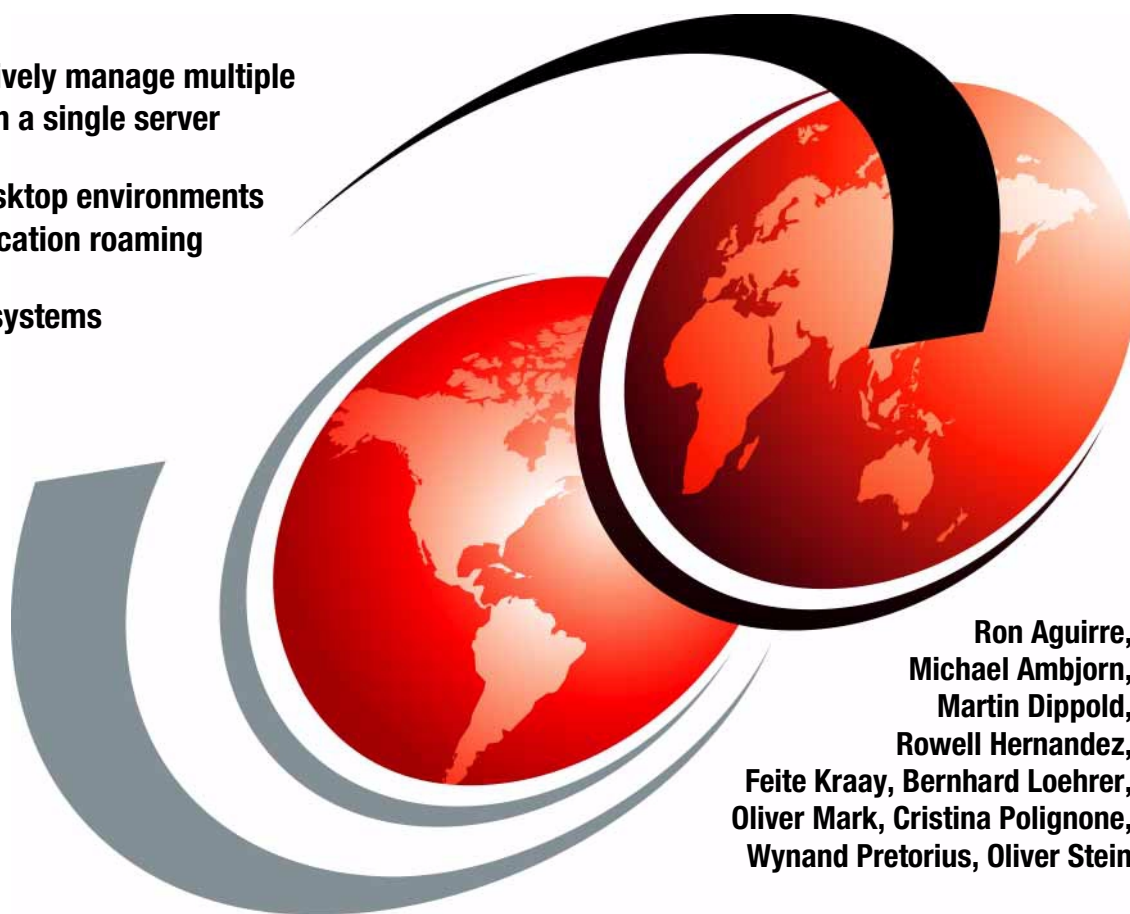


# IBM WorkSpace On-Demand 3.0.1

Cost effectively manage multiple  
clients from a single server

Custom desktop environments  
allow application roaming

Enhanced systems  
security



Ron Aguirre,  
Michael Ambjorn,  
Martin Dippold,  
Rowell Hernandez,  
Feite Kraay, Bernhard Loehrer,  
Oliver Mark, Cristina Polignone,  
Wynand Pretorius, Oliver Stein

[ibm.com/redbooks](http://ibm.com/redbooks)

**Redbooks**





International Technical Support Organization

**IBM WorkSpace On-Demand 3.0.1**

November 2000

**Take Note!**

Before using this information and the product it supports, be sure to read the general information in Appendix C, "Special notices" on page 475.

**First Edition (November 2000)**

This edition applies to Version 3, Release Number 0.1 of IBM WorkSpace On-Demand, Program Number 5648-D47 for use with the Microsoft Windows NT Server 4.0 and Microsoft Windows 2000 Server.

Comments may be addressed to:  
IBM Corporation, International Technical Support Organization  
Dept. OSJA Building 003 Internal Zip 2834  
11400 Burnet Road  
Austin, Texas 78758-3493

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

**© Copyright International Business Machines Corporation 2000. All rights reserved.**

Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

## Contents

|  |       |
|--|-------|
| <b>Figures</b> .....   | xi    |
| <b>Tables</b> .....  | xvii  |
| <b>Preface</b> .....   | xix   |
| The team that wrote this redbook. ....                         | xix   |
| The team that completed this redbook. ....                     | xxiii |
| Comments welcome. ....   | xxv   |
| <b>Chapter 1. Introduction</b> .....                           | 1     |
| 1.1 Benefits of IBM WorkSpace On-Demand 3.0.11 .....           | 1     |
| 1.1.1 Effective software and data management. ....             | 2     |
| 1.1.2 Easier end-user support. ....                            | 2     |
| 1.1.3 Enhanced security .....                                  | 2     |
| 1.1.4 Broad application support. ....                          | 3     |
| 1.1.5 End-user mobility .....                                  | 3     |
| 1.1.6 Support for multiple server operating systems .....      | 3     |
| 1.1.7 Support for multiple client operating systems .....      | 4     |
| 1.2 Architecture of IBM WorkSpace On-Demand 3.0.1 .....        | 4     |
| 1.2.1 The Administration Console .....                         | 5     |
| 1.2.2 The Configuration Server .....                           | 6     |
| 1.2.3 The Deployment Server .....                              | 6     |
| 1.2.4 Design considerations .....                              | 7     |
| 1.3 Client support. ....                                       | 7     |
| 1.3.1 Windows 2000 Professional Edition .....                  | 8     |
| 1.3.2 Windows NT Workstation 4.0 .....                         | 8     |
| 1.3.3 Windows 98 Second Edition .....                          | 8     |
| 1.3.4 IBM OS/2 Client .....                                    | 8     |
| <b>Chapter 2. Installation</b> .....                           | 9     |
| 2.1 IBM WorkSpace On-Demand 3.0.1 hardware prerequisites ..... | 9     |
| 2.1.1 Supported network adapters. ....                         | 9     |
| 2.1.2 Client hardware considerations. ....                     | 10    |
| 2.1.3 Remote boot capability. ....                             | 10    |
| 2.1.4 Windows 2000 client hardware .....                       | 11    |
| 2.1.5 Windows NT Workstation 4.0 client hardware .....         | 11    |
| 2.1.6 Windows 98 client hardware .....                         | 12    |
| 2.1.7 IBM OS/2 client hardware .....                           | 13    |
| 2.2 Software prerequisites .....                               | 14    |
| 2.2.1 IBM OS/2 .....   | 14    |
| 2.2.2 Windows 2000 Professional Edition .....                  | 14    |

|                   |  |           |
|-------------------|--|-----------|
| 2.2.3             | Windows NT Workstation 4.0                                 | 14        |
| 2.2.4             | Windows 98 Second Edition                                  | 14        |
| 2.3               | Installation roadmap                                       | 15        |
| 2.4               | Java Runtime Environment V1.3                              | 15        |
| 2.5               | The DHCP PXE boot mechanism                                | 16        |
| 2.5.1             | Supporting a DHCP PXE boot environment                     | 16        |
| 2.5.2             | The PXE Boot Process                                       | 18        |
| 2.6               | Installing IBM WorkSpace On-Demand 3.0.1                   | 24        |
| 2.6.1             | Installing WorkSpace On-Demand 3.0.1 on Windows Servers    | 25        |
| 2.6.2             | Installing IBM WorkSpace On-Demand 3.0.1 Client Enablement | 34        |
| 2.6.3             | Removing client-machine operating system support           | 40        |
| 2.6.4             | Directory structure for IBM WorkSpace On-Demand 3.0.1      | 40        |
| 2.7               | Installing and configuring IP services                     | 42        |
| 2.7.1             | DHCP.DCF   | 42        |
| 2.7.2             | DHCP configuration - The GUI way                           | 43        |
| 2.7.3             | BINL service   | 47        |
| 2.7.4             | TFTP service   | 49        |
| 2.7.5             | Other IP services  | 50        |
| 2.7.6             | Protected disk driver                                      | 51        |
| 2.8               | Uninstalling IBM WorkSpace On-Demand 3.0.1                 | 52        |
| <b>Chapter 3.</b> | <b>Administration</b>                                      | <b>57</b> |
| 3.1               | Resources and tasks  | 58        |
| 3.2               | The administration GUI                                     | 59        |
| 3.3               | Task flow  | 60        |
| 3.4               | The administration CLI                                     | 62        |
| 3.5               | Command automation   | 66        |
| 3.5.1             | Simplifying commands using JavaScript                      | 66        |
| 3.5.2             | Automating repetitive commands                             | 71        |
| 3.5.3             | Automating the console                                     | 73        |
| 3.6               | Administering users  | 76        |
| 3.6.1             | Managing users in the IBM WorkSpace On-Demand 3.0.1 GUI    | 78        |
| 3.6.2             | Managing users in the IBM WorkSpace On-Demand 3.0.1 CLI    | 82        |
| 3.6.3             | Defining a user in the CLI                                 | 83        |
| 3.6.4             | Deleting a user in the CLI                                 | 89        |
| 3.6.5             | Defining home directory for users                          | 91        |
| 3.6.6             | Defining user-specific logon scripts                       | 92        |
| 3.6.7             | Importing users from Windows 2000 or NT in the CLI         | 93        |
| 3.6.8             | Managing groups in the CLI                                 | 94        |
| 3.7               | Administering the server                                   | 94        |
| 3.7.1             | Connect to a WorkSpace On-Demand configuration server      | 95        |
| 3.7.2             | Disconnect from a WorkSpace On-Demand configuration server | 95        |
| 3.7.3             | Administering the configuration server                     | 95        |

|                   |  |            |
|-------------------|--|------------|
| 3.7.4             | Administering the deployment server . . . . .                  | 96         |
| 3.7.5             | Loading JavaScript files on WorkSpace On-Demand servers . . .  | 99         |
| 3.7.6             | Changing WorkSpace On-Demand server ports . . . . .            | 99         |
| 3.7.7             | Changing network and server time out values . . . . .          | 101        |
| <b>Chapter 4.</b> | <b>Defining machines . . . . .</b>                             | <b>103</b> |
| 4.1               | General machine representation. . . . .                        | 103        |
| 4.2               | Defining Windows 2000 client machines. . . . .                 | 105        |
| 4.2.1             | Windows 2000 client parameters . . . . .                       | 105        |
| 4.2.2             | The machine define command for Windows 2000. . . . .           | 111        |
| 4.2.3             | Windows 2000 client installation and boot . . . . .            | 112        |
| 4.3               | Defining Windows NT client machines . . . . .                  | 124        |
| 4.3.1             | Windows NT client parameters . . . . .                         | 124        |
| 4.3.2             | The machine define command for Windows NT . . . . .            | 130        |
| 4.3.3             | Windows NT client installation and boot . . . . .              | 131        |
| 4.4               | Defining Windows 98 client machines. . . . .                   | 143        |
| 4.4.1             | Windows 98 client parameters . . . . .                         | 143        |
| 4.4.2             | The machine define command for Windows 98. . . . .             | 149        |
| 4.4.3             | Windows 98 client installation and boot . . . . .              | 149        |
| 4.5               | Defining OS/2 client machines . . . . .                        | 160        |
| 4.5.1             | OS/2 client parameters . . . . .                               | 161        |
| 4.5.2             | The machine define command for OS/2 . . . . .                  | 168        |
| 4.5.3             | OS/2 client boot . . . . .                                     | 168        |
| 4.5.4             | The OS/2 client machine FIT file. . . . .                      | 173        |
| 4.6               | Machine templates . . . . .                                    | 177        |
| 4.7               | Other machine commands . . . . .                               | 179        |
| 4.8               | Managing client machines using the administration GUI. . . . . | 181        |
| 4.8.1             | Defining a machine. . . . .                                    | 182        |
| 4.8.2             | Defining a template . . . . .                                  | 191        |
| 4.9               | Customizing Windows client machine images. . . . .             | 191        |
| 4.9.1             | Customizing state.ini . . . . .                                | 191        |
| 4.9.2             | Executing commands after installation . . . . .                | 197        |
| 4.10              | Distributing client machines in a network . . . . .            | 198        |
| <b>Chapter 5.</b> | <b>Managing client desktops . . . . .</b>                      | <b>201</b> |
| 5.1               | Planning your desktop . . . . .                                | 201        |
| 5.2               | General desktop representation . . . . .                       | 202        |
| 5.3               | Windows 2000 Professional desktops . . . . .                   | 203        |
| 5.3.1             | Windows 2000 desktop structure . . . . .                       | 203        |
| 5.3.2             | Assigning a Windows 2000 desktop to a user. . . . .            | 205        |
| 5.3.3             | Defining a new desktop for Windows 2000 Professional . . . . . | 207        |
| 5.4               | Windows NT Workstation 4.0 desktops . . . . .                  | 216        |
| 5.4.1             | Windows NT desktop structure . . . . .                         | 216        |

|                   |  |            |
|-------------------|--|------------|
| 5.4.2             | Assigning a Windows NT desktop to a user . . . . .                       | 218        |
| 5.4.3             | Defining a new desktop for Windows NT 4.0 Workstation . . . . .          | 220        |
| 5.5               | Windows 98 desktops . . . . .  | 227        |
| 5.5.1             | Windows 98 desktop structure . . . . .                                   | 227        |
| 5.5.2             | Assigning Windows 98 desktops to users . . . . .                         | 230        |
| 5.5.3             | Defining a new desktop for Windows 98 . . . . .                          | 231        |
| 5.6               | Defining Windows 2000, NT and 98 desktops in the GUI . . . . .           | 239        |
| 5.7               | Modifying the logon window for Windows 2000, NT and 98 clients . . . . . | 242        |
| 5.8               | OS/2 desktops . . . . .  | 243        |
| 5.8.1             | The OS/2 desktop object . . . . .  | 244        |
| 5.8.2             | Defining an OS/2 desktop in the CLI . . . . .                            | 245        |
| 5.8.3             | Defining an OS/2 desktop in the GUI . . . . .                            | 247        |
| 5.8.4             | The PMLOGON shell . . . . .  | 249        |
| 5.8.5             | Modifying the PMLOGON shell . . . . .                                    | 250        |
| 5.8.6             | Replacing the logon bitmap . . . . .                                     | 252        |
| 5.8.7             | Replacing the background bitmap . . . . .                                | 253        |
| 5.8.8             | Locking up the desktop . . . . .   | 254        |
| 5.8.9             | Logoff and shutdown options . . . . .                                    | 255        |
| 5.8.10            | Customizing program objects . . . . .                                    | 255        |
| 5.8.11            | Replacing the PMLOGON shell . . . . .                                    | 257        |
| <b>Chapter 6.</b> | <b>Applications . . . . .</b>  | <b>259</b> |
| 6.1               | The general application object structure . . . . .                       | 259        |
| 6.1.1             | Windows 2000, NT and 98 application attributes . . . . .                 | 259        |
| 6.1.2             | OS/2 application attributes . . . . .                                    | 260        |
| 6.1.3             | Application object representation . . . . .                              | 261        |
| 6.1.4             | Application sets . . . . .   | 263        |
| 6.2               | Introduction to defining applications . . . . .                          | 264        |
| 6.3               | The CRTPKG tool . . . . .  | 268        |
| 6.3.1             | The configuration file for CRTPKG . . . . .                              | 268        |
| 6.3.2             | Specifying the path of the application package . . . . .                 | 271        |
| 6.4               | The WINSTALL tool . . . . .  | 271        |
| 6.5               | Application packages . . . . .   | 272        |
| 6.5.1             | File structure of an application package . . . . .                       | 272        |
| 6.5.2             | Windows 2000, NT and 98 application packages . . . . .                   | 274        |
| 6.5.3             | OS/2 application packages . . . . .                                      | 275        |
| 6.6               | Installing Windows 2000 applications . . . . .                           | 277        |
| 6.6.1             | Adobe Acrobat Reader as a Server-based Application . . . . .             | 277        |
| 6.6.2             | Adobe Acrobat Reader as a Local Application . . . . .                    | 278        |
| 6.6.3             | Norton AntiVirus 7.0 as a Local Application . . . . .                    | 279        |
| 6.6.4             | Host On-Demand as a Server-based Application . . . . .                   | 281        |
| 6.6.5             | IBM Data Base 2 as a Server-based Application . . . . .                  | 283        |
| 6.6.6             | Microsoft Internet Explorer 5.0 as a Local Application . . . . .         | 284        |



|        |  |     |
|--------|--|-----|
| 6.6.7  | Paint Shop Pro 5 as a Server-Based Application . . . . .         | 286 |
| 6.6.8  | Lotus Notes 5.04 as a Local Application . . . . .                | 287 |
| 6.6.9  | Lotus SmartSuite Millennium as a Server-Based Application . .    | 290 |
| 6.6.10 | Microsoft Office 2000 . . . . .                                  | 292 |
| 6.6.11 | Netscape 4.7 as a Server-based Application . . . . .             | 296 |
| 6.7    | Installing Windows NT applications . . . . .                     | 299 |
| 6.7.1  | Adobe Acrobat Reader as a Server-based Application . . . . .     | 299 |
| 6.7.2  | Adobe Acrobat Reader as a Local Application . . . . .            | 301 |
| 6.7.3  | Netscape 4.7 as a Server-based Application . . . . .             | 302 |
| 6.7.4  | Netscape 4.7 as a Local Application . . . . .                    | 304 |
| 6.7.5  | Microsoft Internet Explorer 5.0 as a Local Application. . . . .  | 306 |
| 6.7.6  | Paint Shop Pro 5 as a Server-Based Application . . . . .         | 308 |
| 6.7.7  | Lotus Notes 5.04 as a Local Application . . . . .                | 310 |
| 6.7.8  | Lotus SmartSuite Millennium as a Server-Based Application . .    | 312 |
| 6.7.9  | Lotus SmartSuite Millennium as a Local Application . . . . .     | 314 |
| 6.7.10 | Microsoft Office 2000 . . . . .                                  | 316 |
| 6.8    | Installing Windows 98 applications . . . . .                     | 318 |
| 6.8.1  | Microsoft Internet Explorer . . . . .                            | 318 |
| 6.8.2  | Adobe Acrobat Reader as a Server-based Application . . . . .     | 318 |
| 6.8.3  | Adobe Acrobat Reader as a Local Application . . . . .            | 320 |
| 6.8.4  | Netscape 4.7 as a Server-based Application . . . . .             | 321 |
| 6.8.5  | Netscape 4.7 as a Local Application . . . . .                    | 323 |
| 6.8.6  | Paint Shop Pro 5 as a Server-Based Application . . . . .         | 325 |
| 6.8.7  | Lotus Notes 5.04 as a Local Application . . . . .                | 327 |
| 6.8.8  | Lotus SmartSuite Millennium as a Server-Based Application . .    | 330 |
| 6.8.9  | Lotus SmartSuite Millennium as a Local Application . . . . .     | 331 |
| 6.8.10 | Microsoft Office 2000 . . . . .                                  | 333 |
| 6.9    | Installing OS/2 applications . . . . .                           | 335 |
| 6.9.1  | The OS/2 sandbox . . . . .                                       | 335 |
| 6.9.2  | The OS/2 application FIT file . . . . .                          | 338 |
| 6.9.3  | Netscape Communicator 4.61. . . . .                              | 341 |
| 6.9.4  | Lotus SmartSuite 1.5 . . . . .                                   | 344 |
| 6.9.5  | Lotus Notes client 4.5 / 4.6 . . . . .                           | 349 |
| 6.10   | Making System Changes to the Win32 Clients . . . . .             | 352 |
| 6.11   | Defining applications in IBM WorkSpace On-Demand 3.0.1 GUI . .   | 354 |
| 6.12   | Problem determination for Win32 Application and Desktop. . . . . | 356 |
| 6.12.1 | Application . . . . .  | 356 |
| 6.12.2 | Desktop . . . . .  | 358 |
| 6.13   | Win32 Client Tools Problem determination . . . . .               | 358 |
| 6.13.1 | CRTPKG . . . . .   | 358 |
| 6.13.2 | Winstall . . . . .   | 360 |
| 6.14   | Moving entire application packages . . . . .                     | 361 |

|  |     |
|--|-----|
| <b>Chapter 7. Updating the client images</b>                     | 363 |
| 7.1 The install scripts  | 363 |
| 7.1.1 Windows NT Workstation and 2000 Professional install files | 364 |
| 7.1.2 Unattend.txt   | 365 |
| 7.1.3 Copying the \$OEM\$ directory and its subdirectories       | 366 |
| 7.1.4 The cmdlines.txt and other files                           | 367 |
| 7.1.5 The \$\$Rename.txt files                                   | 369 |
| 7.1.6 Windows 98 installation flow                               | 370 |
| 7.1.7 The msbatch.inf file                                       | 372 |
| 7.2 Updating Service Packs, fixpacks, and device drivers         | 373 |
| 7.2.1 Windows NT 4.0   | 373 |
| 7.2.2 Windows 2000   | 375 |
| 7.2.3 Windows 98   | 378 |
| 7.2.4 OS/2   | 378 |
| 7.3 Updating drivers   | 383 |
| 7.3.1 Installing support for video adapters                      | 383 |
| 7.3.2 Installing support for network interface cards             | 391 |
| 7.4 Modifying a preboot image                                    | 395 |
| <b>Chapter 8. Migration and product services</b>                 | 399 |
| 8.1 Migration planning   | 399 |
| 8.1.1 Requirements phase   | 399 |
| 8.1.2 Design phase   | 401 |
| 8.2 Testing phase  | 402 |
| 8.2.1 Deployment phase   | 403 |
| 8.3 Migration services   | 404 |
| 8.3.1 Independent project review                                 | 404 |
| 8.3.2 Product services   | 404 |
| 8.3.3 Customized product enhancements                            | 405 |
| 8.3.4 Find out more about migration services                     | 405 |
| <b>Chapter 9. Comparisons</b>                                    | 407 |
| 9.1 Differences between IBM WorkSpace On-Demand 3.0.1 and TMA    | 407 |
| 9.2 The IBM WorkSpace On-Demand 3.0.1 and Tivoli combination     | 408 |
| 9.2.1 Introducing the TMA  | 409 |
| 9.2.2 The endpoint (TMA)   | 409 |
| 9.2.3 The endpoint gateway                                       | 409 |
| 9.2.4 The endpoint manager                                       | 410 |
| 9.2.5 The role of the TMA  | 410 |
| 9.2.6 TMA installation prerequisites                             | 410 |
| 9.2.7 Enabling the TMA   | 411 |
| 9.2.8 Installing the TMA   | 414 |
| 9.2.9 Configuring the clients                                    | 415 |

|  |            |
|--|------------|
| 9.2.10 Stopping and starting the TMA . . . . .                           | 416        |
| <b>Appendix A. Software Announcement 200-211 June 27, 2000 . . . . .</b> | <b>419</b> |
| A.1 IBM WorkSpace On-Demand 3.0 . . . . .                                | 419        |
| A.2 Overview . . . . .   | 420        |
| A.3 Description . . . . .  | 421        |
| A.4 Key Requirements of Network Computing and e-business . . . . .       | 423        |
| A.5 Technical Information . . . . .                                      | 427        |
| <b>Appendix B. WorkSpace On-Demand 3.0 Performance White Paper .</b>     | <b>435</b> |
| B.1 Abstract . . . . .   | 435        |
| B.2 Introduction . . . . .   | 436        |
| B.2.1 Functional Overview . . . . .                                      | 437        |
| B.2.2 Architectural Overview . . . . .                                   | 440        |
| B.2.3 Server requirements . . . . .                                      | 443        |
| B.2.4 Client requirements . . . . .                                      | 443        |
| B.2.5 Scope of this paper . . . . .                                      | 444        |
| B.3 Boot Process Overview . . . . .                                      | 444        |
| B.3.1 Initial Boot (for all clients) . . . . .                           | 445        |
| B.3.2 Remote Boot/Install (Windows NT 4.0 and 98 SE clients only) . .    | 446        |
| B.4 Performance data & analysis . . . . .                                | 449        |
| B.4.1 Measurement environment . . . . .                                  | 450        |
| B.4.2 Boot performance for Windows NT 4.0 clients . . . . .              | 452        |
| B.4.3 Boot performance for Windows 98 clients . . . . .                  | 457        |
| B.4.4 Boot performance for OS/2 Warp clients . . . . .                   | 461        |
| B.4.5 Log-on performance . . . . .                                       | 465        |
| B.4.6 Server memory usage . . . . .                                      | 467        |
| B.4.7 Factors affecting application performance . . . . .                | 469        |
| B.5 Performance recommendations . . . . .                                | 470        |
| B.6 Additional information sources . . . . .                             | 472        |
| B.7 Acknowledgments . . . . .  | 472        |
| B.8 Disclaimer . . . . .   | 473        |
| B.9 Trademarks & Copyrights . . . . .                                    | 473        |
| <b>Appendix C. Special notices . . . . .</b>                             | <b>475</b> |
| <b>Appendix D. Related publications . . . . .</b>                        | <b>479</b> |
| D.1 IBM Redbooks . . . . .   | 479        |
| D.2 IBM Redbooks collections . . . . .                                   | 479        |
| D.3 Referenced Web sites . . . . .                                       | 479        |
| <b>How to get IBM Redbooks . . . . .</b>                                 | <b>481</b> |
| IBM Redbooks fax order form . . . . .                                    | 482        |

|   |     |
|---|-----|
| <b>Abbreviations and acronyms</b> . . . . . | 483 |
| <b>Index</b> . . . . .                      | 485 |
| <b>IBM Redbooks review</b> . . . . .        | 497 |

---

## Figures

|  |       |
|--|-------|
| 1. Redbook development team . . . . .                              | xx    |
| 2. Workshop development team . . . . .                             | xxii  |
| 3. The Completion Team . . . . .                                   | xxiii |
| 4. Architecture of IBM WorkSpace On-Demand 3.0.1.1 . . . . .       | 5     |
| 5. Roadmap to installing IBM WorkSpace On-Demand 3.0.1 . . . . .   | 15    |
| 6. PXE remote boot sequence . . . . .                              | 17    |
| 7. Software Licence Agreement . . . . .                            | 26    |
| 8. IBM WorkSpace On-Demand 3.0.1 . . . . .                         | 27    |
| 9. Choose Destination Location window . . . . .                    | 28    |
| 10. IBM WorkSpace On-Demand 3.0.1 components . . . . .             | 29    |
| 11. Select Program Folder . . . . .                                | 30    |
| 12. User ID and password prompt . . . . .                          | 31    |
| 13. Start Copying Files . . . . .                                  | 32    |
| 14. DHCP manual setup is needed . . . . .                          | 33    |
| 15. Configuration Server not started . . . . .                     | 33    |
| 16. Check the Services status . . . . .                            | 34    |
| 17. Select the operating system to be enabled . . . . .            | 35    |
| 18. Request to select the directory for TDMCLNT.INF file . . . . . | 36    |
| 19. Select the directory for TDMCLNT.INF file . . . . .            | 36    |
| 20. Image name . . . . .   | 37    |
| 21. Machine Management Directory . . . . .                         | 38    |
| 22. Client Operating System . . . . .                              | 39    |
| 23. Configuration Server not started . . . . .                     | 39    |
| 24. IBM WorkSpace On-Demand 3.0.1 directory structure . . . . .    | 41    |
| 25. DHCP Server Configuration . . . . .                            | 44    |
| 26. Subnet configuration . . . . .                                 | 45    |
| 27. DHCP options . . . . .   | 47    |
| 28. TFTP Configuration . . . . .                                   | 50    |
| 29. IBM WorkSpace On-Demand 3.0.1 services . . . . .               | 53    |
| 30. Add/Remove Programs . . . . .                                  | 54    |
| 31. Components of the administration console . . . . .             | 58    |
| 32. IBM WorkSpace On-Demand 3.0.1 administration GUI . . . . .     | 60    |
| 33. Logging on to a server from the administration GUI . . . . .   | 61    |
| 34. The administration CLI . . . . .                               | 62    |
| 35. Logging on to a server from the CLI . . . . .                  | 64    |
| 36. Sample commands and returned information . . . . .             | 65    |
| 37. A complex command - Defining a machine . . . . .               | 67    |
| 38. Encapsulation of a command . . . . .                           | 68    |
| 39. Handling a TDMConsoleResult object . . . . .                   | 69    |
| 40. Executing a JavaScript function . . . . .                      | 70    |

|   |     |
|---|-----|
| 41. Creating multiple machines in one function . . . . .                    | 71  |
| 42. Generic machine deletion . . . . .                                      | 72  |
| 43. TDMConsoleResult values for the machine list command . . . . .          | 73  |
| 44. Launching the CLI from a batch file . . . . .                           | 74  |
| 45. Dynamically creating and running JavaScript . . . . .                   | 76  |
| 46. Users and groups added by IBM WorkSpace On-Demand 3.0.1 . . . . .       | 77  |
| 47. Defining a user. . . . .  | 79  |
| 48. User logon assignments and home directories . . . . .                   | 80  |
| 49. Setting logon assignments. . . . .                                      | 81  |
| 50. Importing users from Windows NT. . . . .                                | 82  |
| 51. User information in the data store . . . . .                            | 85  |
| 52. JavaScript to delete a user . . . . .                                   | 90  |
| 53. JavaScript to delete all users. . . . .                                 | 91  |
| 54. Monitoring a deployment server. . . . .                                 | 98  |
| 55. Machine directory hierarchy. . . . .                                    | 104 |
| 56. Defining a Windows 2000 machine in the CLI . . . . .                    | 111 |
| 57. A Windows W2K machine definition in the data store . . . . .            | 113 |
| 58. tdmclnt.inf and tdmDOS.inf for a Windows 2000 client . . . . .          | 114 |
| 59. Contents of a Windows 2000 machine directory after creation . . . . .   | 116 |
| 60. The Windows 2000 header of state.ini. . . . .                           | 117 |
| 61. Sample partition command section of state.ini . . . . .                 | 118 |
| 62. The format section of state.ini for Windows 2000 . . . . .              | 119 |
| 63. The copy section of Windows 2000 state.ini . . . . .                    | 120 |
| 64. The install section of Windows 2000 state.ini . . . . .                 | 122 |
| 65. Contents of the WRKFILES alias for a Windows 2000 client. . . . .       | 123 |
| 66. Defining a Windows NT machine in the CLI . . . . .                      | 130 |
| 67. A Windows NT machine definition in the data store . . . . .             | 132 |
| 68. tdmclnt.inf and tdmDOS.inf for a Windows NT client. . . . .             | 133 |
| 69. Contents of a Windows NT machine directory after creation. . . . .      | 135 |
| 70. The Windows NT header of state.ini . . . . .                            | 136 |
| 71. Sample partition command section of state.ini . . . . .                 | 137 |
| 72. The format section of state.ini for Windows NT and Windows 98 . . . . . | 138 |
| 73. The copy section of Windows NT state.ini. . . . .                       | 139 |
| 74. The install section of Windows NT state.ini . . . . .                   | 141 |
| 75. Contents of the WRKFILES alias for a Windows NT client . . . . .        | 142 |
| 76. Defining a Windows 98 machine in the CLI . . . . .                      | 149 |
| 77. Data store representation of a Windows 98 client . . . . .              | 151 |
| 78. tdmclnt.inf and tdmDOS.inf for a Windows 98 client . . . . .            | 152 |
| 79. Contents of a Windows 98 machine directory after creation . . . . .     | 154 |
| 80. The Windows 98 header of state.ini. . . . .                             | 155 |
| 81. The copy section of state.ini for Windows 98. . . . .                   | 157 |
| 82. Windows 98 installation from state.ini . . . . .                        | 159 |
| 83. Contents of the WRKFILES alias for a Windows 98 client. . . . .         | 160 |

|  |     |
|--|-----|
| 84. The machine define command for an OS/2 client . . . . .                    | 168 |
| 85. Data store entry for an OS/2 client machine . . . . .                      | 169 |
| 86. tdmclnt.inf and tdmoss2.inf for an OS/2 client . . . . .                   | 170 |
| 87. Selections from the common descriptor file . . . . .                       | 171 |
| 88. Default machine read-only and machine read-write OS/2 client images .      | 172 |
| 89. Sample machine FIT entries . . . . .                                       | 174 |
| 90. The FIT default alias . . . . .  | 174 |
| 91. Additional machine FIT entries . . . . .                                   | 175 |
| 92. Java and Netscape support in the machine FIT file . . . . .                | 176 |
| 93. Sample entries in the default FIT file . . . . .                           | 177 |
| 94. Creating a machine template . . . . .                                      | 178 |
| 95. Data store representation of a machine template . . . . .                  | 178 |
| 96. Machine actions in the administration GUI . . . . .                        | 182 |
| 97. Operating system selection in the administration GUI . . . . .             | 183 |
| 98. Windows 2000 machine definition - Identity tab . . . . .                   | 184 |
| 99. Windows 2000 machine definition - OS Image tab . . . . .                   | 185 |
| 100. Windows 2000 machine definition - Hardware tab . . . . .                  | 186 |
| 101. Windows NT machine definition - Hardware tab . . . . .                    | 187 |
| 102. Windows 2000 machine definition - Preboot tab . . . . .                   | 188 |
| 103. Windows 2000 machine definition - TCP/IP tab . . . . .                    | 189 |
| 104. The machines resource list . . . . .                                      | 190 |
| 105. State.ini modified for large partition support . . . . .                  | 193 |
| 106. State.ini modifications for multiple partitions . . . . .                 | 194 |
| 107. Formatting a second partition . . . . .                                   | 196 |
| 108. The MCOPY command and the misc.lst file . . . . .                         | 197 |
| 109. cmdlines.txt file for a Windows 2000 and a Windows NT client . . . . .    | 197 |
| 110. Transferring a machine definition between deployment servers . . . . .    | 200 |
| 111. The default desktop directories for Windows 2000 clients . . . . .        | 204 |
| 112. The default Windows 2000 desktop in the data store . . . . .              | 204 |
| 113. Data store representation of a user with a desktop . . . . .              | 207 |
| 114. Modify the background wallpaper properties . . . . .                      | 209 |
| 115. Modify the color scheme . . . . .   | 210 |
| 116. Modify the screen saver properties . . . . .                              | 211 |
| 117. The modified Windows 2000 profile information . . . . .                   | 212 |
| 118. POLEDIT: Open the policy file of the new desktop . . . . .                | 213 |
| 119. POLEDIT: Modify the policy of the new desktop to fit your needs . . . . . | 214 |
| 120. Data store representation of a custom desktop for Windows 2000 . . . . .  | 215 |
| 121. The newly-defined desktop EyeDesk . . . . .                               | 216 |
| 122. The default desktop directories for Windows NT clients . . . . .          | 217 |
| 123. The default Windows NT desktop in the data store . . . . .                | 217 |
| 124. Data store representation of a user with a desktop . . . . .              | 220 |
| 125. Modify the background wallpaper properties . . . . .                      | 222 |
| 126. Modify the color scheme . . . . .   | 222 |

|   |     |
|---|-----|
| 127.Modify the screen saver properties . . . . .                                  | 223 |
| 128.The modified Windows NT profile information. . . . .                          | 224 |
| 129.POLEEDIT: Open the policy file of the new desktop . . . . .                   | 225 |
| 130.POLEEDIT: Modify the policy of the new desktop to fit your needs . . . . .    | 225 |
| 131.Data store representation of a custom desktop for Windows NT . . . . .        | 226 |
| 132.The newly-defined desktop EyeDesk . . . . .                                   | 227 |
| 133.The default desktop directories for Windows 98 clients . . . . .              | 228 |
| 134.The default Windows 98 desktop in the data store . . . . .                    | 228 |
| 135.Data store representation of a user with a desktop. . . . .                   | 231 |
| 136.Windows 98: Modify the background wallpaper properties . . . . .              | 233 |
| 137.Windows 98: Modify the color scheme . . . . .                                 | 234 |
| 138.Windows 98: Modify the screen saver properties . . . . .                      | 234 |
| 139.The modified Windows 98 profile information . . . . .                         | 235 |
| 140.Windows 98 Policy Editor: Open the policy file of the new desktop . . . . .   | 237 |
| 141.Windows 98: Modify the policy of the new desktop to your needs . . . . .      | 237 |
| 142.Data store representation of a custom desktop for Windows 98. . . . .         | 238 |
| 143.The newly-defined Windows 98 desktop SnowDesk. . . . .                        | 239 |
| 144.Defining a Windows desktop - Identity tab . . . . .                           | 240 |
| 145.Defining a Windows desktop - Properties tab . . . . .                         | 241 |
| 146.Assigning a Windows desktop to a user . . . . .                               | 242 |
| 147.Windows 2000: Sample registry file to change logon bitmap and color . . . . . | 243 |
| 148.Windows NT: Sample registry file to change logon bitmap and color. . . . .    | 243 |
| 149.Windows 98: Sample registry file to change logon bitmap and color . . . . .   | 243 |
| 150.Data store representation of the default OS/2 desktop . . . . .               | 244 |
| 151.A custom OS/2 desktop in the data store . . . . .                             | 246 |
| 152.Defining an OS/2 desktop in the administration GUI. . . . .                   | 247 |
| 153.Icon tab for the OS/2 desktop definition . . . . .                            | 248 |
| 154.Assigning an OS/2 desktop to a user . . . . .                                 | 249 |
| 155.Sample of an OS/2 setup string with desktop settings . . . . .                | 256 |
| 156.Modifying the SET RUNWORKPLACE statement . . . . .                            | 257 |
| 157.An application package directory . . . . .                                    | 262 |
| 158.A sample application representation in the data store . . . . .               | 262 |
| 159.How applications are made available to users . . . . .                        | 264 |
| 160.How to make applications available to an end user . . . . .                   | 267 |
| 161.File structure of application packages . . . . .                              | 273 |
| 162.Sample of a MODFILE.TXT (Adobe Acrobat Reader) . . . . .                      | 274 |
| 163.Sample of a NEWFILE.TXT (Adobe Acrobat Reader) . . . . .                      | 275 |
| 164.Sample of a WINSTALL.CSF (Adobe Acrobat Reader) . . . . .                     | 275 |
| 165.Changes for Notes in the PRTDSK2K.INI file . . . . .                          | 290 |
| 166.Changes for Microsoft Office 2000 in the PRTDSK2K.INI file. . . . .           | 295 |
| 167.Changes for Netscape in the PRTDSK2K.INI file . . . . .                       | 298 |
| 168.Changes for Netscape in the PRTDSKNT.INI file . . . . .                       | 304 |
| 169.Changes for Notes in the PRTDSKNT.INI file . . . . .                          | 312 |



|   |     |
|---|-----|
| 170.Changes for Notes in the PRTDSKNT.INI file . . . . .                            | 329 |
| 171.Defining an OS/2 sandbox machine . . . . .                                      | 336 |
| 172.The OS/2 sandbox duplicate operating system tree . . . . .                      | 336 |
| 173.The operating system section of the OS/2 sandbox FIT . . . . .                  | 337 |
| 174.Selections from newfiles.lst for a Lotus Notes installation . . . . .           | 340 |
| 175.The apppost.fit file for Netscape Communicator . . . . .                        | 344 |
| 176.Application packages for the SmartSuite components . . . . .                    | 346 |
| 177.The apppost.fit file for Lotus SmartSuite . . . . .                             | 348 |
| 178.The apppost.fit file for Lotus Notes . . . . .                                  | 352 |
| 179.Application Identity tab for Netscape . . . . .                                 | 354 |
| 180.Application Properties tab for Netscape . . . . .                               | 355 |
| 181.Assigning applications to a user in the GUI. . . . .                            | 356 |
| 182.Modified WINSTALL.CSF on the new server . . . . .                               | 362 |
| 183.Windows install process . . . . .   | 364 |
| 184.Contents of preboot image TDMW32UT.IMG . . . . .                                | 396 |
| 185.IBM WorkSpace On-Demand 3.0.1 and TMA . . . . .                                 | 407 |
| 186.Steps to enable the TMA. . . . .  | 411 |
| 187.The endpoint login process . . . . .  | 413 |
| 188.Architectural overview of IBM WorkSpace On-Demand 3.0 . . . . .                 | 442 |
| 189.Installation boot performance for NT 4.0 on Token Ring. . . . .                 | 453 |
| 190.Installation Boot Performance for NT 4.0 - 100 - Mbps Ethernet . . . . .        | 454 |
| 191.Installation Boot Performance for NT 4.0 Clients - 10-Mbps Ethernet . . . . .   | 455 |
| 192.Boot performance for NT 4.0 clients on token ring . . . . .                     | 456 |
| 193.Installation boot performance for Win98 clients on token ring. . . . .          | 457 |
| 194.Installation boot performance for Win98 clients - 100-Mbps Ethernet. . . . .    | 458 |
| 195.Installation boot performance for Win98 clients - 10-Mbps Ethernet. . . . .     | 460 |
| 196.Boot performance for Win98 clients on token ring . . . . .                      | 461 |
| 197.Remote boot performance for OS/2 clients on token ring . . . . .                | 462 |
| 198.Remote boot performance for OS/2 clients - 100-Mbps Ethernet . . . . .          | 463 |
| 199.Remote boot performance for OS/2 clients - 10-Mbps Ethernet . . . . .           | 464 |
| 200.Remote boot performance for OS/2 clients on token ring . . . . .                | 464 |
| 201.Total peak memory working set for setting up clients . . . . .                  | 467 |
| 202.Total peak memory working set for client boot on NT 4.0, OS/2 clients . . . . . | 469 |



---

## Tables

|   |     |
|---|-----|
| 1. Types of command parameters . . . . .                        | 64  |
| 2. Desktop lockup - PMLOGON parameters . . . . .                | 254 |
| 3. OS/2 application setup strings . . . . .                     | 257 |
| 4. Location of CRTPKG.INI file on the server (origin) . . . . . | 268 |
| 5. Sections in the configuration file CRTPKG.INI. . . . .       | 270 |
| 6. Variables in the application FIT file . . . . .              | 339 |
| 7. OSs and respective image names . . . . .                     | 363 |
| 8. Parameters of the winnt command . . . . .                    | 364 |
| 9. Cmdlines.txt parameters . . . . .                            | 370 |
| 10. Windows 98 setup switches. . . . .                          | 370 |
| 11. Install switches for Windows NT Service Packs . . . . .     | 374 |
| 12. Location of the SETUP.ISS file . . . . .                    | 415 |
| 13. IBM WorkSpace On-Demand 3.0 log-on performance . . . . .    | 466 |



---

## Preface

This redbook will help you install, configure, and administer an IBM WorkSpace On-Demand 3.0.1 environment. It is the result of residencies conducted at the International Technical support Organization, Austin Center, during the final development of IBM WorkSpace On-Demand 3.0.1. This redbook will help you to:

- Understand the IBM WorkSpace On-Demand 3.0.1 product, and the situations in which it is most appropriate to deploy the product.
- Understand the remote boot concepts behind IBM WorkSpace On-Demand 3.0.1, and their implementations under IBM WorkSpace On-Demand 3.0.1.
- Plan and install IBM WorkSpace On-Demand in your enterprise.
- Define client workstations on your IBM WorkSpace On-Demand 3.0.1 servers, and install/boot these client workstations remotely over the network using IBM WorkSpace On-Demand 3.0.1.
- Install network applications in your IBM WorkSpace On-Demand 3.0.1 environment, and make them available to client workstations and end users over the network.
- Add support for additional network adapters and other types of hardware (video adapters), to expand the hardware support provided by IBM WorkSpace On-Demand 3.0.1.
- Understand the planning steps necessary to deploy IBM WorkSpace On-Demand 3.0.1 in your enterprise.
- Understand the performance issues when installing in your enterprise.

The residents working with IBM WorkSpace On-Demand 3.0.1 have taken their experiences and knowledge gained during these residencies and captured it in this redbook. By using this redbook, you can capitalize on their experiences with IBM WorkSpace On-Demand 3.0.1 in your environment

---

### The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.



Figure 1. Redbook development team

In Figure 1, the team members are, left to right, Feite Kraay, Ron Aguirre, Michael Ambjorn, Rowell Hernadez, and Berhard Loehrer.

**Ron Aguirre** is a Advisory Software Engineer at the International Technical Support Organization, Austin Center. He has 23 years of experience ranging from mainframes to workstations, hardware and software. He has experience in large account management, service research, and development. He writes extensively and teaches IBM classes worldwide on all areas of server-managed clients.

**Michael Ambjorn** is a Technical Account Manager in the IBM Software Business. As the Technical Advocate for the biggest OS/2 OEM customer worldwide, Michael has responsibility for resolving all software-related issues affecting the account. Prior to joining the IBM Software Business, he worked in IBM Global Services as Team Leader for Lotus Technical Support. He holds a number of industry certifications, including Novell Certified Internet Professional.

**Rowell Hernandez** is a System Services Representative with the Global Services Division in IBM Philippines. He provides technical support for Netfinity, Windows NT, clustering, and Linux. Rowell is also an IBM Certified Professional Server Expert, Microsoft Certified Systems Engineer + Internet, and Comptia A+ Certified Technician. He holds a Bachelor of Science degree in Computer Science.

**Feite Kraay** is a Senior IT Specialist in IBM Canada. He holds a Bachelor of Mathematics degree from the University of Waterloo. Feite has 11 years of experience in a wide variety of areas related to OS/2 and Network Computing, including application design and development, application migration, and support of large-scale OS/2 and LAN server-based RIPL environments. He has taught IBM classes on OS/2 programming and has written an OS/2 programming manual.

**Bernhard Loehrer** is an IT Specialist in IBM Germany. He holds a degree in electrical engineering from the technical university of Kempten. Bernhard has nine years of experience in a wide variety of areas related to OS/2, Network Management, Project Management, Second Level support, as well as design and distribution of a customer-specific WorkSpace On-Demand implementation.



*Figure 2. Workshop development team*

In Figure 2, the team members are, left to right, Ron Aguirre, Vusi Nhlapo, Michael Ambjorn, and Brett Pearce

WorkShop Development Team:

**IBM Scotland**

Michael Ambjorn

**IBM South Africa**

Vusi Nhlapo

**IBM Australia**

Brett Pearce



---

## The team that completed this redbook



*Figure 3. The Completion Team*

In Figure 3, the team members are, left to right, Martin Dippold, Ron Aguirre, Cristina Polignone, Oliver Mark, Wynand Pretorius, Oliver Stein.

**Ron Aguirre** is a Advisory Software Engineer at the International Technical Support Organization, Austin Center. He has 23 years of experience ranging from mainframes to workstations, hardware and software. He has experience in large account management, service research, and development. He writes extensively and teaches IBM classes worldwide on all areas of server-managed clients.

**Cristina Polignone** is an I/T Security Manager in IBM Global Services Italy. She has 17 years of experience ranging from software technical support to IT support management for IBM Microelectronics Division. She holds a degree in Electronic Engineering from the University of Padua. She is OS/2 certified Engineer, IBM Lan Server certified Engineer and certified Lotus Specialist.

**Martin Dippold** is a Systems Engineer with IBM Heidelberg, Germany. He has ten years of experience in networking with a wide variety of computing devices and operating systems. His areas of specialization include Workspace On-Demand, OS/2, LAN Server and OS/2 Warp Server, Windows

NT and 2000, Netfinity, REXX and software distribution using NetViewDM/2 and CID.

**Oliver Mark** is a Advisory I/T Specialist for Client/Server at IBM Global Services Germany. He has more than ten years of experience in OS/2 and OS/2 Warp environments as well as experiences with OS/2 Warp Server, OS/2 Warp Server for e-business, Workspace On-Demand, CID Installations, NetFinity and Remote Access Services. He holds certifications as IBM Certified Expert for OS/2 Warp 2.x, 3.x and 4.x, OS/2 Warp Server and OS/2 Warp Server for e-business as well as IBM Certified Instructor for OS/2 Warp and OS/2 Warp Server. He is a Microsoft Certified System Expert, and IBM Advanced Server Expert for Netfinity Hardware. He has written extensively on updates of Application Installations, Service Packs and Boot Process.

**Oliver Stein** is an Advisory I/T Specialist with IBM Global Services in Germany. He has more than ten years of experience in software engineering, specializing on OS/2 Warp Server, WorkSpace On-Demand and device driver technologies. He holds a degree in Computer Science as well as various professional certifications in the OS/2 area. He has written extensively on server-managed clients and is the Technical Lead for this topic in IBM Germany, teaching classes and creating the architecture for customer solutions.

**Wynand Pretorius** is an IBM Customer. He currently contracts to Standard Bank of South Africa, where he plays a key role in the Support and Development Teams of the Banks OS/2 platform. He holds certifications as IBM Certified Expert for OS/2 Warp 3, IBM LAN Server Administration, and Netfinity Hardware.

Thanks to the following people for their invaluable contributions to this project:

**IBM Austin**

Indran Naick, Oscar Cepeda, Tsutomu Oya, Scott Johnson, John Owczarzak, Milos Radosavljevic, Wade Wallace

**IBM e-business Operating System Solutions, Austin, TX**

CJ Paul, PhD, Randy George, Marc-Arthur Pierre-Louis, Aidon Jennery, Glenn Brew, Jesus Ibanez, Steven French, Don Kutach, Kerry Fehlis, Ernest Covey, Rosie Rocke, Bill Purtill, Brian Howe, David Dutcher, Mei-Hsiang Chang, Betty Truong, Jim Brisson, Chuck Linnea, Peg Clark, David Kerr, Charles Shriver, Jacob Zachariah, Khoa Huynh, Boyd Jasperson, Telford Knox, Anna-Marie Phelps, Carolyn Easter, Dick Reardon, Mark Darholt

**IBM e-business Operating System Solutions Rapid Deployment Team**  
Jim Williams, Bob Spratt

**IBM Software Group, Canada**  
Shelley Renaud

---

## **Comments welcome**

### **Your comments are important to us!**

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks review” on page 497 to the fax number shown on the form.
- Use the online evaluation form found at [ibm.com/redbooks](http://ibm.com/redbooks)
- Send your comments in an Internet note to [redbook@us.ibm.com](mailto:redbook@us.ibm.com)



---

## Chapter 1. Introduction

IBM WorkSpace On-Demand comes in two different versions: IBM WorkSpace On-Demand 3.0.1, that only works on Windows NT, and IBM WorkSpace On-Demand 3.0.1, that works both on Windows 2000 and Windows NT.

In this book, we always refer to IBM WorkSpace On-Demand 3.0.1, but most of its content can also be applied to IBM WorkSpace On-Demand 3.0

IBM WorkSpace On-Demand 3.0.1 is the IBM cross-platform network operating system designed for a network computing environment. It is a simple and economical alternative to traditional client/server environments that reduces the total cost of ownership by eliminating the need to directly manage client machines.

IBM WorkSpace On-Demand 3.0.1 replaces WorkSpace On-Demand 2.0 and extends its proven benefits of server-managed clients to multiple server and client operating systems.

This chapter provides an overview of IBM WorkSpace On-Demand 3.0.1. It provides functional descriptions of the product components and also introduces the different client operating systems that are supported.

---

### 1.1 Benefits of IBM WorkSpace On-Demand 3.0.11

IBM WorkSpace On-Demand 3.0.1 implements a Server-Managed Client environment using Intel-based PCs and network stations over a TCP/IP network. It extends the proven benefits of WorkSpace On-Demand into heterogeneous environments where multiple server types must coexist with multiple client operating systems. These benefits include:

- Effective software and data management
- Easier end-user support
- Enhanced security
- Broad application support
- End-user mobility
- Support for multiple server operating systems
- Support for multiple client operating systems

The following sections describe each of these benefits in more detail.

### **1.1.1 Effective software and data management**

IBM WorkSpace On-Demand 3.0.1 stores all client operating system files on the server. For some clients, such as IBM OS/2, the operating system is downloaded to the client machine at boot time. For others, such as Windows 98, Windows 2000 or Windows NT, the operating system files are installed to the client machine upon its first boot. The installation is an automated process managed from the server and requires no end-user intervention. Application software is also typically stored on a server in the network and not locally on the client machine.

This allows the servers to be the logical endpoints for managing the network. All updates, fixes, or changes to client operating systems and applications need only be downloaded to the servers. The client machines can then be automatically updated or reinstalled as necessary, in a process that is transparent to the end user.

IBM WorkSpace On-Demand 3.0.1 allows administrators to restrict access to the client machine's local hard drive, forcing users to store all data on the server. This allows the data to be more effectively managed and protected. It also allows client operating systems or hardware to be serviced or replaced without risk of losing critical data.

### **1.1.2 Easier end-user support**

A large part of the cost of traditional client/server environments is the support of end-user systems. IBM WorkSpace On-Demand 3.0.1 protects the client operating system, applications, and data from accidental or malicious corruption, reducing the incidence and impact of software-related problems on the client system. By centralizing a pristine copy of the operating system files on the server and restricting access to the client hard drive, IBM WorkSpace On-Demand 3.0.1 reduces the cost and complexity of diagnosing and rectifying client-side problems.

An administrator can design and deploy simple, standardized desktop interfaces for different types of users. The end user only gains access to those applications for which he or she is authorized and is not permitted to change the interface or install unauthorized software on the system. This reduces the overall cost of end-user support by simplifying training and reducing support to a limited, known quantity of applications and interfaces.

### **1.1.3 Enhanced security**

IBM WorkSpace On-Demand 3.0.1 facilitates and enhances both system and data security. When a client machine is started, it immediately displays a

logon window. Access to the client machine and to the network is restricted to users with a valid user ID and password to the network domain. Access is limited to each user's authorized set of applications.

By restricting access to the client desktop and local storage devices, including diskette drive or CD-ROM, IBM WorkSpace On-Demand 3.0.1 restricts the user's ability to introduce malicious applications to the client system or to the network. If a client machine does become corrupted, it can easily be replaced or reinstalled with no adverse effects to the end user, since applications and data reside on the server.

#### **1.1.4 Broad application support**

IBM WorkSpace On-Demand 3.0.1 provides built-in support for OS/2, Windows 2000, Windows NT, and Windows 98 clients. This enables support for the broadest range of applications available today including DOS, OS/2, 16-bit Windows, and 32-bit Windows applications. Java applications are supported through a native Java Virtual Machine on each client platform.

#### **1.1.5 End-user mobility**

IBM WorkSpace On-Demand 3.0.1 allows the administrator to define a customized desktop environment, including a set of specific authorized applications for each user in a domain. When a user logs on to a client machine, that user's desktop and applications are supplied from the server. The user could, therefore, log on to any machine in the domain and see the same desktop and applications.

This capability, known as *application roaming*, is very useful in environments where users do not always work at the same assigned workstation. For example, customer service representatives in retail banking or airline check-in may use different workstations at different times depending on availability.

#### **1.1.6 Support for multiple server operating systems**

IBM WorkSpace On-Demand 3.0.1 was initially released with support for Windows NT Server. Support is now available for Windows 2000 Advanced Edition Server, with other server operating systems actively considered. Many customers have production environments that use multiple types of server operating systems. IBM WorkSpace On-Demand 3.0.1 allows management of clients in such heterogeneous environments and will allow client management to be scaled up beyond the departmental level to regional or enterprise levels as well.

### 1.1.7 Support for multiple client operating systems

In many cases, customers need to be able to support more than one kind of client operating system due to application needs, user preferences, or hardware compatibility issues. With support for IBM OS/2, Windows 2000 Professional Edition, Windows NT Workstation 4.0, and Windows 98 Second Edition clients provided, IBM WorkSpace On-Demand 3.0.1 can control a wide variety of client operating systems from the same server.

---

## 1.2 Architecture of IBM WorkSpace On-Demand 3.0.1

IBM WorkSpace On-Demand 3.0.1 consists of three logical components: the *Administration Console*, the *Configuration Server*, and the *Deployment Server*. These components may all be installed on separate machines, or any combination of components may reside on a single machine in the network. The Configuration Server, however, must reside on the Primary Domain Controller of the network. Figure 4 on page 5 shows the relationship between the three logical components.



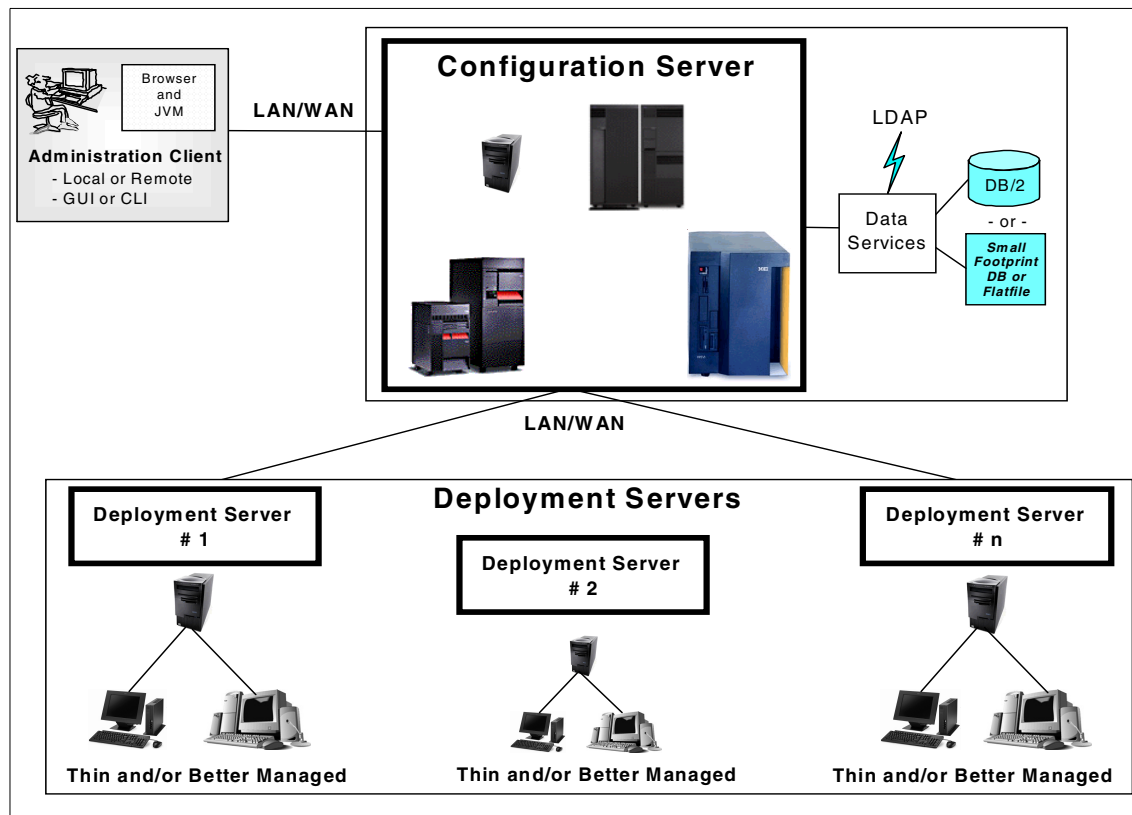


Figure 4. Architecture of IBM WorkSpace On-Demand 3.0.1.1

### 1.2.1 The Administration Console

The Administration Console is a Java application that allows the administrator to manage machines, users, and applications in the network. The console has two components: a graphical user interface (GUI) and a command line interface (CLI). The console can be installed on any machine in the network, although it is typically installed on the Configuration Server.

The Administration GUI is based on the IBM Common Systems Administration (CSA) model and provides task-based and resource-based views of the system. The GUI runs as a stand-alone Java application, which will be enabled to run in a browser in a later release of IBM WorkSpace On-Demand 3.0.1. Through the GUI, the administrator logs on to a specific Configuration Server in order to execute tasks or manage resources.

The Administration CLI is also implemented in Java. It allows the administrator to perform exactly the same functions as in the administration GUI, but through commands entered at a prompt. Using JavaScript, the administrator can automate repetitive tasks, such as multiple machine definitions, and have these tasks executed in an unattended fashion.

The use of the administration console with descriptions of resources and tasks is discussed in detail in Chapter 3, “Administration” on page 57.

### **1.2.2 The Configuration Server**

The Configuration Server stores all machine, user, and application configuration information entered by the administrator via the GUI or CLI console. The information is stored in a data store on the same machine where the Configuration Server resides. This data store is currently implemented as a flat file, but could be implemented in other formats in later versions of IBM WorkSpace On-Demand 3.0.1. Clients never access the data store information directly; rather, the information is sent to the deployment server via a Remote Method Invocation (RMI).

The Configuration Server is written in Java as a cross-platform application, allowing it to be easily ported to other server platforms beyond Windows NT Server 4.0 (for which it has been initially developed). It is a pluggable framework that allows new clients, desktops, or other application-specific configurations to be added over time.

One Configuration Server could manage client information for multiple Deployment Servers. This allows the Configuration Server to be scaled up to a much larger server and to be centrally located. Critical information, such as user and application profiles, would therefore not be spread across a large enterprise and could be better managed and protected.

### **1.2.3 The Deployment Server**

The Deployment Server is responsible for managing the client machines in the network. Windows clients will have their operating system automatically installed from the Deployment Server, while IBM OS/2 Clients will boot their operating system directly from the Deployment Server. The Deployment Server also typically manages applications for the client machines. The client executes applications from shared drives on the Deployment Server, realizing the benefits of local execution combined with server-based management.

Java tasks on the Deployment Server transform the information received from the Configuration Server into client-specific formats, such as Windows registry entries or OS2.INI changes. Just like the Configuration Server, the

Deployment Server is designed in a modular, pluggable format so that new clients or applications can be added as needed. Because the clients do not access the Configuration Server's data store directly, clients can be moved from one Deployment Server to another without having to change the configuration.

#### **1.2.4 Design considerations**

The division of responsibility between the three components provides for flexibility and scalability in the implementation of IBM WorkSpace On-Demand 3.0.1. The Administration Console, for example, can reside on any machine in the network and connect to the Configuration Server via TCP/IP. This gives the administrator the flexibility to manage any server from any physical location. The Configuration Server, which holds critical data (such as user and machine profiles) could reside in a central, secure location.

---

### **1.3 Client support**

The following sections describe the exact versions of the supported client operating systems.

IBM WorkSpace On-Demand 3.0.1 supports the following client operating systems:

- Windows 2000 Professional Edition
- Windows NT Workstation 4.0
- Windows 98 Second Edition (SE)
- IBM OS/2 Client

IBM WorkSpace On-Demand 3.0.1 is bundled with the IBM OS/2 Client. The Microsoft client operating systems need to be licensed and obtained separately.

The IBM OS/2 Client can be run in a true thin-client environment, that is, no local hard disk is required, whereas Windows 2000 Professional Edition, Windows NT Workstation 4.0 and Windows 98 Second Edition require a hard disk to store client OS-specific files.

It should be noted that while a subset of DOS is supplied and used for installation of the Win32 clients, it is not provided as a target operating system for clients. The subset of DOS provided is the same as for the Client Feature for Windows with the same NLV model.

The boot mechanism used for booting the operating systems of the various clients is Preboot Execution Environment (PXE) only.

### **1.3.1 Windows 2000 Professional Edition**

The version of Windows 2000 client currently supported is the Professional Edition. You need to have an original CD to install it on your IBM WorkSpace On-Demand 3.0.1 Deployment Server. No specific Service Pack is required.

### **1.3.2 Windows NT Workstation 4.0**

The NT 4.0 Workstation client is rolled out and updated from the IBM WorkSpace On-Demand 3.0.1 Deployment Server. Service Pack 4 or later should also be installed on the boot. How to update service packs, as well as adding extra display, network, or printer drivers is covered in Chapter 7, “Updating the client images” on page 363.

### **1.3.3 Windows 98 Second Edition**

There are multiple versions of Windows 98 currently available. The version of Windows 98 supported in this environment is Windows 98 Second Edition. Other versions of Windows 98 are not supported. You need to have an original CD to install it on your IBM WorkSpace On-Demand 3.0.1 Deployment Server.

