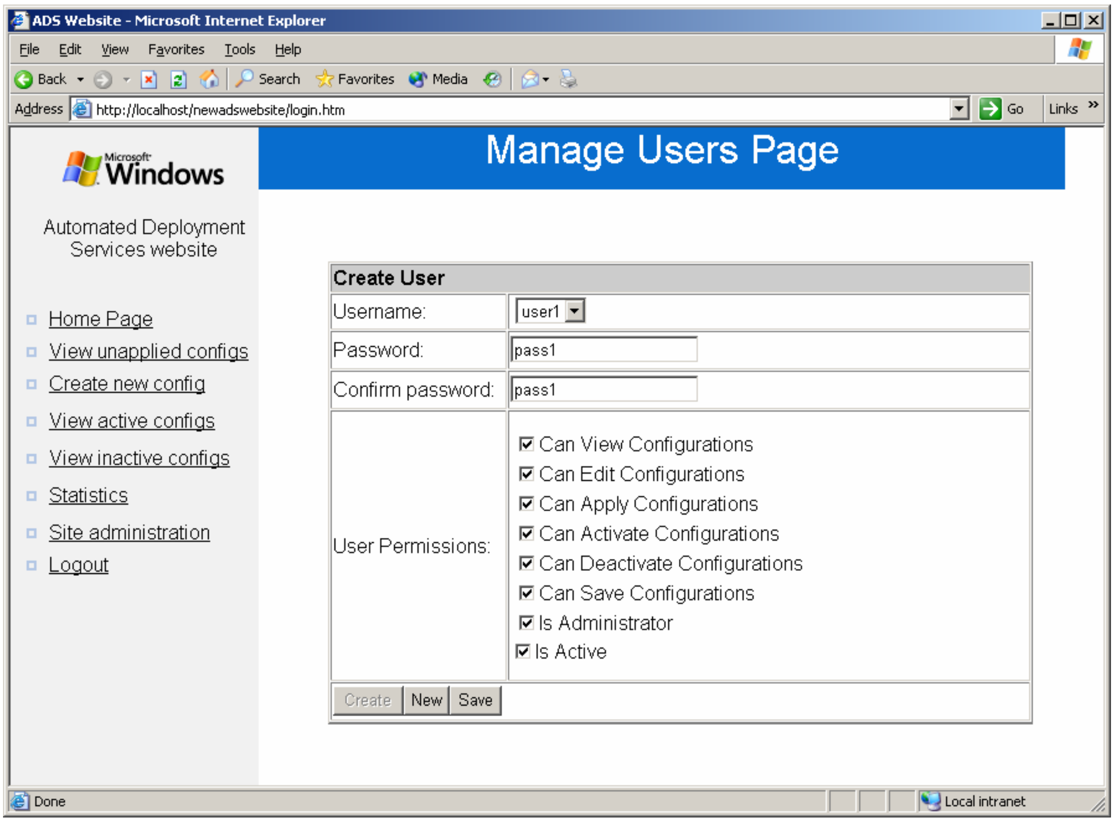
Automated Deployment Services website

- Home Page
- View unapplied configs
- Create new config
- View active configs
- View inactive configs
- Statistics
- Site administration
- Logout

Create User

Username:	
Password:	
Confirm password:	
User Permissions:	□ Can View Configurations □ Can Edit Configurations □ Can Apply Configurations □ Can Activate Configurations □ Can Deactivate Configurations □ Can Save Configurations □ Is Administrator

Appendix 5.20 – Final Manage users page in edit mode screenshot

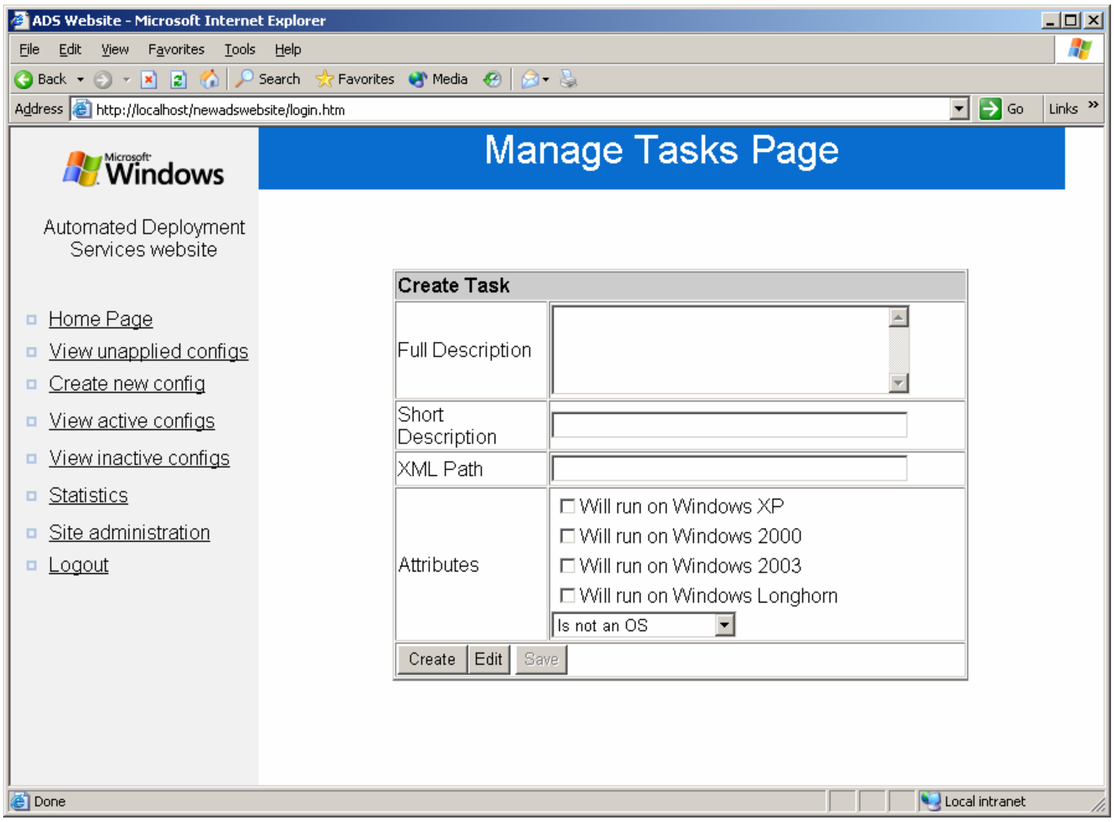

ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Automated Deployment Services website
Home Page
View unapplied configs
Create new config
View active configs
View inactive configs
Statistics
Site administration
Logout
Done
Local intranet