### **1.3.4 IBM OS/2 Client**

As mentioned earlier, IBM OS/2 can operate as a true thin client OS. Alternatively, if there is a local hard disk present, this can also be used. The two approaches are outlined as follows:

1. IBM OS/2 is downloaded over the network from the IBM WorkSpace On-Demand 3.0.1 Deployment Server and held in RAM. No local hard disk is required.
2. A combination of local boot and remote boot loads the operating system. By this method, all read-only files are installed on a local media to reduce network traffic and improve performance during boot storms. The read-only files are *cached*.

All read/write files will be loaded over the network and are stored on the IBM WorkSpace On-Demand 3.0.1 Deployment Server.

---

## Chapter 2. Installation

This chapter discusses the necessary steps in installing IBM WorkSpace On-Demand 3.0.1. It covers the hardware and software requirements for the server and clients.

This chapter also covers the Java Virtual Machine on IBM WorkSpace On-Demand 3.0.1.

---

### 2.1 IBM WorkSpace On-Demand 3.0.1 hardware prerequisites

IBM WorkSpace On-Demand 3.0.1 places a number of requirements on both the server and client hardware. The following sections describe the minimum hardware requirements and, where appropriate, recommendations that will help you derive greater performance from the product.

Hardware requirements differ significantly between environments. Usually, you need to take into account what applications are required, how often they are used, how many are used concurrently, and so on. There are some basic hardware considerations you should check before deploying any of the new functions. For a large corporate rollout, it is best to simulate a real world environment before beginning deployment. This environment can be used to gather data for the current rollout and future expansion.

Further, you should consider to plan your rollout in a phased approach. Every phase will give you the chance to discover new aspects of your implementation and if they meet your needs.

#### 2.1.1 Supported network adapters

At the time of printing, the IBM WorkSpace On-Demand 3.0.1 supports the following network adapters:

- IBM EtherJet
- IBM Token-Ring
- Other network adapters that support the unattended installation and the DHCP PXE Boot mechanism. The driver's INF file must be available if the adapter is not directly supported by Windows 2000 or Windows NT.
- IBM WorkSpace On-Demand 3.0.1 servers

IBM WorkSpace On-Demand 3.0.1 installs on Windows 2000 Advanced Edition Server or Windows NT Server 4.0 with Service Pack 4.

The Server where the Configuration Server component will be installed needs to be a Primary Domain Controller.

We recommend that the following hardware requirements should be met:

- Intel Pentium II 200MHz Processor or greater
- 256 MB RAM or greater
- 150 MB hard disk space available (Configuration Server)
- 200 MB hard disk space available for each operating system image (Deployment Server only)
- SVGA graphics (256 colors or more)

The following is a list of the minimum hardware requirements for a WorkSpace On-Demand 3.0.1 remote administration console:

- Intel Pentium II 200MHz Processor or greater
- 128 MB RAM
- At least 10 MB hard disk space available
- SVGA graphics (256 colors or more)

### **2.1.2 Client hardware considerations**

These vary depending on which client operating system you choose to deploy. See the following individual subsections in addition to the paragraph about remote boot capability.

### **2.1.3 Remote boot capability**

All clients that are going to be used in this environment need to be able to boot from the network using the PXE boot mechanism. All network adapters in your environment must be remote install enabled. This implies that the card should have a remote boot chip, and the chip must be enabled. This is usually done using DIP switches or a software setup program.

The client machine must also have a BIOS setting that allows the network adapter to be one of the boot devices. The normal configuration is: 1) diskette, 2) network, 3) CD, and 4) hard drive.

#### **Note**

For support, IBM requires that both requirements are met, and the network adapters selected are on the list of supported adapters.

## **2.1.4 Windows 2000 client hardware**

The following is a list of the base requirements for Windows 2000 clients. Note that these specifications are dependent on your application requirements.

### **2.1.4.1 Processor**

The Windows 2000 client will run on any Intel-based system (personal computer or network computer) that is capable of running a standard Windows 2000 client installation. The bare minimum specification is an Intel 486-33 MHZ, but for performance reasons, we suggest you use at least an Intel Pentium.

### **2.1.4.2 Memory**

A minimum of 32 MB of RAM is required. However, the exact amount of RAM required on your client depends on the number of applications that you intend to run concurrently.

### **2.1.4.3 Disk space**

As mentioned earlier in this chapter, a hard drive is required on your clients. The default installation partition size is 600 MB.

The maximum partition you can specify during setup of the client is 2 GB. See Section 4.2.3, "Windows 2000 client installation and boot" on page 112 for details on how to change this if necessary.

## **2.1.5 Windows NT Workstation 4.0 client hardware**

The following is a list of the base requirements for Windows NT Workstation 4.0 clients. Note that these specifications are dependent on your application requirements.

### **2.1.5.1 Processor**

The Windows NT client will run on any Intel-based system (personal computer or network computer) that is capable of running a standard NT Workstation installation. The bare minimum specification is an Intel 486-33 MHZ, but for performance reasons, we suggest you use at least an Intel Pentium.

### **2.1.5.2 Memory**

A minimum of 12 MB of RAM is required on x86-based computers, and 32 MB is recommended for a better performance. However, the exact amount of

RAM required on your client depends on the number of applications that you intend to run concurrently.

The more memory you have available for your clients, the better performance you get. The Windows NT Client Feature in an office environment should perform well with 64 MB of memory installed. Even better performance can be achieved with 128 MB or more.

#### **2.1.5.3 Disk space**

As mentioned earlier in this chapter, a hard drive is required on your clients. The default installation partition size is 400 MB. It is recommended that you increase the default partition size for Windows NT Workstation machines to 600 MB to accommodate the service pack.

The maximum partition you can specify during setup of the client is 2 GB. See Section 4.3.3, “Windows NT client installation and boot” on page 131 for details on how to change this if necessary.

### **2.1.6 Windows 98 client hardware**

The following is a list of the base hardware requirements for Windows 98 clients. Note that these specifications are dependent on your application requirements.

#### **2.1.6.1 Processor**

The Windows 98 client runs on any Intel-based system (personal computer or network computer) that is capable of running a standard Windows 98. As with the Windows NT client, the bare minimum specification is an Intel 486 33 MHZ, but for performance reasons, we suggest you use at least an Intel Pentium.

#### **2.1.6.2 Memory**

The client's operating system requires at least 8 MB of memory. We found 32 MB of memory is sufficient. The amount of memory also depends on the number of applications that you are running on your clients. Because memory prices have dropped, it is the most cost effective method of increasing workstation performance. With 64 MB of memory or more, the Windows 98 client will perform best in a standard office environment.

#### **2.1.6.3 Disk space**

As mentioned earlier in this chapter, a hard drive is required on your clients. The default installation partition size is 600 MB. The maximum partition you can specify during setup of the client is 2 GB. See Section 4.3.3, “Windows



NT client installation and boot” on page 131 for details on how to change this if necessary. By default, the partition on the target client will be set up as FAT, but it is possible to use NTFS instead. Again, see Section 4.3.3, “Windows NT client installation and boot” on page 131.

## **2.1.7 IBM OS/2 client hardware**

The following is a list of the base requirements for IBM OS/2 clients. Note that these specifications are dependent on your application requirements.

### **2.1.7.1 Processor**

An Intel 486/33 is the minimum but we recommend that a Intel Pentium 90 MHz or greater is used.

### **2.1.7.2 Memory**

As with the Microsoft clients, 12 MB of memory is required on x86-based computers, and 32 to 64 MB is recommended for better performance. The exact amount of RAM required on your client depends on the number of applications that you intend to run concurrently. The more RAM, the better performance.

### **2.1.7.3 Disk space**

If you set up your IBM OS/2 client as a remote boot client, no hard disk is necessary. The swapping will be handled over the network, and the SWAPPER.DAT is stored on the IBM Workspace On-Demand 3.0.1 boot server.

#### **Note**

To use the Local Cache option for an IBM OS/2 client, you must have one of the following installed on the client machine:

- A partitioned hard disk
- A Compact Flash card for the IBM Network Station Series 2800 computers

The Compact Flash card or the C partition of the hard disk must be empty, formatted with the Fat 16 file system, and have a minimum of 48 MB of free space.

To use the Local Cache option:

1. Create the client machine without using the Local Cache option.

2. After the client machine boots up, run FORMAT.COM from an IBM OS/2 prompt to format the local hard disk.
3. Delete the client machine.
4. Recreate the client machine using the Local Cache option.

If your client uses local caching, a minimum of 100 MB up to 300 MB, depending on the options selected for installation, is required for an IBM OS/2 client. The default installation requires 200 MB of free disk space.

---

## 2.2 Software prerequisites

The following are software prerequisites:

- Windows 2000 Advanced Edition Server or Windows NT Server 4.0 with Service Pack 4 or above
- NTFS file system on partition used for WorkSpace On-Demand 3.0.1 server
- TCP/IP configured on every server

### Note

The IBM WorkSpace On-Demand 3.0.1 installation will fail if the protocol Novel IPX has been installed on the server.

### 2.2.1 IBM OS/2

The software prerequisites for IBM OS/2 are a licensed copy of IBM OS/2.

### 2.2.2 Windows 2000 Professional Edition

The software prerequisite for Windows 2000 is licensed copy of Microsoft Windows 2000 Professional Edition. No Service Pack is required.

### 2.2.3 Windows NT Workstation 4.0

The software prerequisite for Windows NT is licensed copy of Microsoft Windows NT Workstation 4.0. Service Pack 4 or later.

### 2.2.4 Windows 98 Second Edition

The software prerequisite for Windows 98 is a licensed copy of Windows 98 Second Edition.

---

## 2.3 Installation roadmap

Figure 5 illustrates the roadmap to installing IBM WorkSpace On-Demand 3.0.1.

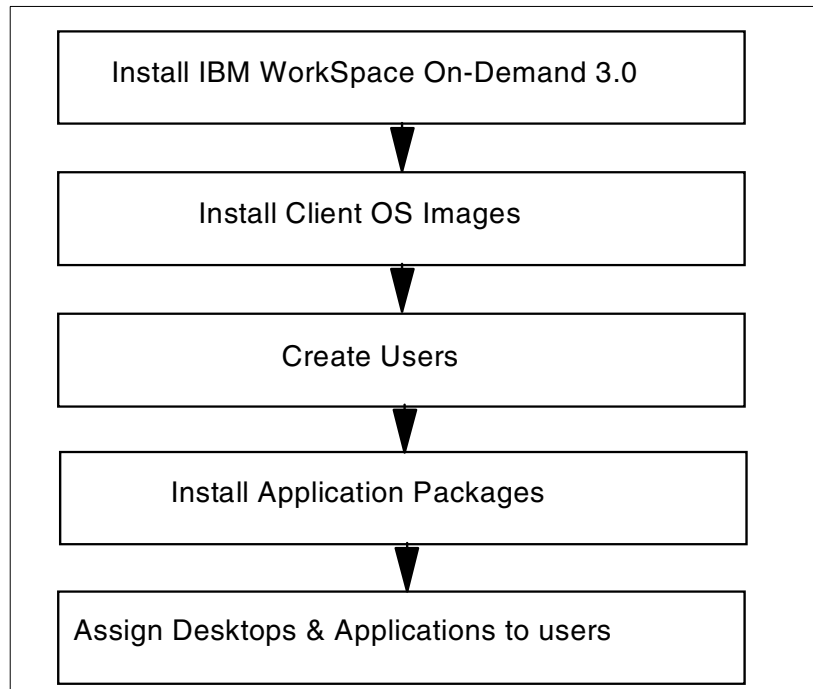


Figure 5. Roadmap to installing IBM WorkSpace On-Demand 3.0.1

---

## 2.4 Java Runtime Environment V1.3

Because most of the IBM WorkSpace On-Demand 3.0.1 components are written in Java, a Java Runtime Environment (JRE) is installed. The JRE is placed in the IBM WorkSpace On-Demand 3.0.1 tree. This installation does not change Java-related variable statements, such as the classpath. This means that the Java Virtual Machine (JVM) runtimes on the same system are not affected.

At the time of printing, the IBM WorkSpace On-Demand 3.0.1 used IBM JRE 1.3.0.

---

## 2.5 The DHCP PXE boot mechanism

DHCP PXE is a boot mechanism developed to take advantage of the Dynamic Host Configuration Protocol (DHCP) functionality on a TCP/IP network. It utilizes Intel's Preboot eXecution Environment (PXE) technology.

More general information about the PXE initiative can be found on Intel's Web site at: <http://developer.intel.com/ial/wfm/wfm20/design/pxedt/>

### 2.5.1 Supporting a DHCP PXE boot environment

In order for a client workstation to load its operating system from the IBM WorkSpace On-Demand 3.0.1 deployment server, both the client and the server must satisfy certain requirements as described in the following sections.

#### 2.5.1.1 Client workstation

The client must have the ability to determine, during the power-on self test (POST), that it must load its operating system code from a network rather than a local device. On a PC, this ability is typically provided by the system BIOS and is configured by selecting the network as the startup device.

The client then broadcasts a request over the network to ask a server to provide it with a boot image. This function is performed by the Boot PROM. The DHCP PXE client follows these steps to achieve this:

1. An IP configuration (including IP address, router address, and so on) is loaded from the DHCP server.
2. The address of the Boot Information Negotiation Layer (BINL) server containing the boot block that the client should use is returned. The client must request this information in addition to the normal DHCP request.
3. The name of the file on the server that contains the client's bootstrap routine is returned to the client. The client obtains this information from the boot server itself.

Figure 6 on page 17 gives a more detailed overview of the process.

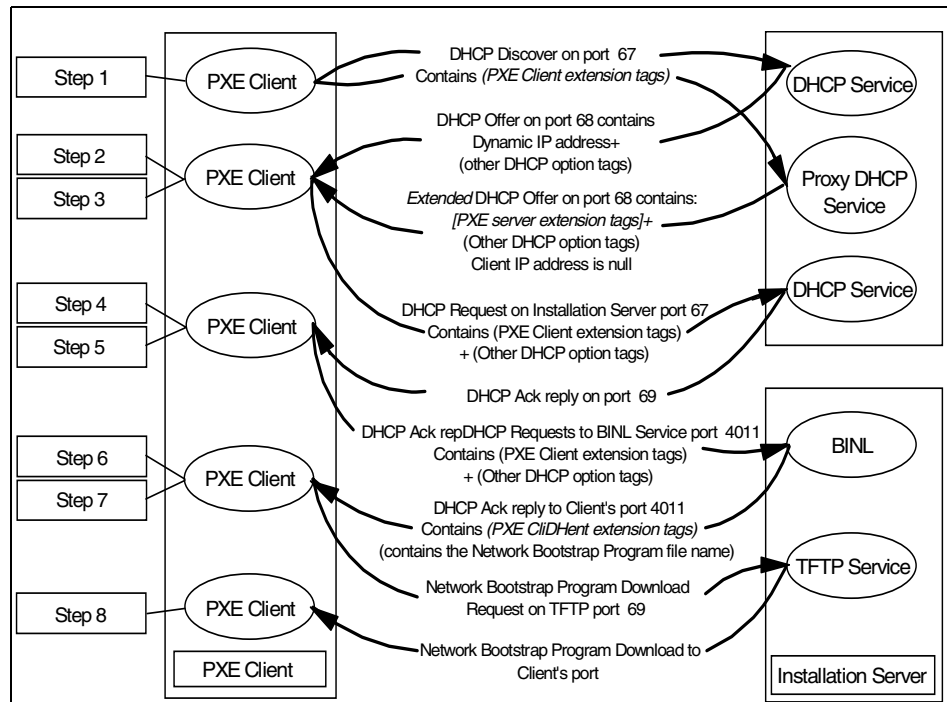


Figure 6. PXE remote boot sequence

### 2.5.1.2 Boot server

To use the DHCP PXE boot mechanism, the requirements for the server are somewhat more complex. The server must be able to receive a request from a DHCP PXE client and respond with the information that the client requires. The server must provide the following functions:

1. The server must provide an IP configuration to the client. In order to boot over a TCP/IP network, a client must first have its own IP address. This enables the boot server to route its responses to the client.

In a DHCP environment, IP addresses are dynamically allocated by a DHCP server when requested. A client may use a different IP address each time it boots.

The allocation of IP addresses is provided by the normal DHCP service of a DHCP server. IBM WorkSpace On-Demand 3.0.1 for Windows 2000 and Windows NT includes a DHCP service that supports this function, or you can use an existing DHCP server on your network.

2. The server must provide the IP address of the BINL server containing the client's bootstrap routine.

A normal DHCP server provides only an IP configuration to the client. However, the PXE standard introduces an extension to the DHCP protocol that provides the client with the address of the Boot Information Negotiation Layer (BINL) server it must use to obtain its bootstrap routine.

The updated DHCP service allows Windows 2000 and Windows NT to act as their own DHCP server. This DHCP service is a replacement for any DHCP server that does not support PXE.

3. The server must provide the name of the file containing the client's bootstrap routine. The BINL server provides this information to the client. IBM WorkSpace On-Demand 3.0.1 for both Windows 2000 and Windows NT includes a BINL service.
4. The server must provide a mechanism whereby the client can download the bootstrap routine. This mechanism is provided by a TFTP server. IBM WorkSpace On-Demand 3.0.1 for both Windows 2000 and Windows NT includes a TFTP server service (TFTPD).

Note that these functions can be performed by different services running on the same server or by different servers on the network. In our test environment, the DHCP, BINL, and TFTPD services are typically combined on a single server.

### **2.5.2 The PXE Boot Process**

Upon power on, the client uses the DHCP PXE protocol to obtain the IP address of the boot server and the name of the initial bootstrap. The same initial bootstrap is used for all clients. After getting downloaded to the memory of the client, this initial bootstrap obtains the client configuration information from a client-specific configuration file. Based on this information, it then determines which second bootstrap to download. The second bootstrap then does what is required to:

- Initiate the remote boot process (for OS/2 Warp clients), or
- Start the remote boot/install process (for Windows clients), or
- Boot the local hard disk in the case where a local boot of an installed operating system is required.

The following descriptions are simply overviews of the initial boot and of each operating system boot.

### 2.5.2.1 The initial boot process

When a client workstation is configured to boot from the network, the BIOS passes control to the Boot PROM when the machine is turned on and the power-on self test (POST) sequence completes.

The Boot PROM then contacts the required server services to obtain a bootstrap routine. The code within the Boot PROM initializes the network adapter and sends out a DHCP DISCOVER packet on the standard DHCP port (67). An option field in this packet contains the following:

- A tag for client identifier (if the client identifier is known)
- A tag for the client Network Interface Identifier
- A tag for the client system architecture

IBM WorkSpace On-Demand 3.0.1 supports only those network adapters that include a PXE-compliant Boot PROM or offer such a Boot PROM as an option. Other network adapters that may provide BOOT-P support are not supported by IBM WorkSpace On-Demand 3.0.1.

A DHCP server will respond by offering an IP configuration in a DHCPOFFER packet on port 68. The DHCPOFFER packet sent by the PXE DHCP server contains the address of the BINL server in the siaddr field. If the DHCP server is running the PXE DHCP service, it may respond with a single DHCPOFFER packet on port 68 containing both the IP configuration and the address of the BINL server. Note that any DHCP server on the network can reply to the client's request with a DHCPOFFER packet. The client can choose which DHCPOFFER packet it will accept.

The time-out for a reply from a DHCP server is standard. The time-out for rebroadcasting to receive a DHCPOFFER packet with PXE extensions packet is based on the standard DHCP time-out, but is substantially shorter to allow reasonable operation of the client in standard BOOTP or DHCP environments that do not provide a DHCPOFFER packet with PXE extensions. The PXE time-out for rebroadcast is 4, 8, and 16 seconds, yielding three broadcasts and a time-out after 28 seconds. The PXE time-out for rebroadcast is 4 seconds after receiving a DHCPOFFER without PXE extensions but with a valid boot file name option.

The client uses the IP address contained in the siaddr field of the DHCPOFFER packet to contact the BINL server by sending a DHCPREQUEST packet to the BINL server on port 4011. This packet contains the following information:

- The IP address assigned to the client by the IP configuration received from the main PXE DHCP server
- All the PXE options fields received from the selected DHCP OFFER packet that contained the PXE options

The BINL server sends a DHCPACKNOWLEDGE packet to the client, also on port 4011. This reply packet contains:

- The bootstrap file name and location
- The Client UUID/GUID option in the PXE DHCP offer
- MTFTP configuration parameters

The client downloads the bootstrap routine using standard TFTP or MTFTP protocols through TFTP client code built into the DHCP PXE Boot PROM. The file downloaded and the placement of the bootstrap routine in memory is dependent on the client's CPU architecture. Those Informations are found on the Server in a file called TDMCLNT.INF.

The Boot PROM now passes control to the bootstrap routine, which now determines the behavior of the client for the next phase of the boot process. The bootstrap routine uses the TFTP client interface built into the Boot PROM to reconnect to the server and access the files it requires.

- The PXE DHCP service (DHCP) provides an IP configuration to a client workstation. It provides an IP configuration and the address of a BINL server, thereby combining the functions of the PXE Proxy and the DHCP services.
- The BINL service (BINL) provides the name and server location of the file containing the client workstation's bootstrap routine.
- The TFTP server service (TFTPD) accepts connection requests from the TFTP client interface built into the Boot PROM and downloads files to the client workstation using the TFTP protocol.
- The File and Print Sharing service (SERVER) accepts connection requests from the client workstation's protect mode redirector and downloads any remaining files using NetBEUI or TCPBEUI protocols. This service is installed by default on Microsoft Windows 2000 and NT 4.0.

#### **2.5.2.2 Boot Process for Windows 2000, NT and 98**

Remote Boot/Install (Windows 2000, NT 4.0 and 98 SE clients only) is as follows:



After the initial boot, new Windows 2000, NT 4.0 and 98 SE clients always start with an installation boot, where the client operating system is downloaded and installed on the client's local disk.

### ***Installation Boot***

The second bootstrap gains control of the client.

1. The second bootstrap downloads the TDMDOS.INF file from the boot server using TFTP.
2. Based on information in the TDMDOS.INF file, the second bootstrap determines the boot image's name and location. This boot image is a DOS boot image.
3. The second bootstrap downloads the DOS boot image from the boot server using TFTP.
4. The second bootstrap applies required fix-ups to the DOS boot image.
5. The second bootstrap intercepts BIOS disk I/O to redirect floppy access to the DOS boot image.
6. The second bootstrap "boots" the DOS image.
7. A network driver and network support code are loaded.
8. A DOS executable, called CONNECT.EXE, establishes two connections to the boot server: one to the Read-Only share and the other to the Read-Write share.
9. An installation shell is started.
10. The installation shell repeatedly calls another DOS executable, called STATE.EXE, and processes commands sent to it by STATE.EXE.
11. STATE.EXE uses a file called STATE.INI to sequence through all necessary phases in the boot process. STATE.INI is really a script file, containing all of the commands to be executed by STATE.EXE to set up the Windows operating system, the Win32 Java Virtual Machine (JVM), the Tivoli Management Agent (TMA), and the WorkSpace Log-on Client (which facilitates the user log-on process to the domain controller). STATE.EXE also updates the COUNTER.INI file to record what it has completed, so it knows what to do the next time it gains control. Here are the phases which STATE.EXE goes through in the boot sequence (in the order of execution):
  - a. Delete the partition table, create new partition on the client's local disk, and reboot.
  - b. Format the installation partition on the client's local disk.

- c. Execute the copy commands specified in STATE.INI to copy the Windows operating system installation files, device drivers, response files, and all other files required to set up the WorkSpace Log-on Client, the Win32 JVM, and the TMA.
- d. Execute commands to release the redirected floppy drive A, load the disk caching program to speed up the copying of Windows files, write a trigger file to the boot server, and call the Windows setup program to start the installation of the Windows operating system on the client's local disk in unattended mode. The trigger file is used to notify a daemon running on the boot server to switch the bootstrap program for the client to the hybrid boot image, so that the next time the client boots, it will go through the hybrid boot process instead (to be discussed later). After the server daemon has switched the bootstrap image for the client, it deletes the trigger file.

There may be cases where the network drivers consume such a significant amount of conventional memory that the memory requirement of the Windows setup program cannot be satisfied. In such cases, STATE.EXE will install DOS (without the network support) on the client's local disk, write a trigger file to the boot server (so a daemon running on the server can switch the bootstrap program to the hybrid boot image), and copy the following from the boot server to the client's local disk: the Windows setup files, the STATE.INI (script) file, and other support files and executables. The client is then rebooted. After the client is rebooted, STATE.EXE resumes the execution of commands in STATE.INI locally on the client and starts the Windows setup program without network connections.

After the Windows operating system has been set up successfully, executables required to set up the WorkSpace Log-on Client, the Win32 JVM, and the TMA are then executed (also in unattended mode).

### ***Hybrid Boot***

Once the daemon running on the boot server detects that a trigger file has been placed on the server by a client, it will modify the TDMCLNT.INF file on the server to point to a hybrid boot image, instead of the second bootstrap. The hybrid boot image is a small bootstrap executable which, once downloaded to the client, will allow the client to boot directly from the client's local disk.

The hybrid boot process begins with the initial boot sequence as described at the very beginning of this section. During this initial boot sequence, the initial bootstrap will download the TDMCLNT.INF file from the boot server using TFTP (please see Step 9 in the initial boot sequence). Based on the

information in the TDMCLNT.INF file, which now points to the hybrid boot image, the initial bootstrap will next download the hybrid boot image from the boot server using TFTP, and pass control to the hybrid boot image. The hybrid boot image will then switch to the Windows bootstrap program on the client's local disk and the Windows operating system is booted from there.

Hybrid boots are very fast and do not require much network bandwidth because the Windows operating system is booted from the client's local disk. All subsequent boots after the installation boot use this very fast hybrid boot process.

### **2.5.2.3 Remote Boot (OS/2 Warp clients only)**

After the initial boot, the boot process for OS/2 Warp clients continues as follows:

1. The second bootstrap gains control of the client.
2. The second bootstrap downloads the TDMOS2.INF file from the boot server using TFTP.
3. Using TFTP, the second bootstrap downloads all files specified in TDMOS2.INF from the boot server to a RAM DISK (including the BPCOMMON file, which contains a list of additional files that are needed).
4. The second bootstrap downloads all files specified in BPCOMMON from the boot server to the RAM DISK using TFTP.
5. The second bootstrap executes the OS/2 Warp loader (OS2LDR) with itself as the mini-Installation File System (min-IFS). It fulfills all disk requests using the RAM DISK.
6. The OS/2 operating system, network, and redirected file system are loaded.
7. The redirected file system replaces the RAM DISK. The OS/2 boot process continues with the redirected file system using the System Message Block (SMB) protocol. SMB connections are made to RPLFILES and WRKFILES shares on the boot server.
8. Boot continues until OS/2 Warp is up and the log-on window is displayed.

If the caching option is used, the client is required to have a local disk. Non-writable, common OS/2 files will be copied from the boot server to this disk, so that the next time the client boots, these files will be loaded from the client's local disk, instead of loaded from the boot server across the network. As a result, the first boot to initialize the cache takes more time than subsequent cached boots. During the first boot with caching, the following takes place:

- A device driver, called NCCACHED.SYS, is loaded and executed early in the boot sequence, just after the network drivers and support are loaded. This driver creates a trigger file and places it on the boot server.
- At a later time in the boot sequence, an executable, called NCCACHEE.EXE, runs and waits for an action file to be created by the caching daemon running on the boot server. This action file contains instructions on which files to cache. NCCACHEE.EXE then copies these files from the boot server to the client's local disk. The client is then rebooted.

During the first remote boot with caching, the action file will denote that every file listed should be copied from the boot server to the client's local disk. Subsequently, it will denote that there is either nothing to do or that files need to be refreshed. New or refreshed files will be copied to the cache (client's local disk) and the client will reboot. If there is nothing to do, the boot just continues on to completion.

---

## 2.6 Installing IBM WorkSpace On-Demand 3.0.1

Both attended and unattended installation of IBM WorkSpace On-Demand 3.0.1 server components is possible. The standard IBM WorkSpace On-Demand 3.0.1 installation requires an administrator to answer prompts provided by the installation GUI.

Some circumstances call for unattended installation, where no one is physically present to answer questions. For example, administrators of large networks may prefer to use a shared network to install a product on many computers. Software distribution packages, such as Tivoli, can also be used.

For unattended installations, you will need to prepare a response file in advance. This chapter covers both using the GUI for the attended installation, as well as how to use response files for unattended installations of IBM WorkSpace On-Demand 3.0.1. Whether your installation is attended or unattended, you must have Windows 2000 or Windows NT administrator privilege to install IBM WorkSpace On-Demand 3.0.1. Information about Tivoli Software Distribution can be found in Chapter 9, "Comparisons" on page 407, but it is beyond the scope of this book to go into detail about how to implement the Tivoli Management Environment. Unattended installation is discussed in detail in the *IBM WorkSpace On-Demand 3.0.1 Administrator Guide*, which is shipped with the product.

Section 2.6.1, "Installing WorkSpace On-Demand 3.0.1 on Windows Servers" on page 25, Section 2.6.2, "Installing IBM WorkSpace On-Demand 3.0.1

Client Enablement” on page 34 and Section 2.6.4, “Directory structure for IBM WorkSpace On-Demand 3.0.1” on page 40 refer to both Windows 2000 and Windows NT Server installation

### 2.6.1 Installing WorkSpace On-Demand 3.0.1 on Windows Servers

Before beginning installation, it is strongly recommended that you close all open applications. It is also advised that you have all the prerequisite hardware and software components installed.

#### Note

Even if a DNS Server is available in your LAN and your Server is defined there, be sure to have the explicit address resolution inside the local HOST file for your Server.

To install IBM WorkSpace On-Demand 3.0.1, complete the following steps:

1. Insert the IBM WorkSpace On-Demand 3.0.1 CD into the CD-ROM driver.
2. Select **Start->Run**.
3. Enter the drive letter of your CD ROM and type `SETUP` For example, if your IBM WorkSpace On-Demand 3.0.1 CD-ROM on your system is in drive D, enter: `D:\SETUP`

The installation program will display an IBM WorkSpace On-Demand 3.0.1 installation window. The next window is the Software Licence Agreement window as shown in Figure 7 on page 26



Figure 7. Software Licence Agreement

4. Read the License Agreement; you can use the scroll bar to read onto the next page. If you agree with the licensing, click the **Accept** button.



Figure 8. IBM WorkSpace On-Demand 3.0.1

5. Figure 8 shows the setup introductory window. Click the **Next** button to begin installation.

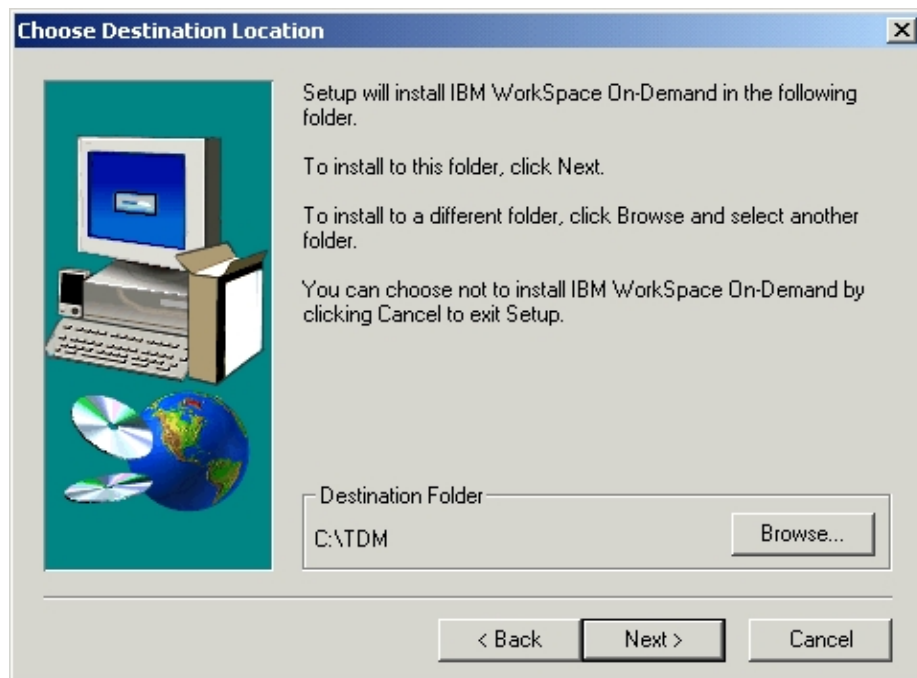


Figure 9. Choose Destination Location window

6. The window shown in Figure 9 prompts you for the destination directory. By default, it points to C:\TDM. You can change which drive and directory you want it to go by clicking the **Browse** button. Please select a drive that has at least 1 GB free space. Your actual disk usage will depend on which component you want to install. The installation directory can get quite big, so it is necessary to select a drive with sufficient space for all application and operating system images.



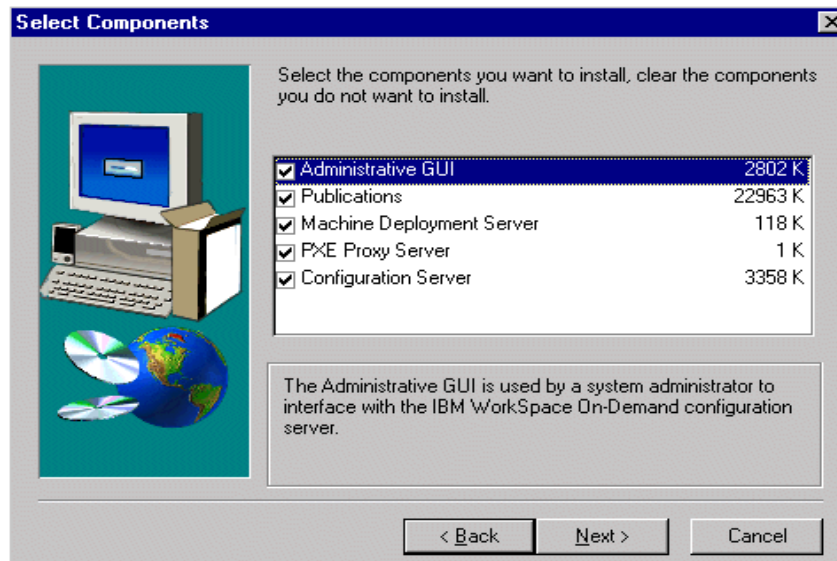


Figure 10. IBM WorkSpace On-Demand 3.0.1 components

7. On the window shown in Figure 10, you can select which components you want to install on this server. The options for this window are:

- a. Administrative GUI

Select this option if you would like to use the built-in feature. It is a good start for new administrator since this GUI is very user-friendly and the administration tasks are straightforward.

- b. Publications

Select this option if you want the online documentation copied onto your server.

- c. Machine Deployment Server

Selecting this option will add the deployment service to your server. This also includes installing the necessary IP support services.

- d. PXE Proxy Server

Windows 2000 and Windows NT Server do not come with the PXE Server service by default. You can make use of this component or the one made by Intel. We recommend you use this one. This service is

used to remote boot the clients. See Section 2.5, “The DHCP PXE boot mechanism” on page 16 for more details.

e. Configuration Server

This option does not appear if your server is not a Primary Domain Controller. This adds the IBM WorkSpace On-Demand 3.0.1 configuration server service to Windows 2000 or Windows NT. This service is responsible for administering your IBM WorkSpace On-Demand 3.0.1 system. Without a configuration server service, you will not be able to log on to the IBM WorkSpace On-Demand 3.0.1 console.

**Note**

Whichever components you choose, the administration console and the necessary IBM WorkSpace On-Demand 3.0.1 support files will be copied automatically to your server.

As discussed in Chapter 1, “Introduction” on page 1, these components can be installed on one or many servers depending on the servers role. For simplicity, we have installed all the services in a single server.

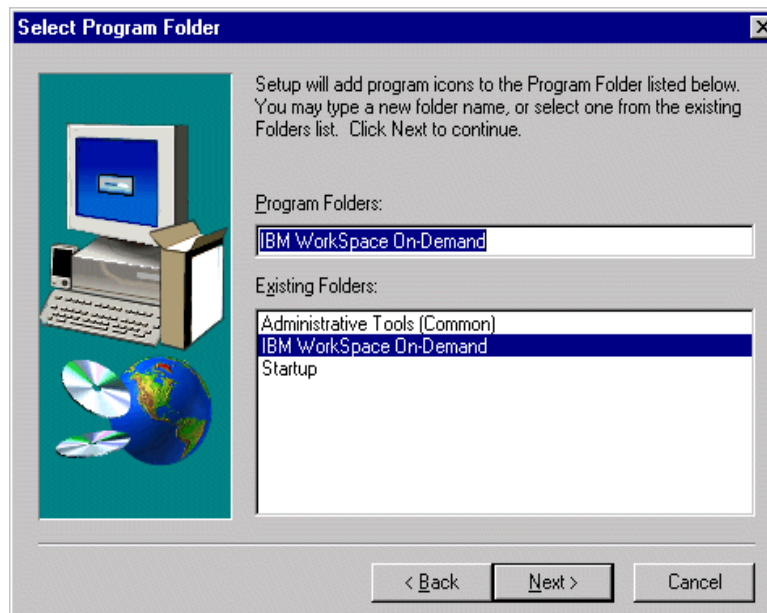


Figure 11. Select Program Folder

8. Setup will then ask you for a program folder as shown in Figure 11 on page 30. By default, it will install the icon and shortcut into the program folder “IBM WorkSpace On-Demand 3.0.1”.

Click **Next** to continue.

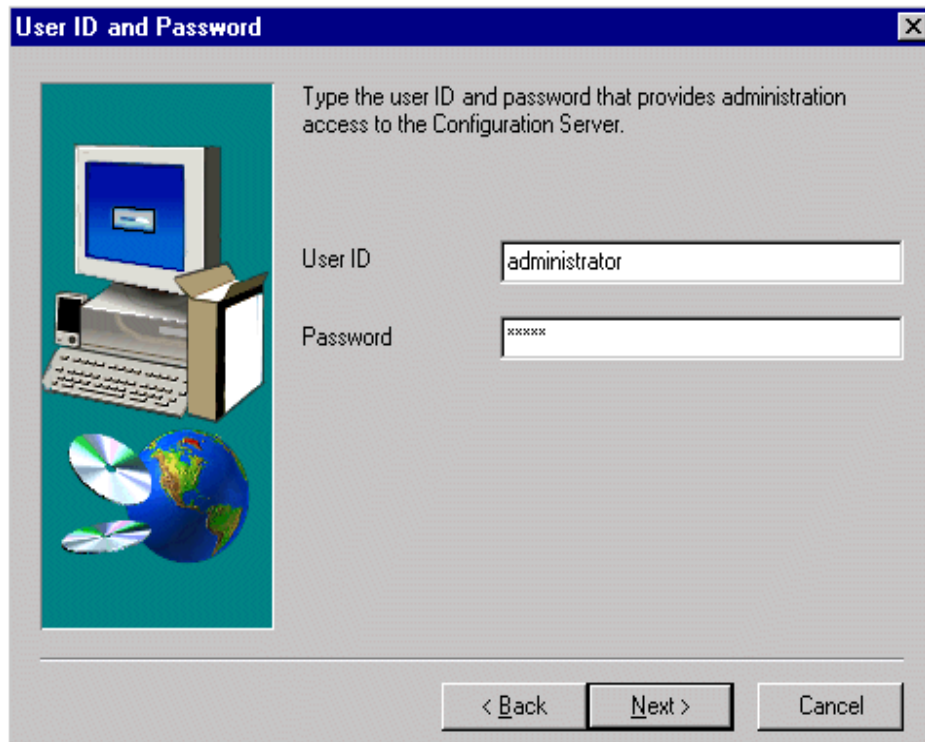
A screenshot of a Windows-style dialog box titled "User ID and Password". The dialog has a blue title bar with a close button (X) in the top right corner. On the left side, there is a graphic showing a computer monitor, a tower unit, a keyboard, and a CD-ROM. To the right of the graphic, the text reads: "Type the user ID and password that provides administration access to the Configuration Server." Below this text are two input fields. The first field is labeled "User ID" and contains the text "administrator". The second field is labeled "Password" and contains a series of "x" characters. At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel". The "Next >" button is highlighted with a black border.

Figure 12. User ID and password prompt

9. A user ID with administrator rights is required to continue with the installation. Enter a User ID and Password in the provided fields, as shown in Figure 12. Click **Next** to continue.

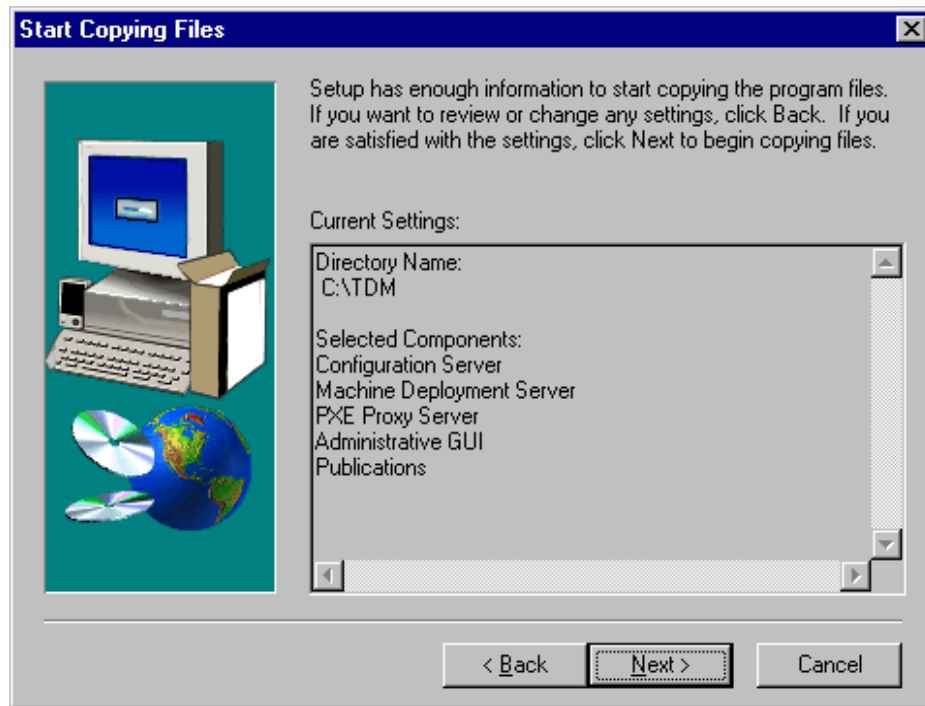


Figure 13. Start Copying Files

10. Right before copying the files, the installation program will prompt you for confirmation. The selected components are listed in Figure 13. If the component selection is correct, click **Next** to continue copying files.

After clicking **Next**, the installation of IBM Intermediate Support Drivers and the components selected starts.

**Note**

If the installation program seems to be dead for a while, check if any error message is displayed on the background by moving the installation window.

11. You could get the message shown in Figure 14 on page 33; this is to remind you to update the DHCP Server configuration.

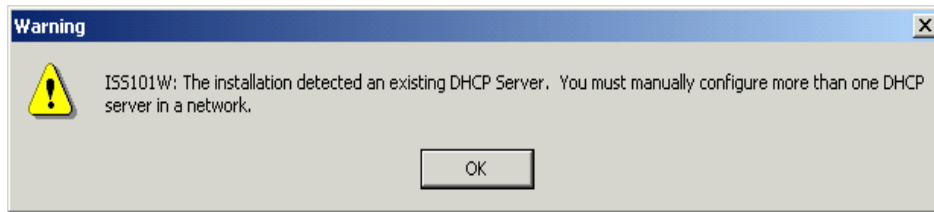


Figure 14. DHCP manual setup is needed

12. Sometime the installation program is not able to properly start the Configuration Server service: in this case you will get the message shown in Figure 15, prompting you to run from the CLI, at the end of the installation, the commands listed inside the TDMCmds.srv file.

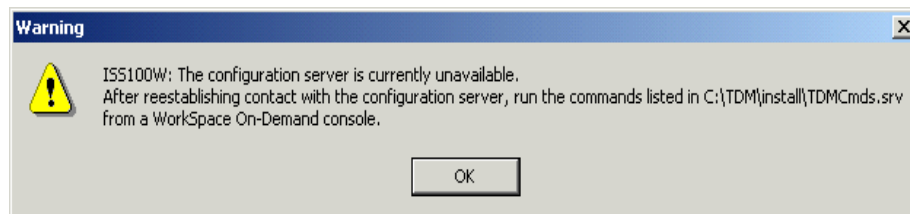


Figure 15. Configuration Server not started

**Note**

If you are facing this problem, you could try to restart the installation program after a reboot; in this way the Configuration Server service should automatically start and the second installation trial should run without problems, avoiding you to do the requested manual job.

After the file copy is complete, setup will automatically start the IBM WorkSpace On-Demand 3.0.1 Client Enablement. This will allow you install client operating system images. You can choose to cancel the client enablement install at this time. Note that it is recommended that you install at least one client operating system image.

The IBM WorkSpace On-Demand 3.0.1 server install is complete and you will be prompted to reboot.

After reboot, check that all the needed services are automatically started, as noted in Figure 16:

- IBM BINL Server
- IBM DHCP Server
- IBM WorkSpace On-Demand Configuration Service
- IBM WorkSpace On-Demand Deployment Service
- IBM WorkSpace On-Demand Local Cache Server

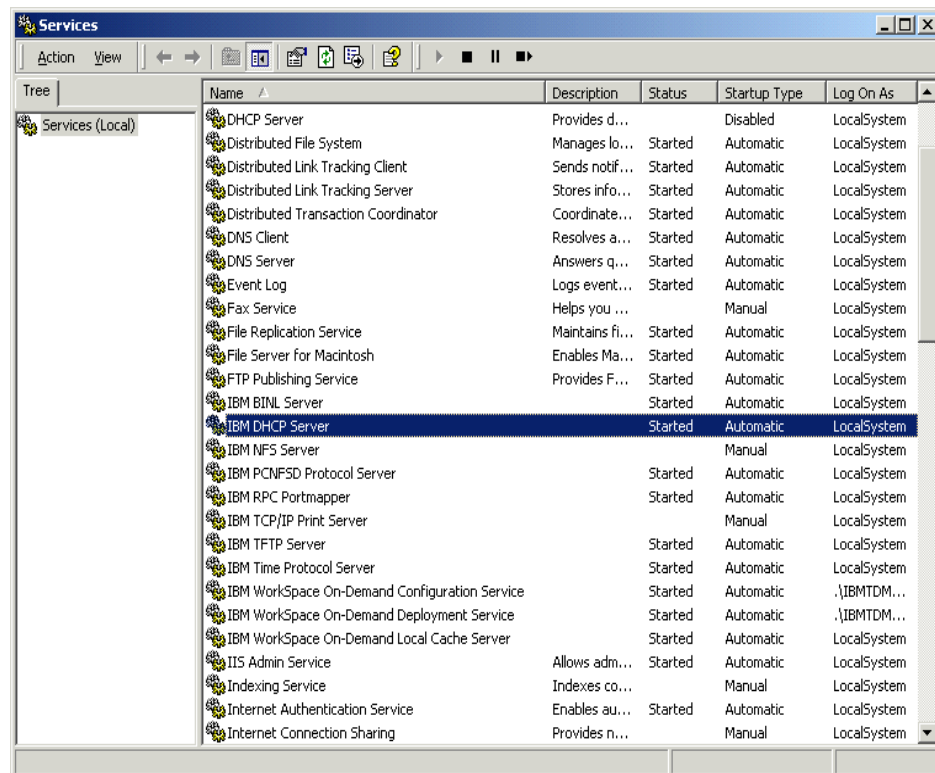


Figure 16. Check the Services status

Turn the *Startup Type* from *Manual* to *Automatic* if necessary.

## 2.6.2 Installing IBM WorkSpace On-Demand 3.0.1 Client Enablement

To be able to install the different client operating systems, the Client Enablement program is included with IBM WorkSpace On-Demand 3.0.1. This program is run right after the IBM WorkSpace On-Demand 3.0.1 setup. If

you did not setup a client and would like to add another supported operating system, you can do this by completing the following steps:

1. Insert the IBM WorkSpace On-Demand 3.0.1 CD into the CD-ROM drive.
2. Select **Start->Run**.

Enter the drive letter of your CD ROM and type `SETUP`. For example, if your IBM WorkSpace On-Demand 3.0.1 CD-ROM is in drive D, enter:

`D:\CLNTINST\DISK1\SETUP`

or select

**Start->Programs->IBM WorkSpace On-Demand->Install Client Enabler**

3. In both cases, an information pop-up window will display while Install Shield loads.

If this Client Enablement installation immediately follows the IBM WorkSpace On-Demand 3.0.1 installation, the next window will be similar to the one shown in Figure 17.

You can check which client operating system you want to create. Select an operating system by clicking the button beside the operating system. As an example, we used Windows 98 Second Edition.

Click **Next** to continue.

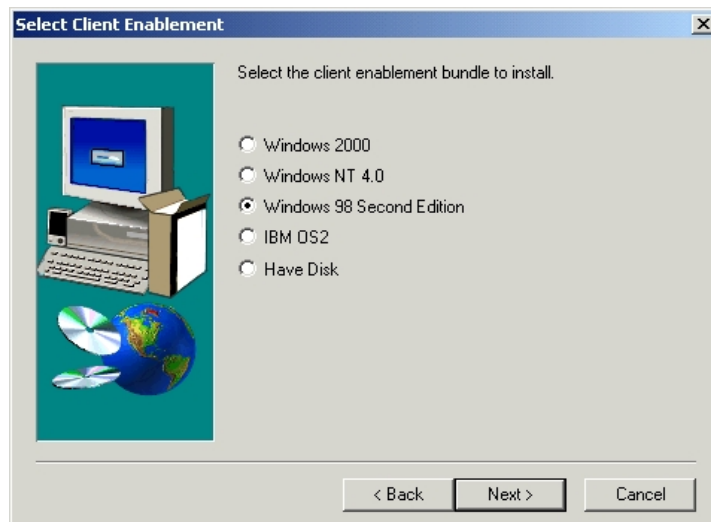


Figure 17. Select the operating system to be enabled

If you are installing the Client Enablement at different time than the IBM WorkSpace On-Demand 3.0.1 installation, you will see the window shown in Figure 17 on page 35.

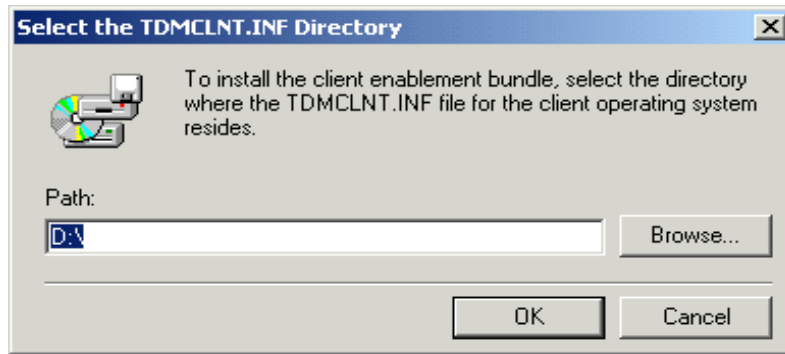


Figure 18. Request to select the directory for TDMCLNT.INF file

Be sure to have the IBM WorkSpace On-Demand 3.0.1 CD in the CD-ROM drive and select the correct directory, as shown in Figure 18

Select the directory where you will store the TDMCLNT.INF file, as shown in Figure 19.

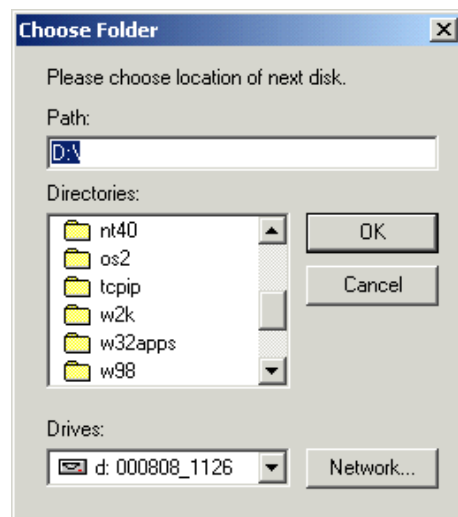


Figure 19. Select the directory for TDMCLNT.INF file



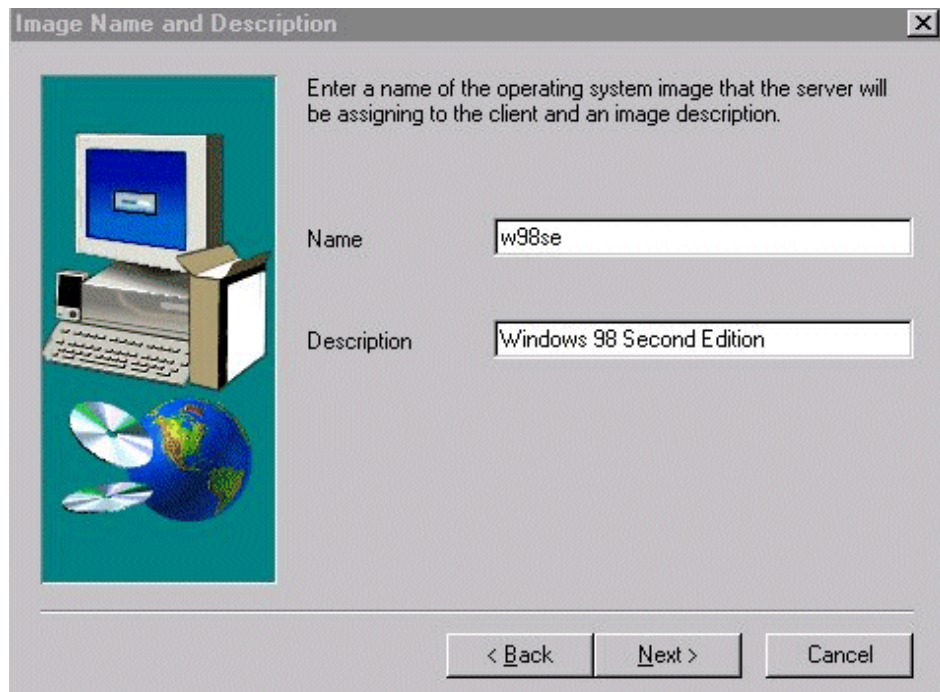


Figure 20. Image name

4. In both cases, the next window, shown in Figure 20, will ask you for an image name. This name can be anything; however, we suggest that you make it as descriptive as possible, since you can make multiple images of the same operating system. In our example, we used W98SE.
5. Type the image name in the space provided. Click **Next** to continue.



Figure 21. Machine Management Directory

The Machine Management Directory takes a lot of space. Most of the operating system images use about 200 MB. Remember that this is the directory where all your client operating system images will be stored; allocate more free space for future client operating system images.

6. If you are installing the first client operating system, then select a directory for your Machine Management Directory. By default, the directory is C:\TDM\MM, as shown in Figure 21. Click **Next** to continue. This will start copying the operating support files from the IBM WorkSpace On-Demand 3.0.1 CD.

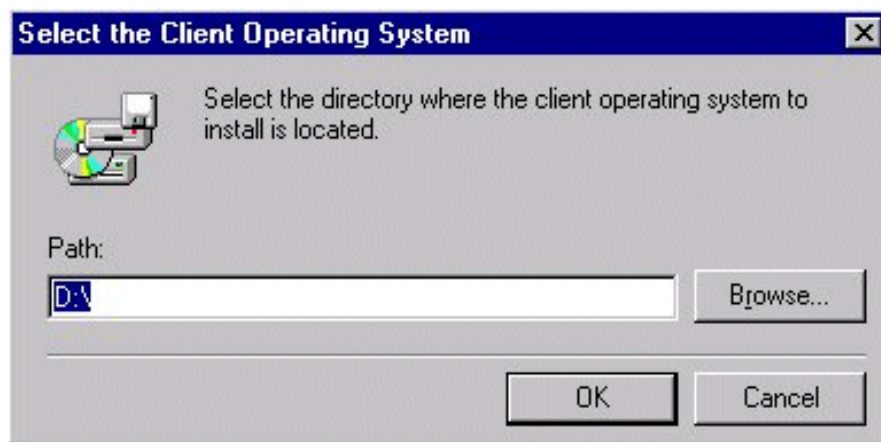


Figure 22. Client Operating System

7. After copying the IBM WorkSpace On-Demand 3.0.1 support files, the next prompt is the Select the Client Operating System prompt shown in Figure 22. Enter the drive letter on the space provided or click **Browse**, then enter the path of your CD-ROM drive or the shared network drive (whichever contains the OS image files you want to use).

For Windows 2000 and Windows NT, select the directory that contains the WINNT.EXE file, usually the I386 directory. For Windows 98, select the directory that contains the \*.CAB files, usually the WIN98\_SE \SETUP\WIN98 directory. For IBM OS2, select the directory that contains the OS2KRNL file, usually the WS3CLNT directory.

Clicking **Next** will start copying the operating system installation files.

Sometime the installation program is not able to properly start the Configuration Server service; in this case, you will get the message shown in Figure 23, asking you to run from the CLI the commands listed inside the TDMCmnds.clt file, at the end of the installation.

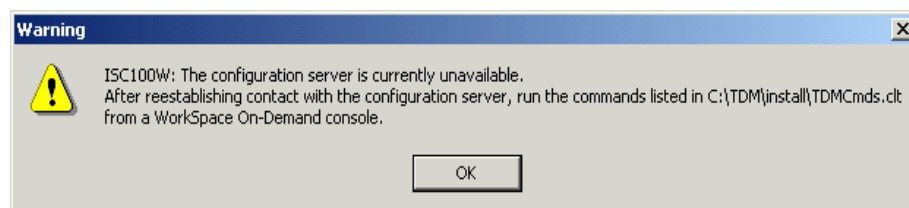


Figure 23. Configuration Server not started

If you want to add another client operating system, repeat the procedure again.

### 2.6.3 Removing client-machine operating system support

To remove a client-machine operating system from a server you have to unassign the operating system from the server using the `SERVER UNASSIGN` command, then delete the operating system image.

Proceed as follows:

1. From a Windows 2000 or NT server or from a Windows 2000 or NT client machine, select **Start->Programs->WorkSpace On-Demand->Administration CLI** to open the CLI.
2. At a command prompt, type:

```
CONNECT
```

3. Type your user ID, password, and the configuration-server name.
4. Type the `SERVER UNASSIGN` command. For example:

```
SERVER UNASSIGN IMAGENAME="w98se" NAME="workshop" LANG=US OS=W98
```

To delete the operating system image and support files, delete the directory where the operating system image and support files reside. For example:

```
\tdm \mm \client \ro \os \imagename \lang
```

where:

OS represents the client operating system.

Imagename represents the image name.

Lang represents the two-character language code for the operating system.

### 2.6.4 Directory structure for IBM WorkSpace On-Demand 3.0.1

After the installation, the following directories are created in the destination drive. We will discuss the underlying directories to give administrators an overview of the file location. Please refer to Figure 24 on page 41 for an overview of the file directories.

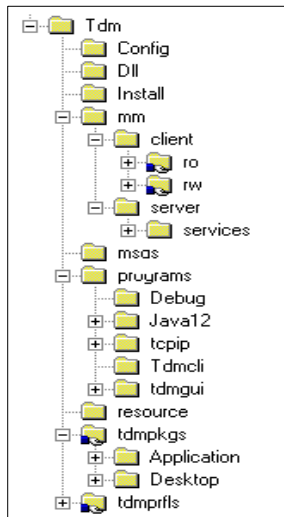


Figure 24. IBM WorkSpace On-Demand 3.0.1 directory structure

The directories are as follows:

|                 |  |
|-----------------|--|
| <b>TDM</b>      | Contains the logs and uninstall information.   |
| <b>CONFIG</b>   | Contains the data store (DATASTORE.INI) wherein all machine, desktop, and application definitions are stored.  |
| <b>DLL</b>      | Contains the dynamic link library of the TDM deployment and configuration services.  |
| <b>INSTALL</b>  | Contains the installation portion of services.   |
| <b>MM</b>       | Contains the machine management binaries and libraries. The RO subdirectory contains read-only files for clients, such as OS images and installation files. The RW subdirectory contains temporary directories where clients write the necessary files needed for installation. Also, the RO directory is shared as RPLFILES and the RW directory as WRKFILES. Chapter 4, “Defining machines” on page 103 discusses this directory in more detail. |
| <b>MSGS</b>     | Contains language prompts for the different dialog boxes.  |
| <b>PROGRAMS</b> | Contains the executable binaries, libraries, and Java class. It also contains subdirectories holding programs for TDMGUI, TDMCLI, JRE, and TCP/IP services.  |

- TDMPKG** Contains desktop and application packages. A more detailed look at the contents and use of this directory can be found in Chapter 7, “Updating the client images” on page 363.
- TDMPRFLS** Contains (in a shared directory) user profiles. Once a user is created and logged on, a subdirectory for the user is created.

**Note**

It is a good idea to verify, at the end of the installation, that the TDM\MM\CLIENT directory has the proper permissions set.

---

## 2.7 Installing and configuring IP services

IBM WorkSpace On-Demand 3.0.1 uses a number of services on the server to support an IBM WorkSpace On-Demand 3.0.1 client booting using the DHCP PXE boot mechanism. These are automatically installed by IBM WorkSpace On-Demand 3.0.1, but we include an overview of the manual configuration in case you wish to customize settings at a later date.

The services used by IBM WorkSpace On-Demand 3.0.1 are as follows:

There are two ways of configuring the DHCP service with IBM WorkSpace On-Demand 3.0.1. One is by editing the file located in DHCP.DHCP.D. It is an ordinary text file and can be edited by any ASCII text editor, such as NOTEPAD or EDIT.

The other way of configuring the DHCP service is by using the DHCP Configuration GUI, which we will discuss later.

### 2.7.1 DHCP.DHCP.D

This file is located in \TDM\PROGRAMS\TCPIP\ETC by default.

```
# sample dhcpd.conf
# Global options. Passed to every client unless overridden at a lower
scope.
logFileName dhcpd.log    #logging options
logFileSize 100
numLogFiles 10
logItem SYSERR
logItem OBJERR
logItem WARNING
logItem INFO
leaseExpireInterval 1 Minutes
```

```

leaseTimeDefault 24 Hours
pingTime 0.5 Seconds
reservedTime 5 Minutes
usedIPAddressExpireInterval 1000 Seconds
statisticSnapshot 1
updatedDNSA "nsupdate -f -h%s -s"d;a;*;a;a;s;s;s;3110400;q"
releasedDNSA "nsupdate -f -h%s -s"d;a;s;s;s;0;q"
supportBOOTP no
supportUnlistedClients both
allRoutesBroadcast no
appendDomainName yes
canonical no
proxyARec standard
bootstrapServer 10.0.0.1
imageserver 10.0.0.1# default BINL server
option 15 atlas1.austin.ibm.com # domain name
# Setup to send options to the pxeclient. Sent in the encapsulated
option 43.
vendor PXEClient
{
    option 60 PXEClient
}

subnet 10.0.0.0 255.255.255.0 10.0.0.50-10.0.0.80 (alias=np2060
{
    option 28 9.3.240.255 # broadcast address
    option 1 255.255.255.0 # subnet mask
    option 3 10.0.0.1 # router / default gateway
    option 5 10.0.0.1 # old (name server)
    option 6 10.0.0.1 # name server
    # Sample config for a specific client machine
    client 6 002035fe4a7b 10.0.0.50 (alias=test2
    {
        option 51 -1 # IP address lease time -1 is indefinite
    }
}
}
# end of dhcpd.conf

```

### 2.7.2 DHCP configuration - The GUI way

The other way of configuring DHCP is by its GUI. This DHCP configuration feature is installed along with the IBM TCP/IP services.

1. Before accessing the IBM TCP/IP services for the first time, you need to create an Administrative Password, by selecting **Start->Programs->IBM WorkSpace On-Demand->IBM TCP\_IP Services->Create Admin Password**

2. If you have already defined a password, you can logon to the DHCP Service by selecting **Start->Programs->IBM WorkSpace On-Demand->IBM TCP\_IP Services->DHCP Logon Service**

A DHCP Logon Window opens.

3. Then you can start the program by selecting **Start->Programs->IBM WorkSpace On-Demand->IBM TCP\_IP Services->DHCP Server Configuration**

When the DHCP Server Configuration Service starts you will see the window shown in Figure 25:

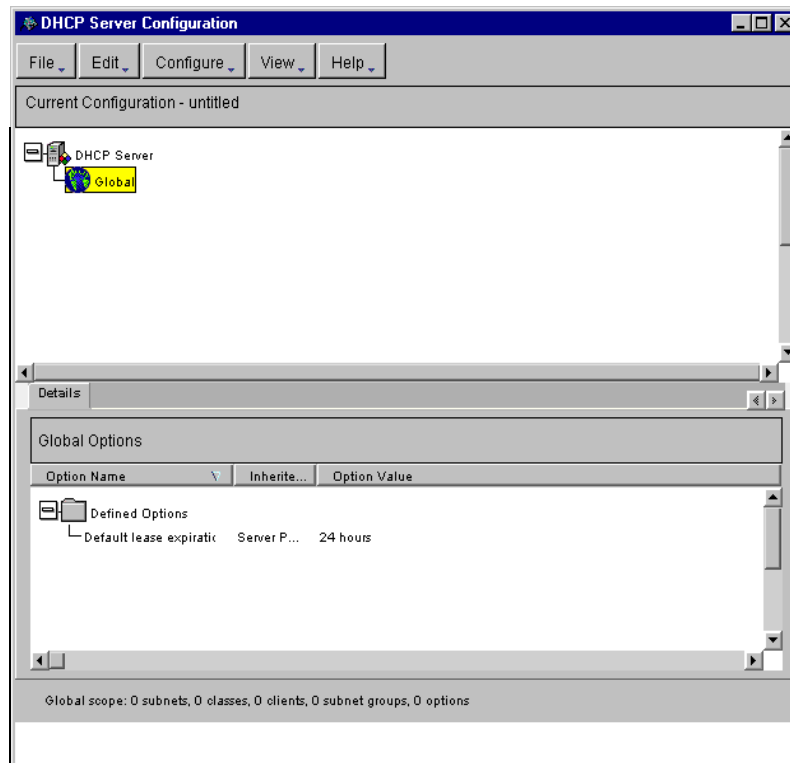


Figure 25. DHCP Server Configuration

4. Most of the default settings work. We just have to change some of them to reflect your network settings.
5. Click **Global**.
6. select **Configure->Add Subnet**. The window that will appear is similar to the one shown in Figure 26 on page 45.



7. Fill in the following information in the appropriate fields. The parameters provided were taken from our test laboratory. Please make the necessary changes to adapt it to your network.

Subnet name: 10.0.0

A descriptive name (can be text or the subnet address).

Subnet Address: 10.0.0.1

The TCP/IP address of the DHCP server.

Subnet mask: 255.255.255.0

A class C subnet mask.

An address range From: 10.0.0.100 to: 10.0.0.200

The IP addresses to be assigned to clients.

The screenshot shows a window titled "Subnet" with four tabs: "Subnet Definition", "DHCP Options", "Miscellaneous", and "Client Support". The "Subnet Definition" tab is active. It contains the following fields and controls:

- Address range for subnet:**
  - Subnet name:
  - Subnet address:
  - Subnet mask:
  - Address range: From:  To:
- Addresses excluded from range:**
  - From:  To:
  - Exclusion list:
- Lease time and comment:**
  - ☒ Default lease time: 24 hours
  - ☐ Enter a lease time:  years  months  days  hours  minutes  seconds
  - ☐ Permanent lease:
  - Comment:

At the bottom right are buttons for "OK", "Cancel", and "Help".

Figure 26. Subnet configuration

8. Configure the DHCP options by clicking the **DHCP Options** tab. The window will be similar to the one shown in Figure 27 on page 47. Do the following:
  - a. Click **Option 1 Subnet Mask**.  
Enter 255.255.255.0 in the space provided.
  - b. Click **Option 3 Router**.  
Enter your gateway address; For example: 10.0.0.1.
  - c. Click **Option 5 Name Server**.  
The IP address of your Domain Name Server (old format).
  - d. Click **Option 6 Domain Name Server**.  
Enter the address of the Domain Name Server.
  - e. Click **Option 12 Host Name**.  
Enter the name of you server.
  - f. Click **Option 15 Domain Name**.  
Substitute the domain name of your organization; we have used austin.ibm.com.
  - g. Click **Option 28 Broadcast Address**.  
For example: 10.0.0.255.

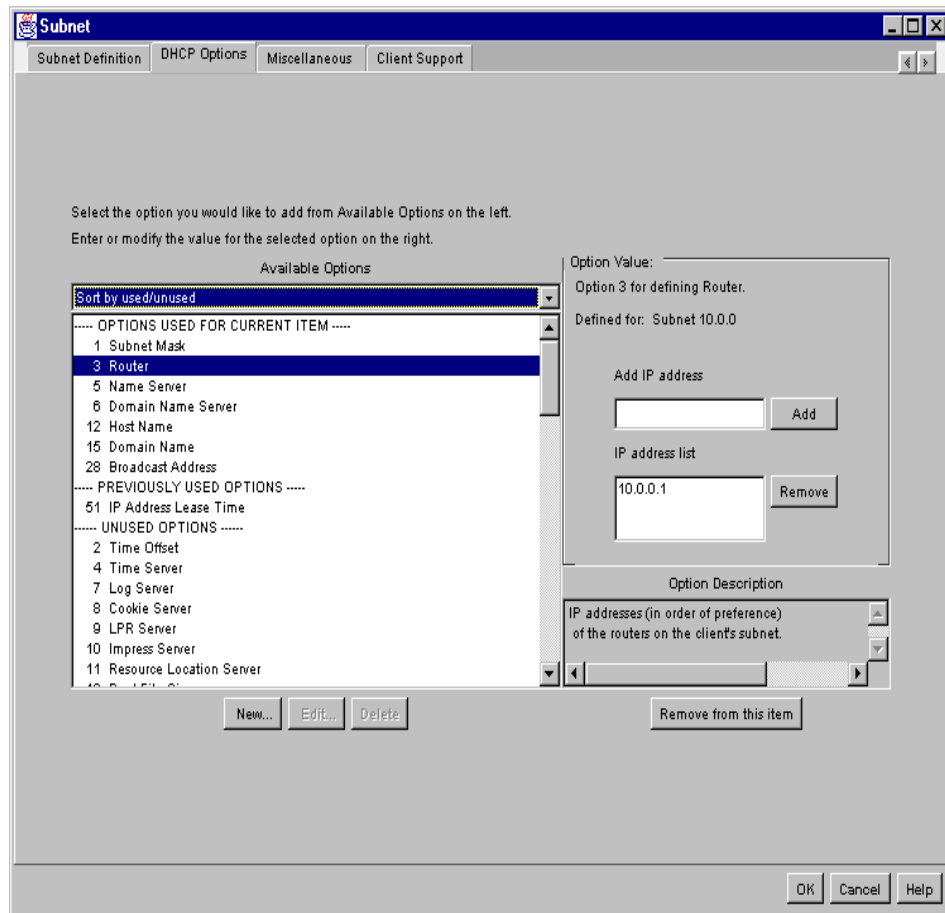


Figure 27. DHCP options

#### Note

A useful reference for this subject is *Beyond DHCP - Work Your TCP/IP Internetwork with Dynamic IP*, SG24-5280/ISBN 0738415073. Available at: <http://www.fatbrain.com>

### 2.7.3 BINL service

The Boot Image Negotiation Layer (BINL) service stores its configuration parameters in a file named BINLSD.CFG, which resides in the \TDM\PROGRAMS\TCPIP\ETC directory. BINLSD.CFG is an ASCII file that can be edited using a normal text editor. A sample BINLSD.CFG file follows:

```

# binlsd.cfg -- BINL (Image Server) Configuration File

# Setup of the log file information. This includes the size and name of
the
# logfile along with number of logfiles maintained and type of
information that
# will be logged.
numLogFiles      0
logFileSize      1000
logFileName      binlsd.log
logItem          SYSERR
logItem          OBJERR
logItem          PROTERR
logItem          WARNING
logItem          EVENT
logItem          ACTION
logItem          INFO
logItem          ACNTING
logItem          TRACE

# default TFTP server
tftp 10.0.0.1

# Fully qualified boot Image pathnmae
bpname MACS\TDMBOOT.SYS

# Global Options
pxevendor
{
    pxecption 128 atlas1
    pxecption 129 10.0.0.1
}

#Excluded client list
client 1 080aab4567 exclude
client 1 000abc45d7 exclude
client 6 080aab4899 exclude

sa 0 nit 1
{
    tftp 10.0.0.1
    bpname MACS\TDMBOOT.SYS
}

sa 0 nit 2
{
    tftp 10.0.0.1

```

```

bpname MACS\TDMBOOT.SYS

#Client excluded from service if it falls into this category.
client 6 08aa343bf3 exclude

# Special configuration for client following into this nit2 type
client 1 12345abcd34
{
    tftp 10.0.0.1
    bpname MACS\TDMBOOT.SYS
}

lsalnic 53
{
    tftp 10.0.0.1
    bpname MACS\TDMBOOT.SYS
}

lsalnic 57
{
    tftp 10.0.0.1
    bpname MACS\TDMBOOT.SYS
}

#
# end of binlsd.cfg

```

#### 2.7.4 TFTP service

The trivial file transfer protocol (TFTP) is a trimmed-down version of the common File Transfer Protocol. Both Microsoft Windows 2000 and Windows NT 4.0 do not, by default, come with TFTP. IBM WorkSpace On-Demand 3.0.1 provides a TFTP service. The boot files are transferred via TFTP from the IBM WorkSpace On-Demand 3.0.1 server to the clients. This version of TFTP service differs somewhat in its security implementation compared to FTP. Instead of using TCP/IP address to authenticate, it uses the client's Windows 2000 or NT user account to login. The configuration is straightforward. The only required parameter is the directory location of the boot files, as shown in Figure 28 on page 50.

Complete the following steps:

1. You can get there by selecting **Start->Programs->IBM WorkSpace On-Demand->IBM TCP\_IP Services->TFTP Configuration**

2. Make sure your C:\TDM\MM\RO is shared. If want to share another directory, you can do it in here. If not, enter the directory on the space provided under Directory Name. Click **Add**.
3. If you want to see who is accessing your TFTP server, you can enable trace, which is basically the logging facility. This can be done by selecting the **Trace** tab. Click the button next to **Enable Trace**.
4. For the changes, you can either stop and start the TFTP server in Windows 2000 or NT services.

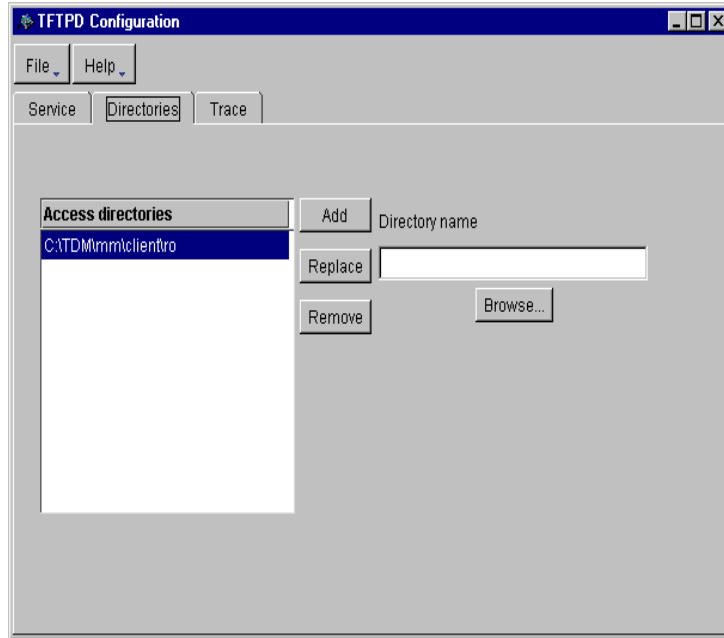


Figure 28. TFTP Configuration

You can also configure this TFTP service to trace and log all file access by clients as shown in Figure 28.

### 2.7.5 Other IP services

There are some other IP services bundled with IBM WorkSpace On-Demand 3.0.1. It is beyond the scope of this book to fully discuss these, but a brief summary follows:

#### IBM NFS Server

NFS stands for Network File System. It is quite common for disk sharing to

occur in a UNIX environment. It is currently not enabled, but it is there for when support for Linux is made available.

#### **IBM PCNFSD Protocol Server**

Many NFS client implementations contain calls to PCNFSD user ID Authentication services. The purpose of these calls is to present a user ID and password to be verified by a host system. As with NFS, it is not currently enabled.

#### **IBM RPC Portmapper**

RPC Portmapper is the service responsible for listening to and redirecting port calls for proper handling.

#### **IBM TCP/IP Print Server**

The print server is the IBM modification of the LPDSRV (Line Printer Daemon) service built into Windows 2000 and NT stack.

### **2.7.6 Protected disk driver**

For Windows 98 client systems, the Protodisk driver is a virtual device driver (\*.VxD) written to the Microsoft Windows 95/98 architecture. The driver registers itself with the operating system during driver initialization, after which all calls to the IFS will go through the function specified to the operating system at driver initialization. For Windows 2000 and NT client systems, the Protodisk device driver is a high-level file system filter driver (\*.SYS) written to the Microsoft Windows NT 4.0 architecture. The driver is loaded during client bootup, where it inserts itself in the device chain above the local file system driver so that calls to the local file system are intercepted by the filter driver before reaching the file system.

The Protodisk driver intercepts all I/O write, copy, move, rename, and delete requests to local drives on the client system and allows those operations to proceed unmolested on drives that have been specified as unrestricted. The driver utilizes a device called the I/O control interface (IOCTL) to control functionality, such as enabling and disabling the restriction functionality, and to pass restriction information to the driver, such as which drives to leave unrestricted and to which files and directories, on restricted drives, write access is explicitly permitted or denied.

The driver will be called by the accompanying PRTDSKCL.EXE executable, which will read a configuration file and communicate information and instructions to the Protodisk driver via the IOCTL interface. One version of the executable will operate with all the versions (2000, NT and 98) of the Protodisk driver.

The Logon Client downloads the Protodisk configuration file to the client system on each boot and also invokes the PRTDSKCL.EXE executable with various command line parameters at the appropriate times to start, stop, and send configuration information to the Protodisk virtual device driver.

You may encounter an application that is required to write application data onto a local hard drive. Protodisk can be configured to allow administrators to specify file names and applications to be given write access.

IBM OS/2 clients use their hard disk for caching purposes only. The swap files are the only ones that are allowed write access.

The Protodisk drivers will be transparent to users. Only sandbox accounts and desktops are allowed to write onto the hard disk.

---

## 2.8 Uninstalling IBM WorkSpace On-Demand 3.0.1

There may be instances where you will want to uninstall the IBM WorkSpace On-Demand 3.0.1 from your Windows 2000 or Windows NT Server.

Most Windows 2000 and NT applications uninstall by using the uninstall feature available within the operating system. Uninstalling an application is a straightforward procedure, but, most of the time, some files and registry entries are left behind.

The product can be uninstalled by following the procedures below.

It is recommended to manually stop the IBM WorkSpace On-Demand 3.0.1 services. Complete the following steps:

1. Select **Start->Settings->Control Panel**.
2. Double-click **Services**.
3. Make sure the following services are stopped. This is done by highlighting the name of the service and clicking the **Stop** button (as shown in Figure 29 on page 53).

IBM BINL Server

IBM DHCP Server

IBM WorkSpace On-Demand 3.0.1 Local Cache Server

IBM WorkSpace On-Demand 3.0.1 Configuration Service

IBM WorkSpace On-Demand 3.0.1 Deployment Service

IBM NFS Server



IBM PCNFSD Protocol Server  
 IBM RPC Portmapper  
 IBM TCP/IP Print Server  
 IBM TPFTP Server  
 Jaaslogon

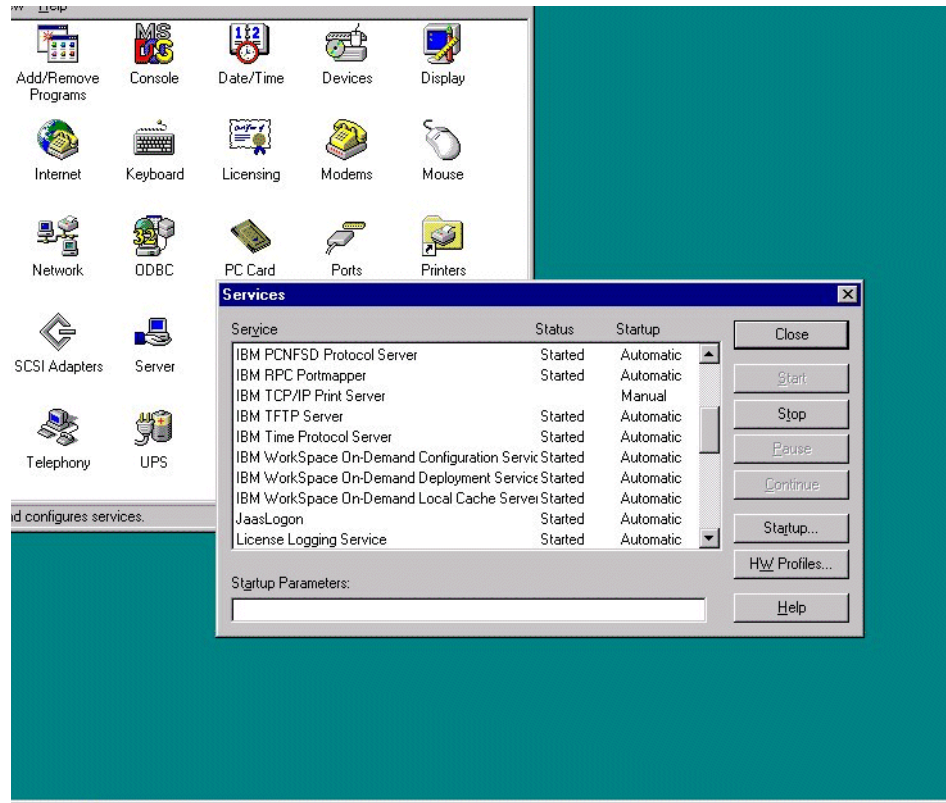


Figure 29. IBM WorkSpace On-Demand 3.0.1 services

Before proceeding with the uninstallation, you should remove all the users accounts defined through the IBM WorkSpace On-Demand 3.0.1. Remember that a user with the machine name has been automatically created at the installation time for each defined machine.

After stopping all services, you can now start uninstalling the IBM WorkSpace On-Demand 3.0.1.

4. Uninstallation is automatically done by Install Shield. Install Shield logs the installation and keeps an \*.ISU file on the main directory of the application (in this case, C:\TDM).

The uninstall program is started by selecting **Start->Settings->Control Panel**.

5. Click **Add/Remove Program** (you should see a window similar to the one shown in Figure 30).
6. Highlight **IBM WorkSpace On-Demand 3.0.1 Client Enablement**.
7. Click the **Add/Remove...** button.

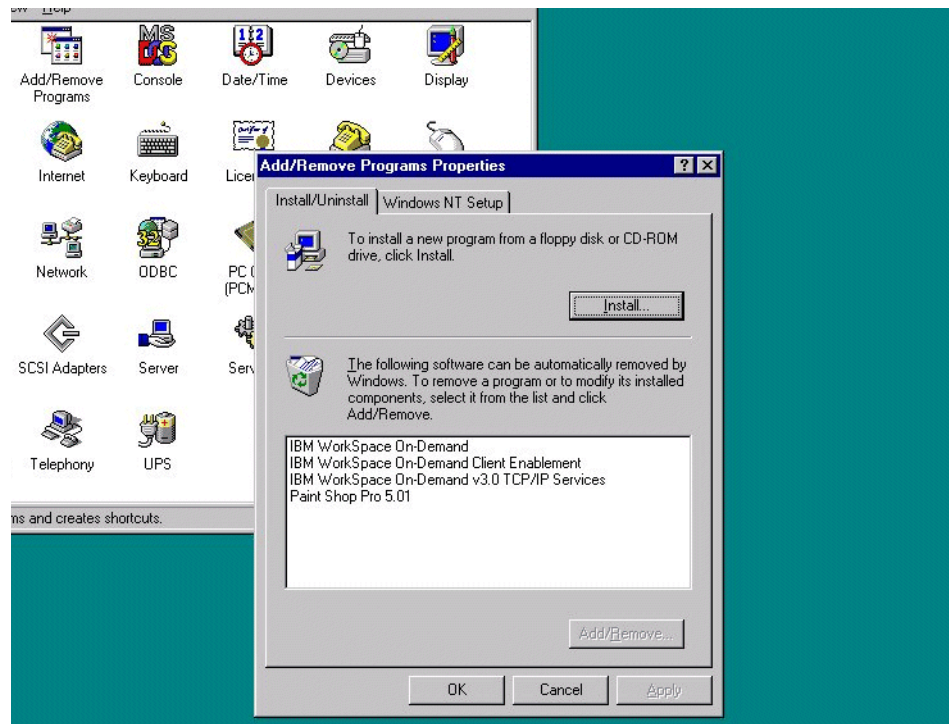


Figure 30. Add/Remove Programs

The uninstallation process will start.

When prompted to reboot at the end of the automated uninstallation routine, click **No**.

8. Repeat steps 4 through 7 for **IBM WorkSpace On-Demand 3.0.1 TCP/IP Servers V1.0**.
9. Repeat steps 4 through 7 for **IBM WorkSpace On-Demand 3.0.1**.

10. The last step in the uninstallation is removing the IBM Intermediate Support Drivers. You can do this by selecting **Start->Settings->Control Panel**.
11. Double-click **Networks**.
12. Go to **Properties**.
13. Click **Protocol**.
14. Highlight **IBM Intermediate Support Driver**.
15. Click **Remove**.
16. Click **Apply**. The networking service of Windows 2000 and Windows NT will do the binding and may need the original Windows 2000 or NT Server CD.
17. Reboot.

IBM WorkSpace On-Demand 3.0.1 should now be fully removed from your system.



---

## Chapter 3. Administration

IBM WorkSpace On-Demand 3.0.1 is administered via an administration console consisting of two parts, a graphical user interface (GUI) and a command line interface (CLI). Both are implemented as Java applications and can be run from any machine with a network connection to a server running IBM WorkSpace On-Demand 3.0.1. Typically, however, the administration console is installed on the same machine where the configuration server resides.

All resources in IBM WorkSpace On-Demand 3.0.1 are managed through the administration console, and all tasks can be accomplished either via the GUI or the CLI. Execution of multiple or repetitive tasks are automated using the JavaScript capability of the CLI. The GUI and CLI share a common communication interface called the command handler. This interface passes commands from the administration console to a configuration server, which in turn invokes a task on the deployment server and responds with return code information to the console.

The GUI and CLI invoke Java methods in the command handler, which in turn uses a remote method invocation (RMI) connection to pass the command to the configuration server. The configuration server resolves the command and executes a configuration task that manipulates object information within the configuration server and results in, for example, additions, deletions, or changes within the data store. The configuration server also executes a transform task that uses an RMI to pass the command on to the deployment server. The deployment server will resolve the command and execute a task (for example: to define a machine.) Return information is passed up to the configuration server and back to the console.

The relationships between these components are shown in Figure 31 on page 58.

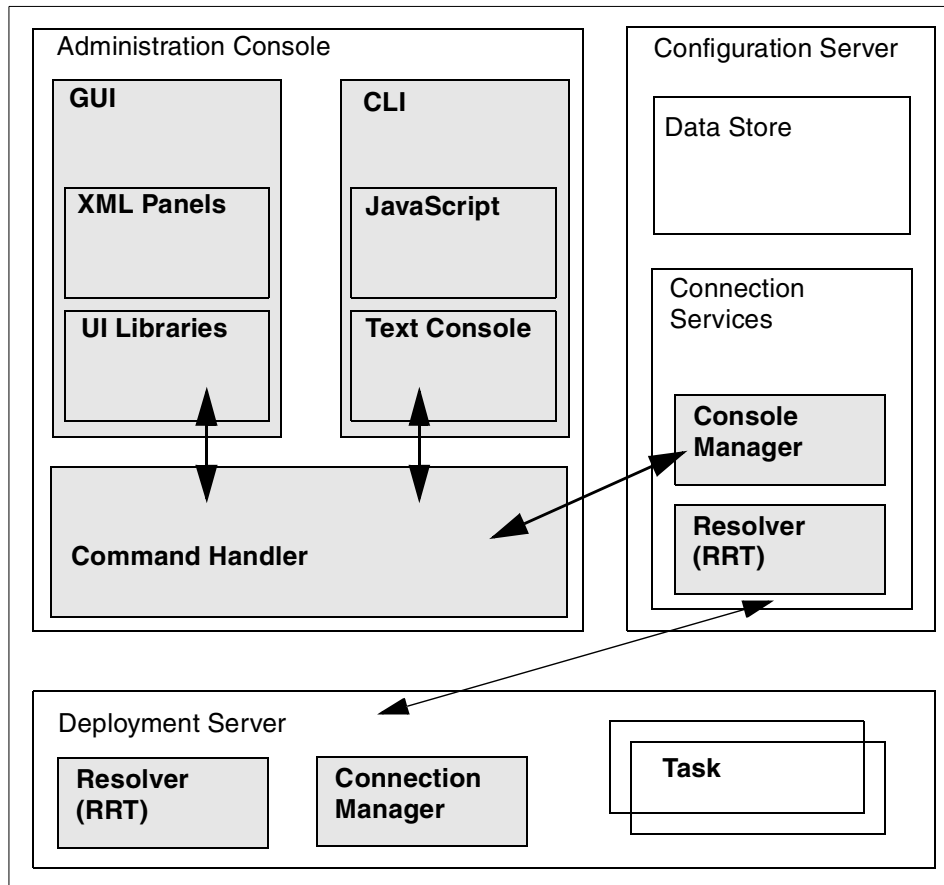


Figure 31. Components of the administration console

### 3.1 Resources and tasks

The resources administered through the console, both GUI and CLI, are:

- Server** Any server in the network that will configure or deploy client machines
- Machine** Any client workstation defined to a specific server
- User** Any user of a client machine who will log on and access applications
- Application** Any application installed on a server or workstation and made available to individual users or groups

**Desktop**      The interface with which a user interacts to access applications or other resources

The following tasks may be performed on all resources, with the exception of Assign, which is valid only for User and Server resources.

|               |   |
|---------------|---|
| <b>Define</b> | Creates a new instance of the resource.   |
| <b>Delete</b> | Removes a specific instance of the resource.  |
| <b>List</b>   | Enumerates all instances of a resource.   |
| <b>Query</b>  | Determines the characteristics of a particular resource.  |
| <b>Modify</b> | Changes the characteristics of a particular resource.   |
| <b>Assign</b> | Assigns a desktop or application to a user, or assign operating system boot function to a server. |

The following sections describe the use of these resources and how typical tasks are performed. Automation of common and repetitive tasks is also covered.

---

## 3.2 The administration GUI

The administration GUI implements the IBM Common Systems Administration (CSA) standard as shown in Figure 32 on page 60. Panels presented by the GUI are defined in XML.

The standard CSA GUI is in the form of a split panel, with a navigation area on the left and a task area on the right. The IBM logo just below the title bar on the GUI is a standard CSA component that functions as a hot spot, which, when selected, will link the user to the IBM home page: <http://www.ibm.com>

The action bar underneath the task area lists the available actions at any point while navigating the GUI. The available tasks will vary depending on the type of resource that is currently selected. In addition, three standard tasks are always available. These are:

|               |   |
|---------------|---|
| <b>Cancel</b> | End the current task                          |
| <b>Edit</b>   | Provide cut and paste capability              |
| <b>Tips</b>   | Provide help for the current task or resource |

Underneath the action bar, a message line presents messages to the user to indicate the success or failure of an action.

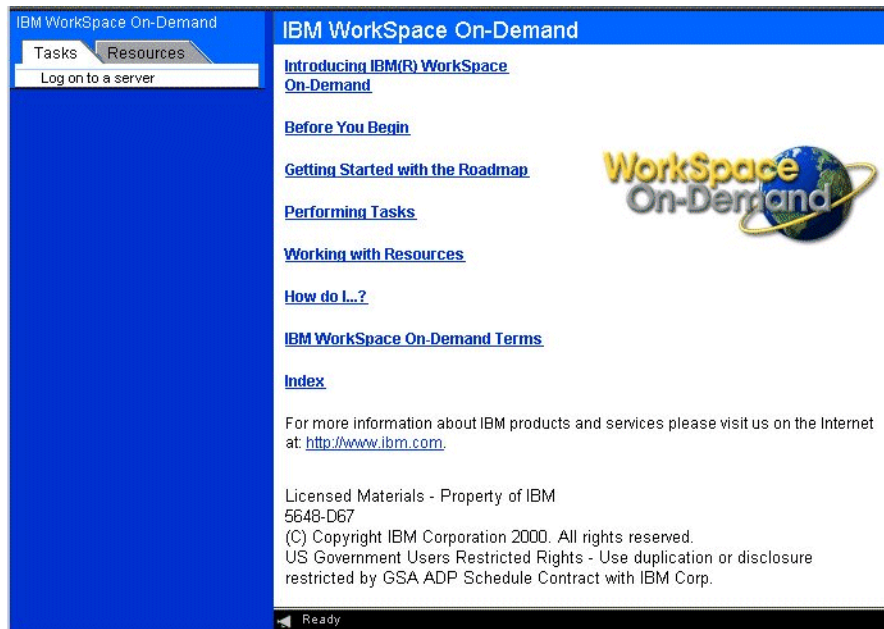


Figure 32. IBM WorkSpace On-Demand 3.0.1 administration GUI

In IBM WorkSpace On-Demand 3.0.1, the navigation area of the GUI presents two categories via tabs. These are **Tasks** and **Resources**. Click the **Tasks** tab to perform a specific job, such as defining or deleting a new user or machine. When a task is selected, the task area on the right presents a wizard-style sequence of panels to complete the task.

When you select the Resource tab, the GUI presents a list of all resources that can be managed. Selecting a specific type of resource, such as Users, will populate the task area on the right with all currently defined resources of that type. You may select single or multiple resources in the list, and the action bar will display the tasks that are valid for the selected resources.

### 3.3 Task flow

Before managing any resources or performing any tasks, the administrator logs on to a configuration server through the GUI. Since any server on the network could be administered, the logon window allows you to specify the server and the TCP/IP port used for a connection, as shown in Figure 33 on page 61. You must log on with a user ID that has administrator authority for the selected server. The password for the administrator user ID cannot be



blank. Note that no other tasks or resources are available until logon has succeeded.

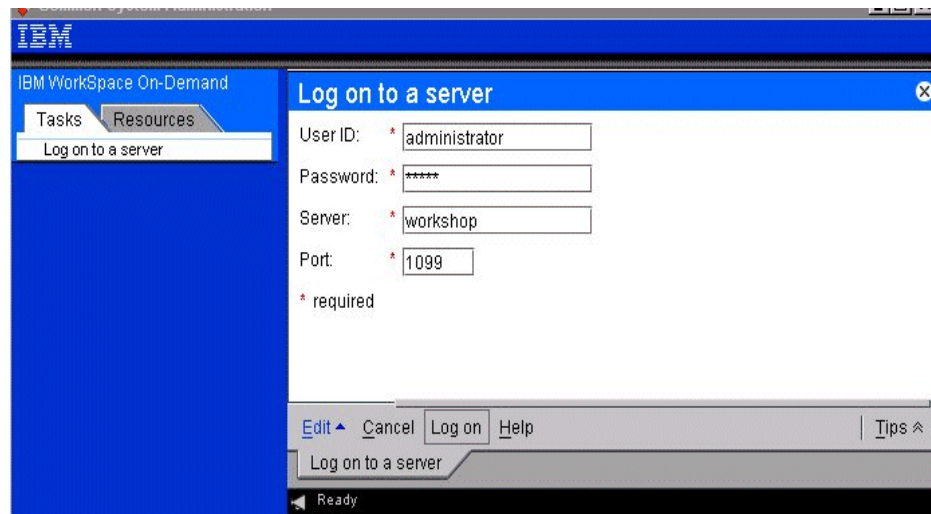


Figure 33. Logging on to a server from the administration GUI

The configuration server might have been updated since the last time it was accessed via the administration GUI and, thus, could contain updates for the administration GUI. As part of the logon process, the administration GUI will automatically retrieve and install new panels from the configuration server.

Logging on to the configuration server establishes the remote method invocation (RMI) communication link between the command handler and the server. Each successful logon creates a new instance of the ConsoleManager object within the configuration server.

It is possible for the administrator to be logged on to more than one server at a time by launching multiple instances of the GUI. It should be noted that each GUI runs in its own JVM. This gives you the flexibility to manage multiple servers in a domain or network.

When the logon process is complete, you may execute commands against the server. Commands are passed from the GUI to the configuration server via the command handler. In the GUI, a command is represented by a panel or sequence of panels. Pressing the **OK** button on the last panel of a sequence invokes a Java method that extracts all necessary information, forms a command, and sends the command to the command handler.

The command handler communicates with the ConsoleManager object in the configuration server via the RMI communication link. The command is passed to the ConsoleManager object. ConsoleManager provides a method called `submitCommand`, which is invoked by the command handler. Then, the command is executed in the configuration server. The results from the command are stored in an object of type `TDMConsoleResult`, which is passed back over the RMI link to the command manager.

When the GUI receives the `TDMConsoleResult` object, it checks a status field, which will contain one of two values: `SUCCESSFUL` or `FAILED`. The GUI then updates its own status field with the message: `Last action completed successfully` or `Last action failed`. In the case of a successful return, the GUI will retrieve additional information from text fields in `TDMConsoleResult` and display this information in a panel. For example, if a new user was successfully defined, the user name and description will be added to a list displaying all defined users. In the case of a failed command, the `TDMConsoleResult` text strings are used to display a message box with an icon indicating the severity of the failure.

---

### 3.4 The administration CLI

The administration GUI provides an easy and intuitive way to perform all commands on a configuration server. However, the GUI is not efficient for execution of many sequential commands or more complex tasks. It is best used for simple isolated tasks, such as adding a new user to an existing configuration. For complex tasks or for the experienced administrator, the command line interface (CLI) will prove to be more useful.

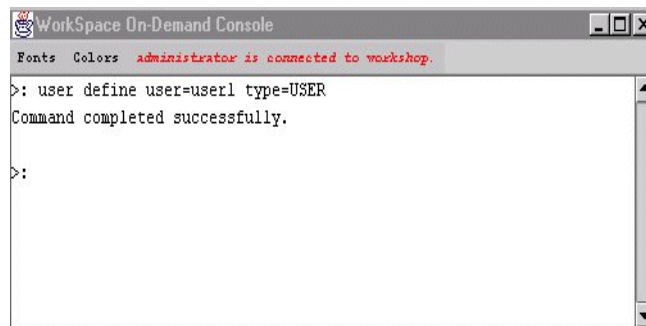


Figure 34. The administration CLI

The administration CLI runs as a Java application presenting only a command prompt to the administrator as shown in Figure 34. When a command is

entered, it is parsed, and if it is syntactically correct, it is passed via a Java method to the command handler for processing. Just as in the case of the administration GUI, the command handler passes the command over the RMI link to the ConsoleManager object in the configuration server. The configuration server executes the command, and the results are passed back to the console via the TDMConsoleResult object.

The console window allows scrolling and retrieval of previously entered commands via the up and down arrow keys. Command output can also be paused or redirected to a file via the `>` and `>>` operators.

Just as in the administration GUI, the administration CLI also requires the administrator to log on to a configuration server prior to executing any commands. This is done via the `connect` command. When the administrator types `connect` at the CLI command prompt, a dialog is displayed to log on to a configuration server as shown in Figure 35 on page 64. A successful logon causes the RMI link to be established to the ConsoleManager object in the configuration server.

Logons to multiple servers are supported through multiple `connect` commands. You can specify either the name or the IP address of the server to which you want to connect. You can log off from any server you are connected to by entering the `disconnect` command and specifying the name of the server. Once disconnected, you can reconnect to a server by using the `connect` command again. However, you will have to provide the authentication information again. This ensures that if the same console is used by multiple administrators, there is no risk of accidental authentication with a previous administrator's logon information to a given server.

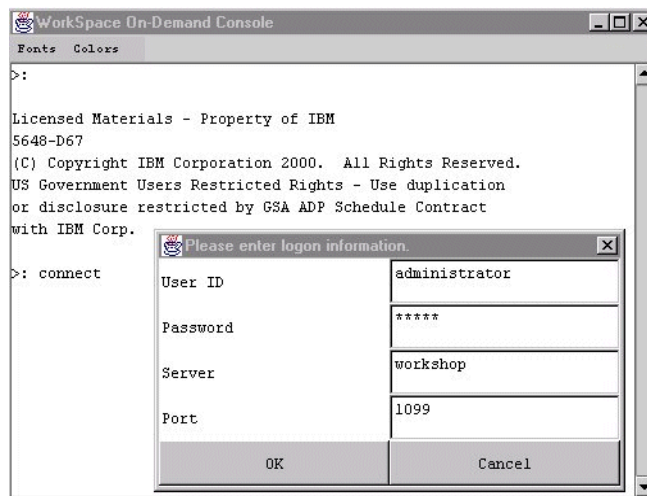


Figure 35. Logging on to a server from the CLI

When you log on to multiple servers from the same console, the console sessions are stacked so that you will only work with the last server to which you logged on. When you disconnect from the current session, you are returned to the last connected session. To manage multiple servers simultaneously, it is easier to open multiple CLI windows and log on to one server from each console.

After logon, any valid commands may be entered in the CLI. The CLI handles commands in an object-oriented fashion. The general syntax for a command is: `object task [parameter] [parameter] ...`. Two commands are shown in Figure 36 on page 65. Note that some commands may have multiple parameters, while other commands may have no parameters. There are three types of parameters for CLI commands as shown in Table 1. Specific commands and their relevant parameters will differ depending on the target object of the command. The most common objects, commands, and parameters will be described later in this chapter as well as the following three chapters of this book.

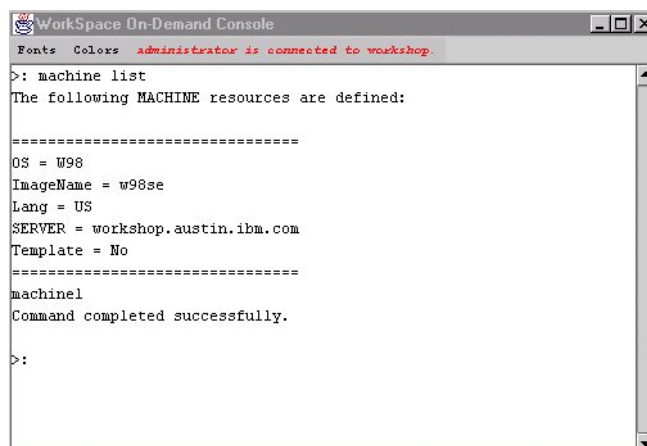
Table 1. Types of command parameters

| Parameter      | Meaning   | Example                          |
|----------------|---|----------------------------------|
| Distinguishing | Servers to distinguish the object target of the command | OS=NT4 , ImageName=SP3 , Lang=US |
| Required       | Must be supplied with the command                       | MAC=00203542986f                 |

| Parameter | Meaning                       | Example             |
|-----------|-------------------------------|---------------------|
| Optional  | Can be left at default values | Description=" ... " |

IBM WorkSpace On-Demand 3.0.1 allows you to extend the command set using JavaScript. Techniques for automation of commands via JavaScript are covered in Section 3.5, “Command automation” on page 66. There are, in fact, three types of commands that can be executed in the CLI console. First, the commands shown in Figure 36 are examples of direct IBM WorkSpace On-Demand 3.0.1 commands. These are passed directly via the command handler to the configuration server for processing. Second, JavaScript instructions can be invoked from the CLI. The console interprets the JavaScript and passes any embedded commands to the command handler for processing.

Third, there are console commands that are used to interact with the console only and not passed on to the configuration server. The commands `connect` and `disconnect` described above are examples of console commands. The order of command processing is as follows: When a command is entered, it is first checked against the list of console commands. If it is not a console command, it is checked to see if it is a valid JavaScript function and will be executed as such. If it is neither a console command nor JavaScript, it will be executed as a direct IBM WorkSpace On-Demand 3.0.1 command.



```

Workspace On-Demand Console
Fonts Colors administrator is connected to workshop.
>: machine list
The following MACHINE resources are defined:
=====
OS = W98
ImageName = w98se
Lang = US
SERVER = workshop.austin.ibm.com
Template = No
=====
machinel
Command completed successfully.
>:

```

Figure 36. Sample commands and returned information

The following list describes the available console commands:

- Connect**      Log on to a configuration server.
- Disconnect**      Log off from a configuration server.

|             |  |
|-------------|--|
| <b>Load</b> | Load a JavaScript file into the CLI. It will be syntax-checked but not executed. After loading, its functions are available for execution from the command line or from other scripts. |
| <b>List</b> | Lists the names of all script files that have been loaded during the current session.  |
| <b>Help</b> | Display general help or help for a specific command.   |
| <b>Exit</b> | Terminate the console session.   |

---

### 3.5 Command automation

The CLI provides a quick way to enter simple commands. However, some commands, such as `machine define`, may have more than 20 parameters and typing these in manually is very cumbersome. Therefore, the CLI can take advantage of JavaScript to help you simplify commands and automate procedures. The IBM WorkSpace On-Demand 3.0.1 implementation of JavaScript is based on the Rhino engine developed by Mozilla. More information about Mozilla and Rhino is available on the Internet at:

<http://www.mozilla.org/rhino>

There are also many good JavaScript references available as books or on the Internet.

In this chapter, we investigate various ways of using JavaScript to extend the power of the IBM WorkSpace On-Demand 3.0.1 CLI.

#### 3.5.1 Simplifying commands using JavaScript

As previously mentioned, many commands in IBM WorkSpace On-Demand 3.0.1 have a large number of parameters, each of which must be specified. Typing such commands directly at the console would be extremely cumbersome and error-prone. One very simple use of JavaScript is to encapsulate complex commands where the values for many of the parameters would normally stay constant.

For example, Figure 37 on page 67 shows the command syntax used to define a new machine. The details of machine definition are explained in Chapter 4, “Defining machines” on page 103; for now, it is sufficient to note that there are many parameters that must be specified. However, in a large enterprise environment, there will be many machines deployed of exactly the same type and defining each of these machines in IBM WorkSpace On-Demand 3.0.1 means that many of the command parameters would be the same for all machines.

```
Machine define OS='NT4' NETADAPTER='TRP' DHCP='Yes' LANG='us'  
PROTOCOLS='TC,NBF' DESCRIPTION='ITSO Test' RESOLUTION='800x600x256@70'  
STATUS='Enabled' JVM='Yes' LOGON='Yes' TMA='No' IMAGENAME='sp3'  
VIDEO='VGA' CDKEY='xxxxx-oem-xxxxxxx-xxxxx' TZ='(GMT-06:00) Central  
Time (US & Canada)' PREBOOTIMAGE='tdmw32ut.img' INSTALLDOS='No'  
LOCALADMPW='getin2see' partition='400' REGUSER='ITSO'  
srcrespfile='ibmfepci.txt' Name='test1' MAC='002035a849cf'  
SERVER='atlas2'
```

*Figure 37. A complex command - Defining a machine*

It is easy to define a JavaScript function that can encapsulate this command and expose only those parameters that are likely to be changed. When defining machines, we know that each machine's network adapter has a unique MAC address. We also know that each machine must have a unique name, and we might further assume that we want the flexibility to define machines with different operating system images—say, either Windows 2000, Windows NT or Windows 98.

Since we are able to manage multiple servers at a time, let's also add the ability to specify on which server we want to define a given machine. The resulting JavaScript is shown in Figure 38 on page 68.

```

function CreateMachinet (Name, Mac, Server,Os)
{
    var NAME=" Name='"+Name+"' ";
    var MAC="MAC='"+Mac+"' ";
    var SERVER="SERVER='"+Server+"' ";
    if (Os=="")
    {
        Os=="W2K"
    }
    if ((Os=="w2k") || (Os=="W2K") || (Os=="W2k") || (Os=="w2K") )
    {
        var cmd ="Machine define OS='W2K' DHCP='Yes' LANG='us' DESCRIPTION='ITSO Test' STATUS='Enabled'
        JVM='Yes' LOGON='Yes' TMA='No'  IMAGENAME='w2kpro' CDKEY='xxx-xxxxxx' TZ='Central America'
        PREBOOTIMAGE='tdnw32ut.img' INSTALLDOS='No' partition='600' srcrespfile='ibmfepci.inf' orgname='ITSO'
        protocol='MS_TCPIP' NETADAPTER='TRP'"
    }
    else if ((Os=="nt4") || (Os=="NT4") || (Os=="Nt4") || (Os=="nT4") )
    {
        var cmd ="Machine define OS='NT4' NETADAPTER='TRP' DHCP='Yes' LANG='us' PROTOCOLS='TC,NBF'
        DESCRIPTION='ITSO Test' RESOLUTION='800x600x256@70' STATUS='Enabled' JVM='Yes' LOGON='Yes'
        TMA='No' IMAGENAME='sp3' VIDEO='VGA' CDKEY='xxxxx-oem-xxxxxxx-xxxxx' TZ='(GMT-06:00)
        Central Time (US & Canada)' PREBOOTIMAGE='tdnw32ut.img' INSTALLDOS='No'
        LOCALADMPW='getin2see' partition='400' REGUSER='ITSO' srcrespfile='ibmfepci.txt'"
    }
    else if ((Os=="w98") || (Os=="W98"))
    {
        var cmd ="Machine define OS='W98' DHCP='Yes' LANG='us' DESCRIPTION='ITSO Test' STATUS='Enabled'
        JVM='Yes' LOGON='Yes' TMA='No'  IMAGENAME='w98se' CDKEY='xxxxx-xxxxx-xxxxx-xxxxx-xxxxx'
        TZ='Central' PREBOOTIMAGE='tdnw32ut.img' INSTALLDOS='No' partition='600' srcrespfile='ibmfepci.inf'
        reguser='test98'"
    }
    else
    {
        exitjs(0) ;
    }
    cmd=cmd+" "+NAME+" "+MAC+" "+SERVER;
    println(cmd);
    var result = execute (cmd);
}

```

Figure 38. Encapsulation of a command

The function `CreateMachinet()` shown in Figure 38 allows us to create either a Windows 2000, a Windows NT or a Windows 98 client by dynamically building the appropriate IBM WorkSpace On-Demand 3.0.1 command. The function `execute()` passes the command to the command handler to be processed by the configuration server. The returned value from `execute()` is an object of type `TDMConsoleResult`. In our example, we built a second function called `printResult()` to handle this object. This function is shown in Figure 39 on page 69.



```

function printResult (theResultObj)
{
    // If TDMConsoleResult is not null and is successful, print
    // its content.
    if (theResultObj != null)
    {
        var ctr;

        for (ctr=0; ctr < theResultObj.data.length; ctr++)
        {
            // Print any messages returned with the result.
            println (theResultObj.data[ctr]);
        }

        // Indicate if the command was successful.
        if (0 == theResultObj.resultCode)
        {
            println (theResultObj.resultCode);
            println ("Command completed successfully.");
        }
        else
        {
            println (theResultObj.resultCode);
            println ("Command did not complete successfully.");
        }
    }
}

```

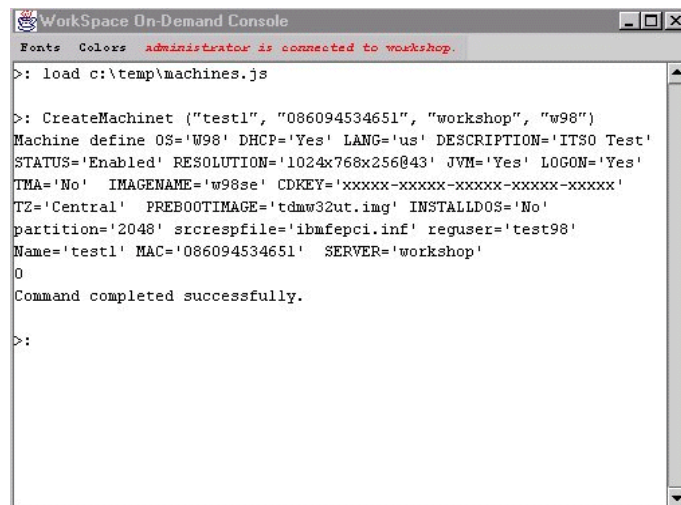
*Figure 39. Handling a TDMConsoleResult object*

Notice that the `TDMConsoleResult` object contains two parts. The first is an array of strings called `data` which contain any information returned from the command. A `list` command such as `machine list`, for example, would contain a separate array element in `data` for each machine listed. In our example, the array is empty. The second part of `TDMConsoleResult` is a numeric value, `resultCode`, which indicates the success or failure of the command. A value of 0 in `resultCode` indicates success, while non-zero values indicate failure.

These two functions also illustrate JavaScript extensions implemented in IBM WorkSpace On-Demand 3.0.1. The functions `execute()`, `print()`, `println()`, and `exitjs()` are all implemented in the file `jsextend.js`, which can be found in `\tdm\programs\tdmcli`. This file is automatically loaded when the CLI is started so that the functions are always available to you.

`Execute()`, as we have seen, passes an IBM WorkSpace On-Demand 3.0.1 command to the command handler for processing. `Print()` and `println()` print information in the CLI window; `println()` adds an extra line feed character at the end of the print. The `exitjs()` function causes the CLI console to terminate, and the numeric argument to `exitjs()` is passed back to the calling program that launched the CLI. We examine this feature in more detail in Section 3.5.3, “Automating the console” on page 73. One additional function implemented in `jsextend.js` is `loadfile()`, which allows one JavaScript file to load another in order to use its functions.

To make the function `CreateMachinet()` available, the administrator only has to issue the `load` command in the console CLI. This will bring the functions in the specified JavaScript file into the CLI so that they can be invoked from the command prompt. You can then call a function by specifying its name and parameters, where the parameters are enclosed in quotation marks and separated by commas as shown in Figure 40. When you load a JavaScript file, you need to specify the fully-qualified path to the file unless it is located in the same directory as the console CLI itself, namely `\tdm\programs\tdmcli`. You can also have commands automatically loaded into the console CLI by editing them into `jsextend.js`, which is automatically loaded when the CLI starts. However, if you reinstall IBM WorkSpace On-Demand 3.0.1, make sure to back up your copy of `jsextend.js` first.



```
Workspace On-Demand Console
Fonts Colors Administrator is connected to workshop.
>: load c:\temp\machines.js

>: CreateMachinet ("test1", "086094534651", "workshop", "w98")
Machine define OS='W98' DHCP='Yes' LANG='us' DESCRIPTION='ITS0 Test'
STATUS='Enabled' RESOLUTION='1024x768x256@43' JVM='Yes' LOGON='Yes'
TMA='No' IMAGE_NAME='w98se' CDKEY='xxxxx-xxxxx-xxxxx-xxxxx-xxxxx'
TZ='Central' PREBOOTIMAGE='tdmw32ut.img' INSTALLDOS='No'
partition='2048' srcrespfile='ibmfepci.inf' reguser='test98'
Name='test1' MAC='086094534651' SERVER='workshop'
0
Command completed successfully.

>:
```

Figure 40. Executing a JavaScript function

### 3.5.2 Automating repetitive commands

Simplifying the execution of a single command using JavaScript is a good start; however, it would be even better to be able to automate the execution of multiple instances of a command. For example, it might be useful to create multiple machines on a single server based on a list of known MAC addresses and using a well-defined convention for naming the machines. Using some very simple JavaScript, we added another wrapper function around the `CreateMachinet()` function that would define machines from an array of MAC addresses. This function, `CreateMultiple()`, is shown in Figure 41.

```
var maclist = new Array("000000000000",
                        "111111111111",
                        "222222222222",
                        "333333333333",
                        "444444444444",
                        "555555555555")

function CreateMultiple ()
{
    var ctr;
    var thisname;
    var prtstring;

    for (ctr=0; ctr < maclist.length; ctr++)
    {
        thisname = "test"+ctr;
        prtstring = "Creating machine "+thisname+" with mac address"+maclist[ctr];
        print (prtstring);

        CreateMachinet (thisname, maclist[ctr], "atlas2", "nt4");
    }
}
```

Figure 41. Creating multiple machines in one function

`CreateMultiple()` can be used to define as many machines as you want, as long as the MAC addresses for all machines are known. This restriction is discussed in Section 3.5.3, “Automating the console” on page 73.

`CreateMultiple()` relies on the `new()` operator in JavaScript to define a new array object that contains the MAC addresses. Note also that JavaScript is an untyped language, meaning that variables have no predefined type at time of initialization. This simplifies the process of building unique machine names

because we can use the + operator to concatenate a text string with a numeric value.

The next example of repetitive automation is more generic. It is often useful to query an object and then use the object's attributes in another function. This allows manipulation of the object without the administrator knowing anything about the object beforehand. For example, to delete a machine definition, you need to specify not just the machine name, but also its operating system, image name, and language. The JavaScript function shown in Figure 42 will delete all machines on a server without the administrator needing to specify any characteristics of those machines.

```
function DeleteAllMachines ()
{
    var cmd, result, ctr, cmdDelete, result2;
    var ctrOS, ctrImage, ctrLang, ctrServer, ctrName, ctrDesc;
    var cmdOS, cmdImage, cmdLang, cmdName;

    cmd = "machine list";
    result = execute (cmd);

    for (ctr = 0; ctr < result.data.length; ctr++)
    {
        ctrOS = result.data[ctr].indexOf("OS");
        ctrImage = result.data[ctr].indexOf("Image");
        ctrLang = result.data[ctr].indexOf("Lang");
        ctrServer = result.data[ctr].indexOf("SERVER");
        ctrName = result.data[ctr].indexOf('"Name"');
        ctrDesc = result.data[ctr].indexOf("Desc");

        cmdOS = result.data[ctr].substring(ctrOS, ctrImage-3);
        cmdImage = result.data[ctr].substring(ctrImage, ctrLang-3);
        cmdLang = result.data[ctr].substring(ctrLang, ctrServer-3);
        cmdName = result.data[ctr].substring(ctrName+1, ctrDesc-3);

        cmdDelete = "machine delete "+cmdName+" "+cmdOS+" "+cmdImage+" "+cmdLang

        print (cmdDelete);
        result2 = execute (cmdDelete);
        printResult(result2);
    }
}
```

Figure 42. Generic machine deletion

The function `DeleteAllMachines()` relies on the `machine list` command in order to determine all the machines to be deleted. This is a good example of a command that returns information via the `TDMConsoleResult` object. The data lines in `TDMConsoleResult` after calling `machine delete` are shown in Figure 43. Each machine is described by a single line of text, although the lines are wrapped in the figure.

```
"OS=NT4", "ImageName=sp3", "Lang=US", "SERVER=ATLAS2", "Template=No", "Name=
test0", "Description=ITSO Test "
"OS=NT4", "ImageName=sp3", "Lang=US", "SERVER=ATLAS2", "Template=No", "Name=
test1", "Description=ITSO Test "
"OS=NT4", "ImageName=sp3", "Lang=US", "SERVER=ATLAS2", "Template=No", "Name=
test2", "Description=ITSO Test "
"OS=NT4", "ImageName=sp3", "Lang=US", "SERVER=ATLAS2", "Template=No", "Name=
test3", "Description=ITSO Test "
"OS=NT4", "ImageName=sp3", "Lang=US", "SERVER=ATLAS2", "Template=No", "Name=
test4", "Description=ITSO Test "
"OS=NT4", "ImageName=sp3", "Lang=US", "SERVER=ATLAS2", "Template=No", "Name=
test5", "Description=ITSO Test "
```

Figure 43. `TDMConsoleResult` values for the `machine list` command

Using the standard JavaScript methods `substring()` and `indexOf()`, we could extract the relevant information for each machine from its associated data string and dynamically build the `machine delete` command.

The functions `CreateMultiple()` and `DeleteAllMachines()` show good examples of how to extend the function of the IBM WorkSpace On-Demand 3.0.1 CLI. It would be easy to provide similar functions to wrap many other commands and thereby automate a large portion of the server administration.

A further capability of this automation is to redirect the output from any command to a file on the hard drive. The CLI supports the redirect operators “>” and “>>” with the same meanings as in a normal Windows command prompt, namely redirect and overwrite the file or redirect and append to a file. If, instead of entering `DeleteAllMachines()` in the CLI, you were to enter `DeleteAllMachines() > c:\delete.log`, then all the output would be recorded in the file and nothing would be displayed in the CLI.

### 3.5.3 Automating the console

One further automation feature supported by the IBM WorkSpace On-Demand 3.0.1 CLI is the ability to automate the starting and stopping of the CLI itself. The CLI is launched from a batch file installed with IBM

WorkSpace On-Demand 3.0.1 called `tdmcli.bat`. `Tdmcli.bat` sets up classpath and other environment variables before launching the CLI and also accepts arguments that can be passed to the CLI upon startup.

The command line syntax for launching `tdmcli.bat` is as follows:

```
tdmcli [server] /u[serid] :userid /p[assword] :password /s[cript] : "javascript  
[function]" /h[elp] /?
```

Not all parameters need be specified, and the parts enclosed in square brackets are optional.

- |                  |   |
|------------------|---|
| <b>server</b>    | Specifies the name of the configuration server to which this CLI session will log on. It can be specified as an IP address or as text.  |
| <b>userid</b>    | Specifies the administrator user ID used to log on to the configuration server.   |
| <b>password</b>  | The password for the administrator user ID.   |
| <b>script</b>    | Specifies a JavaScript that will be loaded when the console is started. Additionally, the name of one function within the script can be specified and this function will automatically be executed when the script is loaded. |
| <b>help or ?</b> | Displays help for the CLI.  |

The batch code in Figure 44 shows a simple example of launching the CLI. In this case, the CLI is automatically loaded with the JavaScript file `example.js`, and the function `CreateMachinet` is invoked. Using the redirection operator from the CLI, the output from `CreateMachinet` is logged to a file and could be read later to perform conditional error handling if necessary.

```
c:  
  
cd \TDM\programs\TDMCLI  
  
call tdmcli.bat atlas2 /u:administrator /p:xxx  
/s:"c:\examples\example.js CreateMachinet('test9', '123456789019',  
'atlas2', 'NT4') > c:\examples\create.log"  
  
cd\examples  
  
REM Read log file and continue
```

Figure 44. Launching the CLI from a batch file

The `CreateMachinet` function is listed in Figure 38 on page 68. One small change is necessary for this function to work correctly in this example. Notice that when `CreateMachinet` successfully creates a machine, the function ends and control is returned back to the CLI. In order to return control all the way back to the batch file, we need to also exit the CLI. This is accomplished by adding the `exitjs()` command at the end of `CreateMachinet` immediately following the call to `PrintResult()`. `Exitjs()` also allows us to pass a return code back to the calling batch program.

Launching the CLI from an external program is a powerful extension to the automation of tasks in IBM WorkSpace On-Demand 3.0.1. Since starting the CLI involves some amount of overhead (to start the JVM), it is best to launch JavaScript functions that do as much work as possible within the same session. To extend the capability of the JavaScript automation further, it would be useful to provide file input capability to the JavaScript functions. One example would be to create multiple machine definitions based on a file that contains a list of MAC addresses.

Although file output is supported through redirection, the JavaScript implementation in IBM WorkSpace On-Demand 3.0.1 does not support file input functions directly. There is no support for a file object, so JavaScript functions are not able to perform any kind of file operations. However, by combining file operations done by external programs or batch files with the ability to automatically launch the CLI console, we can provide sufficient file-based input to JavaScript functions to perform complex tasks within the CLI.

The key is to use file operations in programs outside the CLI to generate syntactically correct JavaScript variable definitions and dynamically add these to a file containing the JavaScript functions before loading it into the CLI. In fact, the JavaScript functions could themselves be dynamically generated if necessary.

As an example, consider the JavaScript code used to create multiple machine definitions shown in Figure 41 on page 71. An external program could be used to collect the data and write a file containing just the array definition of all MAC addresses. We named this file `maclist.txt` and kept the JavaScript functions in a separate file called `functions.txt`. Then, we used a simple `copy` command to combine the two files prior to launching the CLI. The resulting batch script is shown in Figure 45 on page 76.

```
c:
cd \examples

copy maclist.txt+functions.txt multi.js /b

cd \TDM\programs\TDMCLI

call tdmcli.bat atlas2 /u:administrator /p:xxx /s:"c:\examples\multi.js
CreateMultiple () "
```

*Figure 45. Dynamically creating and running JavaScript*

In the `copy` command, we used the `+` operator to append the two text files together. We found that when the files were appended, Windows added an explicit end-of-file character that was not interpreted correctly by the CLI when we loaded the JavaScript. To avoid this, we added the `/b` parameter on the `copy` command that will copy the files as binary without adding any control characters.

---

### 3.6 Administering users

IBM WorkSpace On-Demand 3.0.1 provides a set of tools for administering users within the Windows 2000 and Windows NT Server domains. Users may be added, deleted, or modified through both the administration GUI and the administration CLI. Once a user has been defined in IBM WorkSpace On-Demand 3.0.1, the user may be assigned one or more applications and desktops. Applications and desktops are discussed in more detail in Chapter 6, “Applications” on page 259 and Chapter 7, “Updating the client images” on page 363 of this book.

There is also interaction between IBM WorkSpace On-Demand 3.0.1 and the Windows 2000 or Windows NT User Administration. Figure 46 on page 77 shows the Windows NT administration view of users and groups after the installation of IBM WorkSpace On-Demand 3.0.1. There are several users and groups created as a result of the installation and two points are important to note here.



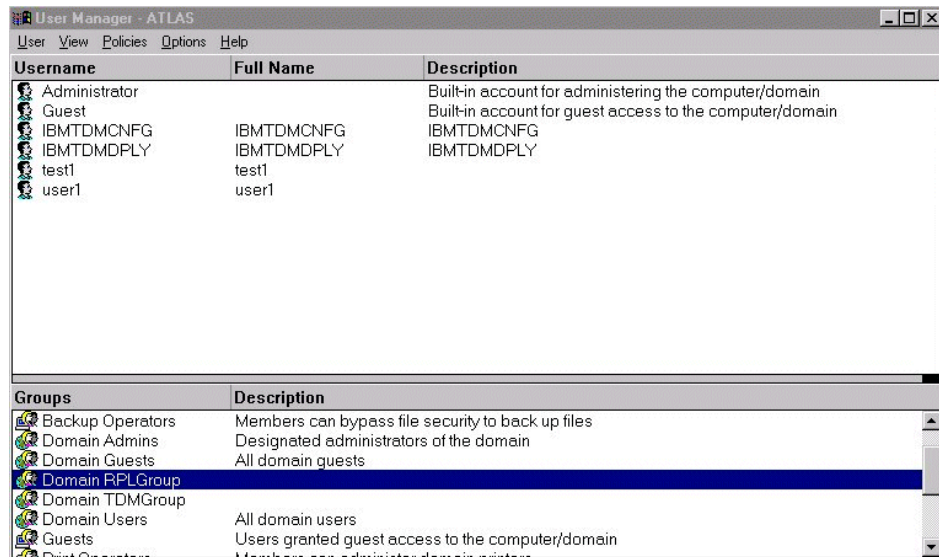


Figure 46. Users and groups added by IBM WorkSpace On-Demand 3.0.1

First, the two users IBMTDMCNFG and IBMTDMDPLY represent the two services installed as part of IBM WorkSpace On-Demand 3.0.1. These are the IBM WorkSpace On-Demand 3.0.1 Configuration Service and the IBM WorkSpace On-Demand 3.0.1 Deployment Service, which can be seen by selecting **Services** in the Windows 2000 or Windows NT Control Panel. These users should not be changed or deleted by the administrator.

Second, the IBM WorkSpace On-Demand 3.0.1 installation caused the creation of two global groups and two local groups. The global groups are called Domain RPLGroup and Domain TDMGroup and are members of the local groups RPLGROUP and TDMGROUP, respectively. The administrator should never modify or delete the groups Domain RPLGroup or RPLGROUP. These groups are defined to hold users created by IBM WorkSpace On-Demand 3.0.1 for each machine defined on the network. If a machine's user is deleted from RPLGROUP, the machine will not be able to boot. The groups Domain TDMGroup and TDMGROUP are used to manage the actual users defined to the server. When a user is created through the console, for example, that user will automatically be added to Domain TDMGroup.