Manage Users Page

Create User

Username:	user1
Password:	pass1
Confirm password:	pass1
User Permissions:	☑ Can View Configurations ☑ Can Edit Configurations ☑ Can Apply Configurations ☑ Can Activate Configurations ☑ Can Deactivate Configurations ☑ Can Save Configurations ☑ Is Administrator ☑ Is Active

Appendix 5.21 – Final manage tasks page in creation mode screenshot


ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Done
Local intranet

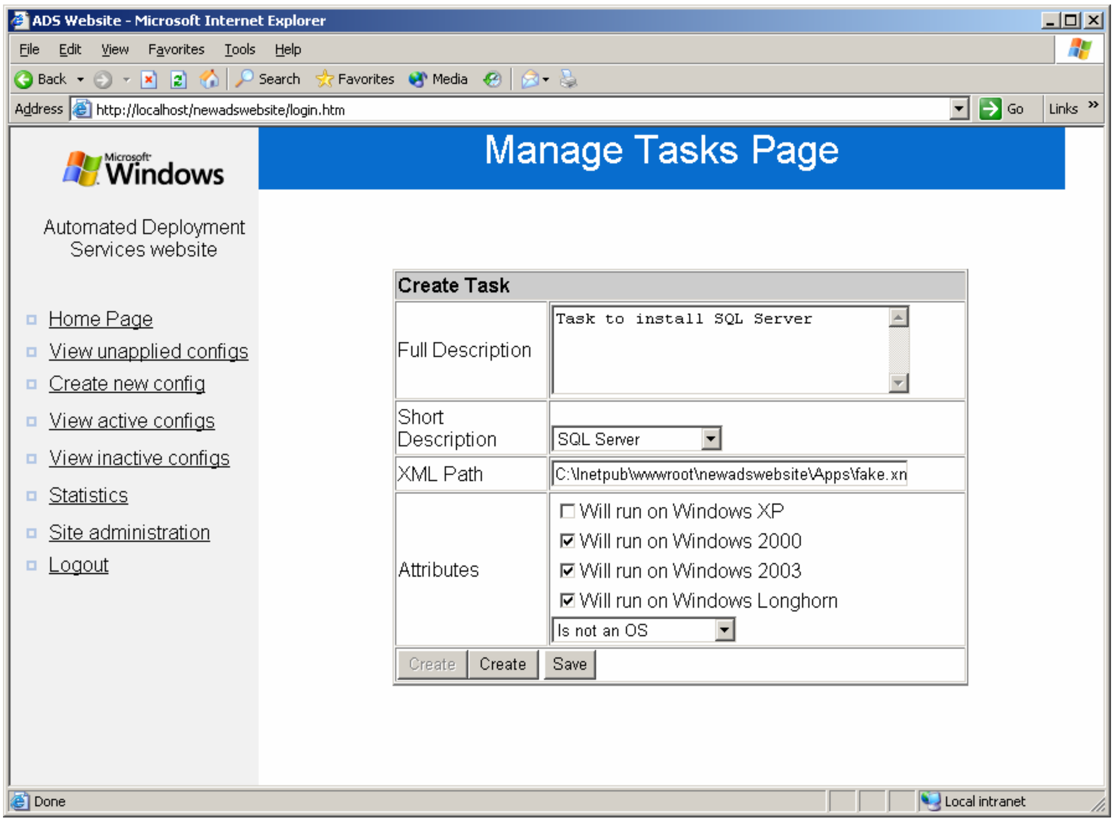
Manage Tasks Page

Automated Deployment Services website

- Home Page
- View unapplied configs
- Create new config
- View active configs
- View inactive configs
- Statistics
- Site administration
- Logout

Create Task	
Full Description	
Short Description	
XML Path	
Attributes	□ Will run on Windows XP □ Will run on Windows 2000 □ Will run on Windows 2003 □ Will run on Windows Longhorn Is not an OS ▾
Create Edit Save	

Appendix 5.22 – Final manage tasks page in edit mode screenshot



ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Done
Local intranet

Manage Tasks Page

Automated Deployment Services website

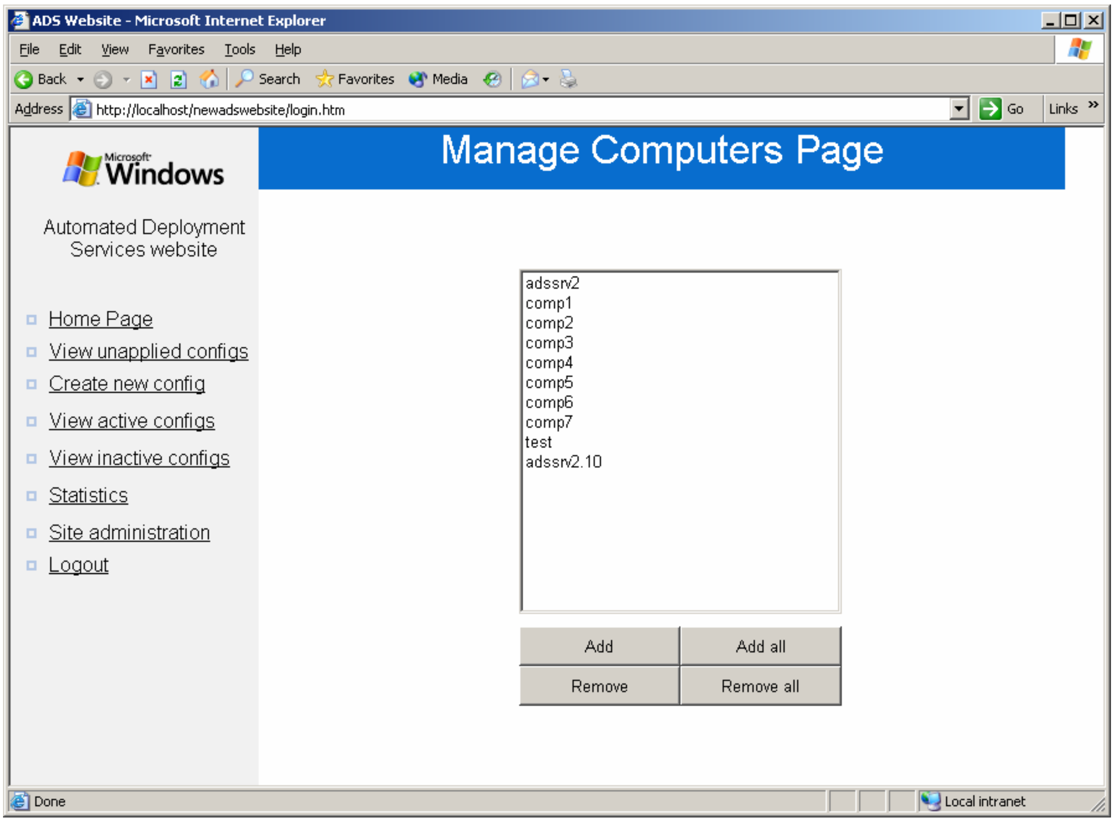
- Home Page
- View unapplied configs
- Create new config
- View active configs
- View inactive configs
- Statistics
- Site administration
- Logout