In Windows 2000 and in Windows NT, a global group is defined for the entire domain, while a local group is defined only for a particular server or computer in the domain. Remember that in IBM WorkSpace On-Demand 3.0.1, the

configuration server must be installed on the primary domain controller, while the deployment server(s) can be installed on any other servers in the domain. When a user is created, the user ID is added as a member of the global group Domain TDMGroup, which, in turn, is a member of the local group TDMGroup. On any other server in the domain, there will also be a local group TDMGroup containing as a member the global group Domain TDMGroup. This way, the user can be propagated to other servers in the domain. The same is true for machines when they are defined as discussed in Chapter 4, “Defining machines” on page 103.

It is important to note the difference between users defined through Windows 2000 or Windows NT administration and users defined in IBM WorkSpace On-Demand 3.0.1. When a user is created in IBM WorkSpace On-Demand 3.0.1, the user can be viewed immediately in the Windows 2000 or Windows NT administration console. However, users created through Windows 2000 or Windows NT administration must be imported into IBM WorkSpace On-Demand 3.0.1 to be administered there, which is discussed later in this chapter. After a user has been imported, it can be modified by either the Windows 2000 (or Windows NT) or the IBM WorkSpace On-Demand 3.0.1 console, but must be deleted in both locations if it is no longer needed.

Importing a user into IBM WorkSpace On-Demand 3.0.1 or defining a user in IBM WorkSpace On-Demand 3.0.1 results in a subdirectory being created in \tdm\tdmprfls. The name of the subdirectory is the same as the name of the user. The directory \tdm\tdmprfls is set up by the IBM WorkSpace On-Demand 3.0.1 installation procedure as a shared directory in Windows 2000 and Windows NT, allowing for remote management of users. When desktops and applications are defined to the user, subdirectories are created in the user’s directory for the operating system and application information, such as profiles and desktop configuration. This is described in more detail in Chapter 5, “Managing client desktops” on page 201 and Chapter 6, “Applications” on page 259.

### 3.6.1 Managing users in the IBM WorkSpace On-Demand 3.0.1 GUI

The administrator can define new users using the IBM WorkSpace On-Demand 3.0.1 GUI console. Once you have logged on to a server, click **Tasks**, then click **Define a user** to display the first panel of the user creation dialog. This panel can also be reached by selecting **Resources**, then **Users**, and clicking **Define user** from the User actions menu at the bottom of the window. The resulting panel is shown in Figure 47 on page 79.

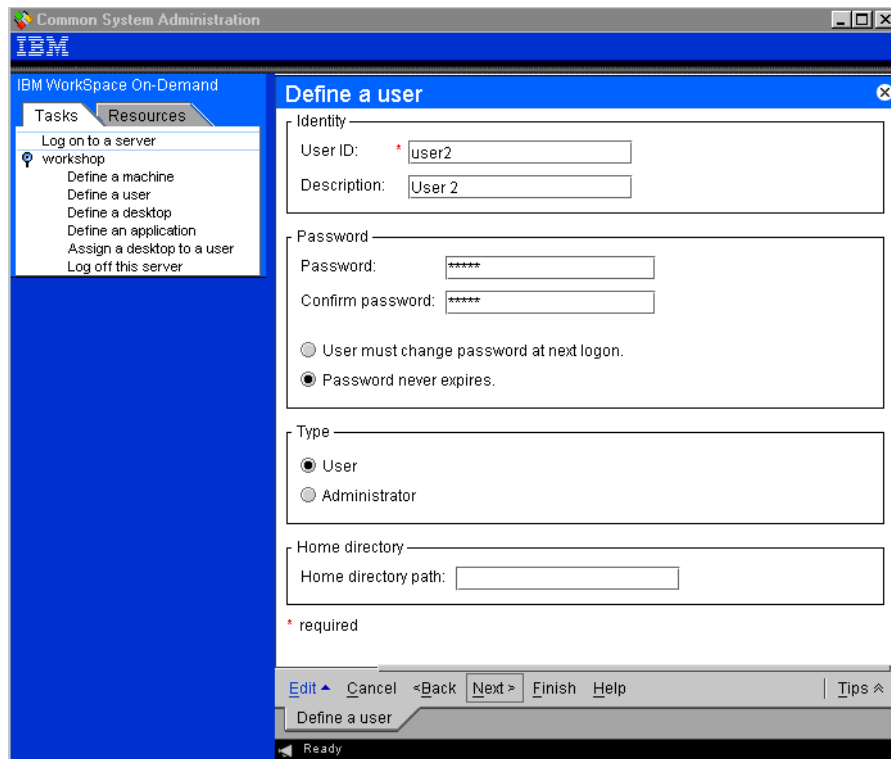


Figure 47. Defining a user

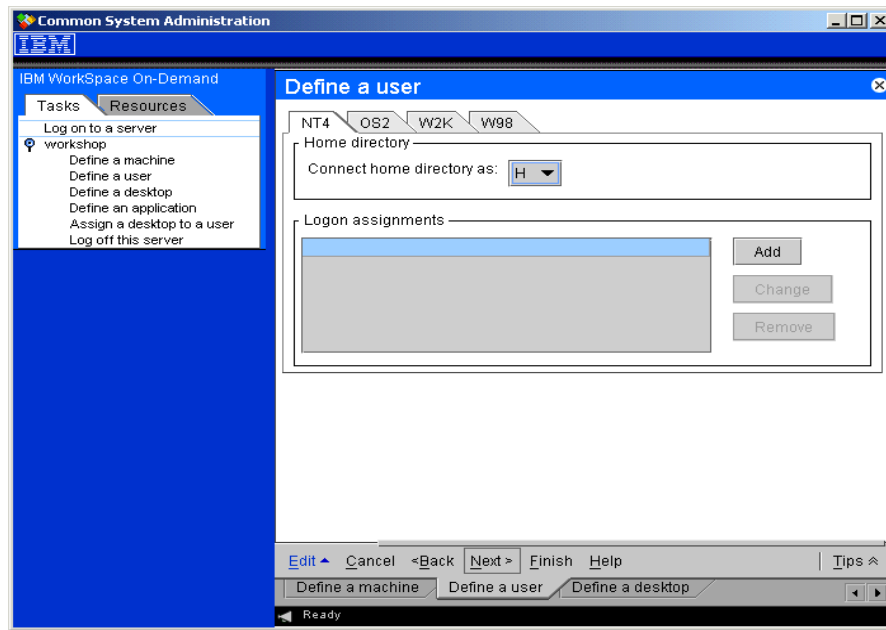


Figure 48. User logon assignments and home directories

When the user ID, password, and description have been entered, clicking **Next** brings you to the second panel of defining a user. Here, you can specify the home directory and any other logon assignments that the user should have as shown in Figure 48. Logon assignments can be any shared resources on the server—for example, shared directories, printers, or serial ports. Each logon assignment is added individually, assigning a drive letter to a shared directory, or a parallel or serial port to other shared devices. Each assignment is added by clicking the **Add** button in the panel shown in Figure 48, then associating the resource to a UNC name in the field *to* of the panel shown in Figure 49 on page 81. The Universal Naming Convention (UNC) is a string with the following format:

```
\\<server name>\<share>.
```

When the user definition is completed click **Finish** and you are returned to a list of all users.

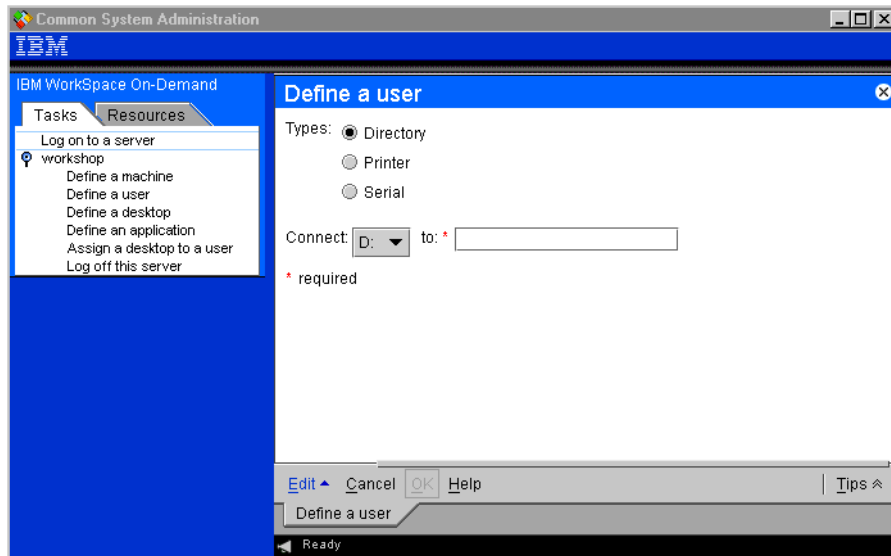


Figure 49. Setting logon assignments

If IBM WorkSpace On-Demand 3.0.1 is installed on a new Windows 2000 or Windows NT Server, the administration GUI could be used to add all users to the server. However, in cases where IBM WorkSpace On-Demand 3.0.1 is installed on an existing Windows 2000 or Windows NT Server with possibly many users already defined, it should not be necessary to redefine these existing users. For this reason, the administration GUI provides a function to import users. From the GUI, click the **Resources** tab, then click **Users**, and from the User actions menu, click **Import from native server users**. This presents a list of all users defined on this server as shown in Figure 50 on page 82. Notice that this list includes the user user1 that we just defined in the previous panels.

In the user list in Figure 50 on page 82, the user user2 had been defined in Windows 2000 but not in IBM WorkSpace On-Demand 3.0.1. Selecting this user and clicking **OK** causes the user to be imported into IBM WorkSpace On-Demand 3.0.1 along with the user's home directory and all logon assignments.

Once users have been added or imported, they can be managed from the user list, which is reached by clicking **Resources**, then **Users** in the GUI. A user could be modified to add or remove logon assignments or desktops, or users could be deleted from this list. Remember, though, that if a user is deleted from IBM WorkSpace On-Demand 3.0.1, the user is not automatically

deleted from the Windows 2000 or Windows NT Server. You must still delete the user from the server via the Windows 2000 or Windows NT User Manager. Conversely, if a user is deleted first from Windows 2000 or from Windows NT, that user must also be explicitly deleted from IBM WorkSpace On-Demand 3.0.1.

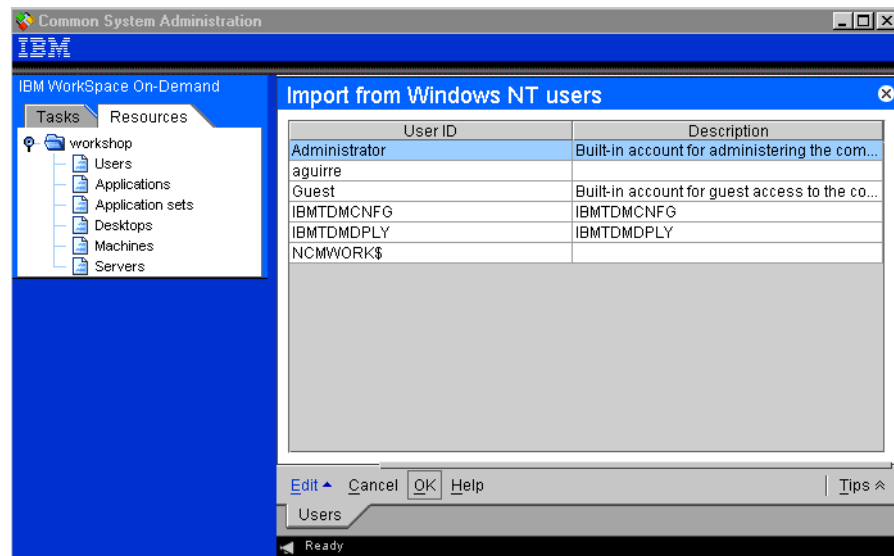


Figure 50. Importing users from Windows NT

### 3.6.2 Managing users in the IBM WorkSpace On-Demand 3.0.1 CLI

Although the IBM WorkSpace On-Demand 3.0.1 GUI provides an easy and intuitive interface for managing users, it can be inefficient for managing large numbers of users or performing many operations on users or groups. The CLI provides a more powerful way to manage users and groups with the added benefit of being able to automate many tasks using JavaScript.

Users are managed in the CLI by performing actions on five basic types of objects. These are:

- user**                      An object representing a user in IBM WorkSpace On-Demand 3.0.1
- nativeuser**             An object representing a user in Windows 2000 or Windows NT
- group**                     A group of users in IBM WorkSpace On-Demand 3.0.1
- nativegroup**            A group of users in Windows 2000 or Windows NT

**groupmember** An object used to represent members of a group, for example, to add or delete users in a group

Most of the operations you will perform will be on the user object. You will occasionally need to work with `nativeuser` as well. The actions you can perform on the user object are as follows:

|                 |  |
|-----------------|--|
| <b>define</b>   | Create a new user.   |
| <b>delete</b>   | Delete a user from IBM WorkSpace On-Demand 3.0.1.                    |
| <b>list</b>     | Show all defined users.  |
| <b>modify</b>   | Change attributes of a user, such as password or home directory.     |
| <b>query</b>    | Display all attributes of a specified user.                          |
| <b>assign</b>   | Assign resources, such as logon assignments or a desktop, to a user. |
| <b>unassign</b> | Remove resources from a user.  |

The following sections describe the actions you can use to manage users and how they can be automated using JavaScript.

### 3.6.3 Defining a user in the CLI

Use the `define` action on the user object to create a new user in IBM WorkSpace On-Demand 3.0.1. The command is as follows:

```
user define user=user1 type=USER
```

This will create the user both in IBM WorkSpace On-Demand 3.0.1 as well as in Windows 2000 or Windows NT. Note that there is a `define` action for the `nativeuser` object as well, but it will only define the user in Windows 2000 or Windows NT and not in IBM WorkSpace On-Demand 3.0.1. We will use the `nativeuser` object only to modify certain attributes of the user and to delete a user from Windows 2000 or Windows NT.

The `define` action takes two parameters, `user=` and `type=`. The `user=` parameter lets you specify the user name. The `type=` parameter has two possible values, `USER` and `MACHINE`, which are case sensitive and must be specified in upper case. The `MACHINE` type would not typically be used by an administrator when defining a user; rather, a user of type `MACHINE` is automatically defined when a machine is created.

Most of the user's attributes cannot be specified in the `user define` command. Instead, they are set using either `user modify`, `user assign`, or `nativeuser`

modify after the user has been created. There is a great deal of overlap between the capabilities of `user modify` and `nativeuser modify`. The typical attributes that you would change with the `user modify` command are:

|                    |  |
|--------------------|--|
| <b>password</b>    | Set as a text string                           |
| <b>homedir</b>     | Set as a UNC string                            |
| <b>homedrive</b>   | The drive letter for the home directory        |
| <b>logonscript</b> | The path and name of the logon script file     |
| <b>workstation</b> | A list of workstations this user may log on to |
| <b>privilege</b>   | ADMINISTRATOR, USER, or SERVER_OPERATOR        |

Specifying a logon script for a user sets a batch file or program that will be executed at every logon for that user. Setting a workstation or workstations for the user will restrict the user to logging on only to the workstations specified. For setting up a typical user, the command is as follows:

```
user modify user=user1 homedir='\\atlas2\users\user1' homedrive=H
password="password"
```

Although the password can be changed using the `user modify` action, the `passwordexpiration` attribute cannot. This can only be changed using the `nativeuser` object. Windows 2000 and Windows NT allow the administrator to set up a user with a password that never expires, or with a password that must be changed at the next logon. In IBM WorkSpace On-Demand 3.0.1, this is set using the `passwordexpiration` parameter on the `nativeuser modify` action. `Passwordexpiration` has two possible values, which must be specified in upper case: they are NEVER and ATNEXTLOGON. Therefore, a further modification of our user would be as follows:

```
nativeuser modify user=user1 passwordexpiration=ATNEXTLOGON
```

When you have defined a user, you will notice some changes that have occurred in IBM WorkSpace On-Demand 3.0.1. A new entry has been made in the data store for this user. The data store is a text file named `datastor.ini`. It is a hidden file and can be found in `\tdm\config`. Figure 51 on page 85 shows a new user's entry in the data store. This information is displayed when you query the user in the CLI using the `user query` action. Notice that the data store entry refers to a profile directory for this user, called `tdmprfls\user1`. This directory was created when the user was defined. As you assign profile information (such as desktop and applications) to the new user, this directory will record the information.



```
[/USER]

[/USER/user1]
USER=user1
Type=USER
RIDid=1187
PROFILEDIR=\\ATLAS2\TDMPRFLS\user1
associated class=com.ibm.tdm.task.principal.TDMUser
```

Figure 51. User information in the data store

Once a user has been defined and modified, you may want to set additional logon assignments for shared directories, printers, or serial ports. Also, the user will not be able to log on to an IBM WorkSpace On-Demand 3.0.1 client machine until you assign a desktop to the user. These operations are done with the `user assign` action.

A user's desktop represents the shell that the user will have as a user interface, including such things as backgrounds, colors, and icon layout. Different types of users will have different desktops. For example, in a banking environment, there might be separate desktops for tellers, financial analysts, and branch managers, each with different layouts. Desktops are specific to the client operating system, so a user could have a different desktop. Desktops are named objects in IBM WorkSpace On-Demand 3.0.1 and are discussed in more detail in Chapter 5, "Managing client desktops" on page 201. For the purpose of user management, let's assume that we have several named desktops available.

With the `user assign` action, you can assign resources to a user. These resources are normally operating system or desktop specific, so for many `user assign` commands, you will have to specify `OS=`, `Lang=`, or `Shell=` to fully define the action you want to perform. The following parameters can be set with the `user assign` action:

|                        |  |
|------------------------|--|
| <b>OS</b>              | The operating system, either W2K, NT4, OS2, or W98                               |
| <b>Desktop</b>         | A string representing the desktop name   |
| <b>Shell</b>           | A shell used with the desktop  |
| <b>Lang</b>            | The language for the user's desktop  |
| <b>Logonassignment</b> | A space-delimited set of strings, each representing a different logon assignment |

For example, the following commands could be used to assign Windows 2000, a default desktop, and two logon assignments to a user:

```
user assign user=user1 os=w2k logonassignment="x=\\atlas2\apps
lpt1=\\atlas2\lanprint"
user assign user=user1 os=w2k desktop=default shell=explorer lang=us
```

Typing four or more commands in the CLI, each with multiple attributes, just to define a single user is somewhat impractical. In a typical enterprise setup, all users in a given domain will likely have the same logon assignments and all home directories will likely be set up in the same way. This would allow administration of a single consistent server image. Also, whenever an administrator creates a new user, it is likely that the user will have a default password that must be changed at the first logon. Therefore, the only things that would be different for each user upon creation of the user are the user name, the server, the operating system, the desktop, and, possibly, the language. Making simple assumptions like these allows us to standardize the user creation process in a JavaScript file that can be executed in the CLI.

The following sample JavaScript code is based on these assumptions. We have defined all users to have a home directory created as a subdirectory of the share users on the server. The name of each user's home directory is the same as the name of the user. Each user will be defined with the same password, which is the word password, and will be forced to change their password when logging on for the first time. We have also set up three logon assignments, two for shared directories and one for a shared printer. Finally, each user will receive a default desktop for the operating system selected for that user, either OS/2, Windows '98, Windows 2000 or Windows NT. Desktop customization is covered in more detail in Chapter 5, "Managing client desktops" on page 201.

In this example, global variables are set up for constant information that would change from implementation to implementation. These variables are the root for all user home directories, the default password, and all the logon assignments. Keeping these variables defined together in the same place makes it easier to modify this code for different implementations. Note also that the function `createUser` makes use of `printResult()`, which is described in Section 3.5, "Command automation" on page 66.

```
// *****
// All user home directories are subdirectories of this share
// *****
var homedirroot = "\\users\\";

// *****
// The default initial password for any new user
```

```

// *****
var firstpw = "password";

// *****
// List any logon assignments in the following arrays
// *****
var assigns = new Array("X", "Y", "LPT1");
var shares = new Array("\\apps", "\\data", "\\LANPRINT");
var numassigns = 3;

// *****
// Encapsulate the actions needed to fully create a user and expose only
// the necessary information of user name, server, desktop, operating
// system and language
// *****

function createUser (name, server, desktop, os, lang)
{
    var cmd, result, message, ctr;
    var username;
    var password, homedir;
    var assignstr, osassign, logonassign;
    var langassign, deskassign, shellassign;

// *****
// First, execute user define to create the user
// *****

    message = "Defining user "+name;
    print (message);

    username = " user="+name;
    cmd = "user define"+username+" type=USER"

    print (cmd);
    result = execute (cmd);
    printResult (result);

// *****
// Next, execute user modify to set password and home directory
// *****

    message = "Setting password and home directory for "+name;
    print (message);

    password = ' password="'+firstpw+'";
    homedir = ' homedrive=H homedir="\\\\'+server+homedirroot+name+'";

```

```

cmd = "user modify"+username+password+homedir;

print (cmd);
result = execute (cmd);
printResult (result);

// *****
// Third, execute nativeuser modify to set password expiration
// *****

message = "Setting password expiration for "+name
print (message);

cmd = "nativeuser modify"+username+" passwordexpiration=ATNEXTLOGON";

print (cmd);
result = execute (cmd);
printResult (result);

// *****
// Fourth, execute user assign to set up logon assignments
// *****

message = "Setting logon assignments for "+name;
print (message);

assignstr = "";
osassign = " os="+os;

// *****
// Set up the logon assignments as a blank-delimited string where each
// substring is in the form X=\\server\share
// *****

for (ctr=0; ctr < numassigns; ctr++)
{
    assignstr=assignstr+assigns[ctr]+"=\\\\\\"+server+shares[ctr]+" ";
}

// *****
// Strip the trailing blank from the logon assignment string
// *****

assignstr=assignstr.substring(0, assignstr.length-1);
logonassign = ' logonassignment="'+assignstr+'";
cmd = "user assign"+username+osassign+logonassign;

```

```

print (cmd);
result=execute (cmd);
printResult (result);

// *****
// Last, execute user assign to set up the desktop. The desktop will
// vary by operating sytem selection.
// *****

message = "Setting default desktop for "+name;
print (message);

langassign = " lang="+lang;
deskassign = " desktop="+desktop;

os = os.toUpperCase();

if (os == "NT4")
{
shellassign=" shell=explorer";
}
if (os == "OS2")
{
shellassign=" shell=pmsHELL";
}
if (shellassign == "")
{
shellassign=" shell=explorer";
}

cmd = "user assign"+username+osassign+langassign+deskassign+shellassign;
print (cmd);
result=execute (cmd);
printResult (result);
}

```

### 3.6.4 Deleting a user in the CLI

Deleting a user requires two actions. First, the user must be deleted from IBM WorkSpace On-Demand 3.0.1. This is done with the `user delete` action. Second, the user should also be deleted from Windows 2000 or Windows NT. This is done via the `nativeuser delete` action. These commands must be performed in the correct order; otherwise, errors will result if you delete a user from IBM WorkSpace On-Demand 3.0.1 that has already been deleted from Windows 2000 or Windows NT.

The following commands could be used to delete a user named user1:

```
user delete user=user1
nativeuser delete user=user1
```

These commands can also be encapsulated into a JavaScript function as shown in Figure 52.

```
function deleteUser (user)
{
  var cmd, message, result;

  message="Deleting user "+user+" from IBM WorkSpace On-Demand 3.0.1";
  print (message);

  cmd = "user delete user="+user;
  result = execute (cmd);
  printResult (result);

  message="Deleting user "+user+" from Windows NT";
  print (message);

  cmd = "nativeuser delete user="+user;
  result = execute (cmd);
  printResult (result);
}
```

*Figure 52. JavaScript to delete a user*

We can extend deleting a single user to deleting multiple users as shown in Figure 53 on page 91. Notice that the loop control counter is initialized at 2. This is because the result of the `user list` action contains two lines of header information before the list of users begins.

```

function deleteAllUsers ()
{
    var cmd, result, ctr;

    cmd = "user list";
    result = execute (cmd);
    for (ctr = 2; ctr < result.data.length; ctr++)
    {
        deleteUser (result.data[ctr]);
    }
}

```

Figure 53. JavaScript to delete all users

### 3.6.5 Defining home directory for users

You can use the WorkSpace On-Demand GUI or CLI to specify a home directory for a user. In order to specify a home directory through the CLI:

1. From a Windows 2000 or NT server or from a Windows 2000 or NT client machine, select:

**Start->Programs->WorkSpace On-Demand->Administration CLI**

to open the CLI.

2. At a command prompt, type:

CONNECT

Type your user ID, password, and the configuration-server name.

3. Type the `user modify` command and specify the HOMEDRIVE and HOMEDIR parameters. For example:

USER MODIFY USER="Myuser" HOMEDIR="\\servername\share" HOMEDRIVE="H"

The HOMEDIR follows the Universal Naming Convention (UNC): no directory must be specified after the share name.

The home directory setting will not be applied to the user until the user is assigned a desktop.

#### Note

The directory corresponding to the “share” name must already exist because it cannot be created from the GUI or CLI.

This is also true if you want to delete the directory: you have to delete it if you no longer need it, using the command available inside the Operating System and not from GUI or CLI.

### 3.6.6 Defining user-specific logon scripts

IBM WorkSpace On-Demand 3.0.1 supports user-specific logon scripts. Follow these requirements when you write logon scripts

#### 3.6.6.1 Windows 2000, NT 4.0, and 98 users

- Name your logon-script file PROFILE.BAT.
- Save the logon-script file to the following directory on the configuration server:

```
\\servername\TDMPRFLS\username\os\lang\PROFILES
```

where:

servername

Represents the name of the configuration server.

username

Represents the name of the user.

os

Represents the operating system. Valid values are W2K, NT4, W98, or OS/2

lang

Represents the two-character country code.

#### 3.6.6.2 OS/2 users

- Name your logon-script file PROFILE.CMD. OS/2 logon scripts must end with the .cmd extension.
- Save the logon-script file to the following directory on the configuration server:

```
\\servername\TDMPRFLS\username\os\lang\PROFILES
```

where:



`servername`

Represents the name of the configuration server.

`username`

Represents the name of the user.

`os`

Represents the operating system. Valid values are W2K, NT4, W98, or OS/2.

`lang`

Represents the two-character country code.

You can use the Windows User Management tool to set the default logon-script file that executes when a user logs on. Refer to Microsoft Administrator's Guides for instructions to use the Windows User Management tool.

When users log on to client machines, the logon process searches the following user profile locations for a logon script. If a default logon script file resides in Windows 2000 or Windows NT User Management, the logon process executes the default logon-script file and the logon-script file in the following locations.

For Windows 2000 users: `W2K\lang\PROFILES\profile.bat`

For Windows NT 4.0 users: `NT4\lang\PROFILES\profile.bat`

For Windows 98 users: `W98\lang\PROFILES\profile.bat`

For OS/2 users: `OS2\lang\PROFILES\profile.cmd`

`lang` still represents the two-character country code.

### 3.6.7 Importing users from Windows 2000 or NT in the CLI

In the administration GUI, a separate action is provided to import users from Windows 2000 or Windows NT. This allows the administrator to select from a list of all users defined in Windows 2000 or NT. In the CLI, there is no separate action to import users from Windows 2000 and Windows NT. Instead, you can use the `nativeuser list` action to display a list of all users defined in Windows 2000 or Windows NT. Note that this list also includes the users already defined in IBM WorkSpace On-Demand 3.0.1. The `user list` action, on the other hand, displays only those users defined in IBM WorkSpace On-Demand 3.0.1. Then, you can use the normal `user define` action to define a Windows 2000 or Windows NT user in IBM WorkSpace On-Demand 3.0.1. If the user named `user1` needed to be imported into IBM WorkSpace On-Demand 3.0.1, the command you would type is:

```
user define user=user1 type=USER
```

You could then use the `user modify` or `user assign` action to set any attributes for the user that are needed in IBM WorkSpace On-Demand 3.0.1. For example, additional logon assignments may be necessary and at least a desktop will have to be assigned.

### 3.6.8 Managing groups in the CLI

The IBM WorkSpace On-Demand 3.0.1 CLI allows you to manage groups as well as users. The distinction between `user` and `nativeuser` previously described is also made for groups, giving you the objects `group` and `nativegroup`. Using the `group define` command, you can create a group that is accessible both in IBM WorkSpace On-Demand 3.0.1 as well as in Windows 2000 or NT. Unlike users, however, the `group delete` action will delete the group from both IBM WorkSpace On-Demand 3.0.1 and Windows 2000 or NT, so there is no need to use the `nativegroup delete` command also. Deleting a group does not cause the members of the group to be deleted.

Users are added and removed from groups using the `groupmember` object. You can use `groupmember define` to add a user to a group and `groupmember delete` to remove a user from a group. The `groupmember list` action shows a list of all members of the group specified. The following commands illustrate group management:

```
group define group=mygroup
groupmember define group=mygroup user=user1
groupmember define group=mygroup user=user2
groupmember define group=mygroup user=user3
groupmember delete group=mygroup user=user2
groupmember list group=mygroup
group delete group=mygroup
```

---

## 3.7 Administering the server

You can administer the server through the IBM WorkSpace On-Demand 3.0.1 CLI using two objects: `server` and `tdmserver`. Actions on the `server` object allow you to manipulate the configuration server, while actions on the `tdmserver` object allow you to manipulate the deployment server. These actions can be performed on a server installed locally on the same machine as the console, or on a remote server over the network. This gives the administrator a great deal of flexibility in managing servers, especially deployment servers, spread out over an enterprise network. The `server` and `tdmserver` objects are automatically created when IBM WorkSpace On-Demand 3.0.1 is installed on a server.

### 3.7.1 Connect to a WorkSpace On-Demand configuration server

In order to connect to a IBM WorkSpace On-Demand 3.0.1 configuration server

1. From a Windows 2000 or NT server or from a Windows 2000 or NT client machine, select **Start->Programs->WorkSpace On-Demand->Administration CLI** to open the CLI.
2. At a command prompt, type:  
`CONNECT`
3. Type your user ID, password, and the configuration server name.

### 3.7.2 Disconnect from a WorkSpace On-Demand configuration server

To disconnect from a IBM WorkSpace On-Demand 3.0.1 configuration server, type:

```
DISCONNECT server_name
```

where `server_name` represents the host name, fully-qualified name or IP address of the server. To disconnect from the server you are currently connected to, you don't have to specify the `server_name`.

### 3.7.3 Administering the configuration server

The following actions are defined for the server object to administer a configuration server:

|                 |   |
|-----------------|---|
| <b>define</b>   | Creates a new configuration server.                                 |
| <b>delete</b>   | Deletes a configuration server.                                     |
| <b>list</b>     | Lists all configuration servers.                                    |
| <b>modify</b>   | Modifies the attributes of a configuration server.                  |
| <b>query</b>    | Displays the attributes of a configuration server.                  |
| <b>assign</b>   | Adds an operating system definition to a configuration server.      |
| <b>unassign</b> | Removes an operating system definition from a configuration server. |

Since the configuration server object is already created upon installation, it is not usually necessary to execute any of the server actions directly.

When you define or modify a server, the following attributes can be specified:

|                       |  |
|-----------------------|--|
| <b>name</b>           | The name for this configuration server. Typically, you would use the same name as the Windows 2000 or NT Server on which the configuration server resides.   |
| <b>authserver</b>     | The name of the authentication server for this configuration server. Again, this is typically the Windows 2000 or NT Server name.  |
| <b>type</b>           | One or more of CONFIGURATION, SMB_AUTHENTICATION, BINL_TFTP_BOOT, DHCP, SMB_FILE, or NFS_FILE. These are case sensitive and must be in upper case. Typically, only CONFIGURATION is set—the rest are Windows 2000 and NT specific. |
| <b>servertimeout</b>  | A timeout interval for commands sent to the server, specified in minutes. The default is 1.  |
| <b>networktimeout</b> | A timeout interval for commands sent to a remote server, also specified in minutes. The default is 1.  |

The following sample command defines a configuration server:

```
server define name=atlas2 authserver=atlas2 type=CONFIGURATION
```

When client support for a new client operating system is installed on the server, the configuration server is assigned the new operating system. Although this is normally done automatically, it can be done manually via the `server assign` command. The operating system, image name, and language must be specified exactly as they were when the client support was installed. For example, Windows 2000 client support can be installed with the operating system name as W2K and the image name as W2KPRO. Therefore, to add Windows 2000 client support to a configuration server, you would use the following command:

```
server assign name=atlas2 os=w2k imagename=w2kpro lang=us
```

Client operating support can be removed from the server using `server unassign` and specifying the same four parameters.

To view a list of all defined configuration servers, type `server list` at the CLI. To delete a specific server, say atlas2, type: `server delete name=atlas2`

### 3.7.4 Administering the deployment server

The `tdmserver` object represents a deployment server. You cannot create or delete a deployment server. You can configure certain attributes of a deployment server and manage it by interacting with the tasks that are being performed or queued on that server. A task refers to any command that might

be running on a server, such as `machine define` or `user assign`. The actions you can execute on a `tdmserver` object are:

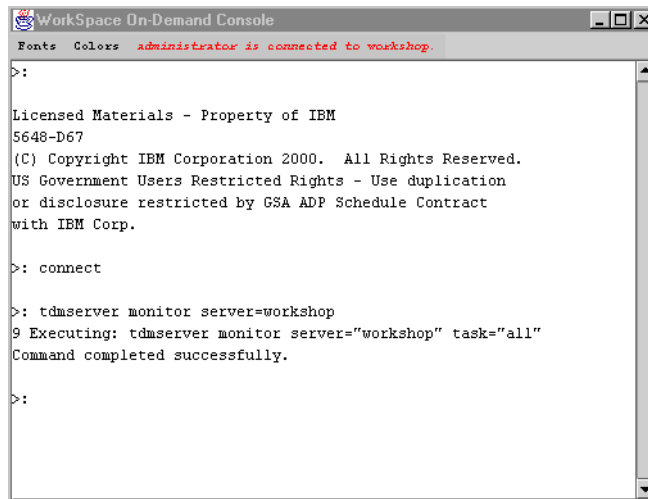
|                     |   |
|---------------------|---|
| <b>configure</b>    | Sets the attributes of a deployment server.   |
| <b>query</b>        | Displays the attributes of a deployment server.   |
| <b>monitor</b>      | Displays the current task running on the deployment server.                               |
| <b>canceltask</b>   | Terminates a task executing on the deployment server.                                     |
| <b>holdtask</b>     | Puts a task on hold.  |
| <b>releasetask</b>  | Releases a previously held task, allowing it to execute.                                  |
| <b>holdqueue</b>    | Holds the entire task queue for the deployment server, so that no tasks execute.          |
| <b>releasequeue</b> | Releases a previously held queue, allowing all tasks in the queue to execute in sequence. |
| <b>shutdown</b>     | Terminates the deployment server.   |
| <b>reloadcrt</b>    | Reloads the command resolution table for the deployment server.                           |

Configuring a `tdmserver` object allows you to set two attributes for the deployment server, the RMI port number and the size of the thread pool. It is normally not necessary to change either value. Querying a `tdmserver` object returns the current value for both attributes as well as the version number of the deployment server as follows:

```
>: tdmserver query server=ATLAS2
The server version is 1.0.0.0.
The current thread pool size is 5.
The current RMI port is 1098.
Command completed successfully.
```

Monitoring a deployment server gives you a real-time view of the status of tasks currently executing on the server. This can be very effective for determining the progress of a long script that is executing on a remote server. For example, we started two instances of the CLI, both connected to the same server. On one, we loaded the JavaScript file `multi.js` described in Section 3.4, “The administration CLI” on page 62. This JavaScript contains the `CreateMultiple()` function to define several machines at once. After executing this function, we executed the `tdmserver monitor` command on the second console as shown in Figure 54 on page 98.

Notice that each task executed by the deployment server is identified by a task number, for example, the `machine define` action displayed in Figure 54 is identified as task 723. You can specify this task number in the `machine monitor` action using the `task=` keyword. The default is `task=all`.



```
WorkSpace On-Demand Console
Fonts Colors administrator is connected to workshop.
>:
Licensed Materials - Property of IBM
5648-D67
(C) Copyright IBM Corporation 2000. All Rights Reserved.
US Government Users Restricted Rights - Use duplication
or disclosure restricted by GSA ADP Schedule Contract
with IBM Corp.
>: connect
>: tdmserver monitor server=workshop
9 Executing: tdmserver monitor server="workshop" task="all"
Command completed successfully.
>:
```

Figure 54. Monitoring a deployment server

The task number also allows you to manipulate a specific task running on a server. You can use the task number to cancel, hold, or release a specific task using the following commands:

```
tdmserver canceltask server=atlas2 task=732
tdmserver holdtask server=atlas2 task=748
tdmserver releasetask server=atlas2 task=748
```

It is also possible to hold or release the entire task queue for a specified deployment server. This is done with the `holdqueue` and `releasequeue` actions. When a server's queue is held, actions can still be sent to the server but they will not be executed until the queue is released. This could be used to schedule intensive work on a deployment server to be performed after business hours. Having the server's queue on hold would guarantee that no work could be done on the server; all actions would remain in the queue. When the queue is released at the administrator's convenience, all actions can be executed. The syntax for holding and releasing a deployment server's queue is:

```
tdmserver holdqueue server=atlas2

tdmserver releasequeue server=atlas2
```

### 3.7.5 Loading JavaScript files on Workspace On-Demand servers

JavaScript files enable you to automate common tasks. You can use the IBM Workspace On-Demand 3.0.1 CLI to load JavaScript files and execute commands in those files. To load a JavaScript file:

1. From a Windows 2000 or NT server or from a Windows 2000 or NT client machine, select **Start** —> **Programs** —> **Workspace On-Demand** —> **Administration CLI** to open the CLI.
2. At a command prompt, type:

```
CONNECT
```

3. Type your user ID, password, and the configuration-server name.
4. Type the `LOAD` command. For example:

```
LOAD my.js
```

To view the JavaScript files that you loaded, type the command `LIST`.

After you load the JavaScript file, you can execute any of the functions in the JavaScript file by typing a function call in the CLI. Your JavaScript file might include a function for the `DEFINE USER` command. For example:

```
function CreateUser (theUser)
{
  USER DEFINE USER="'\"'+theUser+'\"';
}
```

To execute the `CreateUser` function, type:

```
CreateUser ("Amy")
```

After you type the `CreateUser` function into the CLI, the function sends the command `USER DEFINE USER=.Amy...`, to the server. The server defines Amy as a new user in the IBM Workspace On-Demand 3.0.1 environment.

### 3.7.6 Changing Workspace On-Demand server ports

You can change the RMI port that the configuration server uses to communicate with the CLI and GUI, and the RMI port that the machine-deployment server uses to communicate with the configuration server.

To change the RMI port for the configuration server:

1. Stop the IBM Workspace On-Demand 3.0.1 configuration service.

- a. Select **Start** —> **Settings** —> **Control Panel**. If your configuration server is a Windows 2000 server, continue with the next step. If your configuration server is a Windows NT server, skip to 1c.
  - b. Double-click the **Administrative Tools** icon.
  - c. Double-click the **Services** icon.
  - d. Select the **WorkSpace On-Demand Configuration** service.
  - e. Click **STOP**.
2. At a command prompt, change to the directory where you installed IBM WorkSpace On-Demand 3.0.1 and then to the subdirectory CONFIG.
  3. Open the TDMCFG.INI file in an ASCII editor.
  4. Change the RMI port to the port that you want to set as the default port.
  5. Save and close the file.
  6. Start the IBM WorkSpace On-Demand 3.0.1 configuration service.
    - a. Select **Start** —> **Settings** —> **Control Panel**. If your configuration server is a Windows 2000 server, continue with the next step. If your configuration server is a Windows NT server, skip to 6c.
    - b. Double-click the **Administrative Tools** icon.
    - c. Double-click the **Services** icon.
    - d. Select the **WorkSpace On-Demand Configuration** service.
    - e. Click **START**.

To change the RMI port for the machine-deployment server:

1. At a CLI prompt, type the following:

```
TDMSERVER CONFIGURE RMIPORT="4232" SERVER="machdepserv"
```

**Note:** Specify `RMIPORT` to the RMI port number you want to use and `SERVER` to the TCP/IP name of the machine-deployment server you want to configure.

2. Stop the IBM WorkSpace On-Demand 3.0.1 Deployment or WorkSpace On-Demand Application service.
  - a. Select **Start** —> **Settings** —> **Control Panel**. If your configuration server is a Windows 2000 server, continue with the next step. If your configuration server is a Windows NT server, skip to 2c.
  - b. Double-click the **Administrative Tools** icon.
  - c. Double-click the **Services** icon.



- d. Select the **WorkSpace On-Demand Deployment** or **WorkSpace On-Demand Application** service.
  - e. Click **STOP**.
3. Start the WorkSpace On-Demand Deployment or WorkSpace On-Demand Application service.
    - a. Select **Start** —> **Settings** —> **Control Panel**. If your configuration server is a Windows 2000 server, continue with the next step. If your configuration server is a Windows NT server, skip to 3c.
    - b. Double-click the **Administrative Tools** icon.
    - c. Double-click the **Services** icon.
    - d. Select the **WorkSpace On-Demand Deployment** or **WorkSpace On-Demand Application** service.
    - e. Click **START**.

### 3.7.7 Changing network and server time out values

The configuration server executes transform tasks on the machine-deployment server and waits for the results. IBM WorkSpace On-Demand 3.0.1 assigns a number to each task based on the complexity of the task and the approximate time that it takes to complete. To calculate the amount of time that the configuration server waits for a response from the machine-deployment server, multiply the number assigned to the task by the value of `SERVERTIMEOUT` and add the value of `NETWORKTIMEOUT`. Although you cannot change the number assigned to a task, you can accommodate the speed of the servers in your network. To prevent your network from timing out prematurely or to enable your network to spend more time waiting for a response, change the values of `SERVERTIMEOUT` and `NETWORKTIMEOUT`. To change the `SERVERTIMEOUT` and `NETWORKTIMEOUT` values:

1. From a Windows 2000 or NT server or from a Windows 2000 or NT client machine, select **Start** —> **Programs** —> **WorkSpace On-Demand** —> **Administration CLI** to open the CLI.
2. At a command prompt, type:

```
CONNECT
```

3. Type your user ID, password, and the configuration-server name.
4. Type the `SERVER MODIFY` command and specify the `SERVERTIMEOUT` and `NETWORKTIMEOUT` parameters. For example:

```
SERVER MODIFY NAME="server007" NETWORKTIMEOUT="3" SERVERTIMEOUT="2"
```

If the configuration server does not receive a response from the machine deployment server in the specified length of time, the configuration server reports that the transform task has timed out. When a transform task times out, the transform task continues to run and might complete successfully. To view the results of the transform task, follow these instructions:

1. Select **Start —> Programs —> Administrative Tools —> Event Viewer**.
2. At a Windows 2000 configuration server, click **Application Log**. At a Windows NT configuration server, select **Log —> Application**.
3. Double-click an event with a source name of TDMTransformThread.

---

## Chapter 4. Defining machines

As described in Chapter 1, “Introduction” on page 1, IBM WorkSpace On-Demand 3.0.1 allows you to manage different types of client operating systems. Client operating system support can be installed at the same time as the installation of IBM WorkSpace On-Demand 3.0.1, or can be added later using the Client Installation feature. More information about installing client operating systems can be found in Chapter 2, “Installation” on page 9. The following sections discuss defining machines with Windows 2000, Windows NT Workstation, Windows 98, and OS/2 operating systems.

---

### 4.1 General machine representation

The choice of an operating system is tied to the definition of each client machine; therefore, the operating system setup is the most important part of a machine definition. The client operating system support must be installed prior to defining a client machine with that operating system type. Other important information you need to know about the client prior to installing is the MAC address of its network adapter card, and the name of the deployment server to which it will be defined. The rest of the attributes of a machine definition are specific to the operating system selected.

Client machines are represented by the machine object in IBM WorkSpace On-Demand 3.0.1. This object has six defined actions as follows:

- Define**     Creates a new machine.
- Delete**     Removes an existing machine.
- List**        Displays all defined machines.
- Query**       Displays all attributes of a specific machine.
- Modify**     Changes one or more attributes of a machine.
- Parmlist**    Displays further information for a specified machine attribute.

The machine object has a large number of attributes, many of which depend on the operating system selected for the machine. A Windows NT Workstation machine, for example, has over 20 attributes. Many of these are required when you define a machine. It is possible to define a machine through the IBM WorkSpace On-Demand 3.0.1 administration GUI or through the CLI. To minimize effort, it is recommended that you encapsulate common attributes of all your machines in a JavaScript function and only expose those attributes that are likely to vary by client.

When you have successfully defined a client machine, you will find a data store entry for the machine as well as several new directories and subdirectories created in the `\tdm\mm\client` directory (where four machines have been defined) as shown in Figure 55. Different parts of the machine definition are located in `ro\macs`, `ro\machines`, and `rw\machines`. You will also find a user created in Windows 2000 or NT as a member of the global group Domain RPLGroup with the same name as the machine you defined.

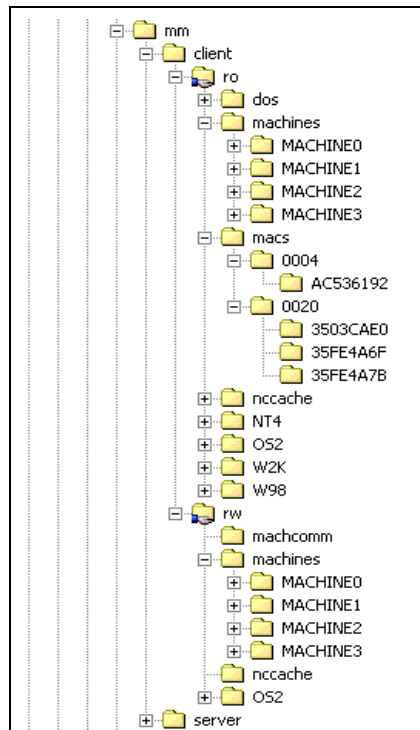


Figure 55. Machine directory hierarchy

The `tdm\mm\client\ro\macs` directory is the starting point of a machine definition. Here, we find subdirectories corresponding to the MAC addresses of the network adapter cards in each machine. When a client machine is booted, the deployment server looks up the MAC address presented by the client and reads the corresponding directory. This directory contains .INF files that tell the server the machine name and operating system. Using the machine name, the server can look up the appropriate directories in `tdm\mm\client\ro\machines` and `tdm\mm\client\rw\machines`. Files in the `ro` directory contain additional operating system-specific boot information, while files in the `rw` directory contain operating system-specific log files describing

the machine's install or boot history as well as other operating system files for which a client machine needs write access. Windows 2000, Windows NT Workstation or Windows 98 client machines will either perform a local boot of their operating system or a remote installation of the operating system based upon the information found in rw, while OS/2 client machines would perform a remote boot.

The ro and rw directories are shared; they have aliases RPLFILES and WRKFILES, respectively. The local group RPLGROUP has read-only access to RPLFILES and full control access to WRKFILES. These aliases and permissions are normally set up during the installation of IBM WorkSpace On-Demand 3.0.1. Each machine you create in IBM WorkSpace On-Demand 3.0.1 is represented by a user ID in the RPLGROUP group. This way, machines have access to the operating system files they need during boot or installation without requiring an actual user to be logged on with access to these aliases.

When referring to a machine using commands in the console CLI, you must specify the machine name as well as the following three parameters: operating system name, image name, and language. These values are determined by the choices you made when installing the client operating system support. You must specify all these values when defining, deleting, querying, or modifying any machine.

---

## 4.2 Defining Windows 2000 client machines

If you have installed Windows 2000 client support in IBM WorkSpace On-Demand 3.0.1 using the Windows 2000 Professional Edition CD, you will be able to define machines with the Windows 2000 operating system. These clients do not perform a remote boot from the server, rather, they perform a remote install from the server once the machine definition is complete. The Windows 2000 image that was installed in IBM WorkSpace On-Demand 3.0.1 is used as a template to install a custom Windows 2000 image on your client machine. The following sections describe the steps involved in creating and booting a Windows 2000 client machine.

### 4.2.1 Windows 2000 client parameters

A Windows 2000 client is created with the `machine define` command. Among all the parameters that can be specified for this command the following are mandatory:

OS, LANG, NAME, SERVER, IMAGENAME, MAC, REGUSER, CDKEY, and TZ

**Name**

This is the name that will uniquely identify the client machine on the deployment server. For example, machine0, machine1, machine2, and machine3 are valid machine names as shown in Figure 55 on page 104. The machine name must follow the 8.3 Naming Convention. This parameter is required when creating, deleting, or modifying a machine. Our sample Windows 2000 client will be named machine0.

**Mac**

This parameter specifies the MAC address of the client machine's network adapter card. It must be specified when creating a machine. The MAC address is expressed as 12 hexadecimal digits and results in a directory structure, such as macs\0004\AC536192, as shown in Figure 55 on page 104.

**Server**

This parameter specifies the name of the deployment server where the client machine will be created. It is required for machine creation only.

**OS**

The OS parameter lets you select the operating system image for your client. This can be W2K, NT4, W98, or OS2. This parameter must be specified for creating, deleting, or modifying a machine. For our sample Windows 2000 client, we will set OS=W2K.

**ImageName**

ImageName must correspond to the image name you selected when you installed the client operating system support. For example, if installing Windows 2000 client, the ImageName value might be w2kpro. This parameter must be specified when creating, deleting, or modifying a machine.

**Lang**

This parameter sets the language of the operating system that will be used for this client machine. It must correspond to the language selected when the client operating system support was installed. Lang is specified as a two-character string, for example, "US" for American English. This parameter is required when defining a machine and cannot be changed after definition.

**CDKey**

This is the CD key supplied by Microsoft for each copy of Windows 2000. It is a 25-digit field formatted as five groups of five characters, as follows:

XXXXX-XXXXX-XXXXX-XXXXX-XXXXX. Each character is alphanumeric. This parameter is required for creation of a new machine.

### **TZ**

The TZ parameter specifies the time zone to be used on the client machine. This parameter is required when creating a machine and must be specified as a string in exactly the same format as would normally be selected from the Windows Date/Time window. For example, to set Central time, you would specify:

```
TZ='(GMT-06:00) Central Time (US & Canada)'
```

### **Template**

This is a yes or no parameter that specifies whether the machine is to be defined as a template. The default value is no. If you specify Template=yes, you can create other machines from this template. Using templates is discussed in Section 4.6, “Machine templates” on page 177.

### **Templatename**

This specifies the name of the template to be used when creating this machine. If you create a machine from a template, all machine attributes will be taken from the template. Using templates is discussed in Section 4.6, “Machine templates” on page 177.

### **Reguser**

The reguser parameter lets you specify the name of the registered Windows 2000 user for the client workstation. This is specified as text and used in the Windows 2000 installation.

### **Netadapter**

This parameter must be specified upon machine creation and indicates the type of network adapter in the client machine. Valid values for this parameter are taken from the \$oem\$\\$1\Drivers\NIC directory in the Windows 2000 client image. In our default installation of IBM WorkSpace On-Demand 3.0.1, there were two subdirectories under \$oem\$\\$1\Drivers\NIC, called TRP for the IBM 16/4 PCI Token-Ring adapter with Wake on LAN, and IBMFE for the IBM 10/100 PCI Ethernet adapter. This yielded two possible valid values for the Netadapter parameter, TRP or IBMFE.

IBM WorkSpace On-Demand 3.0.1 requires client network adapters to support the PXE 2 network boot specification. Network adapters that support this will likely need to have their drivers added to the Windows 2000 client image. You can do this by adding another subdirectory under \$oem\$\\$1\Drivers\NIC in your client image and copying your network adapter

driver files to this new directory. Then, specify the directory name in the Netadapter parameter.

### **DHCP**

DHCP is a yes or no parameter. If you set DHCP to no, you will have to specify the client machine's IP address and other TCP/IP parameters explicitly. Note that even if you set DHCP to no, a DHCP server is still required on your network since the PXE boot protocol requires DHCP for the client to boot at all.

### **IP**

This parameter is required if you set DHCP=No. Set the client machine's IP address in standard IP address format. For example: IP='nnn.nnn.nnn.nnn'.

### **Netmask**

This parameter allows you to specify the client machine's subnet mask and is required if you set DHCP=No.

### **Tcpname**

This parameter specifies the machine's TCP/IP host name. It is required if you set DHCP=No.

### **Tcprouter**

Use this parameter to set the IP address of the IP router in standard IP address format. This parameter is required if you set DHCP=No.

### **Tcpnamesrv**

This parameter is required if you set DHCP=No and is used to specify the IP address of the Primary Domain Name Server.

### **Tcpdomain**

This parameter is required if you set DHCP=No and is used to set the IP domain name.

### **WINS**

This optional parameter specifies the IP addresses of the WINS servers. You can specify a list of IP addresses separated by commas.

### **Protocols**

This parameter is specified as a comma-delimited string and is used to turn on various network protocols. Supported protocol values for Windows 2000 are:

MS\_TCPIP,MS\_NetBEUI,MS\_NWIPX,MS\_DLC,MS\_PPTP,MS\_L2TP,MS\_NE  
TMON,MS\_ATMLANE,MS\_ATMUNI,MS\_ATMARPS,MS\_Streams and



MS\_AppleTalk. A typical protocol setting would be  
Protocols="MS\_TCPIP,MS\_NetBEUI".

### **NBScope**

This parameter specifies the scope ID of NetBIOS over TCP/IP. It is optional and should normally not be specified. The scope can be set as a character string and is used to configure different logical NetBIOS networks on the same TCP/IP network.

### **Status**

A machine's status may be set as enabled or disabled. If a client machine is disabled, it will be prevented from booting. Note that in the GUI, the status is specified by a checkbox labeled Disable Client.

### **JVM**

The JVM parameter lets you specify whether a Java Virtual Machine should be installed with the client operating system. Valid values are Yes or No. Since Java applications are becoming pervasive, it is usually a good idea to install the JVM for all clients.

### **TMA**

The TMA parameter allows you to specify whether the Tivoli Management Agent should be installed to the client workstation. This value is set as yes or no.

### **Logon**

The logon parameter also is a yes or no parameter. If you specify no, the client machine will present only a Windows 2000 local logon window when it boots up after installation. It is recommended to set Logon=Yes so that the client machine will present the IBM WorkSpace On-Demand 3.0.1 logon window instead, which will log on to the server as well as locally.

### **PrebootImage**

This parameter specifies the DOS boot image that will be used prior to installing Windows 2000. The pre-boot DOS image provided with IBM WorkSpace On-Demand 3.0.1 for Windows 2000 clients is tdmw32ut.img and can be found in the boot subdirectory of the Windows 2000 client image installed in \tdm\mm. This boot image uses NetBIOS calls over TCP/IP to communicate with the deployment server to copy the Windows 2000 installation files.

### **Installdos**

If you specify Installdos=Yes, then when the client hard drive is formatted as part of the install, it is formatted with the /s option and DOS is installed. This

is only used to allow network adapter drivers that cannot coexist with Windows to be deleted from memory by booting DOS first with no network support. This keyword should only be used sparingly and is not needed for most network adapters. Specify `Installdos=No` to have no DOS installation on the hard drive. More information about the significance of `Installdos` can be found in Section 4.2.3, “Windows 2000 client installation and boot” on page 112.

### **Partition**

This parameter specifies the size of the primary partition for the client machine. It is set in megabytes, so specifying `Partition=600` will create a primary partition of 600 MB. You can also specify the value `all`, which will create a partition of the maximum possible size on the client machine’s hard drive but no larger than 2048 MB. You cannot specify a partition size greater than 2048 MB when you define a Windows 2000 machine. Bypassing this restriction is discussed in Section 4.9, “Customizing Windows client machine images” on page 191.

### **Srcrespfile**

The `srcrespfile` parameter indicates the name of the response file used to guide the unattended installation of Windows 2000 on the client machine. This could be `unattend.txt` or any other customized response file for the specific client. For example, we used `srcrespfile='ibmfepci.txt'` and customized this response file for various IBM PC300 family PCs.

### **Localadmpw**

This optional parameter lets you set a password for the local administrator user ID on the client machine.

### **Description**

This optional parameter lets you specify a text description of the client machine that will be shown when the machine is queried or a list of machines is displayed.

### **Section**

This parameter lets you optionally define additional sections for the Windows 2000 response file, `unattend.txt`. The format for this parameter is:

```
Section='{SectionName, Key=value,...},{...}...
```

You can set as many sections and key-value pairs as you want.

### Printern

In this parameter, n is a number from 1 to 9, such as printer1= or printer2=. Use this parameter to specify the printer available to the client machine. The printer is specified using a comma-delimited string as follows:

```
printer1="PrinterName, ModelName, PortUNCName, driver, datafile,  
configfile"
```

The printer must be defined locally on your server and must be shared so that the value of PortUNCName is the network name of your server and the alias of the printer. To find the correct values for the driver, datafile, and configfile, it is easiest to use regedit and look up the printer driver's registry entries on the server. For example, the following string could be used to add support for a 4029 laser printer to your Windows 2000 client machine:

```
printer1="ibmprn, 4029, \\atlas2\ibm4029l, rasdd.dll, ibmppsdl.dll,  
rasddui.dll"
```

### Kbdtype

Use this parameter to set the type of keyboard for the client machine. The keyboard type is represented as a string, for example a US keyboard would be specified as kbdtype='United States 101'.

## 4.2.2 The machine define command for Windows 2000

When the Windows 2000 client operating system support has been successfully installed in IBM WorkSpace On-Demand 3.0.1, you can use the define action on the machine object with the parameters as previously described to define a machine using the console CLI. Figure 56 shows the machine define command for a typical token-ring based client.

```
Machine define OS='W2K' NETADAPTER='TRP' DHCP='Yes' LANG='us'  
PROTOCOLS='MS_TCPIP,MS_NETBEUI' DESCRIPTION='ITSO Test'  
STATUS='Enabled' JVM='Yes' TMA='No' IMAGENAME='w2kpro'  
CDKEY='xxxxx-xxxxx-xxxxx-xxxxx-xxxxx' TZ='Central America'  
PREBOOTIMAGE='tdmw32ut.img' INSTALLDOS='No' LOCALADMPW='getin2see'  
partition='800' srcrespfile='ibmfepci.txt' Name='machine0'  
MAC='0004AC536192' SERVER='atlas2' REGUSER='ITSO'
```

Figure 56. Defining a Windows 2000 machine in the CLI

As discussed in Chapter 3, "Administration" on page 57, defining a machine by typing machine define at the console is cumbersome and error-prone, while defining a machine through the GUI is slow and not effective for managing remote servers. You can make reasonable assumptions about many of the

parameters for `machine define` and encapsulate this action in a JavaScript function that only exposes those parameters that do need to change for every machine. For example, the JavaScript function `CreateMachinet()` defined in the `machines.js` file provided on the enclosed CD allows you to create a token-ring based client machine by specifying only the machine name, MAC address, deployment server name, and operating system name. This allows you to simplify the `machine define` command shown in Figure 56 on page 111 to the following:

```
CreateMachinet("machine0", "0004AC536192", "atlas2", "w2k")
```

It is also possible to simplify machine creation using machine templates as discussed in Section 4.6, “Machine templates” on page 177.

### 4.2.3 Windows 2000 client installation and boot

Once you have successfully defined a Windows 2000 client machine, you will notice many changes in the IBM WorkSpace On-Demand 3.0.1 directories to represent the new machine. The directories and files that have been created will direct the process of remotely installing Windows 2000 on the client machine. We investigate several of these changes here.

#### 4.2.3.1 The data store entry

First, a new entry is created in the data store for the new machine. The data store is kept as a text file called `datastor.ini` in `\tdm\config`. Since a full definition of a machine needs to include not just the machine's name but also the operating system name, image name, and language, the data store contains sections for each possible combination of values. All the parameters that you specified when defining the machine are listed under the machine's name in the data store. When a machine is queried, its parameter information is read from the data store and any modifications made using the `machine modify` action are saved here as well.

```

[/MACHINE]

[/MACHINE/W2K]

[/MACHINE/W2K/w2kpro]

[/MACHINE/W2K/w2kpro/US]

[/MACHINE/W2K/w2kpro/US/machine0]
os=W2K
netadapter=TRP
dhcp=Yes
lang=us
protocols=MS_TCPIP
description=ITSO Test
status=Enabled
jvm=Yes
tma=No
imagenamew2kpro
cdkey=xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
tz= Central America
prebootimage=tdmw32ut.img
installldos=No
localadmpw=getin2see
partition=800
srcrespfile=ibmfepci.txt
name=machine0
mac=0004AC536192
SERVER=ATLAS2
associated class=com.ibm.tdm.task.machine.W2KMachine

```

Figure 57. A Windows W2K machine definition in the data store

Figure 57 shows the data store representation of the Windows 2000 machine definition given in Figure 56 on page 111. Other machine definitions, including those for other operating systems, would be listed in the appropriate subsections of the data store. Although the data store is currently a text file, its format may change with future releases of IBM WorkSpace On-Demand 3.0.1; therefore, you should never change a machine definition directly in the data store but use the `machine modify` action instead.

#### 4.2.3.2 The MACS directory

The next change to notice is that a new subdirectory has been created for the MAC address of the client machine in `tdm\mm\client\ro\macs`. In the case of

machine0 defined in Figure 56 on page 111, this subdirectory is \0004\AC536192. When the client machine first boots, IBM WorkSpace On-Demand 3.0.1 will use the client's MAC address to locate this directory. Two files are found here, tdmclnt.inf and tdmDOS.inf. These files were copied from the directory client\ro\w2k\w2kpro\us\defs and customized by the machine define process so that they are specific to the client machine we created. They are used to locate the initial DOS bootable image for the client machine. The .INF files corresponding to our example machine definition are shown in Figure 58.

```
[MESSAGE]
file=MACS\TDMBOOT.MSG
[UNATTENDED]
enabled=yes
file=W2K\w2kpro\us\boot\tdmdos.sys

tdmclnt.inf

PATCH_FILE=AUTOEXEC.BAT,NETWORK.INI,PROTOCOL.INI
MSG_FILE=MACS\TDMBOOT.MSG
IMAGE_FILE=W2K\w2kpro\us\BOOT\tdmw32ut.img
CLIENT_NAME=machine0
SERVER=ATLAS2
LAST_DRIVE=Z
NEXT_DRIVE=Y
LANG=us
OS=W2K
IMAGENAME=w2kpro
DOMAIN=DOMATLAS2
SUBNET_MASK=DHCP_OPT_1
RELEASE_DHCP=NO

tdmdos.inf
```

Figure 58. tdmclnt.inf and tdmDOS.inf for a Windows 2000 client

During the first boot, IBM WorkSpace On-Demand 3.0.1 first locates the file tdmclnt.inf and transfers it to the client machine via TFTP. The client machine's status is listed in this file; the line `enabled=yes` reflects the parameter setting `status=enabled` in the `machine define` command. If a machine were created with disabled status, its boot would not proceed beyond this point.

The line `file=` in `tdmcInt.inf` points to the second boot image, `tdmdos.sys`. This image is copied to the client machine via TFTP and the client boots. This is a DOS boot that will use the file `tdmdos.inf` to locate the actual operating system image that is defined for this client machine. This second boot goes back to the server, locates `tdmdos.inf`, and transfers it to the client via TFTP.

The `prebootimage` parameter specified on the `machine define` command is listed in `tdmdos.inf` as the `IMAGE_FILE` variable. The path specified in `IMAGE_FILE` is relative to the alias `RPLFILES`, or `tdm\mm\client\ro`. This combination completely specifies the path of the Windows 2000 client support installation on the server, where the DOS boot image files are installed. This DOS boot image is now transferred to the client machine via TFTP and the client machine's own directories are located on the server.

These directories are located on the server by means of the `SERVER`, `OS`, `IMAGENAME`, `LANG`, and `CLIENT_NAME` lines defined in `tdmdos.inf`. In our example, these correspond to subdirectories called `machine0` in the `machines` directory of the alias `RPLFILES`, or `tdm\mm\client\ro`, and in the `machines` directory of the alias `WRKFILES`, or `tdm\mm\client\rw`, respectively. The `machine0` subdirectory in `rw` will contain logging information from the installation of Windows 2000 on the client machine. The `machine0` subdirectory in `ro` contains all the control files needed to install Windows 2000 on the client machine.