Create Task

Full Description	Task to install SQL Server
Short Description	SQL Server
XML Path	C:\Inetpub\wwwroot\newadswebsite\Apps\fake.xn
Attributes	☐ Will run on Windows XP ☑ Will run on Windows 2000 ☑ Will run on Windows 2003 ☑ Will run on Windows Longhorn Is not an OS

Create | Create | Save

Appendix 5.23 – Final manage computers page screenshot



ADS Website - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/newadswebsite/login.htm
Go Links
Microsoft Windows
Done
Local intranet

Manage Computers Page

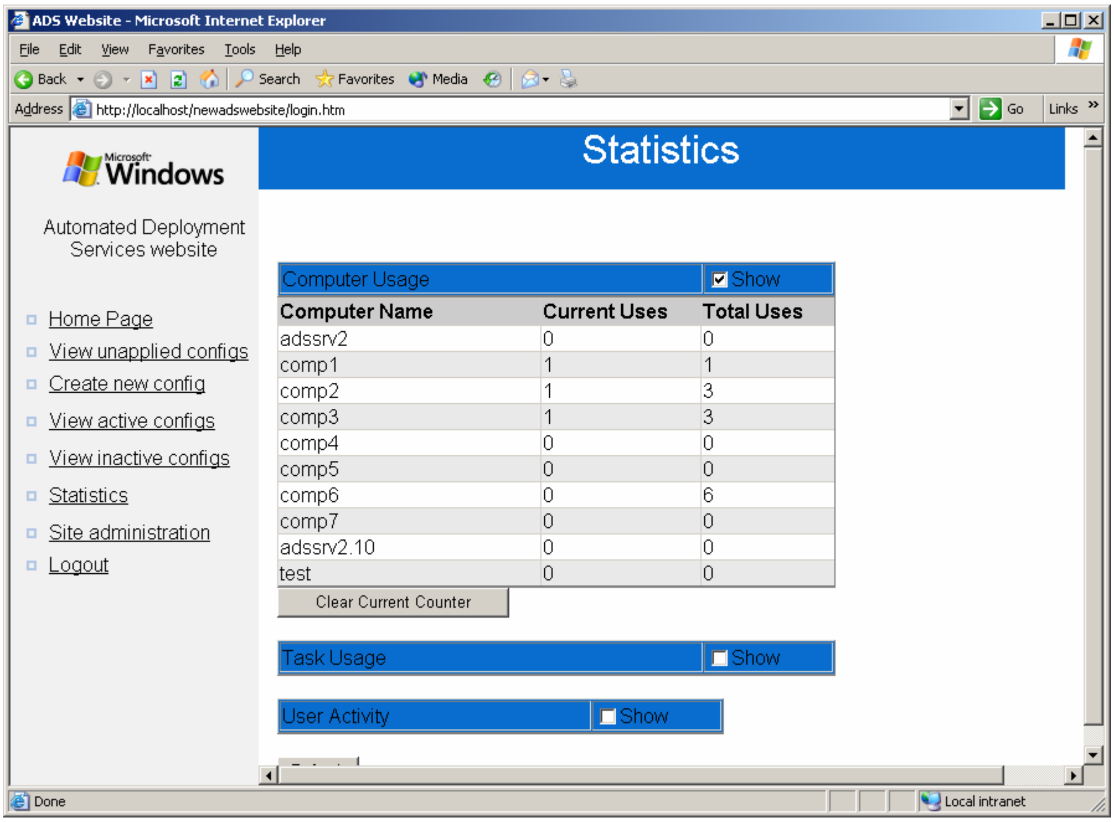
Automated Deployment Services website

- Home Page
- View unapplied configs
- Create new config
- View active configs
- View inactive configs
- Statistics
- Site administration
- Logout

adssrv2
comp1
comp2
comp3
comp4
comp5
comp6
comp7
test
adssrv2.10

Add | Add all
Remove | Remove all

Appendix 5.24 – Final Statistics page screenshot



Statistics

Computer Usage ☑ Show

Computer Name	Current Uses	Total Uses
adssrv2	0	0
comp1	1	1
comp2	1	3
comp3	1	3
comp4	0	0
comp5	0	0
comp6	0	6
comp7	0	0
adssrv2.10	0	0
test	0	0

Clear Current Counter

Task Usage □ Show

User Activity □ Show

Appendix 6.1 - Error Numbers

Error number	**Comment**
e1	You must enter a username
e2	The username or password you entered is incorrect
e3	You must type a password
e4	You must have a description
e5	You do not have the permissions to deactivate a configuration
e6	You do not have the permissions to activate a configuration
e7	The passwords do not match
e8	You must select at least one computer
e9	You must enter a short description
e10	You must enter a XML file path
e11	You must enter a valid XML file path
e12	The unapplied configurations could not be read from the database
e13	The running jobs could not be read from the database
e14	The job errors could not be read from the database
e15	The unapplied configurations could not be read from the database
e16	The OS tasks could not be read from the database
e17	The control permissions could not be applied
e18	The page controls could not be disabled
e19	The configuration could not be loaded
e20	The computers could not be loaded from the database
e21	The job number must be between 1 and 4 in the LoadOtherTasks function
e22	The other tasks could not be loaded from the database
e23	The page could not determine if each computer has only been selected once
e24	It could not be discovered if you have selected a task in all jobs
e25	The ApplyConfig function only takes inputs between zero and four
e26	The configuration could not be applied for an unknown reason
e27	The configuration could not be saved to the database
e28	The current configuration could not be deleted from the database
e29	Some of the page controls could not be cleared
e30	The page controls could not be enabled
e31	The page could not determine if you have selected at least one computer
e32	The active configurations could not be read from the database
e33	The inactive configurations could not be read from the database
e34	The user statistics could not be read the database
e35	The computer statistics could not be read the database
e36	The task statistics could not be read the database
e37	Some of the page controls could not be cleared
e38	The users attributes could not be read from the database

e39	There was an error trying to write the computer to the database
e40	There was an error trying to delete from the database
e41	The task could not be updated
e42	The task could not be created

Appendix 6.2 - Status Numbers

Status Number	**Comment**
s1	There are currently no errors
s2	There are currently no running configurations
s3	You do not have permissions to view configurations
s4	There are no unapplied configurations to view
s5	The configuration was successfully applied
s6	The configuration was successfully saved
s7	The configuration was successfully deactivated
s8	The configuration was successfully activated
s9	The user was successfully created
s10	The user was successfully updated
s11	<computer name> was added to the database
s12	<computer name> was deleted from the database
s13	<computer name> was already present in the database
s14	<computer name> was not present in the database
s15	The task was successfully updated
s16	The task was successfully created

Appendix 6.3 - Login Page test results

Page		
	Test	**Pass/Fail**
t1	Page has correct title	Pass
t2	Page uses correct font	Pass
t3	Error messages Red	Pass
t4	Status messages Black	Pass

Username textbox			
	Test	**Expected Result**	**Pass/Fail**
t5	No text entered	Error number: e1	Pass
t6	Incorrect username	Error number: e2	Pass
t7	Correct username	No error	Pass
t8	21 characters entered	Box should automatically stop at 20 characters	Pass

Password textbox			
	Test	**Expected Result**	**Pass/Fail**
t9	No text entered	Error number: e3	Pass
t10	Incorrect password	Error number: e2	Pass
t11	Correct password	No error	Pass
t12	21 characters entered	Box should automatically stop at 20 characters	Pass

Login button			
	Test	**Expected result**	**Pass/Fail**
t13	Click	Post page	Pass
t14	Check user details	Continue if username and password are correct	Pass

Appendix 6.4 - Home Page test results

Page		
	Test	**Pass/Fail**
t15	Page has correct title	Pass
t16	Page uses correct font	Pass
t17	Error messages Red	Pass
t18	Status messages Black	Pass
t19	Redirect to login.aspx if user has not logged in	Pass
t20	Display username	Pass
t21	Show new configurations link when there are new configurations	Pass
t22	Hide new configurations link when there are new configurations	Pass
t23	Show running configurations table when there are some to show	Pass
t24	Hide running configurations table when there are non to show	Pass
t25	Show errors table when there are some to show	Pass
t26	Hide errors table when there are none to show	Pass

Refresh button			
	Test	**Expected Result**	**Pass/Fail**
t27	Click	Refresh the new, running and error controls	Pass

FindNewConfigs() function			
	Test	**Expected Result**	**Pass/Fail**
t28	FindNewConfigs()	Find out if there are unapplied configurations in the database and inform the user	Pass
t29	Can capture errors	Error Label: e12	Pass

LoadRunningJobs() function			
	Test	**Expected Result**	**Pass/Fail**
t30	LoadRunningJobs()	Find out if there are any running jobs and show the user OR Status Label: s1	Pass
t31	Can capture errors	Error Label: e13	Pass

LoadErrorJobs() function			
	Test	**Expected Result**	**Pass/Fail**
t32	LoadErrorJobs()	Find out if there are any job errors show the user OR Status Label: s2	Pass
t33	Can capture errors	Error Label: e14	Pass

Appendix 6.5 - View unapplied configurations page test results

Page		
	Test	**Pass/Fail**
t34	Page has correct title	Pass
t35	Page uses correct font	Pass
t36	Error messages Red	Pass
t37	Status messages Black	Pass
t38	Redirect to login.aspx if user has not logged in	Pass
t39	Do not show data if user does not have permissions to view the page, Status Label: s3	Pass

View hyperlink			
	Test	**Expected result**	**Pass/Fail**
t40	Click	Link to configuration.aspx	Pass

Refresh button			
	Test	**Expected result**	**Pass/Fail**
t41	Click	Refresh new configurations table	Pass

UpdateTable() function			
	Test	**Expected Result**	**Pass/Fail**
t42	UpdateTable ()	Update the unapplied configurations tables with any unapplied configurations in the database OR Status Label: s4	Pass
t43	Can capture errors	Error Label: e15	Pass

Appendix 6.6 - Create New Configurations Page test results

Page		
	Test	**Pass/Fail**
t44	Page has correct title	Pass
t45	Page uses correct font	Pass
t46	Error messages Red	Pass
t47	Status messages Black	Pass
t48	Redirect to login.aspx if user has not logged in	Pass

Clear Form Button			
	Test	**Expected result**	**Pass/Fail**
t49	Click	Clear all controls	Pass

Description textbox			
	Test	**Expected result**	**Pass/Fail**
t50	No text entered	Error Label: e4	Pass
t51	Text entered	No error	Pass

Include Job 1 Check			
	Test	**Expected result**	**Pass/Fail**
t52	Select	Show job panel 1	Pass
t53	Unselect	Hide job panel 1	Pass

Include Job 2 Check			
	Test	**Expected result**	**Pass/Fail**
t54	Select	Show job panel 1	Pass
t55	Unselect	Hide job panel 1	Pass

Include Job 2 Check			
	Test	**Expected result**	**Pass/Fail**
t56	Select	Show job panel 1	Pass
t57	Unselect	Hide job panel 1	Pass

Include Job 2 Check			
	Test	**Expected result**	**Pass/Fail**
t58	Select	Show job panel 1	Pass
t69	Unselect	Hide job panel 1	Pass

Select OS dropdown list 1			
	Test	**Expected result**	**Pass/Fail**
t60	Select OS	Change other task boxes to show tasks for selected OS only	Pass
t61	No OS selected	Change other task boxes to show all tasks	Pass

Select OS dropdown list 2

	Test	**Expected result**	**Pass/Fail**
t62	Select OS	Change other task boxes to show tasks for selected OS only	Pass
t63	No OS selected	Change other task boxes to show all tasks	Pass

Select OS dropdown list 3

	Test	**Expected result**	**Pass/Fail**
t64	Select OS	Change other task boxes to show tasks for selected OS only	Pass
t65	No OS selected	Change other task boxes to show all tasks	Pass

Select OS dropdown list 4

	Test	**Expected result**	**Pass/Fail**
t66	Select OS	Change other task boxes to show tasks for selected OS only	Pass
t67	No OS selected	Change other task boxes to show all tasks	Pass

Save Button

	Test	**Expected result**	**Pass/Fail**
t68	Click (no errors)	Save the configuration to the database	Pass
t69	Click (with errors)	Show correct error message	Pass

Save and Apply Button

	Test	**Expected result**	**Pass/Fail**
t70	Click (no errors)	Save the configuration to the database then Apply it	Pass
t71	Click (with errors)	Show correct error message	Pass

Apply Button

	Test	**Expected result**	**Pass/Fail**
t72	Click (no errors)	Apply the configuration	Pass
t73	Click (with errors)	Show correct error message	Pass

Reapply Button (For Job 1)

	Test	**Expected result**	**Pass/Fail**
t74	Click (no errors)	Reapply Job 1 only	Pass
t75	Click with errors in Job 1	Show correct error message	Pass
t76	Click with errors in Job 2	Reapply Job 1 only	Fail
t77	Click with errors in Job 3	Reapply Job 1 only	Fail
t78	Click with errors in Job 4	Reapply Job 1 only	Fail

Reapply Button (For Job 2)			
	Test	**Expected result**	**Pass/Fail**
t79	Click (no errors)	Reapply Job 2 only	Pass
t80	Click with errors in Job 1	Reapply Job 2 only	Fail
t81	Click with errors in Job 2	Show correct error message	Pass
t82	Click with errors in Job 3	Reapply Job 2 only	Fail
t83	Click with errors in Job 4	Reapply Job 2 only	Fail