#### **4.2.3.3 The machines directory in RPLFILES**

Figure 59 on page 116 shows a listing of the contents of the `machine0` directory for our sample machine under the `machines` directory in the `RPLFILES` alias. When the DOS pre-boot image is now booted, it uses the control files in this directory to direct the installation of Windows 2000 on the client. These control files were copied and customized from defaults found in the directory `\w2k\w2kpro\us\defs`, also in the `RPLFILES` alias. The DOS boot image establishes connections to both the `ro` and `rw` aliases and uses the utilities in the `\dos\2k\us\tree` subdirectory under the `RPLFILES` alias to perform the installation. An installation shell is started from the DOS image that makes repeated calls to the program `state.exe`. This program uses the file `state.ini` found in `\machines\machine0\w2k\w2kpro\us` to determine each successive step of the installation process.

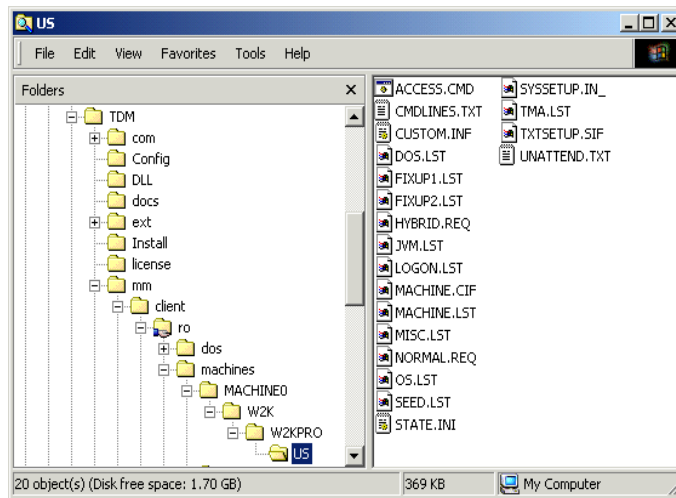


Figure 59. Contents of a Windows 2000 machine directory after creation

#### 4.2.3.4 The state.ini file

The state.ini file is key to the boot and install process from this point on. State.ini was customized from the default provided with the Windows 2000 client support. State.exe processes the contents of state.ini via a series of steps, known as states, that are listed in the StateNames section of state.ini. These states are:

|                |  |
|----------------|--|
| <b>Init</b>    | Initializes the installation.                      |
| <b>New</b>     | Partitions the client hard drive.                  |
| <b>Format</b>  | Formats the hard drive.                            |
| <b>Copy</b>    | Copies system files prior to installation.         |
| <b>Install</b> | Installs Windows 2000 on the client machine.       |
| <b>Hybrid</b>  | Client machine has been installed; reboot locally. |
| <b>Error</b>   | Processes errors from any other states.            |



```
[parms]
PartitionSize=800
InstallDOS=No
MinMemSize=305
MinDiskSize=350
DSize=0
VideoCorrection="NO"
CleanupOSFiles=Yes
InitDiskParms=0
msg1=0
msg2=0
OSDISK=C:
BootImageApps=JVM, LOGON
ClientType=NO

[StateNames]
Init
New
Format
Copy
Install
```

Figure 60. The Windows 2000 header of state.ini

The first few lines of the state.ini from our sample machine are shown in Figure 60. The [parms] section at the top of state.ini contains parameter information that may be substituted in commands listed in other state sections farther down in the file. Some of these parameters were set based upon information provided as parameters in the `machine define` command. In particular, the parameters PartitionSize, InstallDOS, and BootImageApps should look familiar.

For each state, there is a section of state.ini labelled with the state name. Each state section contains commands that are executed by state.exe. These commands are of six possible types. Listed in their search order, these are:

- Internal commands to state.exe
- Internal commands to the install shell
- DOS internal commands
- DOS external commands or executables
- DOS batch files
- Install shell batch files

#### 4.2.3.5 Section [new] of state.ini

Figure 61 illustrates the structure of state.ini by showing one complete section of the file. The section is labelled [new] and thus corresponds to the second state, named new, that directs the install shell to partition the hard drive according to parameters specified by the `machine define` command. State.ini determines the hard drive size and ensures that the disk is at least 350 MB, which is big enough to allow the installation of Windows 2000. The value of the `dsize` variable is logged to the file `counter.ini`, found in machine0's directory in `tdm\mm\client\rw`. In the case of our example machine0, the hard drive size was 6 GB.

```
[new]
dsize=disksize
Message 260 %Dsize%
delay 2
if %dsize% > %MinDiskSize%
if %PartitionSize%="all"
if %dsize%>2048
PartitionSize=2048
else
PartitionSize=%dsize%
endif
else
if %partitionSize%>2048
Log %InvPartSize%
ErrorExit
endif
endif
else
Log %HardDiskTooSmall%
ErrorExit
endif
;Creating a partition of %d Megabytes
Message 261 %PartitionSize%
delay 2
InitdiskParms="/PRI:"&%PARTITIONSIZE%
initdisk 1 %InitDiskParms%
Message 285
delay 5
;echo "Rebooting the system..."
Message 263
delay 3
;pause
reboot
```

Figure 61. Sample partition command section of state.ini

The commands in the [new] section direct state.exe to check what partition size was specified by the `machine define` command. If the administrator had specified a value of all, the partition size is set to the smaller of the physical disk size or 2 GB. If an actual partition size was specified, that size is used or an error is logged if the requested size is larger than 2 GB.

The `initdisk` command is used to create the primary partition of the appropriate size, and when this succeeds, the client machine is rebooted. The state is promoted to state 3, [Format].

#### 4.2.3.6 Section [Format] of state.ini

This state causes the newly created partition to be formatted. The commands for this section are shown in Figure 62. Notice that the commands in this section use the `InstallDOS` parameter from the `machine create` command. If `InstallDOS="Yes"` was specified, or if state.exe determines that the amount of conventional memory on the client machine is insufficient, DOS is installed by using the `/s` parameter on `format`.

```
[Format]
;Save the DOS files in case we need to use them
if %InstallDos%="No"
if convmem < %MinMemSize%
InstallDos="Yes"
endif
endif
;echo "Formatting the partition...please wait..."
Message 264
FormatCustomize "on"
OutputFile "Format.out"
if %InstallDOS%="Yes"
FormatDisk %OSDISK% "/U /V:WNT /s"
else
FormatDisk %OSDISK% "/U /V:WNT /b"
endif
FormatCustomize "off"
;echo "Format Completed Successfully"
Message 265
delay 3
;pause
```

Figure 62. The format section of state.ini for Windows 2000

When the format has successfully completed, there is no reboot specified because it is not necessary. State.exe executes a transition from the [Format]

state to the [Copy] state and copies system files from the server to the client. The [Copy] state for the sample machine machine0 is shown in Figure 63.

#### 4.2.3.7 Section [Copy] of state.ini

Four `copy` commands are executed from this section, since the `InstallDOS` parameter had been set to `No`. These commands copy the files for the client Java Virtual Machine, the IBM WorkSpace On-Demand 3.0.1 logon program, the Tivoli management agent, and other miscellaneous files to the client C: drive. These `copy` commands are directed via list files. The Z: drive letter prefacing each list file name maps to `\tdm\mm\client\ro`, also known as the alias `RPLFILES`. Prior to the `copy` commands, the `chdir` command sets the rest of the path to find the list files to be `\machines\machine0\w2k\w2kpro\us`. Notice that the Tivoli files are copied to the client even though the machine definition had specified `TMA=No`. The files are copied, but the Tivoli Management Agent is simply not enabled as an application for this machine.

```
[Copy]
;Load disk caching software to speed the copying of 2000 files
;echo "Copying Client files...please wait..."
Message 266 "Windows 2000"
cachedisk
chdir %ClientRO%
CopyCustomize "On"
MCOPY /I:Z:JVM.lst /l:y:copy.out /s /r /e
MCOPY /I:Z:Logon.lst /s /r /e /l:copy.out
MCOPY /I:Z:TMA.lst /s /r /e /l:copy.out
MCOPY /I:Z:Misc.lst /s /r /e /l:copy.out
;pause
if %InstallDOS%="Yes"
MCOPY /I:Z:OS.lst /s /r /e /l:copy.out
MCOPY /I:Z:DOS.lst /s /r /e /l:copy.out
MCOPY /I:Z:Fixup2.lst /s /r /e /l:copy.out
endif
;echo "Client Files have been copied sucessfully"
Message 269
delay 3
CopyCustomize "Off"
```

Figure 63. The copy section of Windows 2000 state.ini

The miscellaneous `copy` commands specified in `misc.lst` copy three files from the server to the client. These are `unattend.txt` and `custom.inf`, which are used to drive the installation of Windows 2000, and `access.cmd`, which is used to define the local administrator ID on the client machine.

#### 4.2.3.8 Section [Install] of state.ini

When the `copy` commands have completed, the client system is now ready for the Windows 2000 installation. State.exe exits the [Copy] section of state.ini and enters [Install]. At the beginning of this phase, state.exe checks to see if DOS was installed on the client hard drive due to the `machine define` parameter `Installdos` having been set to Yes. If this is true, it means that there is insufficient conventional memory in the client machine to run the Windows 2000 installation programs along with network drivers to perform the installation directly from the server. Therefore, state.exe will copy the rest of the DOS system files as well as the entire Windows 2000 installation directory to the client machine's local hard drive. It will then boot into DOS with no network connections and begin the installation of Windows 2000. When the installation is finished, the client machine will reboot and restore its network connections. In most cases, however, a local DOS-based install procedure is not required and Windows 2000 installation can proceed directly from the server.

When state.ini proceeds with the installation of Windows 2000, the client machine's state is set to hybrid. A state value of hybrid indicates that the client machine is able to boot from its local hard drive. All client machines in IBM WorkSpace On-Demand 3.0.11 always perform a network boot first when they are rebooted to allow the server to control the machine and perform updates or installations as necessary. When a machine is in the hybrid state and it performs its network boot, the `tdmclnt.inf` file now points to `hdboot.com` instead of `tdmdos.sys`. This redirects the client's boot to the local hard drive.

The lines of state.ini describing the network-based installation are shown in Figure 64 on page 122. State.exe begins the installation using the `unattend.txt` file, which was copied from the server to the client's hard drive. `Unattend.txt` was customized by the `machine define` command to contain the rest of the parameters specified when the client machine was created. For example, the user information, CD key, time zone, video, display resolution, and network protocol information specified by `machine define` are all listed in `unattend.txt`.

```

ChDir "z:\w2k\w2kpro\us\inst\I386"
;save in the environment
;echo "Calling the setup program..."
Message 267 "Windows 2000"
SET BOOT="NTLAUNCH"
;pause
Z:.\winnt /u:c:\ibmwin32\unattend.txt /s:z:
;pause
if %BOOT%="NTLAUNCH"
Log "Installation Failed...."
ErrorExit
else
;Echo "Applying last copies before returning control to the windows
setup program"
Message 313
ChDir %ClientRO%
DELAY 3
MCOPY /I:Z:FIXUP1.LST /L:COPY.OUT
;Echo "Waiting for the server to switch our boot mode to local...."
Message 309
delay 2
if SETBOOT "LOCAL" "machine1"
Log %ErrSetBoot%
ErrorExit
Endif
;Echo "Local boot mode switched, rebooting..."
Message 307
delay 2
Message 263
delay 3
REBOOT
spin
Log %ErrBootMode%
ErrorExit
endif
endif
;if We come back here then something wrong occurred with the Windows 2000
installation
;Echo "This client should have booted locally...an error occurred on the
server"
spin
Log %ErrBootMode%
ErrorExit

```

Figure 64. The install section of Windows 2000 state.ini

When the installation based on `unattend.txt` has completed, `state.exe` uses the file `fixup1.lst` to copy any last fixes to the Windows 2000 installation. `Fixup1.lst` references `txtsetup.sif` and `cmdlines.txt`, which will cause additional files to be copied after the system has rebooted. The commands listed in `cmdlines.txt` will be executed after the last reboot prior to the first user logon.

Finally, when the installation is complete, the client machine remains in its hybrid state and boots to the IBM WorkSpace On-Demand 3.0.1 logon window. When the user logs on, the last copy operations are completed and registry entries set, causing one last reboot of the client machine. The IBM WorkSpace On-Demand 3.0.11 logon window is shown again upon reboot, and the system is ready for the user to log on to their Windows 2000 desktop.

#### 4.2.3.9 The machines directory in WRKFILES

During the entire boot process, `state.exe` must keep track of the client machine's state and log its progress. It does this through the `WRKFILES` alias, defined as `\tdm\mm\client\rw`. Here, the directory `\machines\machine0\w2k\w2kpro\us` contains the work files that need to be written during the installation. A listing of this directory is shown in Figure 65

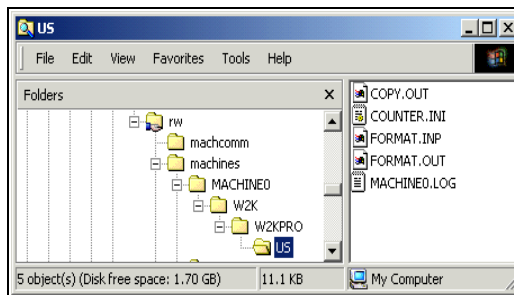


Figure 65. Contents of the `WRKFILES` alias for a Windows 2000 client

The files `format.out` and `copy.out` contain the results of the hard drive format and all `copy` commands of files to the client hard drive. `Format.inp` simply contains the letter `y` to respond to the “Are you sure?” prompt normally presented by the `format` command.

The file `counter.ini` records the state transitions made by `state.exe` during the entire installation process. Within this file is a variable called `state`, which changes value as `state.exe` changes from creating the partition to formatting, copying, and finally installing the operating system. After the user's first logon, the final value of the state variable is 5, indicating that the installation of the client machine has completed.

---

## 4.3 Defining Windows NT client machines

If you have installed Windows NT client support in IBM WorkSpace On-Demand 3.0.1 using the Windows NT Workstation CD, you will be able to define machines with the Windows NT operating system. These clients do not perform a remote boot from the server, rather, they perform a remote install from the server once the machine definition is complete. The Windows NT image that was installed in IBM WorkSpace On-Demand 3.0.1 is used as a template to install a custom Windows NT image on your client machine. The following sections describe the steps involved in creating and booting a Windows NT client machine.

### 4.3.1 Windows NT client parameters

A Windows NT client is created with the `machine define` command. Among all the parameters that can be specified for this command the following are mandatory:

OS, LANG, NAME, SERVER, IMAGENAME, MAC, REGUSER, CDKEY, TZ, and NETADAPTER

#### **Name**

This is the name that will uniquely identify the client machine on the deployment server. For example, `machine0`, `machine1`, `machine2`, and `machine3` are valid machine names as shown in Figure 55 on page 104. The machine name must follow the 8.3 Naming Convention. This parameter is required when creating, deleting, or modifying a machine. Our sample Windows NT client will be named `machine1`.

#### **MAC**

This parameter specifies the MAC address of the client machine's network adapter card. It must be specified when creating a machine. The MAC address is expressed as 12 hexadecimal digits and results in a directory structure, such as `macs\0020\35fe4a7b`, as shown in Figure 55 on page 104.

#### **Server**

This parameter specifies the name of the deployment server where the client machine will be created. It is required for machine creation only.

#### **OS**

The OS parameter lets you select the operating system image for your client. This can be NT4, W98, or OS2. This parameter must be specified for creating, deleting, or modifying a machine. For our sample Windows NT client, we will set `OS=NT4`.



**ImageName**

ImageName must correspond to the image name you selected when you installed the client operating system support. For example, if installing Windows NT Workstation with Service Pack 3 already applied, the ImageName value might be SP3. This parameter must be specified when creating, deleting, or modifying a machine.

**Lang**

This parameter sets the language of the operating system that will be used for this client machine. It must correspond to the language selected when the client operating system support was installed. Lang is specified as a two-character string (for example: "US" for American English). This parameter is required when defining a machine and cannot be changed after definition.

**CDKey**

This is the CD key supplied by Microsoft for each copy of Windows NT. It is either a 10-digit value expressed as xxx-xxxxxxx for retail versions of Windows NT or a 21-character value expressed as xxxxx-oem-xxxxxxx-xxxxx for OEM versions of Windows NT. In both cases, x represents a decimal digit only. This parameter is required for creation of a new machine.

**TZ**

The TZ parameter specifies the time zone to be used on the client machine. This parameter is required when creating a machine and must be specified as a string in exactly the same format as would normally be selected from the Windows Date/Time window. For example, to set Central time, you would specify:

```
TZ='(GMT-06:00) Central Time (US & Canada)'
```

**Template**

This is a yes or no parameter that specifies whether the machine is to be defined as a template. The default value is no. If you specify Template=yes, you can create other machines from this template. Using templates is discussed in Section 4.6, "Machine templates" on page 177.

**Templatename**

This specifies the name of the template to be used when creating this machine. If you create a machine from a template, all machine attributes will be taken from the template. Using templates is discussed in Section 4.6, "Machine templates" on page 177.

**Video**

This parameter identifies the video adapter for the client machine. It can be set to VGA, Auto-Detected, or a specific video driver name. If you select

Auto-detected, or do not set this parameter, the Windows NT installation procedure will try to detect and install the appropriate video driver for you.

If your video adapter is not directly supported by Windows NT, you can specify the name of the video driver that you need to have installed. When you specify your own video driver, you must add this driver to the Windows NT client image on your server. The Windows NT client image can be found in `\tdm\client\ro`. Here, you need to create a new subdirectory in `nt4\sp3\us\inst\i386\%soem%`. (If you installed the Windows NT client image with a different operating system name, image name, or language, replace `nt4`, `sp3`, and `us` in the directory specification.) Create a directory called `display`, and within `display`, create another directory to hold your video driver, namely the `.DLL`, `.INF` and any other files you might need. The name of this directory will be the name you set in the `Video=` parameter. Then, you must also set the `Videoinf=` parameter to point to your driver's `.INF` file.

### **Videoinf**

If you specified a particular video driver in the `Video` parameter, use this parameter to specify the name of the video driver `.INF` file.

### **Resolution**

Video resolution is a required parameter upon machine definition and is specified as a string in the format "`XxYxColors@Refresh`". For example, a video resolution of 800 by 600, with 256 colors, and a refresh rate of 70 MHz would be written as `Resolution="800x600x256@70"`. If the resolution you select is unsupported, the system will default to VGA.

### **Netadapter**

This parameter must be specified upon machine creation and indicates the type of network adapter in the client machine. Valid values for this parameter are taken from the `$oem$\net` directory in the Windows NT client image. In our default installation of IBM WorkSpace On-Demand 3.0.1, there were two subdirectories under `$oem$\net`, called `TRP` for the IBM 16/4 PCI Token-Ring adapter with Wake on LAN, and `IBMFE` for the IBM 10/100 PCI Ethernet adapter. This yielded two possible valid values for the `Netadapter` parameter, `TRP` or `IBMFE`.

IBM WorkSpace On-Demand 3.0.1 requires client network adapters to support the PXE 2 network boot specification. Network adapters that support this will likely need to have their drivers added to the Windows NT client image. You can do this by adding another subdirectory under `$oem$\net` in your client image and copying your network adapter driver files to this new directory. Then, specify the directory name in the `Netadapter` parameter.

**DHCP**

DHCP is a yes or no parameter. If you set DHCP to no, you will have to specify the client machine's IP address and other TCP/IP parameters explicitly. Note that even if you set DHCP to no, a DHCP server is still required on your network since the PXE boot protocol requires DHCP for the client to boot at all.

**IP**

This parameter is required if you set DHCP=No. Set the client machine's IP address in standard IP address format. For example: IP='nnn.nnn.nnn.nnn'.

**Netmask**

This parameter allows you to specify the client machine's subnet mask and is required if you set DHCP=No.

**Tcpname**

This parameter specifies the machine's TCP/IP host name. It is required if you set DHCP=No.

**Tcprouter**

Use this parameter to set the IP address of the IP router in standard IP address format. This parameter is required if you set DHCP=No.

**Tcpnamesrv**

This parameter is required if you set DHCP=No and is used to specify the IP address of the Primary Domain Name Server.

**Tcpdomain**

This parameter is required if you set DHCP=No and is used to set the IP domain name.

**WINS**

This optional parameter specifies the IP addresses of the WINS servers. You can specify a list of IP addresses separated by commas.

**Protocols**

This parameter is specified as a comma-delimited string and is used to turn on various network protocols. Supported protocol values for Windows NT are NBF, NWLNKIPX, TC, DLC, RASPTTP, STREAMS, and ATALK. A typical protocol setting would be Protocols="TC,NBF".

**NBScope**

This parameter specifies the scope ID of NetBIOS over TCP/IP. It is optional and should normally not be specified. The scope can be set as a character

string and is used to configure different logical NetBIOS networks on the same TCP/IP network.

### **Status**

A machine's status may be set as enabled or disabled. If a client machine is disabled, it will be prevented from booting. Note that in the GUI, the status is specified by a checkbox labeled Disable Client.

### **JVM**

The JVM parameter lets you specify whether a Java Virtual Machine should be installed with the client operating system. Valid values are Yes or No. Since Java applications are becoming pervasive, it is usually a good idea to install the JVM for all clients.

### **TMA**

The TMA parameter allows you to specify whether the Tivoli Management Agent should be installed to the client workstation. This value is set as yes or no.

### **Logon**

The logon parameter also is a yes or no parameter. If you specify no, the client machine will present only a Windows NT Workstation local logon window when it boots up after installation. It is recommended to set Logon="Yes" so that the client machine will present the IBM WorkSpace On-Demand 3.0.1 logon window instead, which will log on to the server as well as locally.

### **PrebootImage**

This parameter specifies the DOS boot image that will be used prior to installing Windows NT. The pre-boot DOS image provided with IBM WorkSpace On-Demand 3.0.1 for Windows NT clients is `tdmw32ut.img` and can be found in the boot subdirectory of the Windows NT client image installed in `\tdm\mm`. This boot image uses NetBIOS calls over TCP/IP to communicate with the deployment server to copy the Windows NT installation files.

### **Installdos**

If you specify `Installdos=Yes`, then when the client hard drive is formatted as part of the install, it is formatted with the `/s` option and DOS is installed. This is only used to allow network adapter drivers that cannot coexist with Windows to be deleted from memory by booting DOS first with no network support. This keyword should only be used sparingly and is not needed for most network adapters. Specify `Installdos=No` to have no DOS installation on the hard drive. More information about the significance of `Installdos` can be

found in Section 4.3.3, “Windows NT client installation and boot” on page 131.

### **Partition**

This parameter specifies the size of the primary partition for the client machine. It is set in megabytes, so specifying `Partition='600'` will create a primary partition of 600 MB. You can also specify the value 'all', which will create a partition of the maximum possible size on the client machine's hard drive (but no larger than 2048 MB). You cannot specify a partition size greater than 2048 MB when you define a Windows NT machine. Bypassing this restriction is discussed in Section 4.9, “Customizing Windows client machine images” on page 191.

### **Srcrespfile**

The `srcrespfile` parameter indicates the name of the response file used to guide the unattended installation of Windows NT Workstation on the client machine. This could be `unattend.txt` or any other customized response file for the specific client. For example, we used `srcrespfile='ibmfepci.txt'` and customized this response file for various IBM PC300 family PCs.

### **Reguser**

The `reguser` parameter lets you specify the name of the registered Windows NT user for the client workstation. This is specified as text and used in the Windows NT installation. Although this parameter is optional, if it is not specified, the Windows NT installation will pause and prompt for a registered user name to be entered. For a proper unattended installation, therefore, it is recommended to set this parameter with some value, such as `reguser='ITSO'`.

### **Localadmpw**

This optional parameter lets you set a password for the local administrator user ID on the client machine.

### **Description**

This optional parameter lets you specify a text description of the client machine that will be shown when the machine is queried or a list of machines is displayed.

### **Section**

This parameter lets you optionally define additional sections for the Windows NT response file, `unattend.txt`. The format for this parameter is:

```
Section='{SectionName, Key=value,...},{...}...
```

You can set as many sections and key-value pairs as you want.

### Printern

In this parameter, n is a number from 1 to 9, such as printer1= or printer2=. Use this parameter to specify the printer available to the client machine. The printer is specified using a comma-delimited string as follows:

```
printer1="PrinterName, ModelName, PortUNCName, driver, datafile,  
configfile"
```

The printer must be defined locally on your server and must be shared so that the value of PortUNCName is the network name of your server and the alias of the printer. To find the correct values for the driver, datafile, and configfile, it is easiest to use regedit and look up the printer driver's registry entries on the server. For example, the following string could be used to add support for a 4029 laser printer to your Windows NT client machine:

```
printer1="ibmprn, 4029, \\atlas2\ibm4029l, rasdd.dll, ibmppsdl.dll,  
rasddui.dll"
```

### Kbdtype

Use this parameter to set the type of keyboard for the client machine. The keyboard type is represented as a string, for example a US keyboard would be specified as kbdtype='United States 101'.

## 4.3.2 The machine define command for Windows NT

When the Windows NT Workstation operating system support has been successfully installed in IBM WorkSpace On-Demand 3.0.1, you can use the define action on the machine object with the parameters as previously described to define a machine using the console CLI. Figure 66 shows the machine define command for a typical token-ring based client.

```
Machine define OS='NT4' NETADAPTER='TRP' DHCP='Yes' LANG='us'  
PROTOCOLS='TC,NBF' DESCRIPTION='ITSO Test' RESOLUTION='800x600x256@70'  
STATUS='Enabled' JVM='Yes' LOGON='Yes' TMA='No' IMAGE_NAME='sp3'  
VIDEO='Auto-detected' CDKEY='xxx-xxxxxxx' TZ='(GMT-06:00) Central Time  
(US & Canada)' PREBOOTIMAGE='tdmw32ut.img' INSTALLDOS='No'  
LOCALADMPW='getin2see' partition='800' REGUSER='ITSO'  
srcrespfile='ibmfepci.txt' Name='machine1' MAC='002035FE4A7B'  
SERVER='atlas2'
```

Figure 66. Defining a Windows NT machine in the CLI

As discussed in Chapter 3, "Administration" on page 57, defining a machine by typing machine define at the console is cumbersome and error-prone, while defining a machine through the GUI is slow and not effective for managing

remote servers. You can make reasonable assumptions about many of the parameters for `machine define` and encapsulate this action in a JavaScript function that only exposes those parameters that do need to change for every machine. For example, the JavaScript function `CreateMachinet()` defined in the `machines.js` file provided on the enclosed CD allows you to create a token-ring based client machine by specifying only the machine name, MAC address, deployment server name, and operating system name. This allows you to simplify the `machine define` command shown in Figure 66 on page 130 to the following:

```
CreateMachinet("machine1", "002035FE4A7B", "atlas2", "nt4")
```

It is also possible to simplify machine creation using machine templates as discussed in Section 4.6, “Machine templates” on page 177.

### 4.3.3 Windows NT client installation and boot

Once you have successfully defined a Windows NT client machine, you will notice many changes in the IBM WorkSpace On-Demand 3.0.1 directories to represent the new machine. The directories and files that have been created will direct the process of remotely installing Windows NT on the client machine. We investigate several of these changes here.

#### 4.3.3.1 The data store entry

First, a new entry is created in the data store for the new machine. The data store is kept as a text file called `datastor.ini` in `\tdm\config`. Since a full definition of a machine needs to include not just the machine's name but also the operating system name, image name, and language, the data store contains sections for each possible combination of values. All the parameters that you specified when defining the machine are listed under the machine's name in the data store. When a machine is queried, its parameter information is read from the data store and any modifications made using the `machine modify` action are saved here as well.

```

[/MACHINE]

[/MACHINE/NT4]

[/MACHINE/NT4/sp3]

[/MACHINE/NT4/sp3/US]

[/MACHINE/NT4/sp3/US/machine1]
os=NT4
netadapter=TRP
dhcp=Yes
lang=us
protocols=TC,NBF
description=ITSO Test
resolution=800x600x256@70
status=Enabled
jvm=Yes
logon=Yes
tma=No
imagename=sp3
video=Auto-detected
cdkey=xxx-xxxxxxx
tz=(GMT-06:00) Central Time (US & Canada)
prebootimage=tdmw32ut.img
installdos=No
localadmpw=getin2see
partition=800
reguser=ITSO
srcrespfile=ibmfepci.txt
name=machine1
mac=002035FE4A7B
SERVER=ATLAS2
associated class=com.ibm.tdm.task.machine.NT4Machine

```

Figure 67. A Windows NT machine definition in the data store

Figure 67 shows the data store representation of the Windows NT machine definition given in Figure 66 on page 130. Other machine definitions, including those for other operating systems, would be listed in the appropriate subsections of the data store. Although the data store is currently a text file, its format may change with future releases of IBM WorkSpace On-Demand 3.0.1; therefore, you should never change a machine definition directly in the data store but use the `machine modify` action instead.



#### 4.3.3.2 The MACS directory

The next change to notice is that a new subdirectory has been created for the MAC address of the client machine in `tdm\mm\client\ro\macs`. In the case of machine1 defined in Figure 66 on page 130, this subdirectory is `\0020\35fe4a7b`. When the client machine first boots, IBM WorkSpace On-Demand 3.0.1 will use the client's MAC address to locate this directory. Two files are found here, `tdmclnt.inf` and `tdmdos.inf`. These files were copied from the directory `client\ro\nt4\sp3\us\defs` and customized by the `machine define` process so that they are specific to the client machine we created. They are used to locate the initial DOS bootable image for the client machine. The .INF files corresponding to our example machine definition are shown in Figure 68.

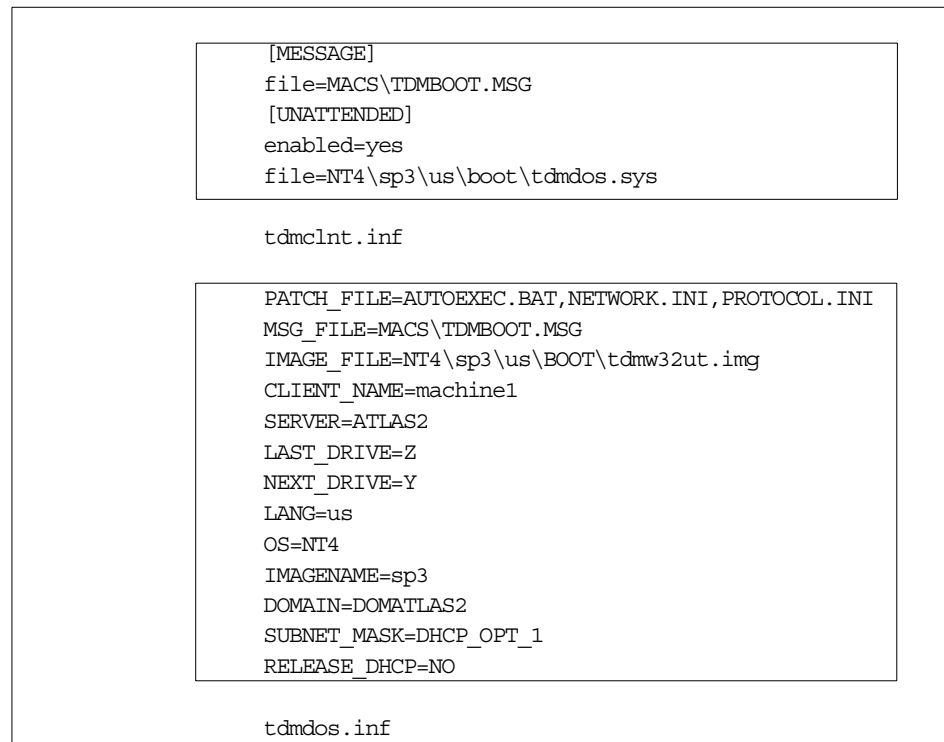


Figure 68. `tdmclnt.inf` and `tdmdos.inf` for a Windows NT client

During the first boot, IBM WorkSpace On-Demand 3.0.1 first locates the file `tdmclnt.inf` and transfers it to the client machine via TFTP. The client machine's status is listed in this file; the line `enabled=yes` reflects the parameter setting `status=enabled` in the `machine define` command. If a

machine were created with disabled status, its boot would not proceed beyond this point.

The line `file=` in `tdmcInt.inf` points to the second boot image, `tdmdos.sys`. This image is copied to the client machine via TFTP and the client boots. This is a DOS boot that will use the file `tdmdos.inf` to locate the actual operating system image that is defined for this client machine. This second boot goes back to the server, locates `tdmdos.inf`, and transfers it to the client via TFTP.

The `prebootimage` parameter specified on the `machine define` command is listed in `tdmdos.inf` as the `IMAGE_FILE` variable. The path specified in `IMAGE_FILE` is relative to the alias `RPLFILES`, or `tdm\mm\client\ro`. This combination completely specifies the path of the Windows NT Workstation client support installation on the server, where the DOS boot image files are installed. This DOS boot image is now transferred to the client machine via TFTP and the client machine's own directories are located on the server.

These directories are located on the server by means of the `SERVER`, `OS`, `IMAGENAME`, `LANG`, and `CLIENT_NAME` lines defined in `tdmdos.inf`. In our example, these correspond to subdirectories called `machine1` in the `machines` directory of the alias `RPLFILES`, or `tdm\mm\client\ro`, and in the `machines` directory of the alias `WRKFILES`, or `tdm\mm\client\rw`, respectively. The `machine1` subdirectory in `rw` will contain logging information from the installation of Windows NT on the client machine. The `machine1` subdirectory in `ro` contains all the control files needed to install Windows NT on the client machine.

#### **4.3.3.3 The machines directory in RPLFILES**

Figure 69 on page 135 shows a listing of the contents of the `machine1` directory for our sample machine under the `machines` directory in the `RPLFILES` alias. When the DOS pre-boot image is now booted, it uses the control files in this directory to direct the installation of Windows NT on the client. These control files were copied and customized from defaults found in the directory `\nt4\sp3\us\defs`, also in the `RPLFILES` alias. The DOS boot image establishes connections to both the `ro` and `rw` aliases and uses the utilities in the `\dos\2k\us\tree` subdirectory under the `RPLFILES` alias to perform the installation. An installation shell is started from the DOS image that makes repeated calls to the program `state.exe`. This program uses the file `state.ini` found in `\machines\machine1\Nt4\sp3\us` to determine each successive step of the installation process.

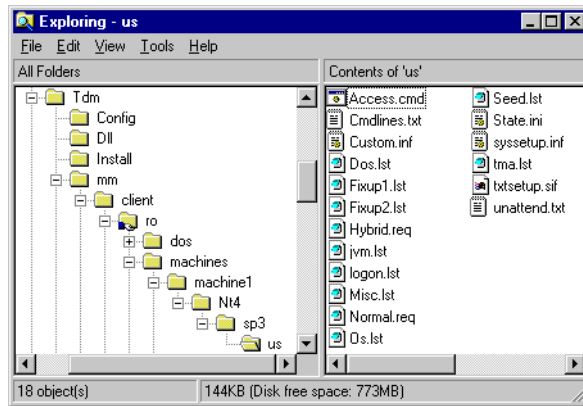


Figure 69. Contents of a Windows NT machine directory after creation

#### 4.3.3.4 The state.ini file

The state.ini file is key to the boot and install process from this point on. State.ini was customized from the default provided with the Windows NT client support. State.exe processes the contents of state.ini via a series of steps, known as states, that are listed in the StateNames section of state.ini. These states are:

|                |  |
|----------------|--|
| <b>Init</b>    | Initializes the installation.                      |
| <b>New</b>     | Partitions the client hard drive.                  |
| <b>Format</b>  | Formats the hard drive.                            |
| <b>Copy</b>    | Copies system files prior to installation.         |
| <b>Install</b> | Installs Windows NT on the client machine.         |
| <b>Hybrid</b>  | Client machine has been installed; reboot locally. |
| <b>Error</b>   | Processes errors from any other states.            |

```

[parms]
PartitionSize=800
InstallDOS=No
MinMemSize=305
MinDiskSize=350
DSize=0
VideoCorrection="NO"
CleanupOSFiles=Yes
InitDiskParms=0
msg1=0
msg2=0
OSDISK=C:
BootImageApps=JVM, LOGON
ClientType=NO

[StateNames]
Init
New
Format
Copy
Install

```

Figure 70. The Windows NT header of *state.ini*

The first few lines of the *state.ini* from our sample machine are shown in Figure 70. The *[parms]* section at the top of *state.ini* contains parameter information that may be substituted in commands listed in other state sections farther down in the file. Some of these parameters were set based upon information provided as parameters in the `machine define` command. In particular, the parameters `PartitionSize`, `InstallDOS`, and `BootImageApps` should look familiar.

For each state, there is a section of *state.ini* labelled with the state name. Each state section contains commands that are executed by *state.exe*. These commands are of six possible types. Listed in their search order, these are:

- Internal commands to *state.exe*
- Internal commands to the install shell
- DOS internal commands
- DOS external commands or executables
- DOS batch files
- Install shell batch files

#### 4.3.3.5 Section [new] of state.ini

Figure 71 illustrates the structure of state.ini by showing one complete section of the file. The section is labelled [new] and thus corresponds to the second state, named new, that directs the install shell to partition the hard drive according to parameters specified by the `machine define` command. State.ini determines the hard drive size and ensures that the disk is at least 350 MB, which is big enough to allow the installation of Windows NT. The value of the `dsize` variable is logged to the file `counter.ini`, found in machine1's directory in `tdm\mm\client\rw`. In the case of our example machine1, the hard drive size was 6 GB.

```
[new]
dsize=disksize
Message 260 %Dsize%
delay 2
if %dsize% > %MinDiskSize%
if %PartitionSize%="all"
if %dsize%>2048
PartitionSize=2048
else
PartitionSize=%dsize%
endif
else
if %partitionSize%>2048
Log %InvPartSize%
ErrorExit
endif
endif
else
Log %HardDiskTooSmall%
ErrorExit
endif
;Creating a partition of %d Megabytes
Message 261 %PartitionSize%
delay 2
InitdiskParms="/PRI:"&%PARTITIONSIZE%
initdisk 1 %InitDiskParms%
Message 285
delay 5
;echo "Rebooting the system...."
Message 263
delay 3
;pause
reboot
```

Figure 71. Sample partition command section of state.ini

The commands in the [new] section direct state.exe to check what partition size was specified by the `machine define` command. If the administrator had specified a value of all, the partition size is set to the smaller of the physical disk size or 2 GB. If an actual partition size was specified, that size is used or an error is logged if the requested size is larger than 2 GB.

The `initdisk` command is used to create the primary partition of the appropriate size, and when this succeeds, the client machine is rebooted. The state is promoted to state 3, [Format].

#### 4.3.3.6 Section [Format] of state.ini

This state causes the newly created partition to be formatted. The commands for this section are shown in Figure 72. Notice that the commands in this section use the `InstallDOS` parameter from the `machine create` command. If `InstallDOS="Yes"` was specified, or if state.exe determines that the amount of conventional memory on the client machine is insufficient, DOS is installed by using the `/s` parameter on `format`.

```
[Format]
;Save the DOS files in case we need to use them
if %InstallDos%="No"
if convmem < %MinMemSize%
InstallDos="Yes"
endif
endif
;echo "Formatting the partition...please wait..."
Message 264
FormatCustomize "on"
OutputFile "Format.out"
if %InstallDOS%="Yes"
FormatDisk %OSDISK% "/U /V:WNT /s"
else
FormatDisk %OSDISK% "/U /V:WNT /b"
endif
FormatCustomize "off"
;echo "Format Completed Successfully"
Message 265
delay 3
;pause
```

Figure 72. The format section of state.ini for Windows NT and Windows 98

When the format has successfully completed, there is no reboot specified because it is not necessary. State.exe executes a transition from the [Format]

state to the [Copy] state and copies system files from the server to the client. The [Copy] state for the sample machine machine1 is shown in Figure 73.

#### 4.3.3.7 Section [Copy] of state.ini

Four `copy` commands are executed from this section, since the `InstallDOS` parameter had been set to `No`. These commands copy the files for the client Java Virtual Machine, the IBM WorkSpace On-Demand 3.0.1 logon program, the Tivoli management agent, and other miscellaneous files to the client C: drive. These `copy` commands are directed via list files. The Z: drive letter prefacing each list file name maps to `\tdm\mm\client\ro`, also known as the alias `RPLFILES`. Prior to the `copy` commands, the `chdir` command sets the rest of the path to find the list files to be `\machines\machine1\nt4\sp3\us`. Notice that the Tivoli files are copied to the client even though the machine definition had specified `TMA=No`. The files are copied, but the Tivoli Management Agent is simply not enabled as an application for this machine.

```
[Copy]
;Load disk caching software to speed the copying of NT files
;echo "Copying Client files...please wait..."
Message 266 "Windows NT"
cachedisk
chdir %ClientRO%
CopyCustomize "On"
MCOPY /I:Z:JVM.lst /l:y:copy.out /s /r /e
MCOPY /I:Z:Logon.lst /s /r /e /l:copy.out
MCOPY /I:Z:TMA.lst /s /r /e /l:copy.out
MCOPY /I:Z:Misc.lst /s /r /e /l:copy.out
;pause
if %InstallDOS%="Yes"
MCOPY /I:Z:OS.lst /s /r /e /l:copy.out
MCOPY /I:Z:DOS.lst /s /r /e /l:copy.out
MCOPY /I:Z:Fixup2.lst /s /r /e /l:copy.out
endif
;echo "Client Files have been copied sucessfully"
Message 269
delay 3
CopyCustomize "Off"
```

Figure 73. The copy section of Windows NT state.ini

The miscellaneous `copy` commands specified in `misc.lst` copy three files from the server to the client. These are `unattend.txt` and `custom.inf`, which are used to drive the installation of Windows NT, and `access.cmd`, which is used to define the local administrator ID on the client machine.

#### 4.3.3.8 Section [Install] of state.ini

When the `copy` commands have completed, the client system is now ready for the Windows NT installation. `State.exe` exits the [Copy] section of `state.ini` and enters [Install]. At the beginning of this phase, `state.exe` checks to see if DOS was installed on the client hard drive due to the `machine define` parameter `Installdos` having been set to Yes. If this is true, it means that there is insufficient conventional memory in the client machine to run the Windows NT installation programs along with network drivers to perform the installation directly from the server. Therefore, `state.exe` will copy the rest of the DOS system files as well as the entire Windows NT installation directory to the client machine's local hard drive. It will then boot into DOS with no network connections and begin the installation of Windows NT. When the installation is finished, the client machine will reboot and restore its network connections. In most cases, however, a local DOS-based install procedure is not required and Windows NT installation can proceed directly from the server.

When `state.ini` proceeds with the installation of Windows NT, the client machine's state is set to hybrid. A state value of hybrid indicates that the client machine is able to boot from its local hard drive. All client machines in IBM WorkSpace On-Demand 3.0.1 always perform a network boot first when they are rebooted to allow the server to control the machine and perform updates or installations as necessary. When a machine is in the hybrid state and it performs its network boot, the `tdmclnt.inf` file now points to `hdboot.com` instead of `tdmdos.sys`. This redirects the client's boot to the local hard drive.

The lines of `state.ini` describing the network-based installation are shown in Figure 74 on page 141. `State.exe` begins the installation using the `unattend.txt` file, which was copied from the server to the client's hard drive. `Unattend.txt` was customized by the `machine define` command to contain the rest of the parameters specified when the client machine was created. For example, the user information, CD key, time zone, video, display resolution, and network protocol information specified by `machine define` are all listed in `unattend.txt`.



```

ChDir "z:\nt4\sp3\us\inst\I386"
;save in the environment
;echo "Calling the setup program..."
Message 267 "Windows NT"
SET BOOT="NTLAUNCH"
;pause
Z:.\winnt /u:c:\ibmwin32\unattend.txt /s:z:
;pause
if %BOOT%="NTLAUNCH"
Log "Installation Failed...."
ErrorExit
else
;Echo "Applying last copies before returning control to the windows
setup program"
Message 313
ChDir %ClientRO%
DELAY 3
MCOPY /I:Z:FIXUP1.LST /L:COPY.OUT
;Echo "Waiting for the server to switch our boot mode to local...."
Message 309
delay 2
if SETBOOT "LOCAL" "machine1"
Log %ErrSetBoot%
ErrorExit
Endif
;Echo "Local boot mode switched, rebooting..."
Message 307
delay 2
Message 263
delay 3
REBOOT
spin
Log %ErrBootMode%
ErrorExit
endif
endif
;if We come back here then something wrong occurred with the NT
installation
;Echo "This client should have booted locally...an error occurred on the
server"
spin
Log %ErrBootMode%
ErrorExit

```

Figure 74. The install section of Windows NT state.ini

When the installation based on `unattend.txt` has completed, `state.exe` uses the file `fixup1.lst` to copy any last fixes to the Windows NT installation. `Fixup1.lst` references `txtsetup.sif` and `cmdlines.txt`, which will cause additional files to be copied after the system has rebooted. The commands listed in `cmdlines.txt` will be executed after the last reboot prior to the first user logon.

Finally, when the installation is complete, the client machine remains in its hybrid state and boots to the IBM WorkSpace On-Demand 3.0.1 logon window. When the user logs on, the last copy operations are completed and registry entries set, causing one last reboot of the client machine. The IBM WorkSpace On-Demand 3.0.1 logon window is shown again upon reboot, and the system is ready for the user to log on to their Windows NT desktop.

#### 4.3.3.9 The machines directory in WRKFILES

During the entire boot process, `state.exe` must keep track of the client machine's state and log its progress. It does this through the `WRKFILES` alias, defined as `\tdm\mm\client\rw`. Here, the directory `\machines\machine1\nt4\sp3\us` contains the work files that need to be written during the installation. A listing of this directory is shown in Figure 75.

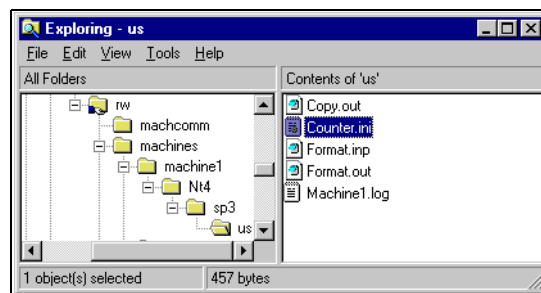


Figure 75. Contents of the `WRKFILES` alias for a Windows NT client

The files `format.out` and `copy.out` contain the results of the hard drive format and all `copy` commands of files to the client hard drive. `Format.inp` simply contains the letter `y` to respond to the “Are you sure?” prompt normally presented by the `format` command.

The file `counter.ini` records the state transitions made by `state.exe` during the entire installation process. Within this file is a variable called `state`, which changes value as `state.exe` changes from creating the partition to formatting, copying, and finally installing the operating system. After the user's first logon, the final value of the `state` variable is 5, indicating that the installation of the client machine has completed.

---

## 4.4 Defining Windows 98 client machines

After you have installed Windows 98 client support in IBM WorkSpace On-Demand 3.0.1 using the Windows 98 CD, you will be able to define machines with the Windows 98 operating system. Just as for Windows NT clients, these machines do not perform a remote boot from the server; rather, they perform a remote install from the server once the machine definition is complete. The Windows 98 image that was installed in IBM WorkSpace On-Demand 3.0.1 is used as a template to install a custom Windows 98 image on your client machine. The following sections describe the steps involved in creating and booting a Windows 98 client machine.

### 4.4.1 Windows 98 client parameters

A Windows 98 client is created with the `machine define` command. Among all the parameters that can be specified for this command the following are mandatory:

OS, LANG, NAME, SERVER, IMAGENAME, MAC, REGUSER, CDKEY, TZ, and NETADAPTER

#### **Name**

This is the name that will uniquely identify the client machine on the deployment server. For example, `machine0`, `machine1`, `machine2`, and `machine3` are valid machine names as shown in Figure 55 on page 104. The machine name must follow the 8.3 Naming Convention. This parameter is required when creating, deleting, or modifying a machine. Our sample Windows 98 machine will be named `machine2`.

#### **MAC**

This parameter specifies the MAC address of the client machine's network adapter card. It must be specified when creating a machine. The MAC address is expressed as 12 hexadecimal digits and results in a directory structure, such as `macs\0020\3503CAE0`, as shown in Figure 55 on page 104.

#### **Server**

This parameter specifies the name of the deployment server where the client machine will be created. It is required for machine creation only.

#### **OS**

The OS parameter lets you select the operating system image for your client. This can be NT4, W98, or OS2. This parameter must be specified for

creating, deleting, or modifying a machine. For our sample Windows 98 client, we set OS=W98.

**ImageName**

ImageName must correspond to the image name you selected when you installed the client operating system support. For example, when we installed the Windows 98 client image, we chose the image name w98se. This parameter must be specified when creating, deleting, or modifying a machine.

**Lang**

This parameter sets the language of the operating system that will be used for this client machine. It must correspond to the language selected when the client operating system support was installed. Lang is specified as a two-character string (for example: "US" for American English). This parameter is required when defining a machine and cannot be changed after definition.

**CDKey**

This is the CD key supplied by Microsoft for each copy of Windows 98. The Windows 98 CD key is a 25-digit field formatted as five groups of five characters, as follows: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX. Each character is alphanumeric. This parameter is required for creation of a new machine.

**TZ**

The TZ parameter specifies the time zone to be used on the client machine. This parameter is required when creating a machine and must be specified as a string in exactly the same format as would normally be selected from the Windows Date/Time window. For example, to set Central time you would specify: TZ='(GMT-06:00) Central Time (US & Canada)'.

**Template**

This is a yes or no parameter that specifies whether the machine is to be defined as a template. The default value is no. If you specify Template=yes, you can create other machines from this template. Using templates is discussed in Section 4.6, "Machine templates" on page 177.

**Templatename**

Specifies the name of the template to be used when creating this machine. If you create a machine from a template, all machine attributes will be taken from the template. Using templates is discussed in Section 4.6, "Machine templates" on page 177.

**Resolution**

Video resolution is a required parameter upon machine definition and is specified as a string in the format “XxYxColors@Refresh”. For example, a video resolution of 800 by 600, with 256 colors, and a refresh rate of 70 MHz would be written as `Resolution='800x600x256@70'`. If the resolution you select is unsupported, the system will default to VGA.

**Netadapter**

This parameter must be specified upon machine creation and indicates the type of network adapter in the client machine. Valid values are “TRP” for a token-ring adapter and “IBMFE” for Ethernet.

**Netadapterdir**

This parameter lets you specify the directory for additional network adapters (if needed). You may specify multiple drivers and directories. The format for this command is: `netadapterdir='driverid=path,driverid=path,...'`.

**DHCP**

DHCP is a yes or no parameter. If you set DHCP to no, you will have to specify the client machine's IP address and other TCP/IP parameters. Note that even if you set DHCP to no, a DHCP server is still required on your network since the PXE boot protocol requires DHCP for the client to boot at all.

**IP**

This parameter is required if you set DHCP=No. Set the client machine's IP address in standard IP address format, such as `IP='nnn.nnn.nnn.nnn'`.

**Tcpname**

This parameter specifies the machine's TCP/IP host name. It is required if you set DHCP=No.

**Netmask**

This parameter allows you to specify the client machine's subnet mask and is required if you set DHCP=No.

**Tcprouter**

Use this parameter to set the IP address of the IP router in standard IP address format. This parameter is required if you set DHCP=No.

**Tcpnamesrv**

This parameter is required if you set DHCP=No and is used to specify the IP address of the Primary Domain Name Server.

**Tcpdomain**

This parameter is required if you set DHCP=No and is used to set the IP domain name.

**WINS**

This optional parameter specifies the IP addresses of the WINS servers. You can specify a list of IP addresses separated by commas.

**Protocols**

This parameter is specified as a comma-delimited string and is used to turn on various network protocols. Supported protocol values for Windows 98 are DEC40, DEC40T, DEC50T, IPXODI, MSDLC, MSTCP, NDISBAN, NDTOKBAN, NETBEUI, NFSLINK, NWLINK, NWNLINK. A typical protocol setting would be Protocols='NETBEUI, MSTCP'.

**NBScope**

This parameter specifies the scope ID of NetBIOS over TCP/IP. It is optional and should normally not be specified. The scope can be set as a character string and is used to configure different logical NetBIOS networks on the same TCP/IP network.

**Status**

A machine's status may be set as enabled or disabled. If a client machine is disabled, it will be prevented from booting. Note that in the GUI, the status is specified by a checkbox labeled Disable Client.

**JVM**

The JVM parameter lets you specify whether a Java Virtual Machine should be installed with the client operating system. Valid values are Yes or No. Since Java applications are becoming pervasive, it is usually a good idea to install the JVM for all clients.

**TMA**

The TMA parameter allows you to specify whether the Tivoli Management Agent should be installed to the client workstation. This value is set as Yes or No.

**Logon**

The logon parameter is also specified as yes or no. If you specify no, the client machine will present only a Windows 98 local logon window when it boots up after installation. It is recommended to set Logon=Yes so that the client machine will present the IBM WorkSpace On-Demand 3.0.1 logon window instead, which will log on to the server as well as locally.

**PrebootImage**

This parameter specifies the DOS boot image that will be used prior to installing Windows NT. The pre-boot DOS image provided with IBM WorkSpace On-Demand 3.0.1 for Windows 98 clients is `tdmw32ut.img`, which can be found in the boot subdirectory of the Windows 98 client image installed in `\tdm\mm`. This boot image uses NetBIOS over TCP/IP calls to communicate with the deployment server to copy the Windows 98 installation files to the client machine.

**Installdos**

If you specify `Installdos=Yes`, when the client hard drive is formatted as part of the install, it is formatted with the `/s` option and DOS is installed. This is only used to allow network adapter drivers that cannot coexist with Windows to be deleted from memory by booting DOS first, with no network support. This keyword should only be used sparingly and is not needed for most network adapters. Specify `Installdos=No` to have no DOS installation on the hard drive. More information about the significance of `Installdos` can be found in Section 4.4.3, “Windows 98 client installation and boot” on page 149.

**Partition**

This parameter specifies the size of the primary partition for the client machine. It is set in megabytes, so specifying `Partition='600'` will create a primary partition of 600 MB. You can also specify the value `all`, which will create a partition of the maximum possible size on the client machine's hard drive but no larger than 2048 MB. You cannot specify a partition size greater than 2048 MB when you define a Windows NT machine. Bypassing this restriction is discussed in Section 4.9, “Customizing Windows client machine images” on page 191.

**Srcrespfile**

The `srcrespfile` parameter indicates the name of the response file used to guide the unattended installation of Windows NT Workstation on the client machine. This could be `unattend.txt` or any other customized response file for the specific client. For example, we used `srcrespfile='ibmfepci.txt'` and customized this response file for various IBM PC300 family PCs.

**Reguser**

The `reguser` parameter lets you specify the name of the registered Windows 98 user for the client workstation. This is specified as text and used in the Windows 98 installation. Although this parameter is optional, if it is not specified, the Windows 98 installation will pause and prompt for a registered user name to be entered. For a proper unattended installation, therefore, it is recommended to set this parameter with some value, such as `reguser='ITSO'`.

**Localadmpw**

This optional parameter lets you set a password for the local administrator user ID on the client machine.

**Description**

This optional parameter lets you specify a text description of the client machine that will be shown when the machine is queried or a list of machines is displayed.

**Options**

This parameter lets you set a list of optional components for Windows 98. Each component can be set to 1 or 0. A value of 1 causes the optional component to be installed. For example, you could specify

`OPTIONS="Paint=1, Mouse Pointers=1, Games=0"`. This would cause Paint and the mouse pointers to be installed but games not to be installed. There are many optional components to choose from; use the `machine parmlist` command described in Section 4.7, "Other machine commands" on page 179 to determine all the available options.

**Section**

This parameter lets you optionally define additional sections for the Windows 98 response file. The format for this parameter is:

```
Section='{SectionName, Key=value,...},{...}...'
```

You can set as many sections and key-value pairs as you want.

**Printern**

In this parameter, *n* is a number from 1 to 9 such as `printer1=` or `printer2=`. Use this parameter to specify the printer available to the client machine. The printer is specified using a comma-delimited string, as follows:

```
printer1='PrinterName, ModelName, PortUNCName'
```

The printer must be defined locally on your server and must be shared so that the value of `PortUNCName` is the network name of your server and the alias of the printer. For example, the following string could be used to add support for a 4029 laser printer to your Windows 98 client machine:

```
printer1="ibmpmn, 4029, \\atlas2\ibm4029l"
```

**Kbdtype**

Use this parameter to set the type of keyboard for the client machine. The keyboard type is represented as a string; for example, a US keyboard would be specified as `kbdtype='United States 101'`.



#### 4.4.2 The machine define command for Windows 98

When the Windows 98 operating system support has been successfully installed in IBM WorkSpace On-Demand 3.0.1, you can use the `define` action on the machine object with the parameters as previously described to define a machine using the console CLI. Figure 76 shows the `machine define` command for a typical token-ring-based client.

```
Machine define OS='W98' DHCP='Yes' LANG='us' DESCRIPTION='ITSO Test'
STATUS='Enabled' RESOLUTION='800x600x256@70' JVM='Yes' LOGON='Yes'
TMA='No' IMAGE_NAME='w98se' NETADAPTER='TRP' PROTOCOLS='NETBEUI, MSTCP'
CDKEY='xxxxx-xxxxx-xxxxx-xxxxx-xxxxx' TZ='(GMT-06:00) Central Time (US
& Canada)' PREBOOTIMAGE='tdmw32ut.img' INSTALLDOS='No' partition='600'
srcrespfile='ibmfepci.inf' reguser='test98' Name='machine2'
MAC='00203503CAE0' SERVER='atlas2'
```

Figure 76. Defining a Windows 98 machine in the CLI

As discussed in Chapter 3, “Administration” on page 57, defining a machine by typing `machine define` at the console is cumbersome and error-prone, while defining a machine through the GUI is slow and not effective for managing remote servers. You can make reasonable assumptions about many of the parameters for `machine define` and encapsulate this action in a JavaScript function that only exposes those parameters that do need to change for every machine. For example, the JavaScript function `CreateMachinet()` defined in the file `machines.js` provided on the enclosed CD allows you to create a token-ring based client machine by specifying only the machine name, MAC address, deployment server name, and operating system name. This allows you to simplify the `machine define` command shown in Figure 66 on page 130 to the following:

```
CreateMachinet("machine2", "00203503CAE0", "atlas2", "w98")
```

It is also possible to simplify machine creation using machine templates as discussed in Section 4.6, “Machine templates” on page 177.

#### 4.4.3 Windows 98 client installation and boot

Once you have successfully defined a Windows 98 client machine, you will notice many changes in the IBM WorkSpace On-Demand 3.0.1 directories to represent the new machine. Just as was discussed for Windows NT clients in Section 4.3.3, “Windows NT client installation and boot” on page 131, the directories and files that have been created will direct the process of remotely installing Windows 98 on the client machine. We investigate several of these

changes here. Much of the client installation process for Windows 98 follows the same sequence as for Windows NT.

#### **4.4.3.1 The data store entry**

First, a new entry is created in the data store for the new machine. The data store is kept as a text file called `datastor.ini` in `\tdm\config`. Since a full definition of a machine needs to include not just the machine's name but also the operating system name, image name, and language, the data store contains sections for each possible combination of values. All the parameters that you specified when defining the machine are listed under the machine's name in the data store. When a machine is queried, its parameter information is read from the data store and any modifications made using the `machine modify` action are saved here as well.

```

[/MACHINE/W98]

[/MACHINE/W98/w98se]

[/MACHINE/W98/w98se/US]

[/MACHINE/W98/w98se/US/machine2]
os=W98
dhcp=Yes
lang=us
description=ITSO Test
status=Enabled
resolution=800x600x256@256
jvm=Yes
logon=Yes
tma=No
imagename=w98se
netadapter=TRP
protocols=NETBEUI, MSTCP
cdkey=xxxxx-xxxxx-xxxxx-xxxxx-xxxxx
TZ=Central
prebootimage=tdmw32ut.img
installdos=No
partition=600
srcrespfile=ibmfepci.inf
reguser=test98
name=machine2
mac=00203503CAE0
SERVER=ATLAS2
associated class=com.ibm.tdm.task.machine.W98Machine

```

Figure 77. Data store representation of a Windows 98 client

Figure 77 shows the data store representation of the machine definition given in Figure 76 on page 149. Other machine definitions, including those for other operating systems, would be listed in the appropriate subsections of the data store. Although the data store is currently a text file, its format may change with future releases of IBM WorkSpace On-Demand 3.0.1; therefore, you should never change a machine definition directly in the data store but use the `machine modify` action instead.

#### 4.4.3.2 The MACS directory

Just as in the creation of a Windows NT machine, the Windows 98 `machine define` command causes a new subdirectory to be created for the MAC

address of the client machine in `tdm\mm\client\ro\macs`. For the client machine named `machine2` that was defined in Figure 76 on page 149, this subdirectory is `\0020\3503cae0`. When the client machine first boots, IBM WorkSpace On-Demand 3.0.1 will use the client's MAC address to locate this directory. Two files are found here, `tdmclnt.inf` and `tdmdos.inf`. These files were copied from the directory `client\ro\w98\w98se\us\defs` and customized by the `machine define` process so that they are specific to the client machine we created. They are used to locate the initial DOS bootable image for the client machine. The .INF files corresponding to our sample Windows 98 machine definition are shown in Figure 78.

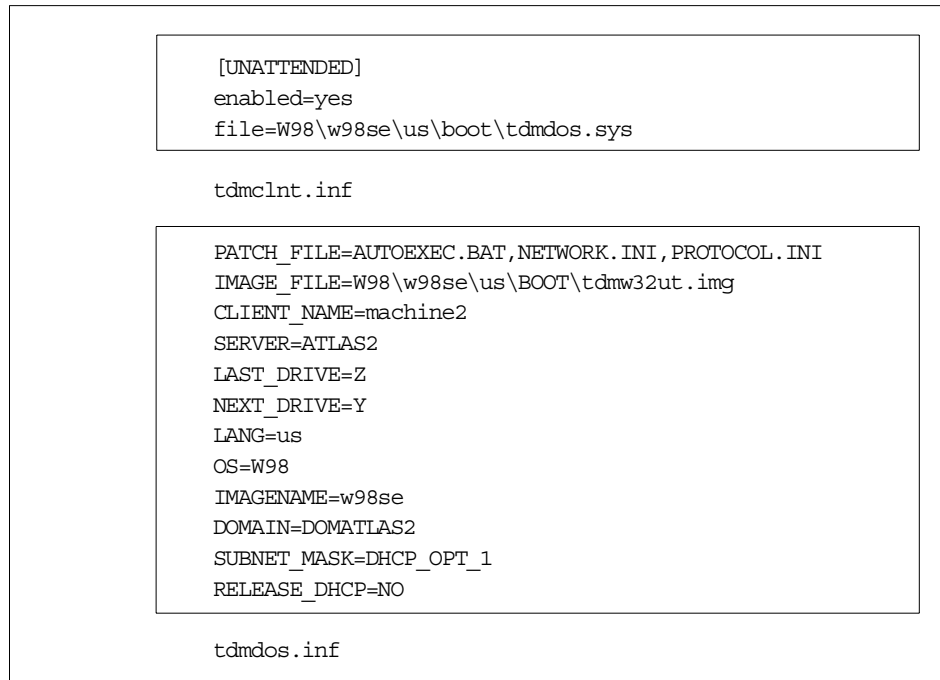


Figure 78. `tdmclnt.inf` and `tdmdos.inf` for a Windows 98 client

During the first boot, IBM WorkSpace On-Demand 3.0.1 first locates the file `tdmclnt.inf` and transfers it to the client machine via TFTP. The client machine's status is listed in this file; the line `enabled=yes` reflects the parameter setting `status=enabled` in the `machine define` command. If a machine were created with disabled status, its boot would not proceed beyond this point.

The line `file=` in `tdmclnt.inf` points to the second boot image, `tdmdos.sys`. This image is copied to the client machine via TFTP and the client boots. This

is a DOS boot that will use the file `tdmdos.inf` to locate the actual operating system image that is defined for this client machine. This second boot goes back to the server, locates `tdmdos.inf`, and transfers it to the client via TFTP.

The `prebootimage` parameter specified on the `machine define` command is listed in `tdmdos.inf` as the `IMAGE_FILE` variable. The path specified in `IMAGE_FILE` is relative to the directory path `tdm\mm\client\ro`. This combination completely specifies the path of the Windows NT Workstation client support installation on the server, where the DOS boot image files are installed. This DOS boot image is now transferred to the client machine via TFTP and the client machine's own directories are located on the server.