Reapply Button (For Job 3)			
	Test	**Expected result**	**Pass/Fail**
t84	Click (no errors)	Reapply Job 3 only	Pass
t85	Click with errors in Job 1	Reapply Job 3 only	Fail
t86	Click with errors in Job 2	Reapply Job 3 only	Fail
t87	Click with errors in Job 3	Show correct error message	Pass
t88	Click with errors in Job 4	Reapply Job 3 only	Fail

Reapply Button (For Job 4)			
	Test	**Expected result**	**Pass/Fail**
t89	Click (no errors)	Reapply Job 4 only	Pass
t90	Click with errors in Job 1	Reapply Job 4 only	Fail
t91	Click with errors in Job 2	Reapply Job 4 only	Fail
t92	Click with errors in Job 3	Reapply Job 4 only	Fail
t93	Click with errors in Job 4	Show correct error message	Pass

Reapply Onwards Button (For Job 1)			
	Test	**Expected result**	**Pass/Fail**
t94	Click (no errors)	Reapply Job 1 onwards	Pass
t95	Click with errors in Job 1	Show correct error message	Pass
t96	Click with errors in Job 2	Show correct error message	Pass
t97	Click with errors in Job 3	Show correct error message	Pass
t98	Click with errors in Job 4	Show correct error message	Pass

Reapply Onwards Button (For Job 2)			
	Test	**Expected result**	**Pass/Fail**
t99	Click (no errors)	Reapply Job 2 onwards	Pass
t100	Click with errors in Job 1	Reapply Job 2 onwards	Fail
t101	Click with errors in Job 2	Show correct error message	Pass
t102	Click with errors in Job 3	Show correct error message	Pass
t103	Click with errors in Job 4	Show correct error message	Pass

Reapply Onwards Button (For Job 3)

	Test	Expected result	Pass/Fail
t104	Click (no errors)	Reapply Job 3 onwards	Pass
t105	Click with errors in Job 1	Reapply Job 3 onwards	Fail
t106	Click with errors in Job 2	Reapply Job 3 onwards	Fail
t107	Click with errors in Job 3	Show correct error message	Pass
t108	Click with errors in Job 4	Show correct error message	Pass

Reapply Onwards Button (For Job 4)

	Test	Expected result	Pass/Fail
t109	Click (no errors)	Reapply Job 4 onwards	Pass
t110	Click with errors in Job 1	Reapply Job 4 onwards	Fail
t111	Click with errors in Job 2	Reapply Job 4 onwards	Fail
t112	Click with errors in Job 3	Reapply Job 4 onwards	Fail
t113	Click with errors in Job 4	Show correct error message	Pass

Clear Selected Computers (For Job 1)

	Test	Expected result	Pass/Fail
t114	Click	Clear selected computers box	Pass

Clear Selected Computers (For Job 2)

	Test	Expected result	Pass/Fail
t115	Click	Clear selected computers box	Pass

Clear Selected Computers (For Job 3)

	Test	Expected result	Pass/Fail
t116	Click	Clear selected computers box	Pass

Clear Selected Computers (For Job 4)

	Test	Expected result	Pass/Fail
t117	Click	Clear selected computers box	Pass

Void LoadOSTasks() function

	Test	Expected result	Pass/Fail
t118	LoadOSTasks()	Load the OS tasks into each OS task list box	Pass
t119	Can capture errors	Error Label: e16	Pass

Void SetControlPermissions(DataGrid) function			
	Test	**Expected result**	**Pass/Fail**
t120	SetControlPermissions()	Will not compile	Pass
t121	SetControlPermissions(DataGrid) Without view permission	IF they are opening a current configuration redirect the user ELSE do nothing	Pass
t122	SetControlPermissions(DataGrid) Without edit permissions	Disable all the page controls ELSE leave them enabled	Pass
t123	SetControlPermissions(DataGrid) Without apply permissions	Disable any control that allow the user to apply a configuration ELSE leave them enabled	Pass
t124	SetControlPermissions(DataGrid) Without save permissions	Disable any control that allows the user to save a configuration ELSE leave them enabled	Pass
t125	SetControlPermissions(DataGrid) With save and apply permissions	Enable the “ Save and Apply” button ELSE disable the “Save and Apply” button	Pass
t126	Can capture errors	Error Label: e17	Pass

Void DisbaleAllControls() function			
	Test	**Expected result**	**Pass/Fail**
t127	DisableAllControls()	Disable all of the controls on the form	Pass
t128	Can capture errors	Error Label: e18	Pass

Void LoadConfig() function			
	Test	**Expected result**	**Pass/Fail**
t129	LoadConfig()	Load a configuration from the database to the form	Pass
t130	Can capture errors	Error Label: e19	Pass

Void LoadComputers() function			
	Test	**Expected result**	**Pass/Fail**
t131	LoadComputers()	Load computers that are not in use from the database and update the computer list controls	Pass
t132	Can capture errors	Error Label: e20	Pass

Void LoadOtherTasks(int) function			
	Test	**Expected result**	**Pass/Fail**
t133	LoadOtherTasks()	Will not compile	Pass
t134	LoadOtherTasks(1)	Load any tasks that is not an OS from the database and update the other task list boxes in Job 1	Pass
t135	LoadOtherTasks(2)	Load any tasks that is not an OS from the database and update the other task list boxes in Job 2	Pass
t136	LoadOtherTasks(3)	Load any tasks that is not an OS from the database and update the other task list boxes in Job 3	Pass
t137	LoadOtherTasks(4)	Load any tasks that is not an OS from the database and update the other task list boxes in Job 4	Pass
t138	LoadOtherTasks(0)	Error Label: e21	Pass
t139	LoadOtherTasks(5)	Error Label: e21	Pass
t140	Can capture errors	Error Label: e22	Pass

Bool ComputersOnlySelectedOnce() function			
	Test	**Expected result**	**Pass/Fail**
t141	ComputersOnlySelectedOnce()	IF each computer has only been selected in one Job return TRUE ELSE return FALSE	Pass
t142	Can capture errors	Error Label: e23	Pass

Bool CheckIsIncludedWithNoTasks() function			
	Test	**Expected result**	**Pass/Fail**
t143	CheckIsIncludedWithNoTasks()	IF a job is checked to be included but does not have any tasks selected return FLASE ELSE return TRUE	Pass
t144	Can capture errors	Error Label: e24	Pass

Void ApplyConfig(int , int) function			
	Test	**Expected result**	**Pass/Fail**
t145	ApplyConfig(0,0)	Apply all jobs, Status Label: s5	Pass
t146	ApplyConfig(1,0)	Apply job one onwards, Status Label: s5	Pass
t147	ApplyConfig(2,0)	Apply job two onwards, Status Label: s5	Pass
t148	ApplyConfig(3,0)	Apply job three onwards, Status Label: s5	Pass
t149	ApplyConfig(4,0)	Apply job four onwards, Status Label: s5	Pass
t150	ApplyConfig(5,0)	Error Label: e25	Pass
t151	ApplyConfig(-1,0)	Error Label: e25	Pass
t152	ApplyConfig(0,1)	Apply job one only, Status Label: s5	Pass
t153	ApplyConfig(0,2)	Apply job two only, Status Label: s5	Pass
t154	ApplyConfig(0,3)	Apply job three only, Status Label: s5	Pass
t155	ApplyConfig(0,4)	Apply job four only, Status Label: s5	Pass
t156	ApplyConfig(0,5)	Error Label: e25	Pass
t157	ApplyConfig(0,-1)	Error Label: e25	Pass
t158	Can capture errors	Error Label: e26	Pass

Void SaveConfigToDatabase() function			
	Test	**Expected result**	**Pass/Fail**
t159	SaveConfigToDatabase()	Save the current configuration to the database, Status Label: s6	Pass
t160	Can capture errors	Error Label: e27	Pass

Void DeleteConfig() function			
	Test	**Expected result**	**Pass/Fail**
t161	DeleteConfig()	Delete the current configuration from the database	Pass
t162	Can capture errors	Error Label: e28	Pass

Void ClearAllControls() function			
	Test	**Expected result**	**Pass/Fail**
t163	ClearAllControls()	Clears all of the controls on the page	Pass
t164	Can capture errors	Error Label: e29	Pass

Void EnableAllControls() function			
	Test	**Expected result**	**Pass/Fail**
t165	EnableAllControls()	Enable all of the controls on the form	Pass
t166	Can capture errors	Error Label: e30	Pass

Bool ComputerIsSelected(ListBox) function			
	Test	**Expected result**	**Pass/Fail**
t167	ComputerIsSelected()	Will not compile	Pass

t168	ComputerIsSelected(ListBox)	IF a computer is selected
		return TRUE
		ELSE return FALSE
			Pass

t169	Can capture errors	Error Label: e31	Pass

Appendix 6.7 - View active configurations page test results

Page		
	Test	**Pass/Fail**
t170	Page has correct title	Pass
t171	Page uses correct font	Pass
t172	Error messages Red	Pass
t173	Status messages Black	Pass
t174	Redirect to login.aspx if user has not logged in	Pass
t175	Redirect to login.aspx if user does not have permissions to view the page	Pass
t176	Hide active configurations table when there are non to show	Pass

Edit Hyperlink			
	Test	**Expected result**	**Pass/Fail**
t177	Click (with edit permissions)	Go to configuration.aspx	Pass
t178	Click (without edit permissions)	Go to configuration.aspx	Pass

Deactivate Hyperlink			
	Test	**Expected result**	**Pass/Fail**
t179	Click (with deactivate permissions)	Deactivate the configuration, Status Label: s7	Pass
t180	Click (without deactivate permissions)	Error Label: e5	Pass

Refresh Button			
	Test	**Expected result**	**Pass/Fail**
t181	Click	Refresh the active configurations list	Pass

Void UpdateTable() function			
	Test	**Expected result**	**Pass/Fail**
t182	UpdateTable()	Reads the active configurations from the database and displays them in a table	Pass
t183	Can capture errors	Error Label: e32	Pass

Appendix 6.8 - View inactive configurations page test results

Page		
	Test	**Pass/Fail**
t184	Page has correct title	Pass
t185	Page uses correct font	Pass
t186	Error messages Red	Pass
t187	Status messages Black	Pass
t188	Redirect to login.aspx if user has not logged in	Pass
t189	Redirect to login.aspx if user does not have permissions to view the page	Pass
t190	Hide inactive configurations table when there are non to show	Pass

View Hyperlink			
	Test	**Expected result**	**Pass/Fail**
t191	Click	Go to configuration.aspx	Pass

Activate Hyperlink			
	Test	**Expected result**	**Pass/Fail**
t192	Click (with activate permissions)	Deactivate the configuration, Status Label: s8	Pass
t193	Click (without activate permissions)	Error Label: e6	Pass

Refresh Button			
	Test	**Expected result**	**Pass/Fail**
t194	Click	Refresh the active configurations list	Pass

Void UpdateTable() function			
	Test	**Expected result**	**Pass/Fail**
t195	UpdateTable()	Reads the deactive configurations from the database and displays them in a table	Pass
t196	Can capture errors	Error Label: e33	Pass

Appendix 6.9 - Statistics page test results

Page		
	Test	**Pass/Fail**
t197	Page has correct title	Pass
t198	Page uses correct font	Pass
t199	Error messages Red	Pass
t200	Status messages Black	Pass
t201	Redirect to login.aspx if user has not logged in	Pass
t202	Hide statistics if the user does not have the permissions to view them	Pass

Clear current counter (Computer) button			
	Test	**Expected result**	**Pass/Fail**
t203	Click	Clear CurrentCounter in database	Pass

Clear current counter (Task) button			
	Test	**Expected result**	**Pass/Fail**
t204	Click	Clear CurrentCounter in database	Pass

Clear current counter (User) button			
	Test	**Expected result**	**Pass/Fail**
t205	Click	Clear CurrentCounter in database	Pass

Show (Computer) checkbox			
	Test	**Expected result**	**Pass/Fail**
t206	Select	Show computer usage panel	Pass
t207	Unselect	Hide computer usage panel	Pass

Show (Task) checkbox			
	Test	**Expected result**	**Pass/Fail**
t208	Select	Show task usage panel	Pass
t209	Unselect	Hide task usage panel	Pass

Show (User) checkbox			
	Test	**Expected result**	**Pass/Fail**
t210	Select	Show user activity panel	Pass
t211	Unselect	Hide user activity panel	Pass

Refresh button			
	Test	**Expected result**	**Pass/Fail**
t212	Click	Refresh all of the statistics	Pass

Void LoadUserStats() function			
	Test	**Expected result**	**Pass/Fail**
t213	LoadUserStats()	Reads the user statistics from the database and displays them in a table	Pass
t214	Can capture errors	Error Label: e34	Pass

Void LoadComputerStats() function			
	Test	**Expected result**	**Pass/Fail**
t215	LoadComputerStats()	Reads the computer statistics from the database and displays them in a table	Pass
t216	Can capture errors	Error Label: e35	Pass

Void LoadTaskStats() function			
	Test	**Expected result**	**Pass/Fail**
t217	LoadTaskStats()	Reads the task statistics from the database and displays them in a table	Pass
t218	Can capture errors	Error Label: e36	Pass