These directories are located on the server by means of the `SERVER`, `OS`, `IMAGENAME`, `LANG`, and `CLIENT_NAME` lines defined in `tdmdos.inf`. In our example, these correspond to subdirectories called `machine2` in the `machines` directory of the alias `RPLFILES`, or `tdm\mm\client\ro` and in the `machines` directory of the alias `WRKFILES`, or `tdm\mm\client\rw`, respectively. The `machine2` subdirectory in `rw` will contain logging information from the installation of Windows 98 on the client machine. The `machine2` subdirectory in `ro` contains all the control files needed to install Windows 98 on the client machine.

#### **4.4.3.3 The machines directory in RPLFILES**

Figure 79 on page 154 shows a listing of the contents of the `machine2` directory for our sample machine in `ro`. When the DOS pre-boot image is now booted, it uses the control files in this directory to direct the installation of Windows 98 on the client. These control files were copied and customized from defaults found in the `client\ro\w98\w98se\us\defs` directory. The DOS boot image establishes connections to both the `ro` and `rw` aliases and uses the utilities in the `ro\dos\2k\us\tree` subdirectory to perform the installation. An installation shell is started from the DOS image that makes repeated calls to the program `state.exe`. This program uses the file `state.ini` found in `ro\machines\machine2\w98\w98se\us` to determine each successive step of the installation process.

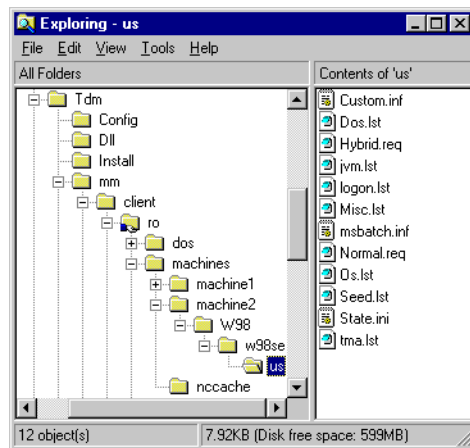


Figure 79. Contents of a Windows 98 machine directory after creation

#### 4.4.3.4 The state.ini file

The state.ini file is key to the boot and install process from this point on. State.ini was customized from the default provided with the Windows 98 client support. State.exe processes the contents of state.ini via a series of steps, known as states, that are listed in the StateNames section of state.ini. These states are:

|                |  |
|----------------|--|
| <b>Init</b>    | Initializes the installation.                      |
| <b>New</b>     | Partitions the client hard drive.                  |
| <b>Format</b>  | Formats the hard drive.                            |
| <b>Copy</b>    | Copies system files prior to installation.         |
| <b>Install</b> | Installs Windows 98 on the client machine.         |
| <b>Hybrid</b>  | Client machine has been installed; reboot locally. |
| <b>Error</b>   | Processes errors from any other states.            |

```

[parms]
PartitionSize=600
InstallDOS=No
MinMemSize=438
MinDiskSize=600
DSize=0
VideoCorrection=NO
CleanupOSFiles=Yes
InitDiskParms=0
msg1=0
msg2=0
OSDISK=C:
ClientType=NO
BootImageApps=JVM, LOGON

[StateNames]
Init
New
Format
Copy
Install

```

Figure 80. The Windows 98 header of state.ini

The Windows 98 state.ini file is very similar to the Windows NT state.ini; however, there are some significant differences. The following paragraphs will document the structure and uses of state.ini for a Windows 98 client and point out the important differences from Windows NT where appropriate.

The first few lines of our sample machine, machine2's state.ini are shown in Figure 80. The [parms] section at the top of state.ini contains parameter information that may be substituted in commands listed in other state sections farther down in the file. Some of these parameters were set based upon information provided as parameters in the `machine define` command. In particular, the parameters PartitionSize, InstallDOS, and BootImageApps should look familiar.

For each state, there is a section of state.ini labelled with the state name. Each state section contains commands that are executed by state.exe. These commands are of six possible types. Listed in their search order, these are:

- Internal commands to state.exe
- Internal commands to the install shell
- DOS internal commands

- DOS external commands or executables
- DOS batch files
- Install shell batch files

#### 4.4.3.5 Section [New] of state.ini

Figure 71 on page 137 illustrates the structure of state.ini by showing one complete section of the file. The section is labelled [new] and thus corresponds to the second state, named new, that directs the install shell to partition the hard drive according to parameters specified by the `machine define` command. State.ini determines the hard drive size and ensures that the disk is at least 600 MB, which is big enough to allow the installation of Windows 98. The value of the `dsize` variable is logged to the file counter.ini, found in machine2's directory in `tdm\mm\client\rw`. In the case of our example machine2, the hard drive size was 6 GB.

The commands in the [new] section direct state.exe to check what partition size was specified by the `machine define` command. If the administrator had specified a value of all, the partition size is set to the smaller of the physical disk size or 2 GB. If an actual partition size was specified, that size is used or an error is logged if the requested size is larger than 2 GB.

The `initdisk` command is used to create the primary partition of the appropriate size, and when this succeeds, the client machine is rebooted. The state is promoted to state 3, [Format].

#### 4.4.3.6 Section [Format] of state.ini

The next state causes the newly created partition to be formatted. The commands for this section are shown in Figure 72 on page 138. Notice that the commands in this section use the `InstallDOS` parameter from the `machine create` command. If `InstallDOS=Yes` was specified, or if state.exe determines that the amount of conventional memory on the client machine is insufficient, then DOS is installed by using the `/s` parameter on `format`.

#### 4.4.3.7 Section [Copy] of state.ini

When the format has successfully completed, there is no reboot specified because it is not necessary. State.exe executes a transition from the [Format] state to the [Copy] state and copies system files from the server to the client. The [Copy] state for the sample machine machine2 is shown in Figure 81 on page 157. The Windows 98 [Copy] state is very similar to that for Windows NT.



```

[Copy]
;Load disk caching software to speed the copying of NT files
Message 266 "Windows 98"
cachedisk
chdir %ClientRO%
;RedirectOutput copy.out
CopyCustomize "On"
MCOPY /I:Z:JVM.lst /l:copy.out
MCOPY /I:Z:Logon.lst /s /r /e /l:copy.out
MCOPY /I:Z:TMA.lst /s /r /e /l:copy.out
MCOPY /I:Z:Misc.lst /s /r /e /l:copy.out
;pause
MCOPY /I:Z:OS.LST /s /r /e /l:copy.out
if %InstallDOS%="Yes"
MCOPY /I:Z:DOS.lst /s /r /e /l:copy.out
endif
;if exist "z:drivers.lst"
;    MCOPY /I:Z:DRIVERS.LST /s /r /e /l:copy.out
;endif
if exist "z:\w98\w98se\us\inst\c"
mcopy z:\w98\w98se\us\inst\c C:\ /s /r /e /l:copy.out
endif
delay 5
Message 269
delay 3
CopyCustomize "Off"

```

Figure 81. The copy section of state.ini for Windows 98

Four `copy` commands are executed from this section since the `InstallDOS` parameter had been set to "No". These commands copy the files for the client Java Virtual Machine, the IBM WorkSpace On-Demand 3.0.1 logon program, the Tivoli management agent and other miscellaneous files to the client C: drive. These `copy` commands are directed via list files. The Z: drive letter prefacing each list file name maps to `\tdm\mm\client\ro`, also known as the alias `RPLFILES`. Prior to the `copy` commands, the `chdir` command sets the rest of the path to find the list files to be `\machines\machine2\w98\w98se\us`. Notice that the Tivoli files are copied to the client, even though the machine definition had specified `TMA=No`. The files are copied, but the Tivoli Management Agent is simply not enabled as an application for this machine.

The miscellaneous `copy` commands specified in `misc.txt` copy two files from the server to the client. These are `msbatch.inf` and `custom.inf`. `Msbatch.inf` will be used to drive the actual installation and setup of Windows 98 in the

[Install] section of state.ini. Custom.inf will be used after installation, prior to the first user login, to customize the desktop and set up the JVM.

Two conditional `copy` commands are added to the [Copy] section of state.ini for Windows 98.

The first checks for the existence of the file drivers.lst in our machine's w98\w98se\us directory. This list would have been created if the netadapterdir parameter had been set on our `machine define` command, allowing the installation of other network drivers.

The second conditional install checks for the existence of a directory named C in the Windows 98 client image under ro\w98\w98se\us\inst. Anything in the C directory will be copied exactly as is to the C: drive of the client machine at this point in the installation. For Windows 98, you will find two subdirectories under C, c\crtpkg and c\windows. These two directories contain programs crtpkg.exe and wininstall.exe. These programs are necessary to define applications and assign applications to users. Managing applications is covered in Chapter 6, "Applications" on page 259.

These package tools are also installed on the client machine during a Windows NT installation; however, they are embedded in the Windows NT client image in \nt4\sp3\inst\i386\soem\$\system32. Therefore, for a Windows NT client, the package tools are automatically installed into the system directory, while for a Windows 98 client, they are explicitly copied to the root of the client's C: drive.

#### **4.4.3.8 Section [Install] of state.ini**

When the `copy` commands have completed, the client system is now ready for the Windows 98 installation. State.exe exits the [Copy] section of state.ini and enters [Install]. At the beginning of this phase, state.exe checks to see if DOS was installed on the client hard drive due to the `machine define` parameter `Installdos` having been set to yes. If this is true, it means that there is insufficient conventional memory in the client machine to run the Windows 98 installation programs along with network drivers to perform the installation directly from the server. Therefore, state.exe will copy the rest of the DOS system files as well as the entire Windows 98 installation directory to the client machine's local hard drive. It will then boot into DOS with no network connections and begin the installation of Windows 98. When the installation is finished, the client machine will reboot and restore its network connections. In most cases, however, a local DOS-based install procedure is not required and Windows 98 installation can proceed directly from the server.

```

;Release RAM drive A: so Windows can have access to the real drive A:
Message 312
delay 3
RelDriveA
;echo "Drive a: released successfully..."
drive "c:"
ChDir "\ibmwin32"
Message 267 "Windows 98"
delay 5
command.com /c setup /is /id /iq msbatch.inf
spin
Log %ErrOsInstallFailed%
ErrorExit

```

Figure 82. Windows 98 installation from state.ini

When state.ini proceeds with the installation of Windows 98, the client machine's state is set to hybrid. A state value of hybrid indicates that the client machine is able to boot from its local hard drive. All client machines in IBM WorkSpace On-Demand 3.0.1 always perform a network boot first when they are rebooted to allow the server to control the machine and perform updates or installations as necessary. When a machine is in the hybrid state and it performs its network boot, the tdmclnt.inf file now points to hdboot.com instead of tmdos.sys. This redirects the client's boot to the local hard drive.

The lines of state.ini describing the network-based installation are shown in Figure 82. State.exe performs the installation of Windows 98 using the file msbatch.inf, which was copied from the server to the client's hard drive. Msbatch.inf was customized by the `machine define` command to contain the rest of the parameters specified when the client machine was created. For example, the user information, CD key, time zone, display resolution, network protocol information, and optional components specified by `machine define` are all listed in msbatch.inf.

Finally, when the installation is complete, the client machine remains in its hybrid state and boots to the IBM WorkSpace On-Demand 3.0.1 logon window. When the user logs on, the last copy operations are completed and registry entries set and the user's Windows 98 desktop is displayed.

#### 4.4.3.9 The machines directory in WRKFILES

During the entire boot process, state.exe must keep track of the client machine's state and log its progress. It does this through the WRKFILES alias, defined as `\tdm\mm\client\rw`. Here, the directory

\machines\machine2\w98\w98se\us contains the work files that need to be written during the installation. A listing of this directory is shown in Figure 83.

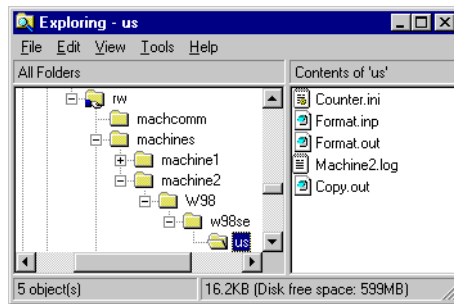


Figure 83. Contents of the WRKFILES alias for a Windows 98 client

The files `format.out` and `copy.out` contain the results of the hard drive format and all `copy` commands of files to the client hard drive. `Format.inp` simply contains the letter `y` to respond to the “Are you sure?” prompt normally presented by the `format` command.

The file `counter.ini` records the state transitions made by `state.exe` during the entire installation process. Within this file is a variable called `state`, which changes value as `state.exe` changes from creating the partition to formatting, copying, and finally installing the operating system. After the user's first logon, the final value of the `state` variable is 5, indicating that the installation of the client machine has completed.

---

## 4.5 Defining OS/2 client machines

If you have installed the OS/2 client support for IBM WorkSpace On-Demand 3.0.1, you will be able to define client machines with the OS/2 operating system. Unlike Windows NT or Windows 98 clients, these machines perform a remote boot from the server rather than an installation of the operating system. The OS/2 client boots from the OS/2 client image installed in IBM WorkSpace On-Demand 3.0.1, although some parts of the operating system that had to be customized for this client will be located in the client machine's directories in the `RPLFILES` and `WRKFILES` aliases. OS/2 client support in IBM WorkSpace On-Demand 3.0.1 is very similar to the standard WorkSpace On-Demand client support. The following sections describe the steps involved in creating and booting an OS/2 client machine.

### 4.5.1 OS/2 client parameters

An OS/2 client is created with the `machine define` command. Among all the parameters that can be specified for this command, the following are mandatory:

OS, LANG, NAME, SERVER, IMAGENAME, MAC, VIDEO, and NETADAPTER

The following parameters are specified for this command:

#### **Name**

This is the name that will uniquely identify the client machine on the deployment server. For example, `machine0`, `machine1`, `machine2`, and `machine3` are valid machine names as shown in Figure 55 on page 104. The machine name must follow the 8.3 Naming Convention. This parameter is required when creating, deleting, or modifying a machine. Our sample OS/2 machine will be named `machine3`.

#### **MAC**

This parameter specifies the MAC address of the client machine's network adapter card. It must be specified when creating a machine. The MAC address is expressed as 12 hexadecimal digits and results in a directory structure, such as `macs\0020\35fe4a6f`, as shown in Figure 55 on page 104.

#### **Server**

This parameter specifies the name of the deployment server where the client machine will be created. It is required for machine creation only.

#### **OS**

The OS parameter lets you select the operating system image for your client. This can NT4, W98, or OS2. This parameter must be specified for creating, deleting, or modifying a machine. For our sample OS/2 client, we set `OS=OS2`.

#### **ImageName**

`ImageName` must correspond to the image name you selected when you installed the client operating system support. For example, when we installed the OS/2 client image, we chose the image name `bb20`. This parameter must be specified when creating, deleting, or modifying a machine.

#### **Lang**

This parameter sets the language of the operating system that will be used for this client machine. It must correspond to the language selected when the client operating system support was installed. `Lang` is specified as a

two-character string, for example, "US" for American English. This parameter is required when defining a machine and cannot be changed after definition.

### **Template**

This is a yes or no parameter that specifies whether the machine is to be defined as a template. The default value is no. If you specify Template=yes, you can create other machines from this template. Using templates is discussed in Section 4.6, "Machine templates" on page 177.

### **Templatename**

Specifies the name of the template to be used when creating this machine. If you create a machine from a template, all machine attributes will be taken from the template. Using templates is discussed in Section 4.6, "Machine templates" on page 177.

### **Netadapter**

This parameter specifies the network adapter type for your client machine. You can determine the supported network adapters by using the `machine parmlist` command as described in Section 4.7, "Other machine commands" on page 179. For example, you could type the following:

```
machine parmlist parameter=netadapter server=atlas2 os=os2  
imagename=bb20 lang=us
```

The value you specify for netadapter must exactly match one of the strings returned from `machine parmlist`. These values are taken from the .NIF files for supported network drivers installed in the IBMCOM\MACS directory of your OS/2 client image. If you need to install support for another network adapter, you must copy the .MSG files to the directory \\RPLFILES\OS2\BB20\US\TREE\IBMCOM and copy the driver and .INF files to \\RPLFILES\OS2\BB20\US\TREE\IBMCOM\MACS on your server. Remember that IBM WorkSpace On-Demand 3.0.1 supports only network adapters that can support the PXE 2 boot protocol.

### **LAA**

Use this parameter to set a locally administered address for your network adapter. This is a 13-character string, where the first character is either 'T' for token ring or 'I' for IEEE Ethernet and the remaining 12 characters are hexadecimal digits.

### **DHCP**

You should set DHCP=Yes to indicate that the client machine should be configured for DHCP. If you do not set DHCP, you will need to configure all the other IP settings for your client machine.

**IP**

Use this parameter to give your client machine a fixed IP address. This will be ignored if you also set DHCP=Yes.

**DDNS**

Set DDNS=Yes to indicate that the client machine should use Dynamic Domain Name Service. If you set this to yes, you must also have DHCP selected.

**Nbddaddr**

This parameter is only needed if your client machine will use NetBIOS over TCP/IP. It is used to specify the IP address (in dotted decimal notation) of the NetBIOS Datagram Distribution Server.

**Nbnamesrv**

This parameter is only needed if your client machine will use NetBIOS over TCP/IP. It is used to specify the IP address of the NetBIOS Name Server.

**Tcpnamesrv**

This parameter is not needed if you set DHCP=Yes. It is used to specify the address of the primary TCP Domain Server.

**Tcpdomain**

Use this parameter to specify the TCP domain name. This value is required if IP or DDNS is specified.

**Tcpname**

This parameter is used to set the TCP host name of the client machine. It is required if both DHCP and DDNS are specified.

**Nodetype**

This parameter specifies the node type for TCPBEUI. The default value is B to set a broadcast node. Other valid values are P for Point-to-Point and H for Hybrid.

**Tcprouter**

Use this parameter to set the IP address of the IP router. This parameter is not required and will be ignored if you set DHCP=Yes.

**Nbscope**

This parameter is used to specify the population of computers in which a registered NetBIOS name is known. It must be less than 128 bytes in length. It is optional and should normally not be specified. The NetBIOS scope is used to configure different logical NetBIOS networks on the same TCP/IP network.

**Netmask**

Use this parameter to specify the subnet mask for your client machine. This parameter is required if IP is specified, but is ignored if DHCP is specified.

**Protocol1**

This parameter specifies the network protocol used at boot time. It can be either NetBEUI or TCPBEUI; the default is NetBEUI.

**Protocol2**

This parameter can be used to specify an additional network protocol. If it has the same value as Protocol1, it is ignored. The default value is NULL.

**Netprtn**

Use this parameter to specify a network printer to which the client machine will connect. Valid values for n are 1 through 9. This is set as a text string consisting of five substrings separated by commas. The entire string must be enclosed in quotation marks. The five substrings are:

|                       |   |
|-----------------------|---|
| <b>Local queue</b>    | The name of the local print queue                               |
| <b>Printer driver</b> | The name of a printer driver and device, as found in prdesc.lst |
| <b>Printer port</b>   | Optional; the name of the printer port, such as LPT2            |
| <b>Remote Queue</b>   | The name of the remote print queue                              |
| <b>Server</b>         | The name of the server on which the remote queue resides        |

The file prdesc.lst is located in the RPLFILES alias in the directory OS2\BB20\US\TREE\OS2\INSTALL.

For example, you could use the following parameter string to install Lexmark Optra Lx+ printer support for your client:

```
netprt1="LASER1,IBMPCL5.Lexmark Optra Lx+,LPT2,REMOTEQ,\\ATLAS2"
```

One optional parameter can be added after the server name. This is the string `RAW` or `META` that can be used to indicate the printer data type. The default data type is `META`.

You can also use this parameter to delete printer support when modifying a machine using the `machine modify` command. For example, setting `netprt1=LASER1:DELETE` will delete the queue created above.

**Bootdrive**

This parameter specifies the drive letter that will be used as the boot drive for the client machine. If you do not set a value, the default is Z.



**Cache**

This is a yes or no parameter and specifies whether or not the client is cached. Cached clients have a copy of the operating system image copied to their local hard drive and boot from the local cache whenever possible to reduce network traffic.

**CDROM**

Use this parameter to specify the CD-ROM type of your client machine. You can use the `machine parmlist` command, setting `parameter=cdrom`, to determine the list of supported CD ROM drives.

**Diskette**

You can use this parameter to enable or disable support for the diskette drive in the remote machine. The default is `diskette=Yes`; set `diskette=No` to disable the diskette drive.

**Hardfiletype**

This parameter sets the type of hard disk supported by your client machine. It can be one of the following values: none, IDE, SCSI, or Both. The default value is IDE.

**Kbdtype**

This parameter sets the keyboard type for your client machine. You can determine the valid possible keyboard types by using the `machine parmlist` command, specifying `parameter=kbdtype`. The valid types of keyboard depend on the language version of OS/2 you installed. For US English, the keyboard type is 101.

**Monitor**

This parameter is used to specify the type of monitor supported by your client machine. This parameter is used in conjunction with the Video parameter and is ignored if Video is set to VGA. Valid values for Monitor can be determined by using `machine parmlist` with `parameter=monitor` and specifying a particular video adapter as follows:

```
machine parmlist os=os2 parameter=monitor video=s3t64 imagename=bb20  
lang=us server=atlas2
```

Normally, the value of monitor can be left at default.

**Video**

Use the Video parameter to set the video driver supported by your client machine. Valid values for Video can be retrieved using `machine parmlist` with `parameter=video`. These values are retrieved from the video directory in your OS/2 image in the WRKFILES alias `\tdm\mm\client\rw`. If you want to add

support for a different video adapter, you need to copy the driver files to a subdirectory in OS2\BB20\US\VIDEO.

### Resolution

The Resolution parameter lets you set the window resolution and refresh rate for your client machine. Valid values for this parameter are dependent upon your selection for both video and monitor. You can determine the valid values by using the `machine parmlist` command with `parameter=resolution` and also specifying your selected monitor and video values, as follows:

```
machine parmlist os=os2 parameter=resolution video=s3t64 monitor=default  
imagename=bb20 lang=us server=atlas2
```

The format for the resolution parameter is a string in the form "XxYxC@R" where X is the horizontal resolution, Y the vertical resolution, C the number of colors, and R the refresh rate in MHz. For example, you could set

```
Resolution="1024x768x256x43".
```

### Mouse

Use this parameter to specify the type of mouse supported on your client machine. The value is a string that must match the returned values of `machine parmlist` with `parameter=mouse`. The default value is "PS/2 Mouse".

### Parallel

The default value for this parameter is 'Y'. Use this to specify whether or not a client machine will support parallel devices.

### Printern

This parameter is used to specify local printer support. Valid values for n are 1 through 4. This parameter is set as a text string to indicate the queue name, device, and printer port. Optionally, you can also specify RAW or META immediately following the printer port. For example, use the following command to install a Lexmark Optra Lx+ printer locally:

```
printer1="LOCAL1,IBMPCL5.Lexmark Optra Lx+,LPT2,RAW"
```

Valid printers are found in the `prdesc.lst` file, which can be found in the OS2\BB20\US\TREE\OS2\INSTALL directory within the RPLFILES alias.

You can also use this parameter when modifying a machine to delete a printer, for example, using `printer1="local1:delete"` will delete the printer support added above.

### Description

This optional parameter is a text string enclosed in quotation marks that will be a comment describing the machine in the data store.

**Sandbox**

This parameter has value 'Y' or 'N' and is used to indicate if the machine being created is a sandbox. This value cannot be modified after a machine has been created. The default value is 'N'. A sandbox machine has no restrictions on access to the hard drive or operating system and is used by an administrator or developer to configure applications prior to deployment.

**SCSI**

Use this parameter to set the SCSI adapter type configured for your remote client. It is set as a text string which must match exactly one of the strings returned from the `machine parmlist` command, setting `parameter=scsi`. Valid values are taken from the `scsi.tbl` file, which is located in the `RPLFILES` alias in the `OS2\BB20\US\TREE\OS2\INSTALL` directory.

**Serial**

This parameter specifies whether or not the client machine supports serial devices. The default is Yes; use `serial=No` to disable serial devices.

**Configsrc**

This text parameter specifies the name of the directory where source files used to create the client machine's image are located. The directory name must be no more than eight characters. A directory named "DEFAULT" is already created as part of the standard OS/2 client image. However, an administrator may add another directory to customize the client image. All files and directories in the existing default directory should be used as a base for creating a new directory to be used with the `configsrc` parameter.

**Status**

The value for this parameter is either "Enabled" or "Disabled". If a client is disabled, it will be prevented from booting.

**Swapper**

The value of this parameter is either "C" or "S". Setting `swapper="C"` indicates that the client's `swapper.dat` file will reside on the client's local hard drive instead of on the server. In this case, the client machine must have a local hard drive and have hard drive support enabled.

**TMA**

This parameter indicates whether the Tivoli management agent should be installed in the client machine's image. The valid values for this parameter are "Enabled" or "Disabled". The default is disabled.

### Uconfigsys

This parameter can be used to add additional text to the config.sys file for the client machine. The text must be enclosed in quotation marks and can be a maximum of 4096 bytes.

## 4.5.2 The machine define command for OS/2

With these parameters, you can use the `machine define` command to create an OS/2 machine. Many parameters can be left at the default values. Figure 84 shows a minimal `machine define` command for a token-ring-based client.

```
Machine define OS='os2' NETADAPTER='IBM Token-Ring PCI Family Adapter  
(IBMTRP.OS2)' LANG='us' IMAGENAME='bb20' VIDEO='VGA' MOUSE='PS/2  
Mouse' Name='machine3' MAC='002035fe4a6f' SERVER='atlas2'
```

Figure 84. The machine define command for an OS/2 client

This command can be simplified further and encapsulated into the `CreateMachinet` function found in the JavaScript file `machines.js` on the CD. If you load `machines.js` into the console and call `CreateMachinet` as follows, it will issue the same `machine define` command:

```
CreateMachinet("machine3", "002035fe4a6f", "atlas2", "os2")
```

## 4.5.3 OS/2 client boot

Once you have successfully defined an OS/2 client machine, you will notice many changes in the IBM WorkSpace On-Demand 3.0.1 directories to represent the new machine. Unlike Windows 98 and Windows NT, however, when the client machine boots, there is no installation of the operating system on its hard drive. Rather, the OS/2 client machine performs a remote boot from the server in much the same way as OS/2 clients do in IBM WorkSpace On-Demand 3.0.1.

### 4.5.3.1 The data store entry

First, when you create an OS/2 client machine, a new entry is created in the data store for the new machine. The data store is kept as a text file called `datastor.ini` in `\tdm\config`. Since a full definition of a machine needs to include not just the machine's name but also the operating system name, image name, and language, the data store contains sections for each possible combination of values. All the parameters that you specified when defining the machine are listed under the machine's name in the data store. When a machine is queried, its parameter information is read from the data

store and any modifications made using the `machine modify` action are saved here as well.

Figure 85 shows the data store representation of the machine definition given in Figure 84 on page 168. Other machine definitions, including those for other operating systems, would be listed in the appropriate subsections of the data store. Although the data store is currently a text file, its format may change with future releases of IBM WorkSpace On-Demand 3.0.1; therefore, you should never change a machine definition directly in the data store; use the `machine modify` action instead.

```
[/MACHINE/OS2]

[/MACHINE/OS2/bb20]

[/MACHINE/OS2/bb20/US]

[/MACHINE/OS2/bb20/US/machine3]
os=os2
netadapter=IBM Token-Ring PCI Family Adapter (IBMTRP.OS2)
lang=us
imagenname=bb20
video=VGA
mouse=PS/2 Mouse
name=machine3
mac=002035fe4a6f
SERVER=ATLAS2
associated class=com.ibm.tdm.task.machine.OS2Machine
```

*Figure 85. Data store entry for an OS/2 client machine*

#### 4.5.3.2 The MACS directory

Just as in the creation of Windows NT and Windows 98 clients, the `OS/2 machine define` command causes a new subdirectory to be created for the MAC address of the client machine in `tdm\mm\client\ro\macs`. For the client machine named `machine3` that was defined in Figure 84 on page 168, this subdirectory is `\0020\35fe4a6f`. When the client machine first boots, IBM WorkSpace On-Demand 3.0.1 will use the client's MAC address to locate this directory. Two files are found here, `tdmclnt.inf` and `tdmos2.inf`. These files were created by the machine definition process for the OS/2 client and customized by the `machine define` process so that they are specific to the client machine. They are used to locate the initial OS/2 bootable image for the client machine. The .INF files corresponding to our sample OS/2 machine definition are shown in Figure 86 on page 170.

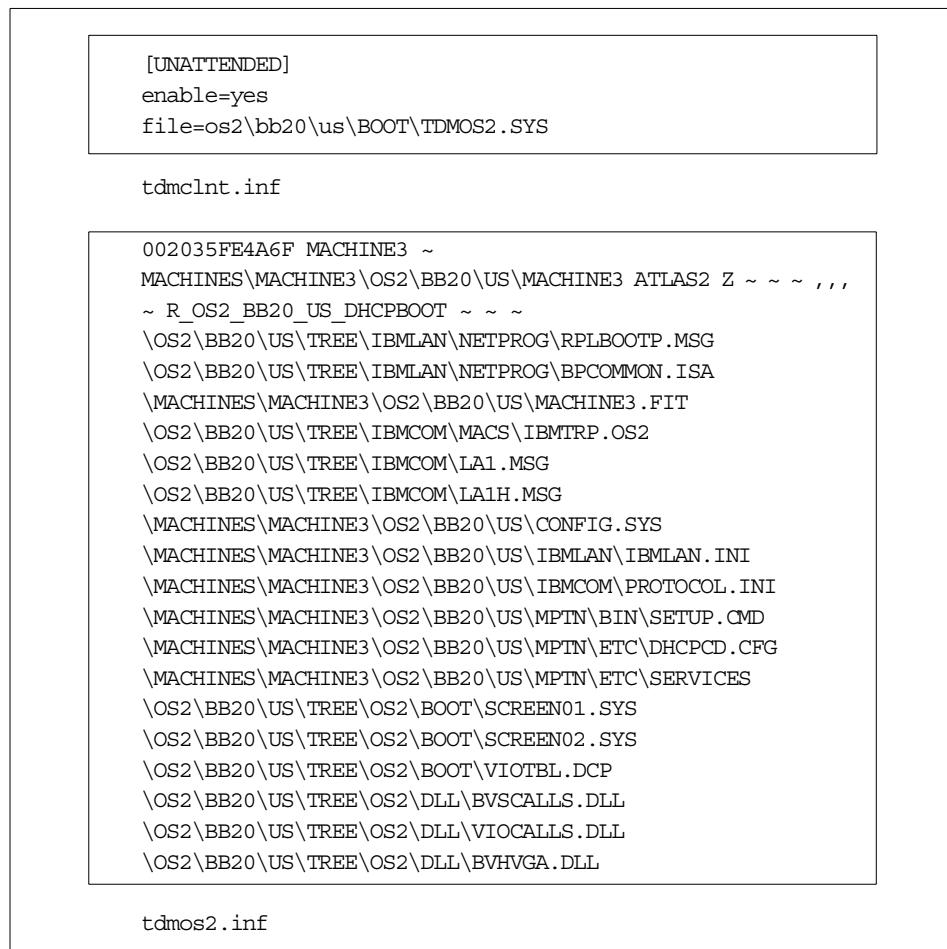


Figure 86. tdmclnt.inf and tdmoss2.inf for an OS/2 client

During the first boot, IBM WorkSpace On-Demand 3.0.1 first locates the file tdmclnt.inf and uses it to continue the remote boot process. The client machine's status is listed in this file; the line enabled=yes reflects the parameter setting status=enabled in the machine define command. If a machine were created with disabled status, its boot would not proceed beyond this point.

The line file= in tdmclnt.inf points to the second boot image, tdmoss2.sys. This image is used to perform the OS/2 remote boot, which will use the file tdmoss2.inf to locate the actual operating system image that is defined for this client machine.

The `tdmos2.inf` file is shown in Figure 86 on page 170. It contains information for `tdmos2.sys` to locate all the files needed to boot the OS/2 image up to the point where it initializes the protect mode redirector. (The protect mode redirector will use the client's FIT file to locate the rest of the operating system files needed to boot the client.) The first line of `tdmos2.inf` is a copy of the client machine's entry from the file `RPL.MAP`, which is located in the `RPLFILES` alias `\tdm\mm\client\ro`. In IBM WorkSpace On-Demand 3.0.1 and IBM LAN Server, the file `RPL.MAP` was used by the server to identify client machines booting with the NetBIOS Remote IPL (RIPL) protocol. The `RPL.MAP` file is not used for PXE boot of OS/2 clients in IBM WorkSpace On-Demand 3.0.1 but only stores information about the OS/2 client machines.

The remainder of the lines in `tdmos2.inf` refer to files residing in the `RPLFILES` alias; the specifications for these files are relative to `RPLFILES` or `\tdm\mm\client\ro`. The first file listed, `rplbootp.msg`, is a text file containing error messages that may be displayed by the bootstrap routine as it boots the client machine. The next reference is to a file called `bpcommon.isa`, which resides in the `ibmlan\netprog` directory within the OS/2 image. This file is known as a common descriptor file and points to all the common files needed to load and boot the OS/2 client. Figure 87 shows a selection of entries from `bpcommon.isa` for our example machine, `machine3`, created in Figure 84 on page 168.

```
\OS2\BB20\US\TREE\OS2KRNL
\OS2\BB20\US\TREE\OS2LDR
\OS2\BB20\US\TREE\OS2LDR.MSG
\OS2\BB20\US\TREE\OS2LOGO
\OS2\BB20\US\TREE\OS2VER
\OS2\BB20\US\TREE\OS2\KEYBOARD.DCP
\OS2\BB20\US\TREE\OS2\SYSTEM\COUNTRY.SYS
\OS2\BB20\US\TREE\OS2\BOOT\RESOURCE.SYS
\OS2\BB20\US\TREE\OS2\BOOT\CLOCK01.SYS
\OS2\BB20\US\TREE\OS2\BOOT\KBDBASE.SYS
```

*Figure 87. Selections from the common descriptor file*

The third important reference in `tdmos2.inf` is to the File Index Table or FIT file, which is used to complete the boot process and is discussed in more detail in Section 4.5.4, "The OS/2 client machine FIT file" on page 173. Each client machine has its own FIT file that redirects all references to operating system files required during the boot process to their real location on the server. In the case of our example client machine called `machine3`, the FIT file is called `machine3.fit` and can be found in the `RPLFILES` alias in the

machines\machine3\os2\bb20\us directory. When you defined the OS/2 client machine, this FIT file was created from a default located in the OS/2 client image. The default FIT file is in the RPLFILES alias in the OS2\BB20\US\CNFG\DEFAULT directory and is named default.fit.

#### 4.5.3.3 The client OS/2 image

For Windows 98 and Windows NT client machines, the amount of information stored on the server to manage the machine is relatively small because the machine boots from a local installation of the operating system. Files on the server are used to manage the installation of the client—nothing more. In contrast, OS/2 client machines keep no operating system information on their local hard drive (in many cases, they will not even have a hard drive), and so, the client machine image on the server is very large. Every time the OS/2 client boots it must find its entire operating system image on the server.

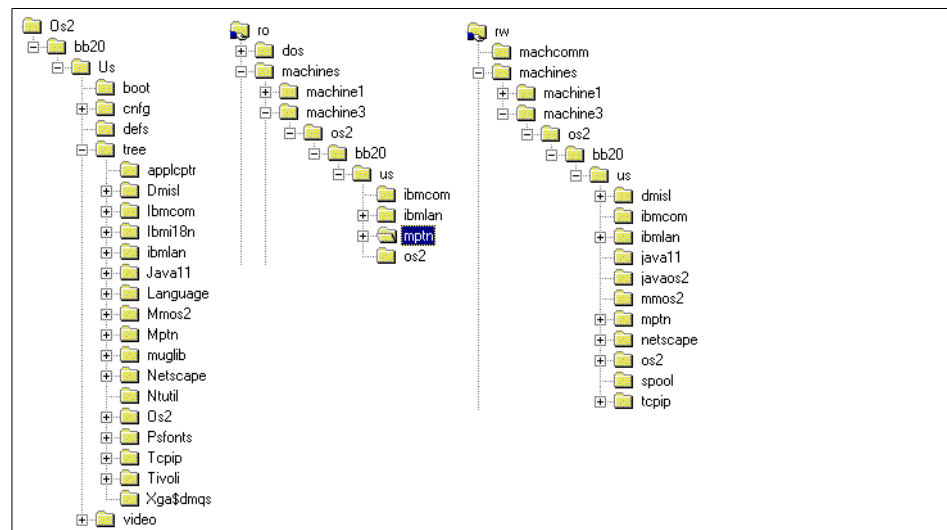


Figure 88. Default machine read-only and machine read-write OS/2 client images

The redirection provided by the FIT file causes the boot process to locate the operating system files in three possible places: the generic OS/2 client image in the RPLFILES alias, the client-specific OS/2 files in the RPLFILES alias and the client-specific OS/2 files in the WRKFILES alias. Figure 88 shows the directory layout of the OS/2 client image. The left column is the default OS/2 client image, kept in both the RPLFILES and WRKFILES aliases. These files are the common OS/2 files that are either the same for all OS/2 clients or used as templates to create client-specific files in the machine directories. For example, video and network drivers are located here.



The center column shows the directory structure of our sample machine, machine3, in RPLFILES. These are the operating system files that are client-specific but read-only, meaning that they do not need to be changed while the client machine is running. The right column displays the same sample machine's directory layout in the WRKFILES alias. These files have read-write access from the client machine, since they may need to change while the client machine is booted. For example, files such as lantran.log and other log files would be stored here.

Portions of the operating system image actually booted on the client machine are taken from all three locations. Determining which operating system files are retrieved from which location and building the client operating system image is the job of the File Index Table or FIT file.

#### **4.5.4 The OS/2 client machine FIT file**

The common files referenced in bpcommon.isa are those files in the OS/2 client image that are common to all OS/2 client machines. However, many operating system files are different per client machine. Each client, for example, could have a different config.sys file or a different desktop that would require it to have different os2.ini and os2sys.ini files. Other differences might include video or other device drivers. In order to locate these client-specific operating system files, tdmoss2.inf contains a pointer to a File Index Table, commonly known as a FIT file. The File Index Table is used to provide file redirection services from a remote boot client to its boot server. FIT files were first introduced with Remote IPL support in IBM LAN Server, and their use in IBM WorkSpace On-Demand 3.0.1 is the same as in WorkSpace On-Demand 2.0.

##### **4.5.4.1 FIT file structure**

WorkSpace On-Demand 2.0 and IBM WorkSpace On-Demand 3.0.1 use three different FIT files, known as the machine FIT, user FIT, and application FIT. User and application FIT files are used to manage desktops and applications, and is described in later sections of this book. The machine FIT file is used to complete the remote boot of the OS/2 client image.

In general, a FIT file entry consists of two parts, mapping a prototype file name on the left to substitution text on the right, which is usually an actual file or directory on the server. Sample entries from the machine FIT file are shown in Figure 89 on page 174.

|                        |   |
|------------------------|---|
| Z:\IBMLAN              | OS2\BB20\US\TREE\IBMLAN                           |
| Z:\MPTN                | OS2\BB20\US\TREE\MPTN                             |
| Z:\IBMCOM              | OS2\BB20\US\TREE\IBMCOM                           |
| Z:\IBMCOM\PROTOCOL.INI | MACHINES\MACHINE3\OS2\BB20\US\IBMCOM\PROTOCOL.INI |

Figure 89. Sample machine FIT entries

The purpose of the machine FIT file is to build a logical boot image for the client workstation using the boot drive specified with the bootdrive parameter when you created the OS/2 machine. The default boot drive value is Z:. This can be overridden upon machine creation. The boot drive letter does not always map to a specific directory, alias, or drive; rather, it is a logical boot drive used by the client machine. Different prototype files or directories on the Z: drive can be mapped to completely different directories or aliases on the server.

In Figure 89, note that the substitution text for each of the sample FIT entries is only a partial specification of the directory or file. This is because the first line of the FIT file provides a default alias that will be used as the prefix for all subsequent FIT entries, unless overridden by a specific alias used in another FIT entry.

|   |
|---|
| <pre> \\ATLAS2\RPLFILES ; The first line of this file MUST be UNC name </pre> |
|---|

Figure 90. The FIT default alias

Figure 90 shows the default alias specification in the FIT file for our sample machine. In this case, all entries in the FIT file that are not prefixed with a specific alias are considered to be relative to the RPLFILES alias, or \tdm\mm\client\ro.

The FIT file is used to map directories as well as specific files from the client operating system image to the server. For example, in Figure 89, the directory prototype Z:\IBMLAN for the client is mapped to the generic OS/2 client image of IBMLAN on the server. The client file PROTOCOL.INI, however, is mapped to a machine-specific PROTOCOL.INI residing, in our example, in the machine3 directory on the server. Different client machines may have different network adapters; therefore, it is necessary to maintain a separate PROTOCOL.INI for each client machine.

Notice that specific file mappings in FIT entries can override the more general directory mappings. For example, in Figure 89, the client's IBMCOM directory is mapped to the generic IBMCOM directory in the OS/2 client image on the

server. However, the specific file `PROTOCOL.INI`, which resides in `IBMCOM`, is mapped to the client-specific `PROTOCOL.INI` in the `machine3` directory on the server.

```
; TCPIP support.
Z:\TCPIP                OS2\BB20\US\TREE\TCPIP
Z:\TCPIP\BIN\SETUP.*    \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\TCPIP\BIN
Z:\TCPIP\BIN\*.LOG      \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\TCPIP\BIN
Z:\TCPIP\BIN\TCPSTART.* \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\TCPIP\BIN
Z:\TCPIP\ETC            \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\TCPIP\ETC
Z:\TCPIP\TMP            \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\TCPIP\TMP
```

Figure 91. Additional machine FIT entries

Figure 91 illustrates other ways of defining FIT entries. Note that most of the TCPIP directory mappings are to a different alias than the default; this is accomplished by specifying the alias in the substitution string in order to override the default alias. Also, wild cards, such as `*` and `?`, are permitted in the FIT file, but only in the prototype string on the left. If you use a wild card in the prototype string, the substitution string must specify a directory.

The effect of the FIT file is to create a logical OS/2 image from the perspective of the client machine. If a user were to log on to our sample client and open an OS/2 command prompt, he or she would see a `Z:>` prompt. There is no physical drive labeled `Z:`, nor does the drive letter `Z:` map to any one specific alias or drive. Rather, the FIT file combines operating system and application files from many different locations on the server to present a logical view of the operating system rooted on `Z:`.

#### 4.5.4.2 Modifying the machine FIT file

The machine FIT file is used to manage the client's operating system files as well as middleware or applications that need to be part of the boot image of the client. For example: Netscape and Java support are built in to the OS/2 client image of IBM WorkSpace On-Demand 3.0.1. The FIT entries to support these are shown in Figure 92 on page 176.

```

; JAVA support
Z:\JAVA11 OS2\BB20\US\TREE\JAVA11
Z:\JAVA11\HOTJAVA \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\JAVA11\HOTJAVA
Z:\JAVA11\WEBLOGS \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\JAVA11\WEBLOGS
Z:\JAVAOS2 OS2\BB20\US\TREE\JAVAOS2
Z:\JAVAOS2\WEBLOGS \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\JAVAOS2\WEBLOGS
Z:\JAVAOS2\HOTJAVA \\ATLAS2\WRKFILES\MACHINES\MACHINE3\OS2\BB20\US\JAVAOS2\HOTJAVA

; NETSCAPE support
Z:\NETSCAPE OS2\BB20\US\TREE\NETSCAPE

```

Figure 92. Java and Netscape support in the machine FIT file

You may wish to modify the location of specific Java or Netscape files, or even add entirely new applications that are needed for all OS/2 client machines. To do this, you will need to modify the machine FIT file. For example, you could use the FIT file to provide support for a PCOMM client to all OS/2 machines.

The following rules apply to modifying the FIT file:

- Directory names must map to directory names.
- Individual file names must map to individual file names.
- Wild cards may only be used in the prototype at the left of each line in the FIT file.
- Prototypes containing wild cards must map to directory names.
- No more than one wild card may appear in a prototype. For example, Z:\OS2\ANY???.\* is invalid.
- If you map files to other aliases than RPLFILES or WRKFILES, ensure that the members of RPLGROUP have the appropriate access to the alias.

If you modify a specific machine's FIT file, your changes will only apply to that specific machine. To apply your modifications to all OS/2 client machines, it is better to modify the default FIT file. This file is called default.fit and can be found in the RPLFILES alias in the os2\bb20\us\cnfg\default directory. The default FIT file looks the same as the machine-specific FIT file except that the substitutions are generic.

Figure 93 on page 177 shows some sample entries in a default FIT file. When you create an OS/2 client machine, that client's FIT file is created using the default FIT file as a template. All references to \\RPLSERVER are modified to be the name of the deployment server on which your machine is created, and all references to IMAGENAME are replaced with the image name you specified when you installed OS/2 client support. In the example used

throughout this chapter, the RPLSERVER reference became ATLAS2, and the IMAGENAME reference became BB20.

```
; TCPIP support.  
Z:\TCPIP OS2\IMAGENAME\US\TREE\TCPIP  
Z:\TCPIP\BIN\SETUP.* \\RPLSERVER\WRKFILES\MACHINES\DEFAULT\OS2\IMAGENAME\US\TCPIP\BIN  
Z:\TCPIP\BIN\*.LOG \\RPLSERVER\WRKFILES\MACHINES\DEFAULT\OS2\IMAGENAME\US\TCPIP\BIN  
Z:\TCPIP\ETC \\RPLSERVER\WRKFILES\MACHINES\DEFAULT\OS2\IMAGENAME\US\TCPIP\ETC  
Z:\TCPIP\TMP \\RPLSERVER\WRKFILES\MACHINES\DEFAULT\OS2\IMAGENAME\US\TCPIP\TMP
```

Figure 93. Sample entries in the default FIT file

When you modify the default machine FIT file, your modifications will be effective for all machines created after that point. If you have already created OS/2 client machines prior to modifying the machine FIT file, you will have to delete and redefine those clients for your changes to take effect.

---

## 4.6 Machine templates

Machine templates give you another way to simplify the machine creation process. A machine template is analogous to the machine class in IBM WorkSpace On-Demand 3.0.1 in that it encapsulates a large number of machine attributes. Machines can be created from the template without having to continually specify the same attributes over and over again. Using templates is a good way to ensure that all machines of the same type are created with the correct values for critical information, such as network adapters, protocols, or video drivers.

Creating a template is very similar to creating a machine (as previously described). Use the `machine define` command to create the template, and set the parameter `Template=Yes`. Also, since a template is not a physical machine, you must not specify a MAC address for the template. The third point to remember is that you do not need to specify the name of the deployment server when creating a template, since no machine definition will actually take place. The `machine define` command for a Windows NT machine template is shown in Figure 94 on page 178.

```
Machine define OS='NT4' NETADAPTER='TRP' DHCP='Yes' LANG='us'
PROTOCOLS='TC,NBF' DESCRIPTION='ITSO Test' RESOLUTION='800x600x256@70'
STATUS='Enabled' JVM='Yes' LOGON='Yes' TMA='No' IMAGENAME='sp3'
VIDEO='Auto-detected' CDKEY='xxx-xxxxxxx' TZ='(GMT-06:00) Central Time
(US & Canada)' PREBOOTIMAGE='tdmw32ut.img' INSTALLDOS='No'
LOCALADMPW='getin2see' partition='800' REGUSER='ITSO'
srcrespfile='ibmfepci.txt' template='yes' Name='NTTempl'
```

Figure 94. Creating a machine template

The template was defined with the template name NTTempl. This name will be used to create machines from the template. Notice that the parameters mac= and server= are missing from the template definition and that the parameter template=Yes was set. You are still required to specify the OS, Imagename, and Lang parameters for a template, just as for any other machine.

```
[/MACHINE/NT4/sp3/US/NTTempl]
os=NT4
netadapter=TRP
dhcp=Yes
lang=us
protocols=TC,NBF
description=ITSO Test
resolution=800x600x256@70
status=Enabled
jvm=Yes
logon=Yes
tma=No
imagename=sp3
video=Auto-detected
cdkey=xxx-xxxxxxx
tz=(GMT-06:00) Central Time (US & Canada)
prebootimage=tdmw32ut.img
installdos=No
localadmpw=getin2see
partition=800
reguser=ITSO
srcrespfile=ibmfepci.txt
template=yes
name=NTTempl
associated class=com.ibm.tdm.task.machine.NT4Machine
```

Figure 95. Data store representation of a machine template

When a template is defined, information about the template is recorded in the data store. There are no directories created in `ro\machines`, `ro\macs`, or `rw\machines`, since no physical machine was created. The data store representation of the template is shown in Figure 95 on page 178.

Once you have created a template, you can easily define a machine based on the template. For example, the following command creates the machine named `machine5` based on the template named `NTTempl` that we just defined:

```
machine define name=machine5 templatename=NTTempl mac=002035fe4a6f os=nt4
imagenname=sp3 lang=us server=atlas2
```

Besides the usual specification of OS, Imagenname, and Lang (that are mandatory even if they have already been specified inside the template), you need only specify three parameters to define a machine: the name of the new machine, the template name from which the new machine is derived, and the MAC address of the new machine's network adapter card. You could further automate the machine definitions by encapsulating the commands to create templates and to create machines from templates in JavaScript functions.

A template is treated just as a machine for any of the other console commands. You would delete a template using `machine delete` in the same way that you would delete any other machine. All the other machine commands documented in Section 4.7, "Other machine commands" on page 179 are also applicable to machine templates.

---

## 4.7 Other machine commands

Once a client machine of any operating system type has been defined, it may be manipulated using other actions that are defined for the machine object. These actions are described in the following sections.

### machine list

The `machine list` command is generic to machines of all operating system types and takes no further parameters. When you type `machine list` at the IBM WorkSpace On-Demand 3.0.1 CLI, you are presented with a list of all machines currently defined to the deployment server. This list is broken down by operating system, image name, and language.

### machine query

The `machine query` command requires you to specify the machine name as well as the operating system, image name, and language for that machine. The correct syntax for this command is as follows:

```
machine query name=machine0 os=w2k imagename=w2kpro lang=us
```

```
machine query name=machine1 os=nt4 imagename=sp3 lang=us
```

```
machine query name=machine2 os=w98 imagename=w98se lang=us
```

The deployment server will query and display all the machine's attributes from the data store.

### **machine modify**

`Machine modify` requires you to specify the machine name and operating system, image name, and language, as well as one or more attributes that you wish to change for that machine. The following command, for example, could be used to change the video resolution for a client machine:

```
machine modify name=machine0 os=w2k imagename=w2kpro lang=us  
resolution=1024x768x256@70
```

```
machine modify name=machine1 os=nt4 imagename=sp3 lang=us  
resolution=1024x768x256@70
```

```
machine modify name=machine2 os=w98 imagename=w98se lang=us  
resolution=1024x768x256@70
```

Be careful with the `machine modify` command. In many cases, modifying a machine will reset the machine's state and cause it to be completely reinstalled.

### **machine delete**

This command also requires you to specify the machine name as well as the operating system, image name, and language. For example:

```
machine delete name=machine0 os=w2k imagename=w2kpro lang=us
```

```
machine delete name=machine1 os=nt4 imagename=sp3 lang=us
```

```
machine delete name=machine2 os=w98 imagename=w98se lang=us
```

Executing this command will completely remove the machine specified from IBM WorkSpace On-Demand 3.0.1.1. The machine's corresponding user will be deleted and its entry in the data store will be removed. Also, all directories and files for this machine in the `\ro\machines`, `ro\macs`, and `rw\machines` directories will be deleted.

### **machine parmlist**

This command displays a list of the valid values for a specific parameter for any client machine. It requires you to specify the operating system, image



name, and language, as well as the parameter for which you want more information. The following commands, for example, will provide a list of all the valid protocols for a Windows 2000, Windows NT or a Windows 98 client:

```
machine parmlist os=w2k imagename=w2kpro lang=us parameter=protocols
```

```
machine parmlist os=nt4 imagename=sp3 lang=us parameter=protocols
```

```
machine parmlist os=w98 imagename=w98se lang=us parameter=protocols
```

---

## 4.8 Managing client machines using the administration GUI

You can define, delete, and modify machines and templates through the administration GUI as well as the CLI. From the Resources tab on the left-hand side of the GUI, click **Machines** to see a list of all machine resources already defined. Then click the **Machine actions** tab at the bottom of the right-hand side of the GUI to perform any actions on the machine object. You can create, modify, copy, or delete a machine or define a template as shown in Figure 96 on page 182. The `copy` command allows you to create a new machine definition based on an existing machine so that all the fields in the GUI are already filled in and you only have to change certain key values, such as machine name or MAC address.

### 4.8.1 Defining a machine

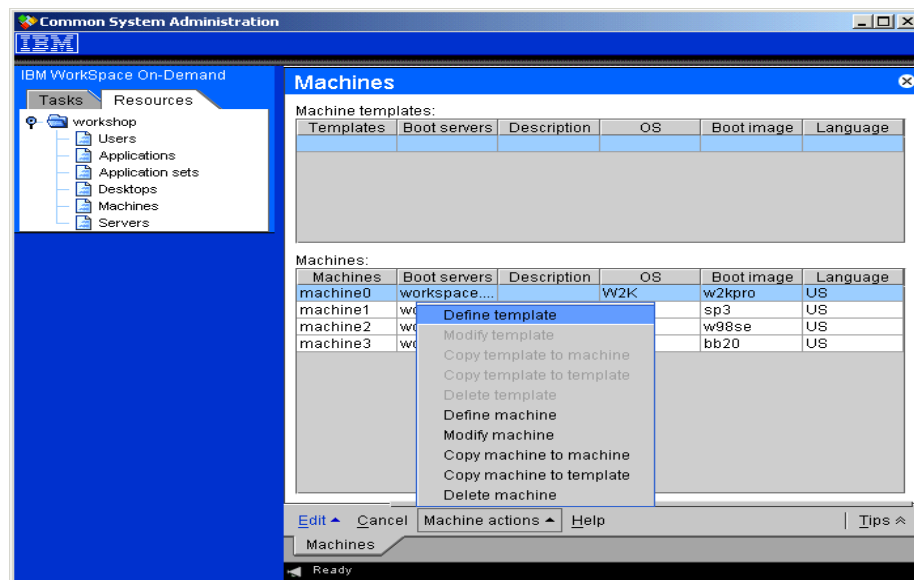


Figure 96. Machine actions in the administration GUI

When you select **Define machine** from the Machine actions menu, you are presented with a list of possible operating systems as shown in Figure 97 on page 183. This list is dynamically built based on the client operating systems that you installed. For example, if you only installed support for Windows 2000 clients, Windows NT, Windows 98 and OS/2 would not appear in the list.

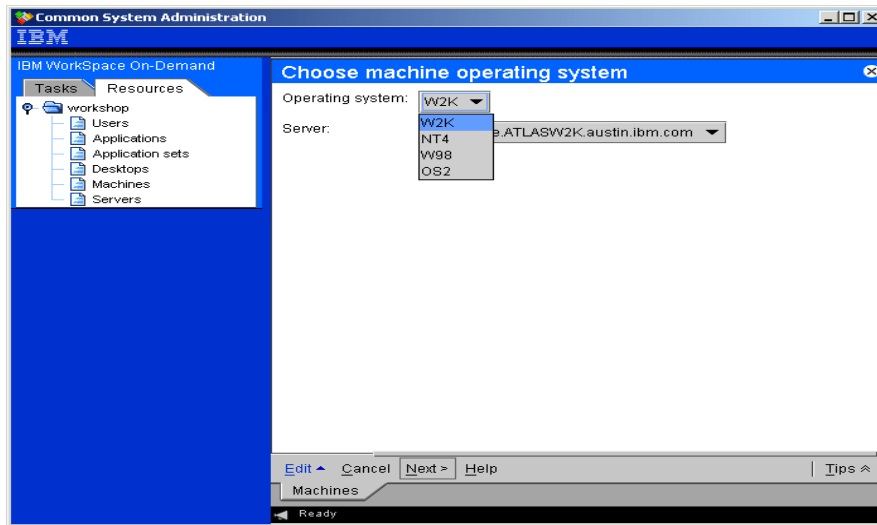


Figure 97. Operating system selection in the administration GUI

When you select the operating system you want for your new machine, you will be presented with a set of dialog boxes that prompt you for all the machine parameters as described in Section 4.2, “Defining Windows 2000 client machines” on page 105, Section 4.3, “Defining Windows NT client machines” on page 124, Section 4.4, “Defining Windows 98 client machines” on page 143 and Section 4.5, “Defining OS/2 client machines” on page 160. The examples on the following pages will demonstrate defining a Windows 2000 client machine. The dialog sequence is similar for other types of clients, although many of the individual fields will differ.

When you define a Windows 2000 client machine in the GUI, you are presented with a dialog consisting of multiple windows selectable by tabs at the top of the window. On each window, required fields are marked with a red asterisk. Other fields may be left blank to take default values. If you miss a required field, you will be prompted to fill it in when you press the OK button to create the machine.

The first window contains parameters relating to the machine identification as shown in Figure 98 on page 184. You must fill in the machine name, which must be different from all other machine names on the deployment server you have selected. You must also select the Imagename and the Language.

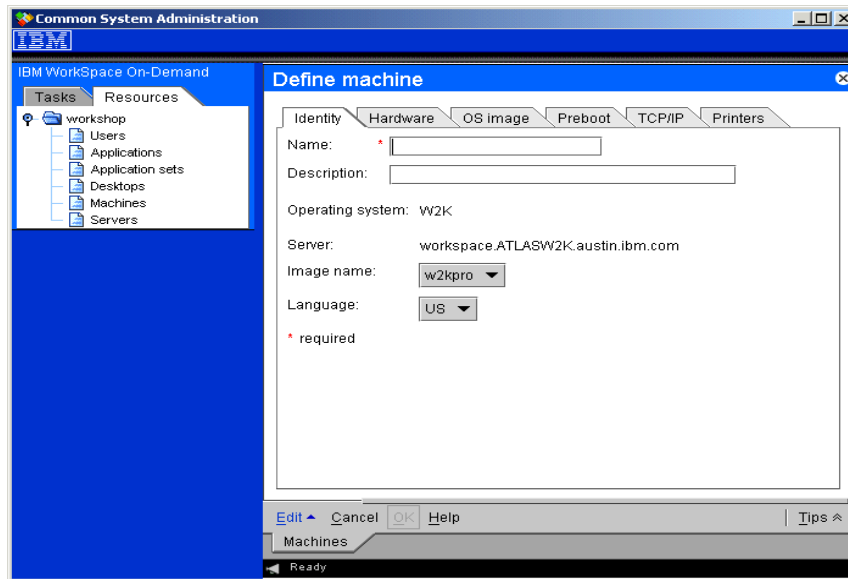


Figure 98. Windows 2000 machine definition - Identity tab

The drop-down list for the base response file allows you to select a response file that will be used to install the Windows 2000 client image. The response files in this list are taken from the w2k\w2kpro\us\resp directory in the RPLFILES alias on your server. If you create your own custom response file, you can place it in this directory so that it will be selectable in the GUI.

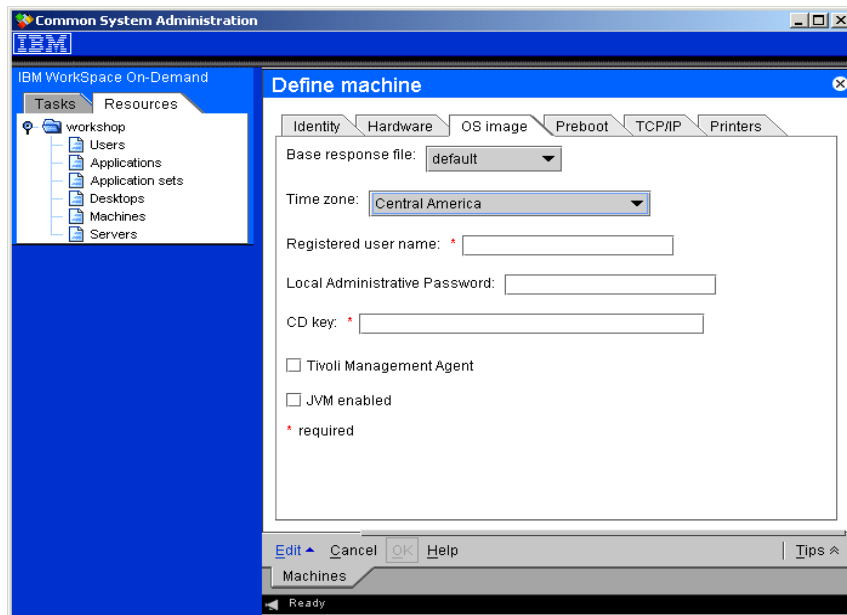


Figure 99. Windows 2000 machine definition - OS Image tab

The third window for a Windows 2000 client definition allows you to define additional machine settings as shown in Figure 99. You must enter a valid Windows 2000 CD key. This is a 25-digit field formatted as five groups of five characters, as follows:

XXXXX-XXXXX-XXXXX-XXXXX-XXXXX.

Each character is alphanumeric. The time zone selection is made via a drop-down list box. You can also select, via check boxes, whether you want the Tivoli management agent and Java support added as part of your client machine's boot image. RegUser is another mandatory field to fill in.

The Hardware tab of the machine definition dialog, shown in Figure 100 on page 186, allows you to specify the keyboard, video resolution and network adapter for your Windows 2000 client machine. You must also fill in the MAC address of the machine's network adapter card. The drop-down list for keyboard type will display all keyboards supported by your Windows 2000 client installation. You can select the national keyboard type you want or leave this field at default.

Windows 2000 should be able to support your client machine's video adapter: you do not need to specify it or supply the driver into the w2k\w2kpro\us\inst\i386\soem\$\\$1\Drivers\Video directory.