Appendix 6.10 - Manage users page test results

Page		
	Test	**Pass/Fail**
t220	Page has correct title	Pass
t221	Page uses correct font	Pass
t222	Error messages Red	Pass
t223	Status messages Black	Pass
t224	Redirect to login.aspx if user has not logged in	Pass
t225	Redirect if user does not have permissions to view the page	Pass

Username textbox			
	Test	**Expected result**	**Pass/Fail**
t226	No text entered	Error: e1	Pass
t227	21 characters entered	Box should automatically stop at 20 characters	Pass
t228	Valid username entered	No error	Pass

Password textbox			
	Test	**Expected result**	**Pass/Fail**
t229	No text entered	Error: e3	Pass
t230	21 characters entered	Box should automatically stop at 20 characters	Pass
t231	Valid password entered	No error	Pass

Confirm password textbox			
	Test	**Expected result**	**Pass/Fail**
t232	No text entered	Error: e7	Pass
t233	21 characters entered	Box should automatically stop at 20 characters	Pass
t234	Matching password entered	No error	Pass

Can view configurations checkbox			
	Test	**Expected result**	**Pass/Fail**
t235	Checked	Save CanView to true in the database	Pass
t236	Unchecked	Save CanView to false in the database	Pass

Can edit configurations checkbox			
	Test	**Expected result**	**Pass/Fail**
t237	Checked	Save CanEdit to true in the database	Pass
t238	Unchecked	Save CanEdit to false in the database	Pass

Can apply configurations checkbox			
	Test	**Expected result**	**Pass/Fail**
t239	Checked	Save CanApply to true in the database	Pass
t240	Unchecked	Save CanApply to false in the database	Pass

Can activate configurations checkbox			
	Test	**Expected result**	**Pass/Fail**
t241	Checked	Save CanActivate to true in the database	Pass
t242	Unchecked	Save CanActivate to false in the database	Pass

Can deactivate configurations checkbox			
	Test	**Expected result**	**Pass/Fail**
t243	Checked	Save CanDeactivate to true in the database	Pass
t244	Unchecked	Save CanDeactivate to false in the database	Pass

Can save configurations checkbox			
	Test	**Expected result**	**Pass/Fail**
t245	Checked	Save CanSave to true in the database	Pass
t246	Unchecked	Save CanSave to false in the database	Pass

Is administrator checkbox			
	Test	**Expected result**	**Pass/Fail**
t247	Checked	Save IsAdmin to true in the database	Pass
t248	Unchecked	Save IsAdmin to false in the database	Pass

Is active checkbox			
	Test	**Expected result**	**Pass/Fail**
t249	Checked	Save IsActive to true in the database	Pass
t250	Unchecked	Save IsActive to false in the database	Pass

Create button			
	Test	**Expected result**	**Pass/Fail**
t251	Click (no errors)	Save the user to the database, Status Label s9	Pass
t252	Click (with errors)	Show correct error message	Pass

Edit button			
	Test	**Expected result**	**Pass/Fail**
t253	Click (when in create mode)	Switch page to edit mode	Pass
t254	Click (when in edit mode)	Switch page to create mode	Pass

Save button			
	Test	**Expected result**	**Pass/Fail**
t255	Click (no errors)	Update the user in database, Status Label s10	Pass
t256	Click (with errors)	Show correct error message	Pass

Void ClearAllControls() function			
	Test	**Expected result**	**Pass/Fail**
t257	ClearAllControls()	Clears all the controls on the page	Pass
t258	Can capture errors	Error Label: e37	Pass

Void UpdateUserAttributes() function			
	Test	**Expected result**	**Pass/Fail**
t259	UpdateUserAttributes()	Reads the users attributes from the database	Pass
t260	Can capture errors	Error Label: e38	Pass

Appendix 6.11 - Manage computers page test results

Page		
	Test	**Pass/Fail**
t261	Page has correct title	Pass
t262	Page uses correct font	Pass
t263	Error messages Red	Pass
t264	Status messages Black	Pass
t265	Redirect to login.aspx if user has not logged in	Pass
t266	Redirect to login.aspx if user does not have permissions to view the page	Pass

Add button			
	Test	**Expected result**	**Pass/Fail**
t267	Click with one computer selected	Add the computer to the database if it is not already present, Status Label: s11	Pass
t268	Click with many computers selected	Add the computers to the database if it is not already present, Status Label: s11	Pass
t269	Click with no computers selected	Error Label: e8	Pass

Remove button			
	Test	**Expected result**	**Pass/Fail**
t270	Click with one computer selected	Remove the computer to the database if it is present, Status Label: s12	Pass
t271	Click with many computers selected	Remove the computers to the database if it is present, Status Label: s12	Pass
t272	Click with no computers selected	Error Label: e8	Pass

Add all button			
	Test	**Expected result**	**Pass/Fail**
t273	Click with one computer selected	Add each computer to the database that is not already present, Status Label: s11	Pass
t274	Click with many computers selected	Add each computer to the database that is not already present, Status Label: s11	Pass
t275	Click with no computers selected	Add each computer to the database that is not already present, Status Label: s11	Pass

Remove all button			
	Test	**Expected result**	**Pass/Fail**
t276	Click with one computer selected	Remove all computers from the database, Status Label: s12	Pass
t277	Click with many computers selected	Remove all computers from the database, Status Label: s12	Pass
t278	Click with no computers selected	Remove all computers from the database, Status Label: s12	Pass

Void AddComputersToDB(ListItem) function			
	Test	**Expected result**	**Pass/Fail**
t279	AddComputerToDB()	Will not compile	Pass
t280	AddComputerToDB(ListItem) with computer already in the database	Adds the computer to the database, Status Label: s11	Pass
t281	AddComputerToDB(ListItem) with computer not already in the database	Do not add the computer to the database, Status Label: s13	Pass
t282	Can capture errors	Error Label: e39	Pass

Void RemoveComputerFromDB(ListItem) function			
	Test	**Expected result**	**Pass/Fail**
t283	**RemoveComputerFromDB**()	Will not compile	Pass
t284	**RemoveComputerFromDB**(ListItem) with computer in the database	Removes the computer from the database, Status Label: s12	Pass
t285	**RemoveComputerFromDB**(ListItem) with computer not in the database	Does not remove the computer from the database, Status Label: s14	Pass
t286	Can capture errors	Error Label: e40	Pass

Appendix 6.12 - Manage tasks page test results

Page		
	Test	**Pass/Fail**
t287	Page has correct title	Pass
t288	Page uses correct font	Pass
t289	Error messages Red	Pass
t290	Status messages Black	Pass
t291	Redirect to login.aspx if user has not logged in	Pass
t292	Redirect to login.aspx if user does not have permissions to view the page	Pass

Full description textbox			
	Test	**Expected result**	**Pass/Fail**
t293	No text entered	No error	Pass
t294	501 characters entered	Box should automatically stop at 500 characters	Pass