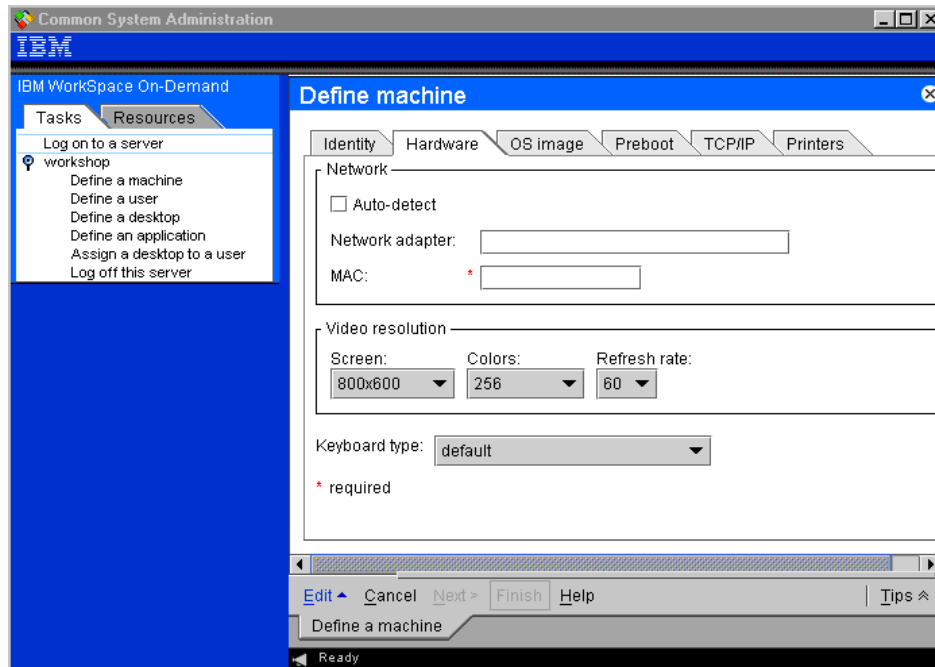


Figure 100. Windows 2000 machine definition - Hardware tab

When installing the Windows NT client, the window is different, as you can see in Figure 101 on page 187. As long as your client machine's video adapter is directly supported by your Windows NT client installation, you can select Auto-detected for the video adapter, and Windows NT will install the appropriate drivers. If it is not directly supported, you will have to select the **Specify** radio button. Then, you can specify the name of the video driver and its .INF file in the entry fields provided. Your video driver must be added to the Windows NT client image on your server in the nt4\sp3\us\inst\i386\soem\$ directory within the RPLFILES alias. You can create a subdirectory containing your video driver and its .INF file in this directory. For either auto-detected or specified video adapters, you must also select a resolution and refresh rate as shown in Figure 101 on page 187

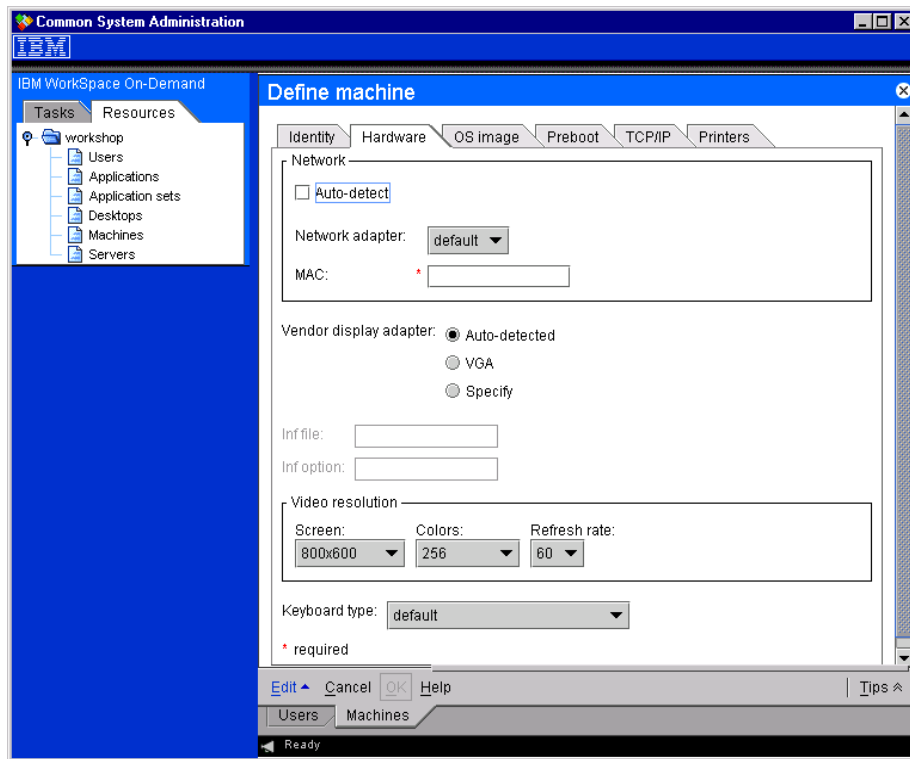


Figure 101. Windows NT machine definition - Hardware tab

You must also specify the type of network adapter in your client machine. Supported network adapters are found in the RPLFILES alias in the w2k\w2kpro\us\inst\i386\soem\$\\$1\Drivers\NIC directory for Windows 2000; while for NT the path is nt4\sp3\us\inst\i386\soem\$\net. The default adapters supported are TRP for the IBM 16/4 PCI Token-Ring adapter with Wake on LAN, and IBMFE for the IBM 10/100 PCI Ethernet adapter. Other adapters can be added in subdirectories in the same location and then specified in the GUI.

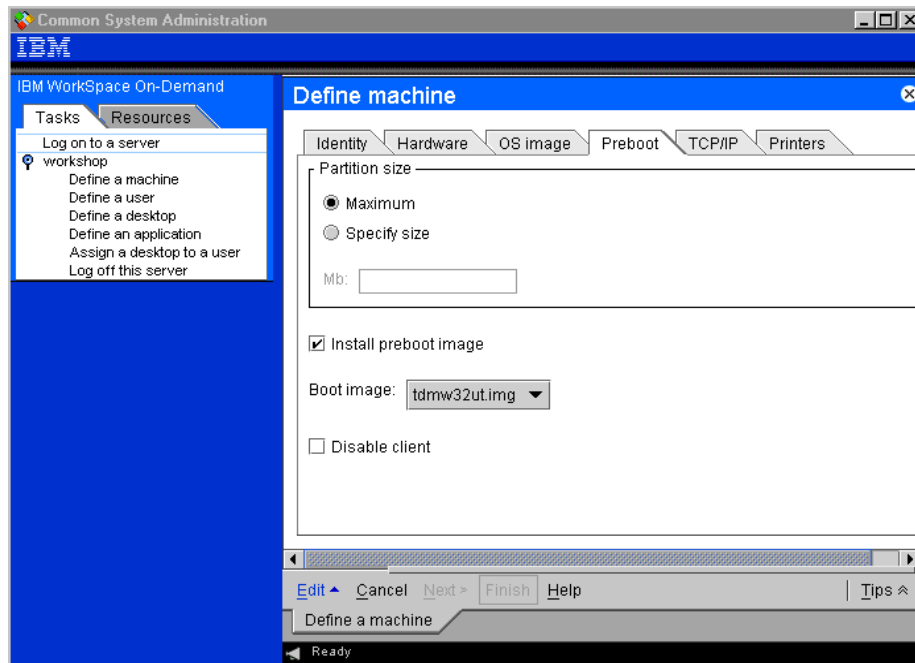


Figure 102. Windows 2000 machine definition - Preboot tab

Figure 102 shows the Preboot tab for a Windows 2000 client definition. This dialog allows you to configure the hard drive for the client machine. You can select your partition size to be the entire client hard drive, or specify a partition size. In either case, the partition cannot be set larger than 2 GB. The preboot image drop-down is populated by the boot images found in the directory w2k\w2kpro\us\boot in the RPLFILES alias. If you select the check box to install the preboot image, DOS will be installed first on your client's hard drive prior to the installation of Windows 2000. This allows the client machine to copy the Windows 2000 files and perform the installation of Windows 2000 while disconnected from the server and is only necessary if the network drivers you are using cannot coexist with the Windows 2000 install programs. For more information about the preboot image installation, please refer to Section 4.2.3, "Windows 2000 client installation and boot" on page 112. If you select the check box for Disable client, the tdmclnt.inf file in your machine's MACS directory will mark the client as disabled so that it will not boot. You will have to modify the machine definition later to enable the client.



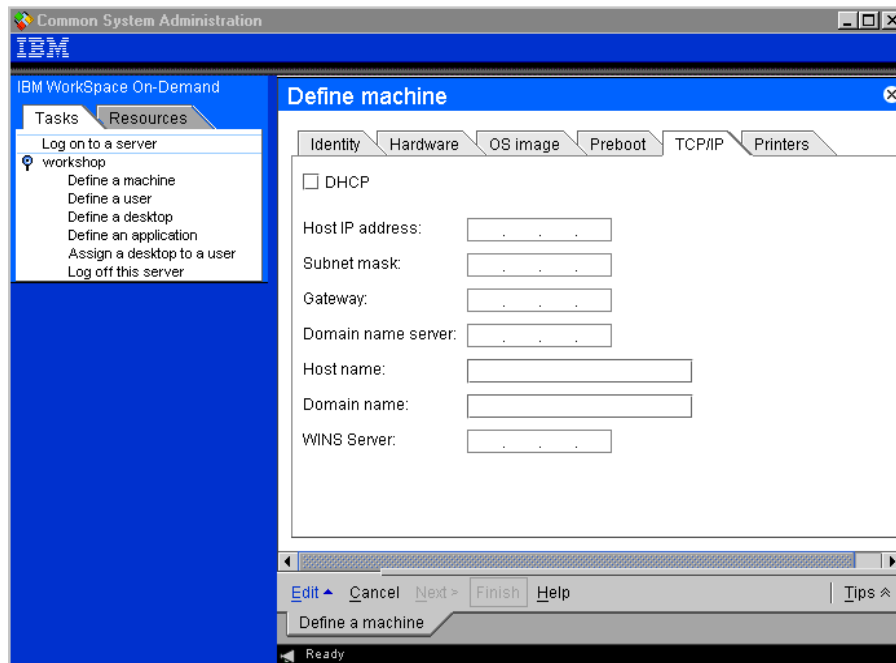


Figure 103. Windows 2000 machine definition - TCP/IP tab

The TCPI/IP page is shown in Figure 103. Specify DHCP to indicate that your client machine will receive its IP address and other parameters from a DHCP server in your network. If you do not specify DHCP, you will have to fill in the remaining network parameters. These parameters are documented in Section 4.2.1, “Windows 2000 client parameters” on page 105 for Windows 2000 client and in the equivalent paragraphs for the other clients.

After you fill in all the machine definition parameters and click **OK**, your client machine will be created on the deployment server. The new machine is automatically added to the machines resource list as shown in Figure 104 on page 190. The message line in the GUI will indicate that the `machine define` command was successful.

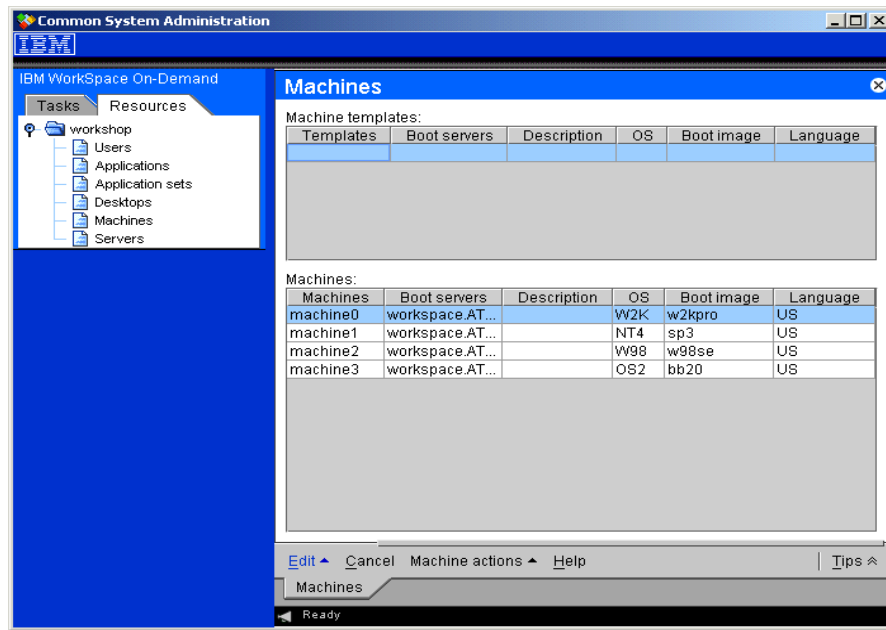


Figure 104. The machines resource list

Defining machines of other operating system types follows exactly the same sequence as was presented for Windows 2000. The dialog windows and fields correspond to the parameters for the `machine define` command as documented in Section 4.2.1, “Windows 2000 client parameters” on page 105, Section 4.3.1, “Windows NT client parameters” on page 124, Section 4.4.1, “Windows 98 client parameters” on page 143, and Section 4.5.1, “OS/2 client parameters” on page 161.

You can perform other machine commands through the administration GUI as well. To delete a machine, simply select the machine in the resource list and click **Delete** from the Machine actions menu. To modify or copy a machine definition, again select the machine you want to work on and then click **Modify** or **Copy** from the Machine actions menu. You will be presented with the same dialog windows as when you created the machine with all values filled in. Change the necessary fields and click **OK** to modify an existing machine or, in the case of **Copy**, create a new machine based on the values set for an existing machine.

### 4.8.2 Defining a template

Defining a template follows exactly the same steps as defining a machine; the only difference is that the MAC address is not a required field.

---

## 4.9 Customizing Windows client machine images

Creating a client machine through the GUI or CLI imposes certain restrictions on how the machine will be built. For example, Windows 2000, Windows NT and Windows 98 clients may not have a hard drive size larger than 2 GB and are restricted to only one primary partition. You may also want to copy additional files to the client hard drive, or install applications or middleware as part of the boot image of the client machine. The following sections describe various ways of modifying the client image prior to installation for Windows 2000, Windows NT and Windows 98 clients, or prior to boot for OS/2 clients.

### 4.9.1 Customizing state.ini

When you define a Windows 2000, Windows NT or Windows 98 client machine, IBM WorkSpace On-Demand 3.0.1 creates a state.ini file for your client based on a default state.ini located in the client operating system directory. State.ini is used to control the installation of the Windows operating system to the client machine's hard drive and is customized prior to installation by the choices made with the `machine define` command. It is possible to add further customization by modifying state.ini prior to creating a machine. You can customize a single machine by modifying the state.ini file found in that machine's client directory, or customize all your Windows clients by modifying the default state.ini in the Windows 2000, Windows NT or Windows 98 client directory.

#### 4.9.1.1 Modifying the disk partition

When you define a Windows 2000, Windows NT or Windows 98 client machine, you can specify the size of the client machine's primary partition. This partition may be no larger than 2 GB, and you cannot create logical drives. Many PCs have hard drives larger than 2 GB, and Windows 98 supports primary partition sizes larger than 2 GB. If you want to define client machines with primary partitions larger than 2 GB, or with an extended partition and logical drives, you can do so by modifying state.ini prior to installing Windows 2000, Windows NT or Windows 98 on the client machine. If you modify either the default or the machine's specific state.ini after the machine has been installed, you will have to delete and redefine the machine for your modifications to take effect.

Figure 70 on page 136 shows a normal state.ini file for a Windows client as produced by the `machine define` command. The parameter PartitionSize in the [parms] section indicates the size of the primary partition as requested by the parameter partition in `machine define`. In Figure 105 on page 193, we have modified the value of PartitionSize in state.ini to be a value larger than 2 GB, in this case, 2.2 GB. We also modified the logic in the [new] section of state.ini to change the maximum partition size checks built into the logic around creating partitions. We selected a new maximum partition size of 2.4 GB, bound by the physical size of the client hard drive. Contrast this with the 2 GB restriction shown in Figure 71 on page 137.

The full state.ini shown in Figure 105 on page 193 is given on the CD accompanying this book as state.ini.bigdrive.

```

[parms]
PartitionSize=2200
InstallDOS=No
MinMemSize=305
MinDiskSize=350
DSize=0
VideoCorrection="NO"
CleanupOSFiles=Yes
InitDiskParms=0
tatusBar "On"
...
[new]
dsize=disksize
Message 260 %Dsize%
delay 2
if %dsize% > %MinDiskSize%
if %PartitionSize%="all"
if %dsize%>2400
PartitionSize=2400
else
PartitionSize=%dsize%
endif
else
if %partitionSize%>2400
Log %InvPartSize%
ErrorExit
endif
endif
else
Log %HardDiskTooSmall%
ErrorExit
endif
;Creating a partition of %d Megabytes

```

*Figure 105. State.ini modified for large partition support*

You can also create more than one partition by modifying state.ini. This requires a little more coding and knowledge of the batch language syntax in state.ini. Figure 106 on page 194 shows selections from a state.ini modified to create a secondary partition and a logical drive in the secondary partition, in addition to the primary partition, resulting in a C: drive and D: drive on the client machine. This state.ini can also be found on the CD accompanying this book as state.ini.multidisk.

```

[parms]
PartitionSize=600
PartitionSize2=600
LogDriveSize=600
...
InitDiskParms=0
DiskParm1=0
DiskParm2=0
DiskParm3=0
DiskParm4=0
OSDISK=C:
OTHERDISK=D:
...
[new]
dsize=disksize
Message 260 %Dsize%
delay 2
...
;Creating a partition of %d Megabytes
Message 261 %PartitionSize%
delay 2
DiskParm1="/PRI:"&%PARTITIONSIZE%
DiskParm2=" /EXT:"&%PARTITIONSIZE2%
DiskParm3=" /LOG:"&%LOGDRIVESIZE%
DiskParm4=%DiskParm1%&%DiskParm2%
InitDiskParms=%DiskParm4%&%DiskParm3%
delay 2
initdisk 1 %InitDiskParms%
Message 285
delay 5
;echo "Rebooting the system...."
...

```

Figure 106. *State.ini* modifications for multiple partitions

State.exe executes in a restricted DOS shell to process the commands in state.ini. Some DOS commands can be used in state.ini and the batch language allows conditional branching, variable substitution, and string concatenation operations. These capabilities allow you to extend the use of state.ini.

In order to create multiple partitions, we first set up additional variables in the [parms] section of state.ini. Variable names are not case sensitive, but all variables must be defined in the [parms] section before they are used in later sections. We created additional variables for the size of the secondary

partition and the size of the logical drive, as well as temporary variables used to build the command string that will create the partitions.

To partition the hard drive, `state.ini` uses an internal command called `initdisk` that resolves to the DOS `FDISK` command. Parameters for `FDISK` can be used on `initdisk` as well. In order to create both the C: and D: drives, we had to extend the default parameters for `initdisk`. Figure 71 on page 137 shows the default `initdisk` in `state.ini`. The default parameter for `initdisk` is the string `/PRI:` followed by the disk size as specified in the variable `PartitionSize`. Variable substitution is done by enclosing the variable name with the `%` symbol, and string concatenation is done using the `&` symbol.

Figure 105 on page 193 shows the modifications we made to the `initdisk` command. The batch language does not allow more than one variable substitution per line, nor does it allow concatenation of more than two strings at once. To extend `initdisk` for a logical drive, we needed to add two more arguments: `/EXT` (to create the extended partition) and `/LOG` (to create the logical drive in the extended partition). Each of these arguments is followed by the size of the partition or drive. We used temporary variables to set up each part of the parameter string by concatenating one literal string with one variable substitution. Then, we used temporary variables to piece the whole string together, two parts at a time. The result is an `initdisk` command that resolves to `fdisk /pri:600 /ext:600 /log:600`.

#### 4.9.1.2 Formatting partitions

To format the logical drive, we chose to extend the `[Format]` section of `state.ini`. This way, the logical drive would be available to copy files or install boot image applications during the Windows client installation. It would also be possible to have the `format` command executed along with other commands at the end of the system installation as described in Section 4.9.2, “Executing commands after installation” on page 197.

Figure 107 on page 196 shows an extension of the `[Format]` section of `state.ini` to format two partitions. We used the DOS `echo` command to display messages indicating the beginning and end of the second format and simply added another `formatdisk` command to format the second partition. The `state.ini` `formatdisk` command resolves to the DOS `FORMAT` command and accepts the same parameters as `FORMAT` does. By extending the `[Format]` section in this way, we were able to display messages to the user indicating that the second format was in progress and show a progress indicator during the second format operation.

```

;echo "Formatting the partition...please wait..."
Message 264
FormatCustomize "on"
OutputFile "Format.out"
if %InstallDOS%="Yes"
FormatDisk %OSDISK% "/U /V:WNT /s"
else
FormatDisk %OSDISK% "/U /V:WNT /b"
endif
FormatCustomize "off"
;echo "Format Completed Successfully"
Message 265
delay 3
echo "Formatting second partition ... please wait ..."
FormatCustomize "on"
OutputFile "Format.out"
FormatDisk %OTHERDISK% "/U /V:TEST /b"
FormatCustomize "off"
echo "Format Completed Successfully"
delay 3

```

Figure 107. Formatting a second partition

#### 4.9.1.3 Modifying other sections of state.ini

State.ini for windows 2000, Windows NT and Windows 98 clients has three additional sections, labelled [Copy], [Install], and [Logon]. The [Install] section transfers control to the Windows installation routines and probably needs little, if any, customization. The [Logon] section of state.ini installs the IBM WorkSpace On-Demand 3.0.1 logon routine, which causes the client to perform a logon to the server rather than the default Windows 2000 or Windows NT local logon. This section is not typically modified.

You can modify the [Copy] section of state.ini to copy additional information to the client hard drive prior to installation. The `MCOPY` commands in this section cause files to be copied based on source and target information provided by a control file. It is possible to add additional `MCOPY` commands and write your own control files, or simply add additional source and target information to a control file already used.

Figure 108 on page 197 shows a sample `MCOPY` command and the contents of its control file, `misc.lst`. We added an additional file to `misc.lst`, called `yes.txt`, to be copied from the machine directory to the root of the C: drive on the client machine. You can specify any valid source and target directories for additional `copy` commands; however, the source directory on the server must



be accessible by the client machine. This means that it must be in an alias to which members of the user group RPLGROUP have access. The Z: drive is already set up to map to the RPLFILES alias on the deployment server.

```
MCOPY /I:Z:Misc.lst /s /r /e /l:copy.out

z:\machines\machine1\NT4\sp3\us\unattend.txt c:\ibmwin32\
z:\machines\machine1\NT4\sp3\us\access.cmd c:\logon\
z:\machines\machine1\NT4\sp3\us\custom.inf c:\logon\
z:\machines\machine1\NT4\sp3\us\yes.txt c:\
```

Figure 108. The MCOPY command and the misc.lst file

#### 4.9.2 Executing commands after installation

IBM WorkSpace On-Demand 3.0.1 gives you the ability to execute other commands on the Windows 2000, Windows NT or Windows 98 client machines after installation but before the first user logon. When you create a machine, IBM WorkSpace On-Demand 3.0.1 places a file in the machine's directory containing commands that it needs to have executed based on parameters you selected in the `machine define` command. For Windows 2000 and for Windows NT clients, this file is called `cmdlines.txt`, while for Windows 98 clients it is called `custom.inf`.

```
[Commands]
"rundll32 setupapi,InstallHinfSection DefaultInstall 128 .\delicons.inf"
"rundll32 setupapi,InstallHinfSection DefaultInstall 128 c:\logon\ntclean.inf"
"cmd /c copy c:\logon\state.ini \winnt"
"cmd /c copy c:\logon\custom.inf \winnt"
"c:\logon\RunOnce.cmd"
"C:\JVM\SETJVM.cmd"
"convert c: /fs:ntfs < c:\yes.txt"
"C:\logon\setlogon.cmd"
```

Figure 109. `cmdlines.txt` file for a Windows 2000 and a Windows NT client

For example: Figure 109 shows the `cmdlines.txt` file for a Windows 2000 or Windows NT client. We modified this file by adding one additional command in the second-to-last line. This is the Windows NT `convert` command, which allows you to convert a partition that has already been formatted as FAT to NTFS. The `convert` command prompts the user for a 'y' or 'n' response, so we

redirected its input to the file yes.txt, which was copied to the client machine in Section 4.9.1.3, “Modifying other sections of state.ini” on page 196. You can use this technique to add any other commands you need to your Windows client installation.

---

#### **4.10 Distributing client machines in a network**

Defining and installing client machines is reasonably easy in a small LAN-based environment. However, care must be taken when deploying many machines over a large enterprise network. In this section, we examine some considerations for large-scale deployments of IBM WorkSpace On-Demand 3.0.1 workstations, concentrating on installation considerations for Windows 2000, Windows NT and Windows 98 clients.

For Windows 2000, Windows NT and Windows 98 client machines, the initial installation of these machines can take a great deal of time. On a 16 MB token-ring network, installing one Windows NT client or a Windows 2000 client from a NetFinity 3000 server took just under half an hour. Installation of a Windows 98 client took closer to 45 minutes. In addition to the time taken for the installation, you need to consider the network traffic implications of remote installation of many client machines simultaneously.

If the client machines are equipped with network cards that have Wake on LAN capability, it would be possible to leave the machines turned off and schedule their installations for staggered times overnight or over a weekend to reduce the impact on the network. This is not always possible, however. For example, if a machine needed to be immediately replaced in the field, the bandwidth impact and installation time can become an issue.

To avoid this problem, it would be good to pre-install a client workstation with Windows 2000, Windows NT or Windows 98 at a central site prior to shipping it to the business location where it will be deployed. Many organizations outsource their network service requirements to third-party service providers who handle the standard machine configuration and install it at their customer's location. This service provider or internal service organization could keep an IBM WorkSpace On-Demand 3.0.1 server and client configuration on behalf of the customer. They would then perform the machine definition and installation for a generic client and deploy it at the business office.

To make this work in IBM WorkSpace On-Demand 3.0.1.1, we need to be able to transfer a specific machine's definition from one deployment server to another, and configure the definition on the second deployment server so that

the client machine will not reinstall when it is connected and booted. Working through the installation processes described in Section 4.2, “Defining Windows 2000 client machines” on page 105, Section 4.3, “Defining Windows NT client machines” on page 124 and Section 4.4, “Defining Windows 98 client machines” on page 143, we found a straightforward procedure that can accomplish this transfer.

The key is knowing that the client machine’s boot sequence is determined primarily by the `tdmclnt.inf` file found in the directory of the client’s MAC address in `tdm\mm\client\ro\macs`. Before installation, `tdmclnt.inf` points to the bootstrap image called `tdmdos.sys`, which causes the client to be installed. When the installation is complete, `tdmclnt.inf` points to a different bootstrap image, called `hdboot.com`. This redirects the client’s boot to the local hard drive. Also, the client machine’s directory in the `WRKFILES` alias, known as `tdm\mm\client\rw\machines`, is initially empty when the machine is defined but contains log files and the file `counter.ini` when the install has completed. The `counter.ini` file indicates the machine state, which should be set to 4.

Knowing that these changes happen as a result of installation, we can easily execute the `machine define` command on both deployment servers, then fix `tdmclnt.inf` and `counter.ini` on the target deployment server to simulate that the client machine has been installed from there. When the client machine finally boots from the target deployment server, its boot will be redirected to its own hard drive and the installation will be bypassed.

It is easier to transfer the machine if its machine name is made the same on both deployment servers. However, if this is not the case, the transfer can still be accomplished by making one more file change. In the client machine’s `MACS` directory, along with `tdmclnt.inf`, is a second file called `tdmdos.inf`. This file records the machine name that would be used for any subsequent rebuild of the client machine. Change the line `CLIENT_NAME=` in `tdmdos.inf` to the new name for the client machine on the target deployment server. It is also possible just to copy `tdmdos.inf` to the target deployment server.

The steps to transfer a machine are listed below and illustrated in Figure 110 on page 200:

1. The service provider executes `machine define` on a deployment server, creating a Windows 98, a Windows 2000 or Windows NT client.
2. The new client machine boots to the service provider’s deployment server, installing its Windows 98, Windows 2000 or Windows NT operating system image.

3. The MAC address of the client machine's network card is communicated to the network administrator.
4. The network administrator executes `machine define` on the production deployment server that will manage the new client machine using the same MAC address.
5. The files `tdmclnt.inf` and `tdmdos.inf` are transferred from the MACS directory on the first deployment server to the MACS directory on the second deployment server.
6. The contents of the client machine's directory from `WRKFILES` on the first deployment server are copied to the client machine's directory in `WRKFILES` on the second deployment server.
7. The client machine is transferred and connected to the second deployment server.
8. The client machine boots, and boot is redirected to its hard drive without installation.

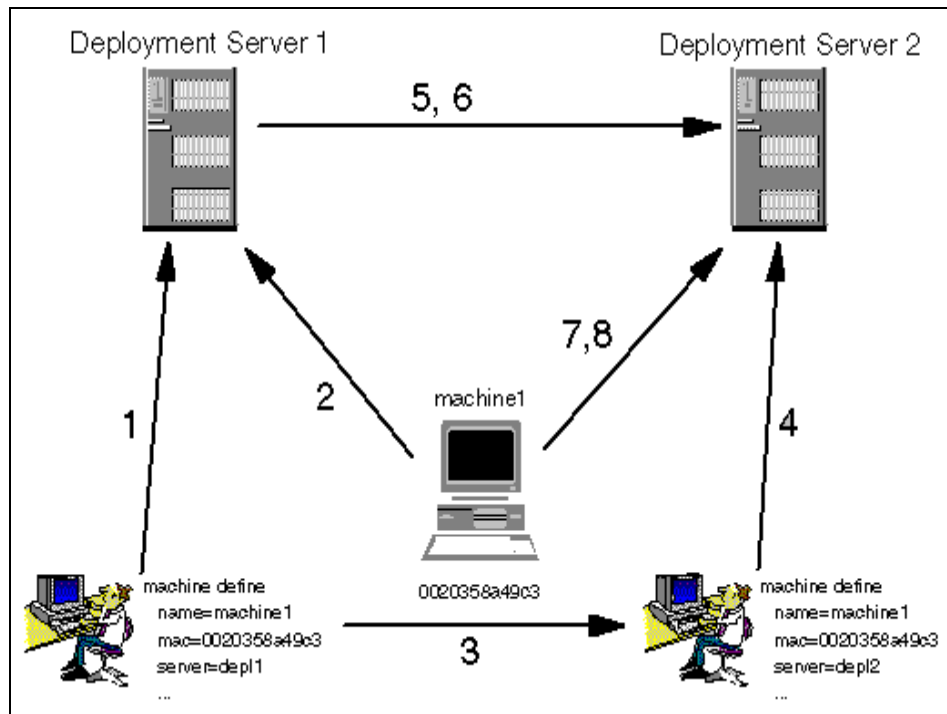


Figure 110. Transferring a machine definition between deployment servers

---

## Chapter 5. Managing client desktops

IBM WorkSpace On-Demand 3.0.1 allows you to define desktops separately from users and machines. The desktop object encapsulates the look and feel of the operating system image with which the user will interact. When a desktop has been defined, it can be assigned to one or more users. This allows roaming users to preserve their user interface when working on different machines, since the user's desktop will be set up at logon. The following sections describe how to define and modify desktops and assign them to users.

---

### 5.1 Planning your desktop

Before starting to create a desktop, you should consider what the user environment should look like. The following questions can help you better prepare your implementation:

- What folders should be on the desktop?
- What should the Start menu look like?
- How should shortcuts and links be displayed to the user?
- How much of the user's environment should be controlled?
- What should the background look like; is there a corporate standard?
- What will the desktop look like on different hardware platforms?

When deciding what a user should be presented with, it is wise to consult with the user community and understand their requirements. As a rule of thumb, users who perform very specific, repetitive tasks daily prefer a simple desktop with few, easily accessible icons. More sophisticated users who perform a variety of different tasks and are able to navigate their way around the desktop will want to have more individual control and, therefore, less restricted desktops.

The more restricted the desktop is made, the easier it is to manage, and this lowers the overall cost of support. However, you need to ensure that this does not hamper the overall productivity of the individual user.

For task-based users, you would want to create a profile geared toward enhancing the work environment for the set of applications that the users need. You would hide functionality from the user for two reasons:

- To simplify the user interface so that the user has easy access to the functions required.

- To focus the user's attention on what he or she must do to get the job done.

Try to create a few common desktop types. Do this by grouping users into the tasks they perform, then create desktops associated with those groups.

---

## 5.2 General desktop representation

Desktops are represented as objects in IBM WorkSpace On-Demand 3.0.1 just like users and machines. They have representations in the data store as well as in the tdm pkg share, which is \tdm\tdmpkgs. The following actions are defined for the desktop object in IBM WorkSpace On-Demand 3.0.1:

**list** Displays a list of all defined desktops.

**define** Creates a new desktop.

**delete** Deletes a desktop from IBM WorkSpace On-Demand 3.0.1.

**modify** Changes one or more attributes of a desktop object.

**query** Shows all the attributes of a desktop object.

Depending upon the operating system you select, the parameters for each of the above actions will differ. The desktop object encapsulates the user shell, which is the program that forms the primary user interface for the operating system. For Windows 2000, Windows NT and Windows 98, the default shell is Explorer. For OS/2, the default shell is pmshell.exe. You can use a different program for the shell in any client operating system.

Each client operating system has default desktops already provided with IBM WorkSpace On-Demand 3.0.1. For Windows 2000 Professional Edition, Windows NT Workstation and Windows 98 clients, there are two desktops provided, one called default and one called sandbox. The default shell can be applied to any user and forms the basis for assigning applications to the user. The sandbox shell is assigned only to a specific administrator defined as a sandbox user. The sandbox user is only used to create application packages for distribution to other users and is explained more fully in Chapter 6, "Applications" on page 259. The OS/2 client only has the built-in default desktop, since sandbox users are managed differently in this environment.

When you create a user in IBM WorkSpace On-Demand 3.0.1, the user, by default, has no desktop. Without a desktop, the user is unable to log on to a client machine. A user also must have a desktop before applications can be assigned. Therefore, one of the first things that must be done for each new user is to assign a desktop.

When you assign a desktop to a user, you also specify the operating system, language, and shell that go with the desktop. The details of assigning a desktop to a user is covered in the following operating system-specific sections. Note, though, that you can assign multiple desktops to the same user but, at most, one per client operating system. Therefore, the same user can have a desktop for Windows 2000, Windows NT, Windows 98, and OS/2, allowing the user to roam between machines of different operating system types. As discussed in Chapter 6, “Applications” on page 259, applications are assigned to the combination of user and desktop so the user’s application set would automatically be customized to the operating system to which the user is logging on.

---

## **5.3 Windows 2000 Professional desktops**

The following sections describe the structure and definition of desktops for the Windows 2000 Professional Edition operating system and how to assign Windows 2000 desktops to users in IBM WorkSpace On-Demand 3.0.1.

### **5.3.1 Windows 2000 desktop structure**

IBM WorkSpace On-Demand 3.0.1 maintains desktops for Windows 2000 clients in three parts. You can think of a desktop as comprising the shell, the profile, and the policy that apply to a given user. Figure 111 on page 204 shows the representation of Windows 2000 desktops in the TDMPKGS share. The default and sandbox desktops are kept as subdirectories within Explorer, and policy and profiles directories are kept in the desktop\w2k\us directory. When you define new desktops, they will also be represented as subdirectories in these locations.

Default policies and profiles are stored in the files policy\default\ntconfig.pol and profiles\default\ntuser.dat. These can be used as the basis for creating custom desktops of your own. A sandbox policy is stored in the file policy\sandbox\ntconfig.pol. These are typically only used by sandbox users to gain unrestricted access to a client machine in order to configure an application and define the application package. The sandbox user has no profile.

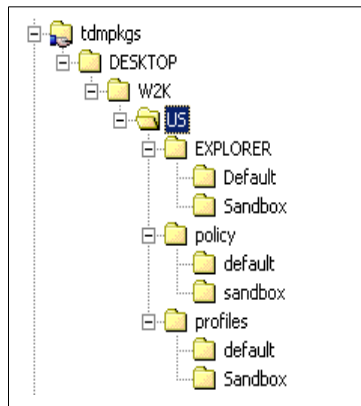


Figure 111. The default desktop directories for Windows 2000 clients

A Windows 2000 desktop also has a data store entry as shown in Figure 112. Only the default desktop is shown; the sandbox desktop entry is identical, except that the profile and policy lines refer to sandbox instead of default. Notice the three location lines, PackageLocation, PolicyLocation, and ProfileLocation, all refer to the directories shown in Figure 111. Therefore, the data store definition of a desktop completely identifies for IBM WorkSpace On-Demand 3.0.1 where to find its shell, policy, and profile data. Any new desktops that you create will have similar entries in the data store.

```

[/DESKTOP]

[/DESKTOP/W2K]

[/DESKTOP/W2K/EXPLORER]

[/DESKTOP/W2K/EXPLORER/US]

[/DESKTOP/W2K/EXPLORER/US/Default]
PackageLocation=C:\TDM\tdmpkgs\DESKTOP\W2K\US\EXPLORER
PolicyLocation=C:\TDM\tdmpkgs\DESKTOP\W2K\US\POLICY
ProfileLocation=C:\TDM\tdmpkgs\DESKTOP\W2K\US\PROFILES
Policy=Default
Profile=Default
associated class=com.ibm.tdm.task.desktop.W2KExplorerDesktop

```

Figure 112. The default Windows 2000 desktop in the data store



The profile component of the desktop is used primarily to customize the user interface. By modifying a profile, the administrator can set background bitmaps, color schemes, screen savers, and similar characteristics of the user's interface. Administrators can customize user profiles and assign them to users to enforce a corporate standard. User profiles are also locked down in IBM WorkSpace On-Demand 3.0.1 so that users cannot change them.

*System policies* are used mainly to restrict what users can do. For example, you can use system policies to restrict what users can do from their desktops, such as disabling certain **Control Panel** options or configuring network settings. The default desktop for Windows 2000 clients has its policies completely restricted so that users with this desktop have no access to the **Control Panel** or any other system settings and are thus unable to change anything in their user interface.

A policy consists of two parts: computer and user. The computer section of a policy contains settings for network, printing, and other system-related values while the user section contains settings for the **Control Panel**, shell, and other interface-related values. Windows 2000 Server provides a utility known as the System Policy Editor (\winnt\poledit.exe) that allows you to view and modify policy information. This can be used to customize the policy portion of the desktop when you build a customized desktop.

Policy information is maintained in a file called ntconfig.pol. This file can be found in each desktop directory under \tdm\tdmpkgs\desktop\w2k\us\policy. When a desktop is assigned to a user, the policy file is copied to the user's profiles directory, \tdm\tdmprfls\<username>\w2k\us\profiles. The policy file contains registry settings that are copied to the user or local machine portions of the registry on the client machine when the user logs on.

There is some overlap between policies and profiles in that the user section of the policy also allows you to modify settings, such as background bitmaps and color schemes. It is recommended that the System Policy Editor be used to make as many changes as possible for a new desktop in order to localize the changes as much as possible in one place. Customizing and modifying desktop policies and profiles are described in more detail in Section 5.3.3, "Defining a new desktop for Windows 2000 Professional" on page 207.

### 5.3.2 Assigning a Windows 2000 desktop to a user

You can add a desktop to a user via the `assign` action on the user object. When you assign a desktop to a user, you must also specify the operating system, language, and shell. For example, the following command assigns the default Windows 2000 desktop to a user named user1:

```
user assign user=user1 desktop=default os=w2k shell=explorer lang=us
```

Without creating a new desktop, you can modify some basic desktop values at the time that you assign a desktop to a user. The values you can set are:

- The desktop bitmap (wallpaper)
- The way the desktop bitmap is displayed (tiled or centered)
- The color of the desktop background

The console support for changing the desktop of a user is limited to these settings. If you want to change anything outside this scope, you need to define your own desktop. This is described in Section 5.3.3, “Defining a new desktop for Windows 2000 Professional” on page 207.

The following parameters are available for changing a user’s desktop:

Bitmap=“bitmap”

The fully qualified path of the desktop bitmap to use.  
For example: Bitmap=“C:\test.bmp”

BitmapDisplay=“C | T”

Specifies how to display the bitmap, where:  
C = Centered  
T = Tiled

Background=“background”

RGB value specifying the desktop background color.  
Format: X X X  
Where X is 0 - 255  
For example: Background=“0 128 0”

For example:

```
user assign user=user1 desktop=default os=w2k lang=us shell=explorer  
bitmap="C:\Winnt\wsoddtbg.BMP" BitmapDiskplay="C" Background="210 210 210"
```

This command assigns the bitmap “C:\Winnt\wsoddtbg.BMP” in a centered display with a grey background color.

When you assign a desktop to a user, the user’s data store entry is modified to reflect the desktop and settings. Figure 113 on page 207 shows the data store representation of our user, user1, after executing the above command. Notice that the bitmap and color information is included with the specification

of the default desktop. These values could, in fact, be set for any desktop for this user.

```
[/USER/user1]
USER=user1
Type=USER
RIDid=1130
PROFILEDIR=\\ATLAS2\TDMPRFLS\user1
HomeDrive=H
associated class=com.ibm.tdm.task.principal.TDMUser

[/USER/user1//DESKTOP/W2K/EXPLORER/us/default]
bitmap=c:\winnt\wsoddtbg.bmp
bitmapdisplay=c
```

Figure 113. Data store representation of a user with a desktop

### 5.3.3 Defining a new desktop for Windows 2000 Professional

To define a new desktop, you can set up both a new policy and a new profile. It is best to copy an existing desktop and modify the attributes to suit your needs for the new desktop. For example, the steps outlined in this section show you how to copy and modify the default Windows 2000 client desktop.

#### 5.3.3.1 Modifying the profile

Modifying the profile is done on a client machine using a user ID that has administrator authority very similar to a sandbox. (See Chapter 6, “Applications” on page 259 for more information about sandbox users.) The profile changes will be saved back to the IBM WorkSpace On-Demand 3.0.1 server. Modifying the policy for a new desktop can be done directly on the server using the System Policy Editor poledit.exe.

To begin, the following steps take you through the process of modifying the profile for a new Windows 2000 desktop:

1. Create a new Windows 2000 Professional client in the IBM WorkSpace On-Demand 3.0.1 console.
2. Create a user on the IBM WorkSpace On-Demand 3.0.1 console. Our example will use the user name UserDesk.
  - a. Create the user itself:

```
user define type=USER user=UserDesk
```
  - b. Reset the password and prevent the change of the password:

```
nativeuser modify password="" passwordexpiration=NEVER user=UserDesk
```

- c. Assign the default desktop to the user so that we can change it later:
- ```
user assign desktop=default os=w2k lang=us shell=explorer
user=UserDesk
```
- d. Make the user a member of the “Domain Admins” group. This turns our user into an administrator so that we have permission to modify settings on the client machine.
- ```
groupmember define group="Domain Admins" user=UserDesk
```
3. Replace the policy file of the user with the sandbox policy. This will remove the restrictions that are in place by default on normal users. You must do this by entering the following command in a DOS command prompt on your server, not in the IBM WorkSpace On-Demand 3.0.1 CLI console:
- ```
copy C:\Tdm\tdmpkgs\Desktop\w2k\us\policy\sandbox\ntconfig.pol
C:\Tdm\tdmprfls\UserDesk\w2k\us\Profiles
```
4. Logon to the client with the user ID UserDesk.
5. Make changes to the desktop as you see fit. Most of these changes can be applied by modifying the desktop properties in the **Control Panel**. Some common changes are as follows:
- a. To change the wallpaper, click the **Background** tab as shown in Figure 114 on page 209. The picture you select is applied as the wallpaper. You can also select to tile the wallpaper across the window. Available wallpaper bitmaps are contained in the C:\Winnt directory. In our example, we used the custom bitmap eyes.bmp.

#### Note

If you want to use your own bitmap that is not supplied with Windows 2000 or IBM WorkSpace On-Demand 3.0.1, you have to ensure that it is copied to the C:\Winnt directory of the client machine first. The best way to do this is to copy your bitmap to the directory w2k\w2kimg\us\inst\i386\soem\$c\winnt in the RPLFILES alias on your IBM WorkSpace On-Demand 3.0.1 deployment server before you define any client machines. If this directory does not already exist, you can create it. Then, when you define a client, your bitmap will be copied to C:\Winnt on the client hard drive along with all the built-in bitmaps.

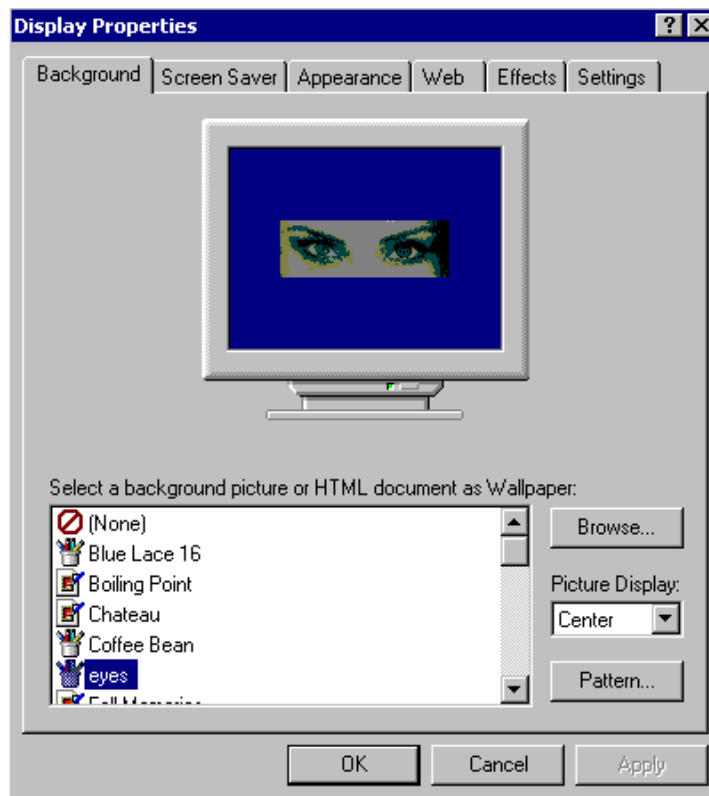


Figure 114. Modify the background wallpaper properties

- b. To change the color scheme, click the **Appearance** tab in the Display Properties dialog as shown in Figure 115 on page 210. You can change individual colors of desktop objects or set an entire scheme. Before you change the color scheme, be sure to consult the user community.

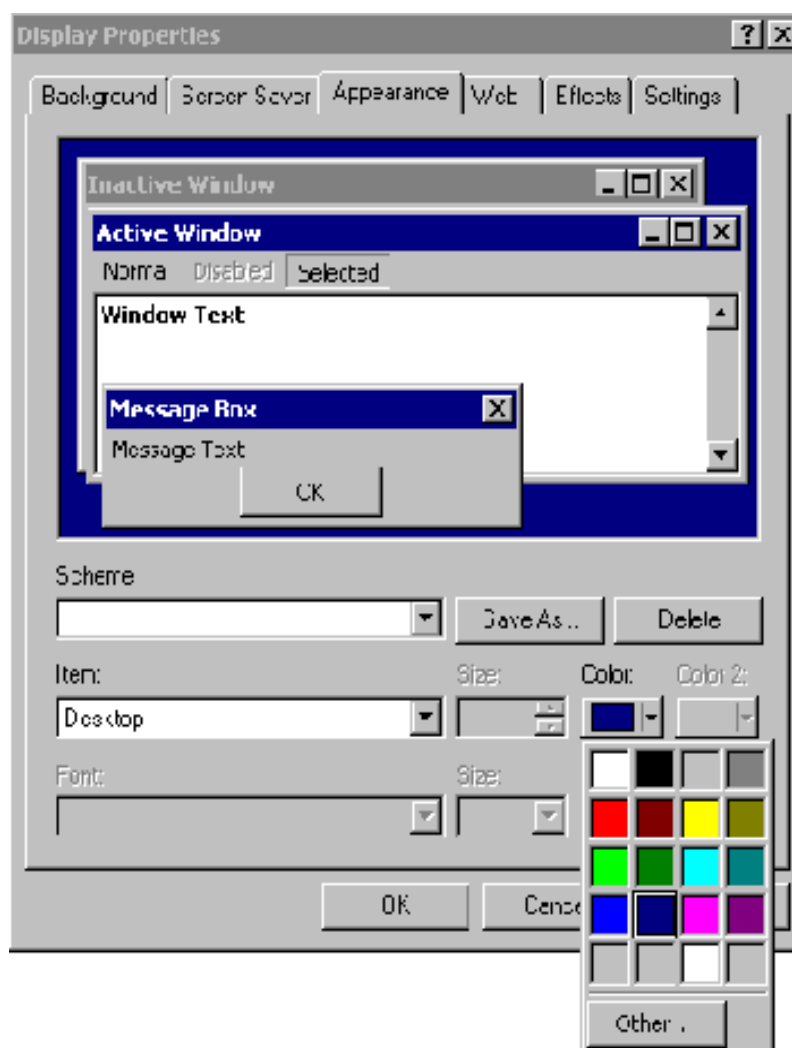


Figure 115. Modify the color scheme

- c. To change the screen saver, click the **Screen Saver** tab as shown in Figure 116 on page 211.

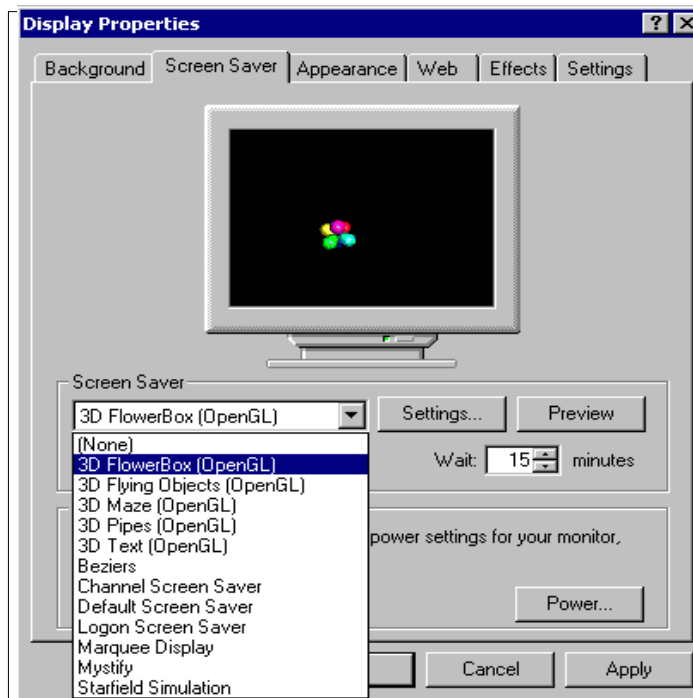


Figure 116. Modify the screen saver properties

6. Click **OK** from **Display Properties** to apply the changes to your desktop.
7. To change the **Start** menu, you will need to change the contents of two directories on your client. These are:

- a. C:\Documents and Settings\UserDesk\Start Menu
- b. C:\Documents and Settings\All Users\Start Menu

The user UserDesk to which we are logged on does not have direct access to a Windows 2000 command prompt or Windows Explorer from the desktop. You can, however, select **Run** from the **Start** menu and enter `explorer` to launch Explorer and access the hard drive. You can also access the **Start** menu directories by right-clicking the mouse on the **Start** icon and selecting either **Open** or **Open All Users**. You can now add, remove, or modify any options on the **Start** menu.

8. When you have finished your modifications, close all windows, empty the Recycle Bin, and remove any documents from the **Taskbar**.
9. Logoff from the client. The changes applied to the desktop are saved in the file *ntuser.dat* as well as other subdirectories in our user's profile

directory C:\Tdm\tdmprfls\UserDesk\w2k\us\Profiles\Desktop, as shown in Figure 117.

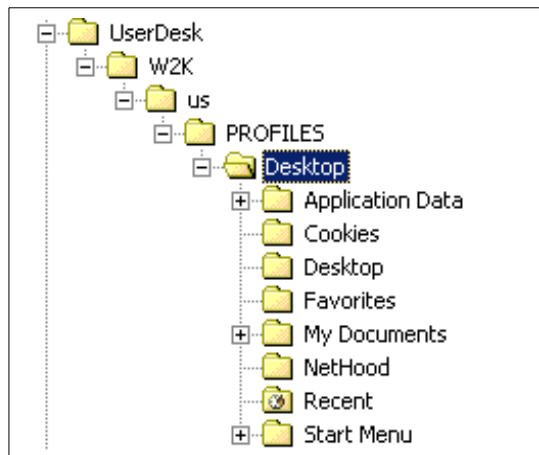


Figure 117. The modified Windows 2000 profile information

#### 5.3.3.2 Creating the desktop directory structure

Having created the profile for our new desktop, we need to create the directory structure to hold the desktop information. In our example, we will name our new desktop EyeDesk. Therefore, we need to create the following three new subdirectories:

- \tdm\tdmpkgs\desktop\w2k\us\explorer\eyedesk
- \tdm\tdmpkgs\desktop\w2k\us\policy\eyedesk
- \tdm\tdmpkgs\desktop\w2k\us\profiles\eyedesk

We also need to copy our new profile to the profiles\eyedesk subdirectory. Copy all the files and subdirectories in the userdesk\nt4\us\profiles\desktop directory (shown in Figure 117) to the directory c:\tdm\tdmpkgs\desktop\w2k\us\profiles\eyedesk. This will ensure that all the changes you made while logged on as UserDesk will be applied to the new desktop.

#### 5.3.3.3 Modifying the desktop policy

Now that we have created a profile for our new desktop, we may also want a new policy. To create a new policy, it is easiest to copy and modify an existing policy. There are two default policy files with which we can start. Both are called ntconfig.pol, but one is a restricted policy intended for normal users,



while the other is an unrestricted policy intended for sandbox users. We will start with the restricted policy and make modifications as required:

1. Copy the default restricted policy to our new desktop directory, using the following command at a Windows command prompt:

```
copy c:\tdm\tdmpkgs\desktop\w2k\us\policy\default\ntconfig.pol  
c:\tdm\tdmpkgs\desktop\w2k\us\policy\eyedesk
```

2. Modify the policy file for the EyeDesk desktop using the System Policy Editor. The System Policy Editor is a graphical tool provided with Windows 2000 that allows you to create or modify the registry entries contained in the policy file. On your IBM WorkSpace On-Demand 3.0.1 server, run `c:\winnt\poledit.exe`. From **File**, select **Open Policy** and browse to locate the policy for EyeDesk as shown in Figure 118.

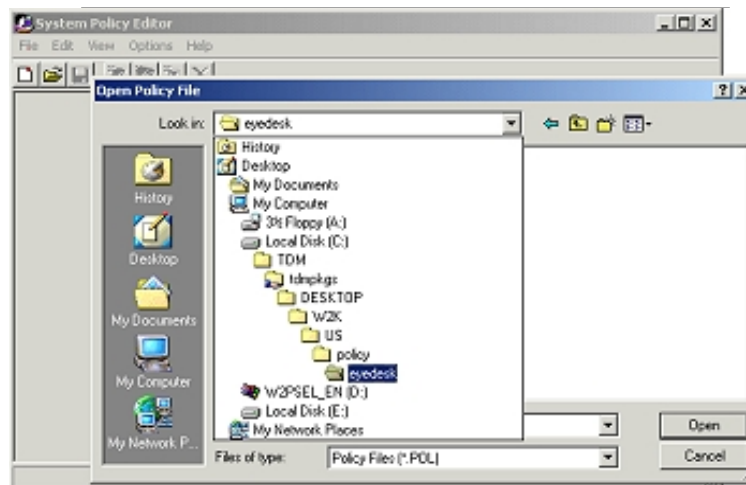


Figure 118. POLEDIT: Open the policy file of the new desktop

3. You can now modify the desktop's policy as shown in Figure 119 on page 214. In this example, we are easing the policy's restrictions by unchecking the selection to remove the **Run** command from the **Start** menu. This means that any user with the assigned desktop EyeDesk will be able to select **Run** from the **Start** menu to execute programs.

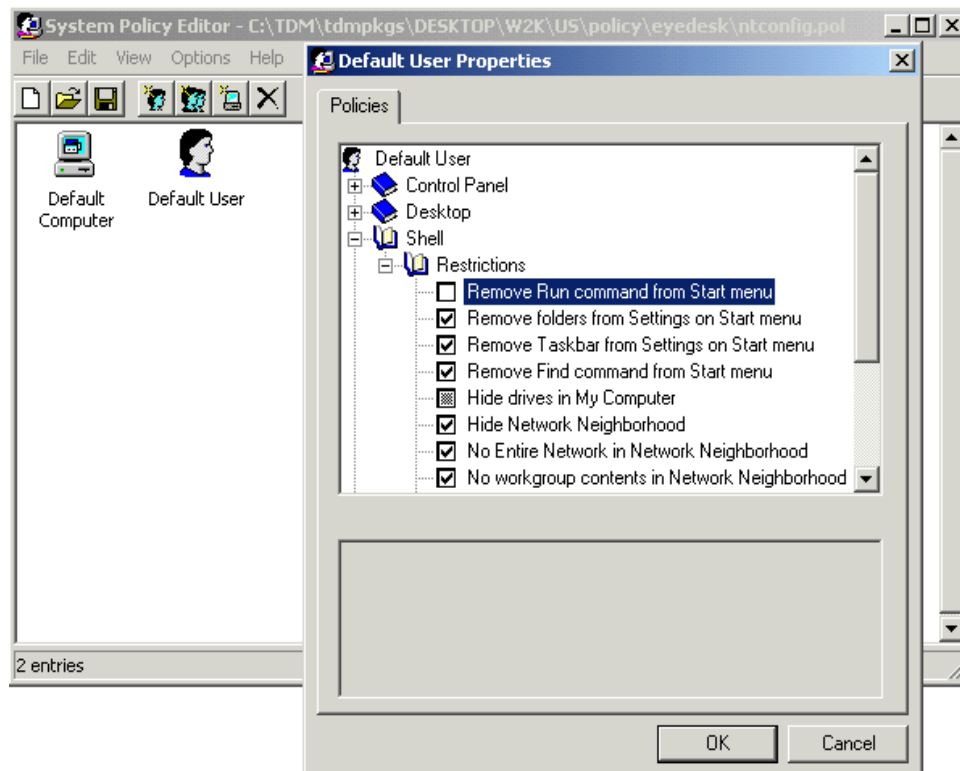


Figure 119. POLEDIT: Modify the policy of the new desktop to fit your needs

4. When you have finished modifying the policy, save it under the file name *ntconfig.pol* in its current directory, `\tdm\tdmpkgs\dekstop\w2k\us\policy\eyedesk`.

#### 5.3.3.4 Defining and assigning the desktop

Now that you have created all the files you need for the new desktop, you need to define it to IBM WorkSpace On-Demand 3.0.1. This is done with the *define* action on the desktop object. Execute the following command in the CLI:

```
desktop define name=EyeDesk os=w2k lang=us shell=explorer profile=EyeDesk
policy=EyeDesk
```

If you did not set a background bitmap and color using the desktop policy, you can specify these at the time that you define the desktop in the same way that we specified them upon assigning a desktop to a user in Section 5.3.2, “Assigning a Windows 2000 desktop to a user” on page 205. For example, the

following command would create our desktop with a different background color and bitmap:

```
desktop define name=EyeDesk os=w2k lang=us shell=explorer profile=EyeDesk  
policy=EyeDesk bitmap="C:\Winnt\wsoddtbg.BMP" BitmapDisplay="C"  
Background="210 210 210"
```

You should see a new data store entry for this desktop as shown in Figure 120.

```
[/DESKTOP/W2K/EXPLORER/US/EyeDesk]  
profile=EyeDesk  
policy=EyeDesk  
PackageLocation=\\ATLAS2\TDMPKGS\DESKTOP\W2K\US\EXPLORER  
PolicyLocation=\\ATLAS2\TDMPKGS\DESKTOP\W2K\US\POLICY  
ProfileLocation=\\ATLAS2\TDMPKGS\DESKTOP\W2K\US\PROFILES  
associated class=com.ibm.tdm.task.desktop.W2KExplorerDesktop
```

*Figure 120. Data store representation of a custom desktop for Windows 2000*

The newly-defined desktop, *EyeDesk*, is now available for other users. The following command assigns the new desktop to the user named user1:

```
user assign os=w2k lang=us shell=explorer desktop=EyeDesk user=user1
```

Figure 121 on page 216 shows our user logged on after having been assigned the customized desktop EyeDesk.

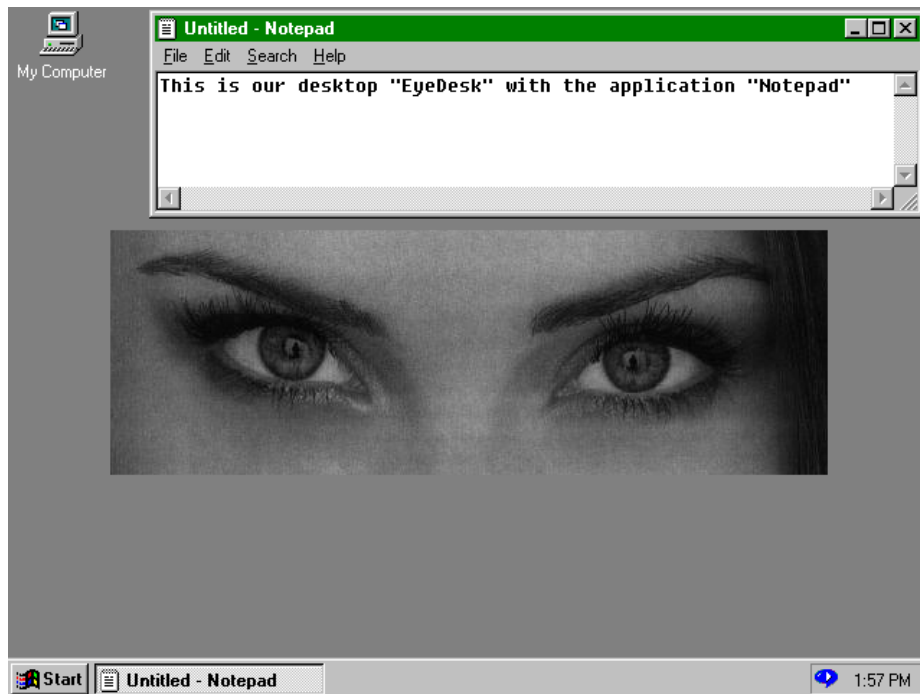


Figure 121. The newly-defined desktop EyeDesk

---

## 5.4 Windows NT Workstation 4.0 desktops

The following sections describe the structure and definition of desktops for the Windows NT Workstation client operating system and how to assign Windows NT desktops to users in IBM WorkSpace On-Demand 3.0.1.

### 5.4.1 Windows NT desktop structure

IBM WorkSpace On-Demand 3.0.1 maintains desktops for Windows NT clients in three parts. You can think of a desktop as comprising the shell, the profile, and the policy that apply to a given user. Figure 122 on page 217 shows the representation of Windows NT desktops in the TDMPKGS share. The default and sandbox desktops are kept as subdirectories within Explorer, policy and profiles directories in the desktop\nt4\us directory. When you define new desktops, they will also be represented as subdirectories in these locations.

Default policies and profiles are stored in the files policy\default\ntconfig.pol and profiles\default\ntuser.dat. These can be used as the basis for creating

custom desktops of your own. A sandbox policy is stored in the file `policy\sandbox\ntconfig.pol`. These are typically only used by sandbox users to gain unrestricted access to a client machine in order to configure an application and define the application package. The sandbox user has no profile.

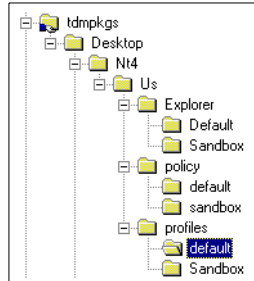


Figure 122. The default desktop directories for Windows NT clients

A Windows NT desktop also has a data store entry as shown in Figure 123. Only the default desktop is shown; the sandbox desktop entry is identical except that the profile and policy lines refer to sandbox instead of default. Notice the three location lines, `PackageLocation`, `PolicyLocation`, and `ProfileLocation`, all refer to the directories shown in Figure 122. Therefore, the data store definition of a desktop completely directs IBM WorkSpace On-Demand 3.0.1 where to find its shell, policy, and profile data. Any new desktops that you create will have similar entries in the data store.

```

[/DESKTOP]

[/DESKTOP/NT4]

[/DESKTOP/NT4/EXPLORER]

[/DESKTOP/NT4/EXPLORER/US]

[/DESKTOP/NT4/EXPLORER/US/Default]
PackageLocation=C:\TDM\tdmpkgs\DESKTOP\NT4\US\EXPLORER
PolicyLocation=C:\TDM\tdmpkgs\DESKTOP\NT4\US\POLICY
ProfileLocation=C:\TDM\tdmpkgs\DESKTOP\NT4\US\PROFILES
Policy=Default
Profile=Default
associated class=com.ibm.tdm.task.desktop.NT4ExplorerDesktop

```

Figure 123. The default Windows NT desktop in the data store

The profile component of the desktop is used primarily to customize the user interface. By modifying a profile, the administrator can set background bitmaps, color schemes, screen savers, and similar characteristics of the user's interface. Administrators can customize user profiles and assign them to users to enforce a corporate standard. User profiles are also locked down in IBM WorkSpace On-Demand 3.0.1 so that users cannot change them.

*System policies* are used mainly to restrict what users can do. For example, you can use system policies to restrict what users can do from their desktops, such as disabling certain **Control Panel** options or configuring network settings. The default desktop for Windows NT clients has its policies completely restricted so that users with this desktop have no access to the **Control Panel** or any other system settings and are thus unable to change anything in their user interface.

A policy consists of two parts: computer and user. The computer section of a policy contains settings for network, printing, and other system-related values while the user section contains settings for the **Control Panel**, shell, and other interface-related values. Windows NT Server provides a utility known as the System Policy Editor (\winnt\poedit.exe) that allows you to view and modify policy information. This can be used to customize the policy portion of the desktop when you build a customized desktop.

Policy information is maintained in a file called ntconfig.pol. This file can be found in each desktop directory under \tdm\tdmpkgs\desktop\nt4\us\policy. When a desktop is assigned to a user, the policy file is copied to the user's profiles directory, \tdm\tdmprfls\<username>\nt4\us\profiles. The policy file contains registry settings that are copied to the user or local machine portions of the registry on the client machine when the user logs on.