Short description textbox			
	Test	**Expected result**	**Pass/Fail**
t295	No text entered	Error: e9	Pass
t296	101 characters entered	Box should automatically stop at 100 characters	Pass

XML path textbox			
	Test	**Expected result**	**Pass/Fail**
t297	No text entered	Error: e10	Pass
t298	201 characters entered	Box should automatically stop at 200 characters	Pass
t299	Invalid path entered	Error: e11	Fail

Will run on Windows XP checkbox			
	Test	**Expected result**	**Pass/Fail**
t300	Checked	Save WillRunOnXP as true in the database	Pass
t301	Unchecked	Save WillRunOnXP as false in the database	Pass

Will run on Windows 2000 checkbox			
	Test	**Expected result**	**Pass/Fail**
t302	Checked	Save WillRunOn2k as true in the database	Pass
t303	Unchecked	Save WillRunOn2k as false in the database	Pass

Will run on Windows 2003 checkbox			
	Test	**Expected result**	**Pass/Fail**
t304	Checked	Save WillRunOn03 as true in the database	Pass
t305	Unchecked	Save WillRunOn03 as false in the database	Pass

Will run on Windows Longhorn checkbox			
	Test	**Expected result**	**Pass/Fail**
t306	Checked	Save WillRunOnLH as true in the database	Pass
t307	Unchecked	Save WillRunOnLH as false in the database	Pass

OS dropdown list			
	Test	**Expected result**	**Pass/Fail**
t308	No OS selected	All OS fields set to false in the database	Pass
t309	Windows XP selected	Save IsXP as true in the database	Pass
t310	Windows 2000 selected	Save Is2k as true in the database	Pass
t311	Windows 2003 selected	Save Is03 as true in the database	Pass
t312	Windows Longhorn selected	Save IsLH as true in the database	Pass

Edit button			
	Test	**Expected result**	**Pass/Fail**
t313	Click (when in create mode)	Switch page to edit mode	Pass
t314	Click (when in edit mode)	Switch page to create mode	Pass

Save button			
	Test	**Expected result**	**Pass/Fail**
t315	Click (no errors)	Update the task in the database, Status Label: s15	Pass
t316	Click (with errors)	Show correct error message	Pass
t317	Can capture errors	Error Label: e 41	Pass

Create button			
	Test	**Expected result**	**Pass/Fail**
t318	Click (no errors)	Create the task in the database, Status Label: s16	Pass
t319	Click (with errors)	Show correct error message	Pass
t320	Can capture errors	Error Label :42	Pass

Appendix 6.13 - Left menu page test results

Page		
	Test	**Pass/Fail**
t321	Page has correct title	Pass
t322	Page uses correct font	Pass
t323	Error messages Red	Pass
t324	Status messages Black	Pass
t325	Redirect to login.aspx if user has not logged in	Pass

Logout hyperlink			
	Test	**Expected result**	**Pass/Fail**
t325	Click	Logout the user and return to login.aspx	Pass

Home hyperlink			
	Test	**Expected result**	**Pass/Fail**
t326	Click	Redirect the user to mainpage.aspx	Pass

New configurations hyperlink			
	Test	**Expected result**	**Pass/Fail**
t327	Click	Redirect the user to configuration.aspx	Pass

View unapplied configurations hyperlink			
	Test	**Expected result**	**Pass/Fail**
t328	Click	Redirect the user to viewunappliedconfigurations.aspx	Pass
t329	User does not have view permissions	The link should not be visible	Pass
t330	User does have view permissions	The link should be visible	Pass

View active configurations hyperlink			
	Test	**Expected result**	**Pass/Fail**
t331	Click	Redirect the user to viewactiveconfigurations.aspx	Pass
t332	User does not have view permissions	The link should not be visible	Pass
t333	User does have view permissions	The link should be visible	Pass

View inactive configurations hyperlink			
	Test	**Expected result**	**Pass/Fail**
t334	Click	Redirect the user to viewinactiveconfigurations.aspx	Pass
t335	User does not have view permissions	The link should not be visible	Pass
t336	User does have view permissions	The link should be visible	Pass

Statistics hyperlink			
	Test	**Expected result**	**Pass/Fail**
t337	Click	Redirect the user to statistics.aspx	Pass
t338	User does not have view permissions	The link should not be visible	Pass
t339	User does have view permissions	The link should be visible	Pass

Site administration hyperlink			
	Test	**Expected result**	**Pass/Fail**
t340	Click	Redirect the user to administration.aspx	Pass
t341	User does not have view permissions	The link should not be visible	Pass
t342	User does have view permissions	The link should be visible	Pass

Appendix 6.14 – Example test case

Test the user management and login screen			
Step No.	**Instruction**	**Expected result**	**Pass/Fail**
Step 1	Go to the login page	The page loads	Pass
Step 2	Enter the following values username: Admin password: Pass Click 'Login'	The user gets logged in and the page redirect to the home page	Pass
Step 3	Click the link to the Site administration page	The page redirect to the Administration page	Pass
Step 4	Click the link to the Manage users page	The page redirects to the Manage users page	Pass
Step 6	Enter the following values username: <blank> password: abc123 confirm password: def456 can view: checked can edit: unchecked can apply: checked can activate: checked can deactivate: chaecked can save: checked Is administrator: checked Click 'Create'	The page displays the errors e1: "You must enter a username" e7: "The passwords do not match"	Pass
Step 7	Change the following values username: test confirm password: abc123 Click 'Create'	The user is created in the database and the page displays s9: "The user was successfully created"	Pass
Step 8	Click 'Logout' link	The user is logged out and returned to the login screen	Pass
Step 9	Enter the following values username: test password: abc123 Click 'Login'		Pass
Step 10	Click the link to the Site administration page	The page redirect to the Administration page	Pass
Step 11	Click the link to the Manage users page	The page redirects to the Manage users page	Pass
Step 12	Click the 'Edit' button	The page switches to edit	Pass

Step 13	Select ‘Admin’ from the usernames list	The information for the ‘Admin’ user is loaded	Pass
Step 14	Uncheck the Is active check box Click ‘Save’	The user is updated in the database and the page displays s10: “The user was successfully updated”	Pass
Step 15	Click ‘Logout’ link	The user is logged out and returned to the login screen	Pass
Step 16	Enter the following values username: Admin password: Pass Click ‘Login’	The page should display the error e2: “The username or password you entered is incorrect”	Pass

Appendix 6.14 – Usability Issues

Major issues
Do not understand what a ‘job’ or a ‘configuration’ is

Minor issues
Move the ‘Refresh’ Button on the view active configurations page
Move the ‘Refresh’ Button on the view inactive configurations page
Move the ‘Refresh’ Button on the view new configurations page
Move the ‘Clear Form’ Button on the configurations page
The error messages don’t tell you how to fix errors
Some of the messages should be green rather than black