There is some overlap between policies and profiles in that the user section of the policy also allows you to modify settings, such as background bitmaps and color schemes. It is recommended that the System Policy Editor be used to make as many changes as possible for a new desktop in order to localize the changes as much as possible in one place. Customizing and modifying desktop policies and profiles are described in more detail in Section 5.4.3, "Defining a new desktop for Windows NT 4.0 Workstation" on page 220.

#### 5.4.2 Assigning a Windows NT desktop to a user

You can add a desktop to a user via the `assign` action on the user object. When you assign a desktop to a user, you must also specify the operating system, language, and shell. For example, the following command assigns the default Windows NT desktop to a user named user1:

```
user assign user=user1 desktop=default os=nt4 shell=explorer lang=us
```

Without creating a new desktop, you can modify some basic desktop values at the time that you assign a desktop to a user. The values you can set are:

- The desktop bitmap (wallpaper)
- The way the desktop bitmap is displayed (tiled or centered)
- The color of the desktop background

The console support for changing the desktop of a user is limited to these settings. If you want to change anything outside this scope, you need to define your own desktop. This is described in Section 5.4.3, “Defining a new desktop for Windows NT 4.0 Workstation” on page 220.

The following parameters are available for changing a user’s desktop:

Bitmap=“bitmap”

The fully qualified path of the desktop bitmap to use.  
For example: Bitmap=“C:\test.bmp”.

BitmapDisplay=“C | T”

Specifies how to display the bitmap, where:  
C = Centered.  
T = Tiled.

Background=“background”

RGB value specifying the desktop background color.  
Format: X X X.  
Where X is 0 - 255.  
For example: Background=“0 128 0”.

For example:

```
user assign user=user1 desktop=default os=nt4 lang=us shell=explorer  
bitmap="C:\Winnt\wsoddtbg.BMP" BitmapDisplay="C" Background="210 210 210"
```

This command assigns the bitmap “C:\Winnt\wsoddtbg.BMP” in a centered window with a grey background color.

When you assign a desktop to a user, the user’s data store entry is modified to reflect the desktop and settings. Figure 124 on page 220 shows the data store representation of our user, user1, after executing the above command. Notice that the bitmap and color information is included with the specification

of the default desktop. These values could, in fact, be set for any desktop for this user.

```
[/USER/user1]
USER=user1
Type=USER
RIDid=1187
PROFILEDIR=\\ATLAS2\\TDMPRFLS\\user1
associated class=com.ibm.tdm.task.principal.TDMUser

[/USER/user1//DESKTOP/NT4/EXPLORER/us/default]
bitmap=c:\\winnt\\wsoddtbg.bmp
bitmapdisplay=c
background=210 210 210
```

Figure 124. Data store representation of a user with a desktop

### 5.4.3 Defining a new desktop for Windows NT 4.0 Workstation

To define a new desktop, you can set up both a new policy and a new profile. It is best to copy an existing desktop and modify the attributes to suit your needs for the new desktop. For example, the steps outlined in this section show you how to copy and modify the default Windows NT client desktop.

#### 5.4.3.1 Modifying the profile

Modifying the profile is done on a client machine using a user ID that has administrator authority very similar to a sandbox. (See Chapter 6, “Applications” on page 259 for more information about sandbox users.) The profile changes will be saved back to the IBM WorkSpace On-Demand 3.0.1 server. Modifying the policy for a new desktop can be done directly on the server using the System Policy Editor poledit.exe.

To begin, the following steps take you through the process of modifying the profile for a new Windows NT desktop:

1. Create a new Windows NT 4.0 Workstation client in the IBM WorkSpace On-Demand 3.0.1 console.
2. Create a user on the IBM WorkSpace On-Demand 3.0.1 console. Our example will use the user name UserDesk.

- a. Create the user itself:

```
user define type=USER user=UserDesk
```

- b. Reset the password and prevent the change of the password:

```
nativeuser modify password="" passwordexpiration=NEVER user=UserDesk
```



- c. Assign the default desktop to the user so that we can change it later:
- ```
user assign desktop=default os=nt4 lang=us shell=explorer
user=UserDesk
```
- d. Make the user a member of the “Domain Admins” group. This turns our user into an administrator, so that we have permission to modify settings on the client machine.
- ```
groupmember define group="Domain Admins" user=UserDesk
```
3. Replace the policy file of the user with the sandbox policy. This will remove the restrictions that are in place by default on normal users. You must do this by entering the following command in a DOS command prompt on your server, not in the IBM WorkSpace On-Demand 3.0.1 CLI console:
- ```
copy C:\Tdm\tdmpkgs\Desktop\nt4\us\policy\sandbox\ntconfig.pol
C:\Tdm\tdmprfls\UserDesk\Nt4\us\Profiles
```
4. Logon to the client with the user ID UserDesk.
5. Make changes to the desktop as you see fit. Most of these changes can be applied by modifying the desktop properties in the **Control Panel**. Some common changes are as follows:
- a. To change the wallpaper, click the **Background** tab as shown in Figure 125 on page 222. The pattern you select is applied to either a wallpaper or a color. The wallpaper selection is on the right side of the box. You can also select to tile the wallpaper across the window. Available wallpaper bitmaps are contained in the C:\Winnt directory. In our example, we used the custom bitmap eyes.bmp.

#### Note

If you want to use your own bitmap that is not supplied with Windows NT or IBM WorkSpace On-Demand 3.0.1, you have to ensure that it is copied to the C:\Winnt directory of the client machine first. The best way to do this is to copy your bitmap to the directory nt4\sp3\us\inst\i386\soem\$c\winnt in the RPLFILES alias on your IBM WorkSpace On-Demand 3.0.1 deployment server before you define any client machines. If this directory does not already exist, you can create it. Then, when you define a client, your bitmap will be copied to C:\Winnt on the client hard drive along with all the built-in bitmaps.

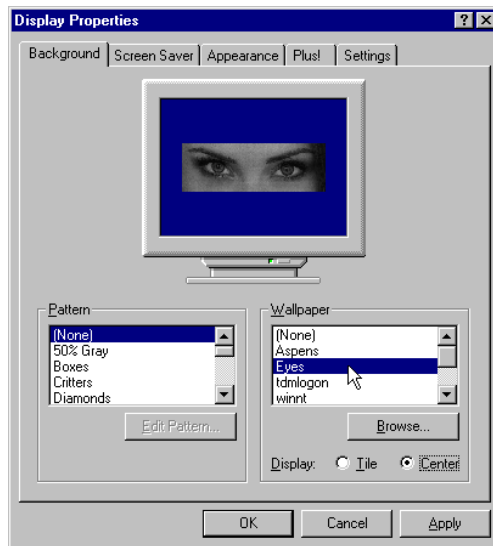


Figure 125. Modify the background wallpaper properties

- b. To change the color scheme, click the **Appearance** tab in the **Display Properties** dialog as shown in Figure 126. You can change individual colors of desktop objects or set an entire scheme. Before you change the color scheme, be sure to consult the user community.

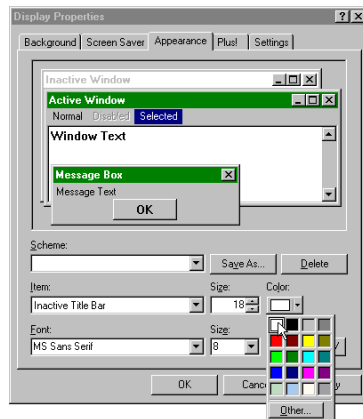


Figure 126. Modify the color scheme

- c. To change the screen saver, click the **Screen Saver** tab as shown in Figure 127 on page 223.

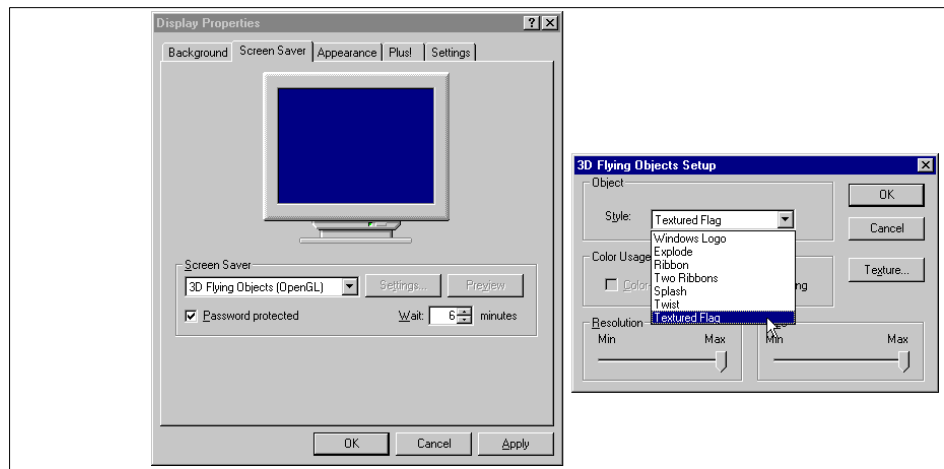


Figure 127. Modify the screen saver properties

6. Click **OK** from **Display Properties** to apply the changes to your desktop.
7. To change the **Start** menu, you will need to change the contents of two directories on your client. These are:
  - C:\Winnt\Profiles\UserDesk\Start Menu
  - C:\Winnt\Profiles\All Users\Start Menu

The user UserDesk to which we are logged on does not have direct access to a Windows NT command prompt or Windows Explorer from the desktop. You can, however, select **Run** from the **Start** menu and enter explorer to launch Explorer and access the hard drive. You can also access the **Start** menu directories by right-clicking the mouse on the **Start** icon and selecting either **Open** or **Open All Users**. You can now add, remove, or modify any options on the **Start** menu.

8. When you have finished your modifications, close all windows, empty the Recycle Bin, and remove any documents from the **Taskbar**.
9. Logoff from the client. The changes applied to the desktop are saved in the file *ntuser.dat* as well as other subdirectories in our user's profile directory C:\Tdm\tdmprfls\UserDesk\Nt4\us\Profiles\Desktop as shown in Figure 128 on page 224.

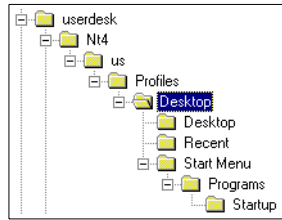


Figure 128. The modified Windows NT profile information

#### 5.4.3.2 Creating the desktop directory structure

Having created the profile for our new desktop, we need to create the directory structure to hold the desktop information. In our example, we will name our new desktop EyeDesk. Therefore, we need to create the following three new subdirectories:

- \tdm\tdmpkgs\desktop\nt4\us\explorer\eyedesk
- \tdm\tdmpkgs\desktop\nt4\us\policy\eyedesk
- \tdm\tdmpkgs\desktop\nt4\us\profiles\eyedesk

We also need to copy our new profile to the profiles\eyedesk subdirectory. Copy all the files and subdirectories in the userdesk\nt4\us\profiles\desktop directory (shown in Figure 128) to the directory c:\tdm\tdmpkgs\desktop\nt4\us\profiles\eyedesk. This will ensure that all the changes you made while logged on as UserDesk will be applied to the new desktop.

#### 5.4.3.3 Modifying the desktop policy

Now that we have created a profile for our new desktop, we may also want a new policy. To create a new policy, it is easiest to copy and modify an existing policy. There are two default policy files with which we can start. Both are called ntconfig.pol, but one is a restricted policy intended for normal users, while the other is an unrestricted policy intended for sandbox users. We will start with the restricted policy and make modifications as required:

1. Copy the default restricted policy to our new desktop directory, using the following command at a Windows command prompt:

```
copy c:\tdm\tdmpkgs\desktop\nt4\us\policy\default\ntconfig.pol
c:\tdm\tdmpkgs\desktop\nt4\us\policy\eyedesk
```

2. Modify the policy file for the EyeDesk desktop using the System Policy Editor. The System Policy Editor is a graphical tool provided with Windows NT Server 4.0 that allows you to create or modify the registry entries contained in the policy file. On your IBM WorkSpace On-Demand 3.0.1

server, run `c:\winnt\poledit.exe`. From **File**, select **Open Policy** and browse to locate the policy for EyeDesk as shown in Figure 129.

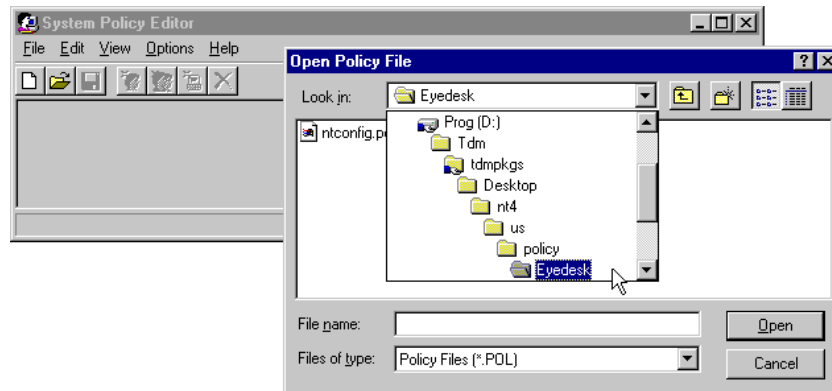


Figure 129. POLEDIT: Open the policy file of the new desktop

3. You can now modify the desktop's policy as shown in Figure 130. In this example, we are easing the policy's restrictions by unchecking the selection to remove the **Run** command from the **Start** menu. This means that any user with the assigned desktop EyeDesk will be able to select **Run** from the **Start** menu to execute programs.

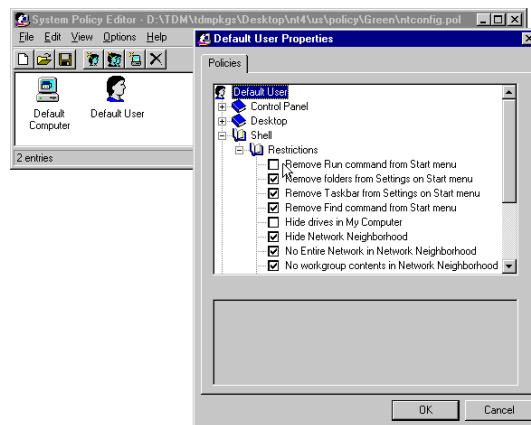


Figure 130. POLEDIT: Modify the policy of the new desktop to fit your needs

4. When you have finished modifying the policy, save it under the file name `ntconfig.pol` in its current directory, `\tdm\tdmpkgs\dekstop\nt4\us\policy\eyedesk`.

#### 5.4.3.4 Defining and assigning the desktop

Now that you have created all the files you need for the new desktop, you need to define it to IBM WorkSpace On-Demand 3.0.1. This is done with the `define` action on the desktop object. Execute the following command in the CLI:

```
desktop define name=EyeDesk os=nt4 lang=us shell=explorer profile=EyeDesk  
policy=EyeDesk
```

If you did not set a background bitmap and color using the desktop policy, you can specify these at the time that you define the desktop in the same way that we specified them upon assigning a desktop to a user in Section 5.4.2, “Assigning a Windows NT desktop to a user” on page 218. For example, the following command would create our desktop with a different background color and bitmap:

```
desktop define name=EyeDesk os=nt4 lang=us shell=explorer profile=EyeDesk  
policy=EyeDesk bitmap="C:\Winnt\wsoddtbg.BMP" BitmapDisplay="C"  
Background="210 210 210"
```

You should see a new data store entry for this desktop as shown in Figure 131.

```
[/DESKTOP/NT4/EXPLORER/US/eyedesk]  
profile=eyedesk  
policy=eyedesk  
PackageLocation=\\ATLAS2\TDMPKGS\DESKTOP\NT4\US\EXPLORER  
PolicyLocation=\\ATLAS2\TDMPKGS\DESKTOP\NT4\US\POLICY  
ProfileLocation=\\ATLAS2\TDMPKGS\DESKTOP\NT4\US\PROFILES  
associated class=com.ibm.tdm.task.desktop.NT4ExplorerDesktop
```

Figure 131. Data store representation of a custom desktop for Windows NT

The newly-defined desktop, *EyeDesk*, is now available for other users. The following command, for example, assigns the new desktop to the user named `user1`:

```
user assign os=nt4 lang=us shell=explorer desktop=EyeDesk user=user1
```

Figure 132 on page 227 shows our user logged on after having been assigned the customized desktop *EyeDesk*.

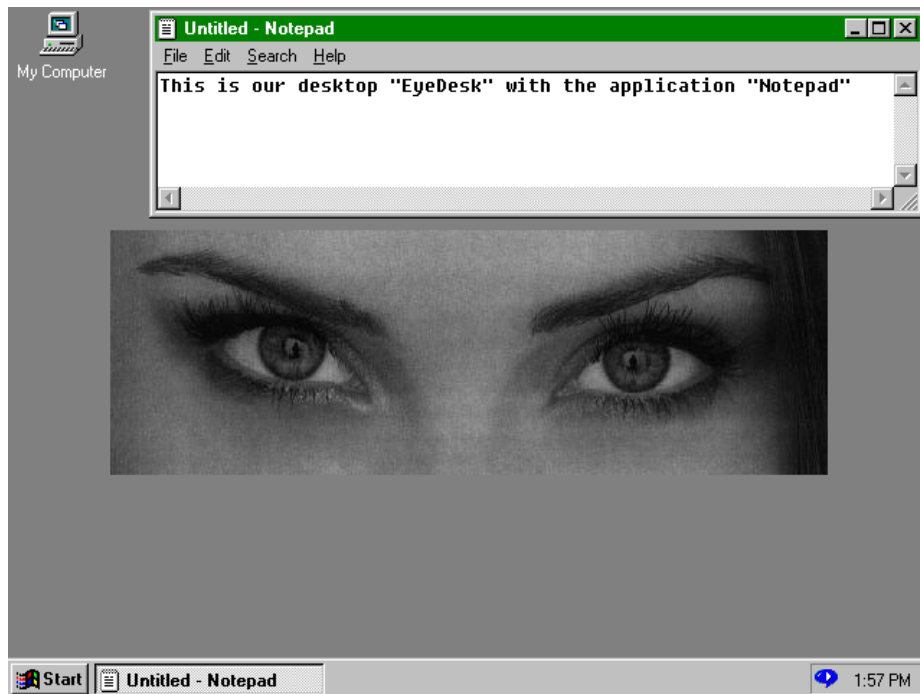


Figure 132. The newly-defined desktop EyeDesk

---

## 5.5 Windows 98 desktops

The following sections describe the structure and definition of desktops for the Windows 98 client operating system and how to assign Windows 98 desktops to users in IBM WorkSpace On-Demand 3.0.1.

### 5.5.1 Windows 98 desktop structure

IBM WorkSpace On-Demand 3.0.1 maintains desktops for Windows 98 clients in three parts. You can think of a desktop as comprising the shell, the profile, and the policy that apply to a given user. Figure 133 on page 228 shows the representation of Windows 98 desktops in the TDMPKGS share. The default and sandbox desktops are kept as subdirectories within the Explorer, policy and profiles directories in the desktop\w98\us directory. When you define new desktops, they will also be represented as subdirectories in these locations.

Default policies and profiles are stored in the files policy\default\config.pol and profiles\default\user.dat. These can be used as the basis for creating

custom desktops of your own. The sandbox policy is stored in the file `policy\sandbox\config.pol`. This is typically only used by sandbox users to gain unrestricted access to a client machine in order to configure an application and define the application package. A sandbox user has no profile.

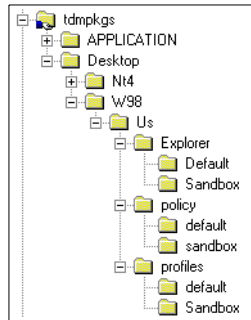


Figure 133. The default desktop directories for Windows 98 clients

A Windows 98 desktop also has a data store entry as shown in Figure 134. Only the default desktop is shown; the sandbox desktop entry is identical except that the profile and policy lines refer to sandbox instead of default. Notice the three location lines, `PackageLocation`, `PolicyLocation`, and `ProfileLocation`, all refer to the directories shown in Figure 133. Therefore, the data store definition of a desktop directs IBM WorkSpace On-Demand 3.0.1 where to find its shell, policy, and profile data. Any new desktops that you create will have similar entries in the data store.

```

[/DESKTOP/W98]

[/DESKTOP/W98/EXPLORER]

[/DESKTOP/W98/EXPLORER/US]

[/DESKTOP/W98/EXPLORER/US/Default]
PackageLocation=C:\TDM\tdmpkgs\DESKTOP\W98\US\EXPLORER
PolicyLocation=C:\TDM\tdmpkgs\DESKTOP\W98\US\POLICY
ProfileLocation=C:\TDM\tdmpkgs\DESKTOP\W98\US\PROFILES
Policy=Default
Profile=Default
associated class=com.ibm.tdm.task.desktop.W98ExplorerDesktop

```

Figure 134. The default Windows 98 desktop in the data store



The profile component of the desktop is used primarily to customize the user interface. By modifying a profile, the administrator can set background bitmaps, color schemes, screen savers, and similar characteristics of the user's interface. Administrators can customize user profiles and assign them to users to enforce a corporate standard. User profiles are also locked down in IBM WorkSpace On-Demand 3.0.1 so that users cannot change them.

*System policies* are used mainly to restrict what users can do. For example, you can use system policies to restrict what users can do from their desktops, such as disabling certain **Control Panel** options or configuring network settings. The default desktop for Windows 98 clients has its policies completely restricted so that users with this desktop have no access to the **Control Panel** or any other system settings and thus are unable to change anything in their user interface.

A policy consists of two parts: computer and user. The computer section of a policy contains settings for the network, printing and other system-related values, while the user section contains settings for the **Control Panel**, shell, and other interface-related values. Windows 98 provides a utility known as the System Policy Editor (poledit.exe) that allows you to view and modify policy information. This can be used to customize the policy portion of the desktop when you build a customized desktop.

Since the IBM WorkSpace On-Demand 3.0.1 server is Windows NT, it is important not to confuse the Windows NT System Policy Editor with that for Windows 98. The Windows 98 System Policy Editor can be found on the Windows 98 CD in the WIN98\_SE\SETUP\TOOLS\RESKIT\NETADMIN\POLEDIT directory.

Policy information is maintained in a file called config.pol. This file can be found in each desktop directory under \tdm\tdmpkgs\desktop\w98\us\policy. When a desktop is assigned to a user, the policy file is copied to the user's profiles directory, \tdm\tdmprfls\<username>\w98\us\profiles. The policy file contains registry settings that are copied to the user or local machine portions of the registry on the client machine when the user logs on.

There is some overlap between policies and profiles in that the user section of the policy also allows you to modify settings, such as background bitmaps and color schemes. It is recommended that the System Policy Editor be used to make as many changes as possible for a new desktop in order to localize the changes as much as possible in one place. Customizing and modifying desktop policies and profiles are described in more detail in Section 5.5.3, "Defining a new desktop for Windows 98" on page 231.

### 5.5.2 Assigning Windows 98 desktops to users

You can add a desktop to a user via the `assign` action on the user object. When you assign a desktop to a user, you must also specify the operating system, language, and shell. For example, the following command assigns the default Windows 98 desktop to a user named `user2`:

```
user assign user=user2 desktop=default os=w98 shell=explorer lang=us
```

Without creating a new desktop, you can modify some basic desktop values when you assign a desktop to a user. The values you can set are:

- The desktop bitmap (wallpaper)
- The way the desktop bitmap is displayed (tiled or centered)
- The color of the desktop background

The console support for changing the desktop of a user is limited to these settings. If you want to change anything outside this scope, you need to define your own desktop, which is described in Section 5.5.3, “Defining a new desktop for Windows 98” on page 231.

The following parameters are available for changing a user’s desktop:

Bitmap=“bitmap”

The fully-qualified path of the desktop bitmap to use.  
For example: Bitmap=“C:\test.bmp”.

BitmapDisplay=“C | T”

Specifies how to display the bitmap, where:  
C = Centered.  
T = Tiled.

Background=“background”

RGB value specifying the desktop background color.  
Format: X X X.  
Where X is 0 - 255.  
For example: Background=“0 128 0”.

For example:

```
user assign user=user2 desktop=default os=w98 lang=us shell=explorer  
bitmap="C:\Windows\wsoddtbg.BMP" BitmapDiskplay="C" Background="210 210  
210"
```

This command assigns the bitmap “C:\Windows\wsoddtbg.BMP” in a centered display with a grey background color.

When you assign a desktop to a user, the user’s data store entry is modified to reflect the desktop and settings. Figure 124 on page 220 shows the data store representation of our user, user2, after executing the above command. Notice that the bitmap and color information is included with the specification of the default desktop. These values could, in fact, be set for any desktop for this user.

```
[/USER/user2]
USER=user2
Type=USER
RIDid=1266
PROFILEDIR=\\ATLAS2\TDMPRFLS\user2
associated class=com.ibm.tdm.task.principal.TDMUser

[/USER/user2//DESKTOP/W98/EXPLORER/us/default]
bitmap=c:\windows\wsoddtbg.bmp
bitmapdisplay=c
background=0 128 0

[/USER/user2//DESKTOP/NT4/EXPLORER/us/default]
bitmap=c:\winnt\wsoddtbg.bmp
bitmapdisplay=c
background=0 128 0
```

*Figure 135. Data store representation of a user with a desktop*

Notice also that this user’s data store representation contains a second desktop, defined for Windows NT. You can use the `user assign` command to add one desktop for each client operating system to the same user. Each desktop is listed after the user definition in the data store as shown in Figure 135. This permits our user, user2, to log on to both Windows 98 and Windows NT client machines. You cannot, however, assign more than one desktop of the same operating system type to the same user.

### 5.5.3 Defining a new desktop for Windows 98

To define a new desktop, you can set up both a new policy and a new profile. It is best to copy an existing desktop and modify the attributes to suit your needs for the new desktop. For example, the steps outlined in this section show you how to copy and modify the default Windows 98 client desktop.

### 5.5.3.1 Modifying the profile

Modifying the profile is done on a client machine using a user ID that has administrator authority very similar to a sandbox. (See Chapter 6, “Applications” on page 259 for more information about sandbox users.) The profile changes will be saved back to the IBM WorkSpace On-Demand 3.0.1 server. Modifying the policy for a new desktop can be done directly on the server using the System Policy Editor poledit.exe.

To begin, the following steps take you through the process of modifying the profile for a new Windows 98 desktop:

1. Create a new Windows 98 client in the IBM WorkSpace On-Demand 3.0.1 console.
2. Create a user on the IBM WorkSpace On-Demand 3.0.1 console. Our example uses the user name UserDesk.
  - a. Create the user itself:  
`user define type=USER user=UserDesk`
  - b. Reset the password and prevent the change of the password:  
`nativeuser modify password="" passwordexpiration=NEVER user=UserDesk`
  - c. Assign the default desktop to the user so that we can change it later:  
`user assign desktop=default os=w98 lang=us shell=explorer  
user=UserDesk`
  - d. Make the user a member of the “Domain Admins” group. This turns our user into an administrator so that we have permission to modify settings on the client machine.  
`groupmember define group="Domain Admins" user=UserDesk`
3. Replace the policy file of the user with the sandbox policy. This will remove the restrictions that are in place by default on normal users. You must do this by entering the following command in a DOS command prompt on your server, not in the IBM WorkSpace On-Demand 3.0.1 CLI console:  
`copy C:\Tdm\tdmpkgs\Desktop\w98\us\policy\sandbox\config.pol  
C:\Tdm\tdmprfls\UserDesk\w98\us\Profiles`
4. Logon to the client with the user ID UserDesk.
5. Make changes to the desktop as you see fit. Most of these changes can be applied by modifying the desktop properties in the **Control Panel**. Some common changes are as follows:
  - a. To change the wallpaper, click the **Background** tab as shown in Figure 136 on page 233. The pattern you select is applied to either a wallpaper or a color. The wallpaper selection is on the right side of the box. You can also select to tile the wallpaper across the window.

Available wallpaper bitmaps are contained in the C:\Windows directory. In our example, we used the bitmap image called snowflake.

**Note**

If you want to use your own bitmap that is not supplied with Windows 98 or IBM WorkSpace On-Demand 3.0.1, you have to ensure that it is copied to the C:\Winnt directory of the client machine first. The best way to do this is to copy your bitmap to the w98\w98se\us\inst\c\windows directory in the RPLFILES alias on your IBM WorkSpace On-Demand 3.0.1 deployment server before you define any client machines. If this directory does not already exist, you can create it. Then, when you define a client, your bitmap will be copied to C:\Windows on the client hard drive along with all the built-in bitmaps.

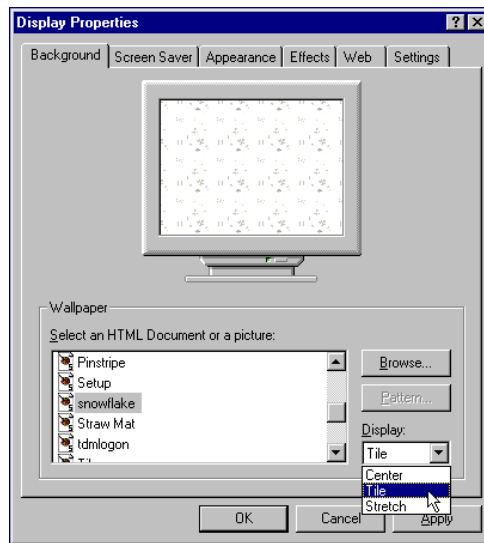


Figure 136. Windows 98: Modify the background wallpaper properties

- b. To change the color scheme, click the **Appearance** tab in the **Display Properties** dialog as shown in Figure 137 on page 234. You can change individual colors of desktop objects or set an entire scheme. Before you change the color scheme, be sure to consult the user community.

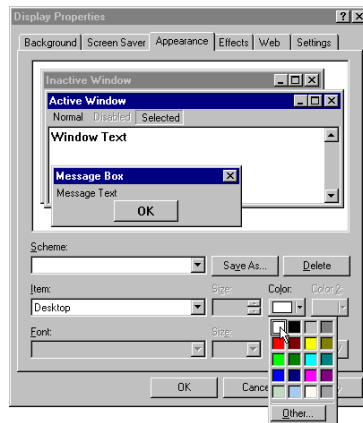


Figure 137. Windows 98: Modify the color scheme

- c. To change the screen saver, click the **Screen Saver** tab as shown in Figure 138.

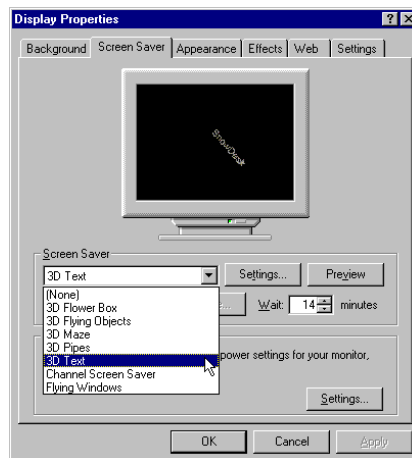


Figure 138. Windows 98: Modify the screen saver properties

6. Click **OK** in **Display Properties** to apply the changes to your desktop.
7. To change the **Start** menu, you will need to change the contents of two directories on your client. These are:
  - C:\Windows\Start Menu
  - C:\Windows\All Users\Start Menu

The user UserDesk to which we are logged on does not have direct access to a Windows 98 command prompt or Windows Explorer from the desktop. You can, however, select **Run** from the **Start** menu and enter Explorer to launch Explorer and access the hard drive. You can also access the **Start** menu directories by right-clicking the mouse on the **Start** icon and clicking either **Open** or **Open All Users**. You can now add, remove, or modify any options on the **Start** menu.

8. When you have finished your modifications, close all windows, empty the Recycle Bin, and remove any documents from the **Taskbar**.
9. Log off from the client. The changes applied to the desktop are saved in the *user.dat* file as well as in several subdirectories in our user's profile directory C:\Tdm\tdmprfls\UserDesk\W98\us\Profiles\Desktop as shown in Figure 139.

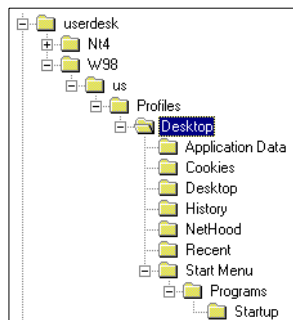


Figure 139. The modified Windows 98 profile information

### 5.5.3.2 Creating the desktop directory structure

Having created the profile for our new desktop, we need to create the directory structure to hold the desktop information. In our example, we name our new desktop SnowDesk. Therefore, we need to create the following three new subdirectories:

- \tdm\tdmpkgs\desktop\w98\us\explorer\snowdesk
- \tdm\tdmpkgs\desktop\w98\us\policy\snowdesk
- \tdm\tdmpkgs\desktop\w98\us\profiles\snowdesk

We also need to copy our new profile to the profiles\snowdesk subdirectory. Copy all the files and subdirectories in the userdesk\w98\us\profiles\desktop directory (shown in Figure 139) to the c:\tdm\tdmpkgs\desktop\w98\us\profiles\snowdesk directory. This will ensure that all the changes you made while logged on as UserDesk will be applied to the new desktop.

### 5.5.3.3 Modifying the desktop policy

Now that we have created a profile for our new desktop, we may also want a new policy. To create a new policy, it is easiest to copy and modify an existing policy. There are two default policy files with which we can start. Both are called config.pol, but one is a restricted policy intended for normal users, while the other is an unrestricted policy intended for sandbox users. We start with the restricted policy and make modifications as required:

1. Copy the default restricted policy to the new desktop directory using the following command at a Windows command prompt:

```
copy c:\tdm\tdmpkgs\desktop\w98\us\policy\default\config.pol  
c:\tdm\tdmpkgs\desktop\w98\us\policy\snowdesk
```

2. Modify the policy file for the SnowDesk desktop using the System Policy Editor. The System Policy Editor is a graphical tool provided with Windows 98 that allows you to create or modify the registry entries contained in the policy file.

#### Note

You are running on a Windows NT Server, so do not confuse the Windows NT System Policy Editor with the Windows 98 System Policy Editor. Both are named poledit.exe, but to modify a Windows 98 policy, you must run the Windows 98 System Policy Editor on a Windows 98 machine. The Windows 98 System Policy Editor can be found on the Windows 98 CD in the \win98\_se\setup\tools\reskit\netadmin\poledit directory. When you have modified your policy, you can copy it back to your desktop location, such as  
c:\tdm\tdmpkgs\desktop\w98\us\policy\snowdesk.

From **File** on the System Policy Editor, select **Open Policy** and browse to locate the policy for SnowDesk as shown in Figure 140 on page 237.



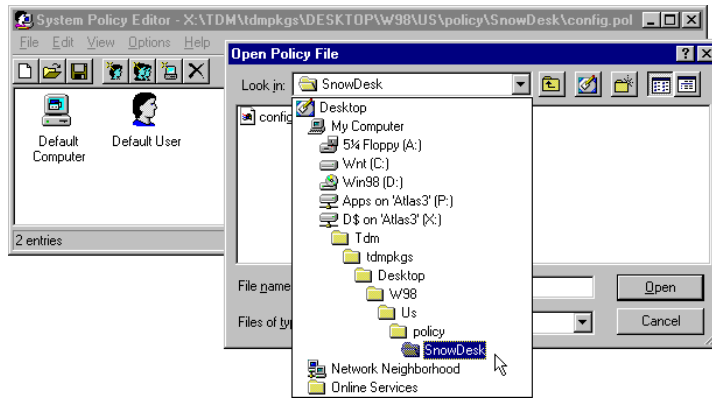


Figure 140. Windows 98 Policy Editor: Open the policy file of the new desktop

3. You can now modify the desktop's policy as shown in Figure 141. In this example, we are easing the policy's restrictions by unchecking the selection to remove the **Run** command from the **Start** menu. This means that any user with the assigned desktop SnowDesk will be able to select **Run** from the **Start** menu to execute programs.

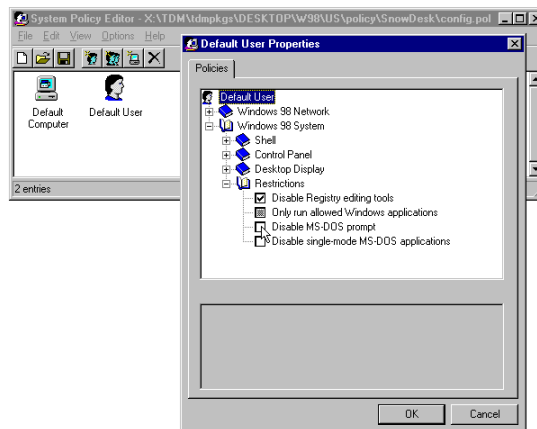


Figure 141. Windows 98: Modify the policy of the new desktop to your needs

4. When you have finished modifying the policy, save it and copy it from your Windows 98 machine to your desktop's location on the IBM WorkSpace On-Demand 3.0.1 server, for example, to the `c:\tdm\tdmpkgs\desktop\w98\us\policy\snowdesk` directory.

#### 5.5.3.4 Defining and assigning the desktop

Now that you have created all the files you need for the new desktop, you need to define it to IBM WorkSpace On-Demand 3.0.1. This is done with the `define` action on the desktop object. Execute the following command in the CLI:

```
desktop define name=SnowDesk os=w98 lang=us shell=explorer profile=SnowDesk  
policy=SnowDesk
```

If you did not set a background bitmap and color using the desktop policy, you can specify these when you define the desktop in the same way that we specified them upon assigning a desktop to a user in Section 5.5.2, “Assigning Windows 98 desktops to users” on page 230. For example, the following command would create our desktop with a different background color and bitmap:

```
desktop define name=SnowDesk os=w98 lang=us shell=explorer profile=SnowDesk  
policy=SnowDesk bitmap="C:\Winnt\wsoddtbg.BMP" BitmapDisplay="C"  
Background="210 210 210"
```

You should see a new data store entry for this desktop as shown in Figure 142.

```
[/DESKTOP/W98/EXPLORER/US/snowdesk]  
profile=snowdesk  
policy=snowdesk  
PackageLocation=\\ATLAS2\TDMPKGS\DESKTOP\W98\US\EXPLORER  
PolicyLocation=\\ATLAS2\TDMPKGS\DESKTOP\W98\US\POLICY  
ProfileLocation=\\ATLAS2\TDMPKGS\DESKTOP\W98\US\PROFILES  
associated class=com.ibm.tdm.task.desktop.W98ExplorerDesktop
```

Figure 142. Data store representation of a custom desktop for Windows 98

The newly-defined desktop, *SnowDesk*, is now available for other users. The following command, for example, assigns the new desktop to the user named `user2`:

```
user assign os=w98 lang=us shell=explorer desktop=SnowDesk user=user2
```

Figure 143 on page 239 shows our user logged on with the customized desktop.

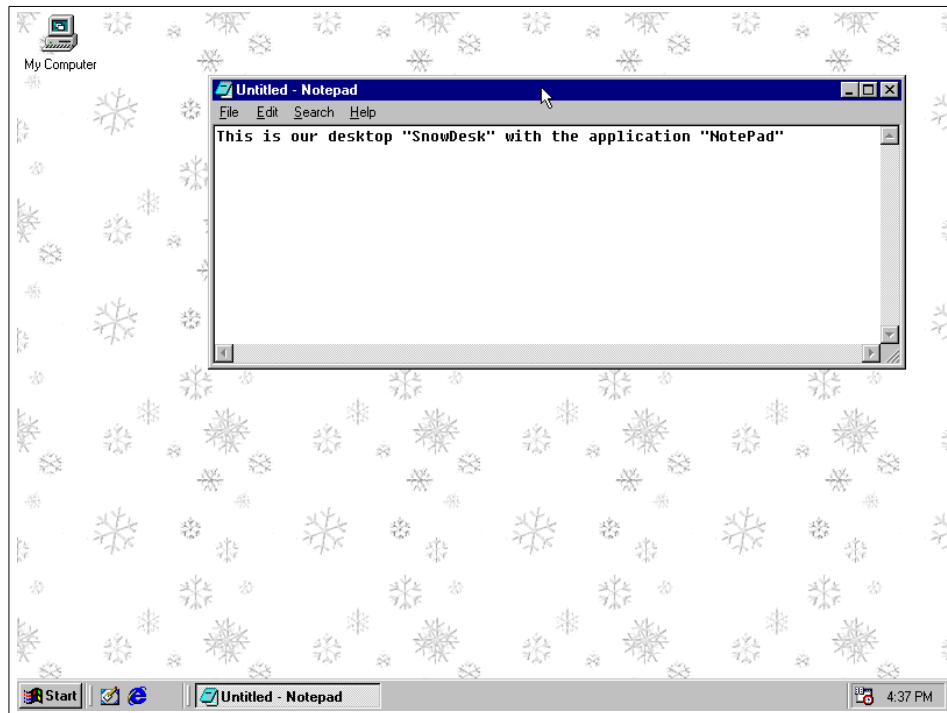


Figure 143. The newly-defined Windows 98 desktop *SnowDesk*

---

## 5.6 Defining Windows 2000, NT and 98 desktops in the GUI

You can also define your desktop using the IBM WorkSpace On-Demand 3.0.1 administration GUI. When you launch the GUI and log on to your server, select **Define a desktop** from the **Tasks** menu. You will be presented with a drop-down list allowing you to select the operating system for your desktop, followed by a dialog to define the desktop. This dialog is the same for Windows 2000, Windows NT and Windows 98. In the first window of this dialog, you can specify the desktop identification as shown in Figure 144 on page 240. Just as with the `desktop define` command, you can set the desktop name and select the language and shell for the desktop.

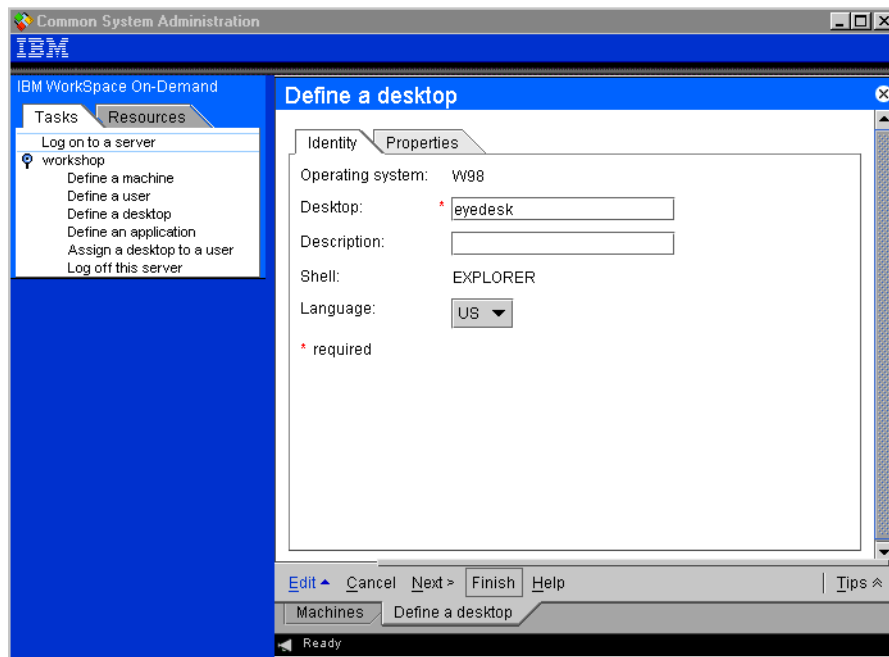


Figure 144. Defining a Windows desktop - Identity tab

When you click the **Properties** tab, you will see a dialog that allows you to set the profile, policy, and background information for your desktop. Figure 145 on page 241 shows the **Properties** dialog where we filled in the profile and policy information for our new desktop, EyeDesk.

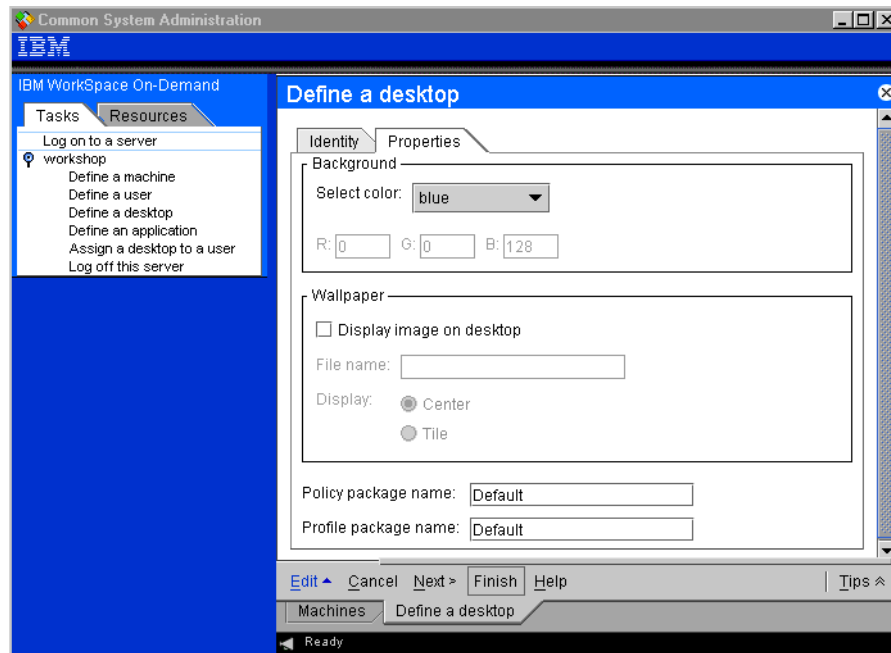


Figure 145. Defining a Windows desktop - Properties tab

When you have defined the desktop in the administration GUI, you still need to assign it to a user. From the **Tasks** menu, select **Assign a desktop to a user**. You will be presented with a list of users defined in IBM WorkSpace On-Demand 3.0.1. Select the user to whom you will assign the desktop and press **Next**. You will see a dialog with tabs for each operating system. When you select **W2K**, **NT 4.0** or **Win 98**, you will be able to select the desktop and applications you want to assign, as shown in Figure 146 on page 242. If you press the **Customize** button, you will be presented with a dialog like the one shown in Figure 145, allowing you to override the background color and bitmap, or override the policy and profile for your desktop.

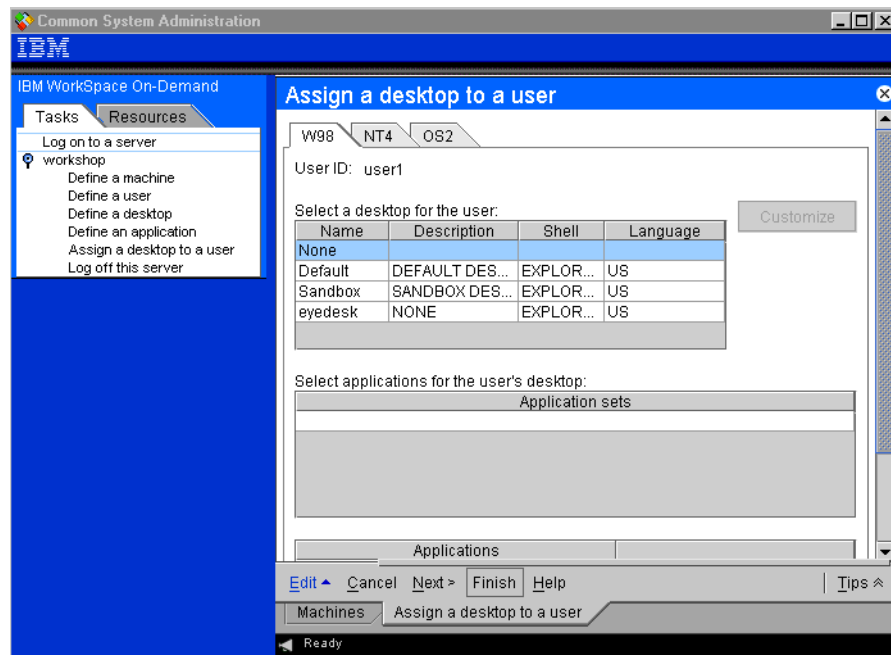


Figure 146. Assigning a Windows desktop to a user

## 5.7 Modifying the logon window for Windows 2000, NT and 98 clients

For Windows 2000, Windows NT and Windows 98 clients, you can also modify the logon window that is presented by IBM WorkSpace On-Demand 3.0.1. You can modify the bitmap that is displayed above the user, password, and domain prompts. To make your own bitmap the logon bitmap, place it in the following locations and name it TDMLOGON.BMP:

- Windows 2000:  
`\TDM\MM\CLIENT\RO\W2k\W2KIMG\US\INST\I386\%OEM%\C\WINNT\`
- Windows NT:  
`\TDM\MM\CLIENT\RO\NT4\SP3\US\INST\I386\%OEM%\C\WINNT\`
- Windows 98:  
`\TDM\MM\CLIENT\RO\W98\US\INST\C\WINDOWS\`

Do this before you define any machines on which you want your new bitmap displayed. When you define a Windows 2000 or Windows NT client, your `tdmlogon.bmp` will be copied to the `C:\Winnt` directory. For a Windows 98

client, it will be copied to C:\Windows. The bitmap will be displayed in the logon window when the client boots.

Alternatively, you could modify the registry with an administrator user after the first logon to the client. You could do this by using the registry editor REGEDIT or by applying the registry file, shown in Figure 148 and Figure 149, to the registry with the `REGEDIT /s <filename>` command.

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IBMNeTNT\Parameters]
"LogonBitmap"="MYBITMAP.BMP"
"LogonColorBBGRR"=dword:00808080
```

*Figure 147. Windows 2000: Sample registry file to change logon bitmap and color*

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IBMNeTNT\Parameters]
"LogonBitmap"="MYBITMAP.BMP"
"LogonColorBBGRR"=dword:00808080
```

*Figure 148. Windows NT: Sample registry file to change logon bitmap and color*

```
REGEDIT4

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\IBMNET32\NetworkProvider]
"LogonColorBBGRR"=dword:00800000
"LogonBitmap"="TLMLOGON.BMP"
```

*Figure 149. Windows 98: Sample registry file to change logon bitmap and color*

Using this registry approach only modifies the logon bitmap for a single client and requires you to log on to that client as an administrator.

---

## 5.8 OS/2 desktops

The following sections describe the desktop object for OS/2 clients and how to create and modify OS/2 desktops.

### 5.8.1 The OS/2 desktop object

OS/2 desktops differ from Windows 200, Windows NT and Windows 98 desktops in that they do not include policy and profile information from the TDMPKGS share. The concepts of policies and profiles as described in Section 5.4, “Windows NT Workstation 4.0 desktops” on page 216 and Section 5.5, “Windows 98 desktops” on page 227 apply only to the Win32 environment. Therefore, you will find that there are no directories for OS/2 desktops in the TDMPKGS share as there are for the Windows client operating systems.

Instead, the data store entry for an OS/2 desktop contains a great deal of the information that would be contained in the profile for a Windows desktop. Figure 150 shows the data store entry for the default desktop, named default, that is created when you install OS/2 client support in IBM WorkSpace On-Demand 3.0.1. Although there are references to package, policy, and profile in this desktop entry, they are not used, because the directories and files do not exist.

```
[/DESKTOP/OS2]

[/DESKTOP/OS2/PMSHELL]

[/DESKTOP/OS2/PMSHELL/US]

[/DESKTOP/OS2/PMSHELL/US/Default]
PackageLocation=C:\TDM\tdmpkgs\DESKTOP\OS2\US\PMSHELL
PolicyLocation=C:\TDM\tdmpkgs\DESKTOP\OS2\US\POLICY
ProfileLocation=C:\TDM\tdmpkgs\DESKTOP\OS2\US\PROFILES
Policy=Default
Profile=Default
Bitmap=\os2\bitmap\tdm.bmp
BitmapDisplay=C
Background=0 0 128
IconText=255 255 255
IconFont=WarpSans
IconFontSize=9
IconBackground=0 0 128
associated class=com.ibm.tdm.task.desktop.PMShellDesktop
```

Figure 150. Data store representation of the default OS/2 desktop

Notice, however, that the data store representation of an OS/2 desktop contains more information about how the desktop is configured. Besides the background bitmap and color information, the desktop also describes the font



and color of the text for icons on the desktop. The values for `IconText` and `IconBackground` set the foreground and background colors for the text that appears under each icon on the desktop, while `IconFont` specifies the font for the same text.

### 5.8.2 Defining an OS/2 desktop in the CLI

When you define a new OS/2 desktop, you can specify each of the described values as parameters with the `machine define` command. The following parameters are valid when you define an OS/2 desktop:

|                       |   |
|-----------------------|---|
| <b>Name</b>           | The name of the new desktop.  |
| <b>OS</b>             | The operating system; in this case, OS2.  |
| <b>Shell</b>          | The user interface for the desktop, typically PMSHELL.  |
| <b>Lang</b>           | The language for the desktop; for example: US.  |
| <b>Background</b>     | The color of the desktop background, expressed as "RGB," where each of R, G, and B is a number from 0 to 255.   |
| <b>Iconfont</b>       | The font name for the desktop icons.  |
| <b>Iconfontsize</b>   | The point size of the desktop icon font.  |
| <b>Icontext</b>       | The foreground color of the icon text, expressed as "RGB," where each of R, G, and B is a number from 0 to 255.   |
| <b>Iconbackground</b> | The background color of the icon text, expressed as "RGB," where each of R, G, and B is a number from 0 to 255.   |
| <b>Bitmap</b>         | The path and file name of the background bitmap; for example, <code>\os2\bitmap\wsoddtbg.bmp</code> . You must specify the fully-qualified path without a drive letter; the OS/2 client will interpret this relative to its boot drive. |
| <b>Bitmapdisplay</b>  | Either C, T, or "S,n." C indicates centered, T indicates tiled, and "S,n" indicates that the bitmap should be scaled where n is an integer scaling factor.  |
| <b>Description</b>    | An optional text description for your desktop.  |

For example, the following command will create a new desktop with a different background bitmap and different colors:

```
desktop define name="mydesk" os=os2 shell=pmshell lang=us background="0 128  
0" iconfont="WarpSans" iconfontsize="9" icontext="0 0 128"  
iconbackground="128 0 0" bitmap="\os2\bitmap\wsoddtbg.bmp"  
bitmapdisplay="C"
```

This command results in a new desktop entry being created in the data store as shown in Figure 151. The custom desktop has a green background with blue icon text on a red icon background. The background bitmap we selected is `wsoddtbg.bmp`, which is installed with the IBM WorkSpace On-Demand 3.0.1 OS/2 client support.

```
[/DESKTOP/OS2/PMSHELL/US/mydesk]
background=0 128 0
iconfont=WarpSans
iconfontsize=9
icontext=0 0 128
iconbackground=128 0 0
bitmap=\os2\bitmap\wsoddtbg.bmp
bitmapdisplay=C
PackageLocation=\\ATLAS2\TDMPKGS\DESKTOP\OS2\US\PMSHELL
PolicyLocation=\\ATLAS2\TDMPKGS\DESKTOP\OS2\US\POLICY
Policy=Default
ProfileLocation=\\ATLAS2\TDMPKGS\DESKTOP\OS2\US\PROFILES
Profile=Default
associated class=com.ibm.tdm.task.desktop.PMShellDesktop
```

*Figure 151. A custom OS/2 desktop in the data store*

To add your own bitmap to the OS/2 desktop instead of the default bitmaps supplied with IBM WorkSpace On-Demand 3.0.1, you need to copy your bitmap to a location on the deployment server where it can be found by the OS/2 client machine. It is probably easiest to place your bitmap in the `\os2\bitmaps` directory for the client, which is where the system bitmaps are also located.

Since the OS/2 client does a remote boot from the server, you will need to check the FIT file to verify where the client directory `\os2\bitmaps` maps to on the server. The default FIT file provided with OS/2 client support maps the client's `\os2` directory to `os2\bb20\us\tree\os2` in the RPLFILES alias of the deployment server. If you copy your bitmap to this directory, it will be available for all your OS/2 client machines and can be added to your client desktops.

When you have defined your OS/2 desktop, you can assign it to a user in order to allow the user to log on to an OS/2 client machine. Do this with the `user assign` command, as follows:

```
user assign user=user3 desktop=mydesk os=os2 shell=pmsHELL lang=us
```

### 5.8.3 Defining an OS/2 desktop in the GUI

The `desktop define` command to create a new OS/2 desktop can also be executed in the IBM WorkSpace On-Demand 3.0.1 GUI. From the **Tasks** tab, select **Define a desktop**, then select **OS2** from the drop-down list that appears. You will be presented with a dialog to define the desktop as shown in Figure 152.

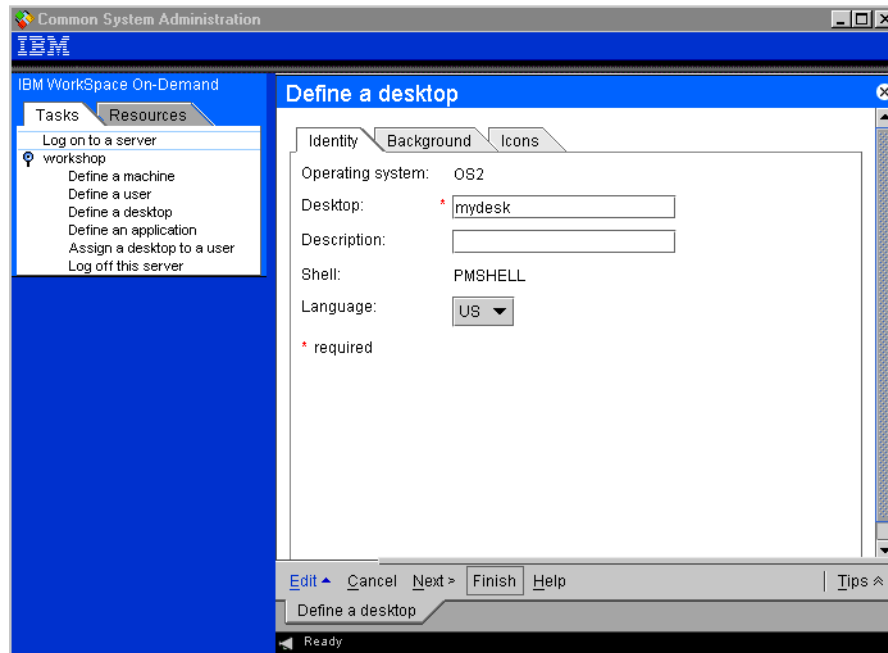


Figure 152. Defining an OS/2 desktop in the administration GUI

You can name your desktop and select the language and shell. When you click the **Identity** tab, you will be presented with a dialog that lets you set the fonts and colors.

All the remaining parameters of the `desktop define` command can be set via this dialog. Instead of explicitly setting the RGB values for each of your colors, you can select predefined colors from a drop-down list for the background, icon text, and icon background. If you select transparent for the icon background, the icon text will appear over the desktop background color or bitmap. This is most useful if the entire desktop is covered with a bitmap.

You can also specify the path and name of the bitmap you want for the background. This path must not include a drive letter, and the client must be able to locate the path via the FIT file.

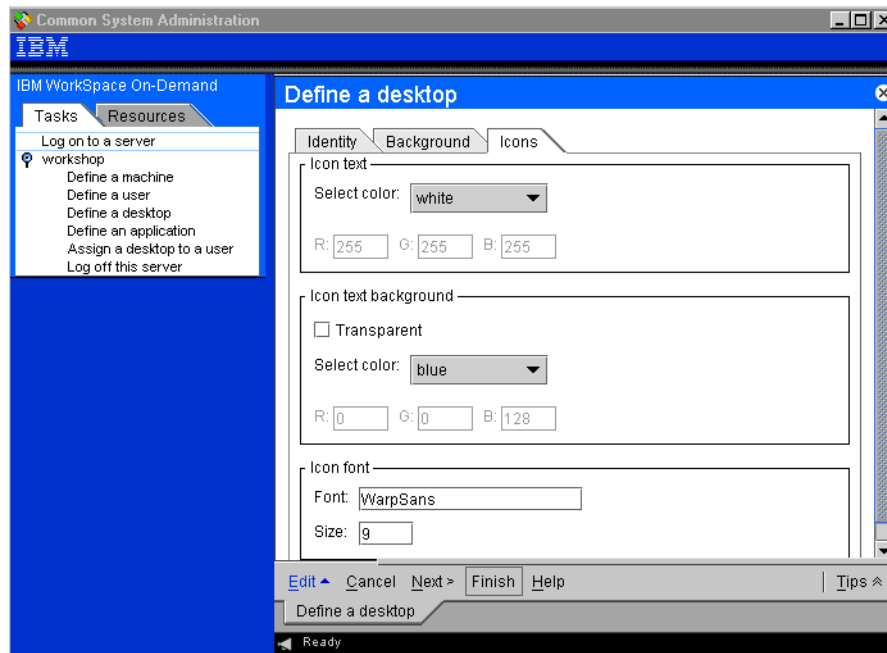


Figure 153. Icon tab for the OS/2 desktop definition

When you have finished defining the desktop, you can assign it to a user via the administration GUI as well. From the **Tasks** tab, select **Assign a desktop to a user**. When you select the user from the resulting list and then select the operating system, you will see a dialog as shown in Figure 154 on page 249. If you have OS/2 applications defined, you can assign them to the user at the same time. If you click the **Customize** button, you can customize the desktop for this user via a dialog.

When you have finished assigning the desktop to the user, the user will be able to log on to an OS/2 client machine.

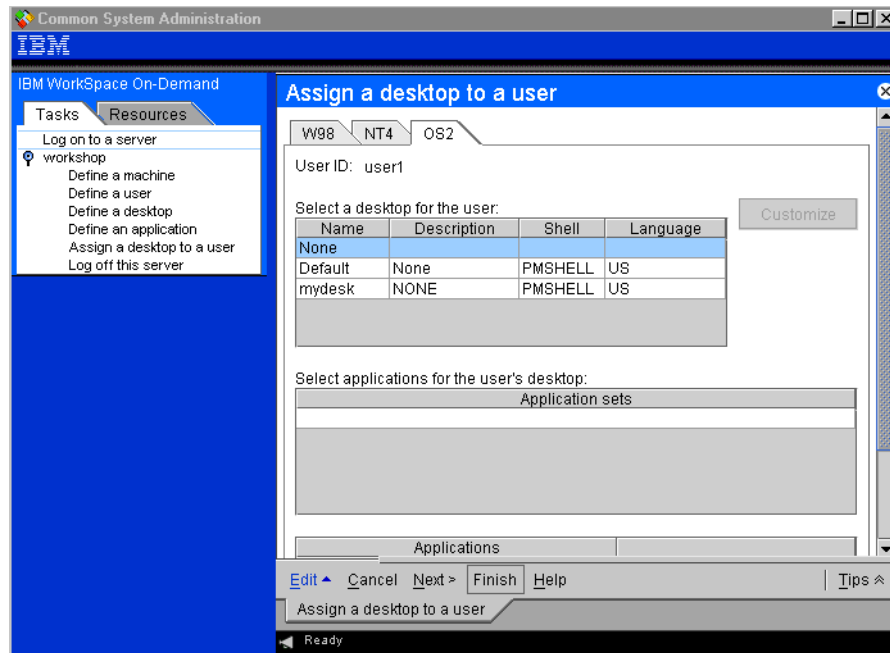


Figure 154. Assigning an OS/2 desktop to a user

#### 5.8.4 The PMLOGON shell

IBM WorkSpace On-Demand 3.0.1 provides a simplified version of the OS/2 Workplace Shell, known as the PMLOGON shell. This chapter describes the PMLOGON shell and discusses ways in which you can modify the shell to better suit your requirements, or even replace it with a shell of your own.

For many years, OS/2 has provided a user interface known as the Workplace Shell (WPS). This shell is object oriented and can be easily customized. Programmers can build common functions once and, using object-oriented programming techniques, reuse them everywhere that such functionality is required. This allows you to easily modify or enhance the shell by making a major change in one place, and that change will automatically be utilized throughout the system.

With IBM WorkSpace On-Demand 3.0.1, IBM has modified the Workplace Shell to restrict users' access to its features. IBM WorkSpace On-Demand 3.0.1 uses the identical basis of the Workplace Shell. The new shell eliminates pop-up menus and settings notebooks and prevents access to any type of configuration or customization by the end user. The new shell

eliminates any means to obtain access to the rest of the Workplace Shell, including command prompts.

The only icons that are accessible are those that are defined for the user as network applications in the IBM Workspace On-Demand 3.0.1 server's domain. These icons are dynamically created on the desktop for the user during the logon process. End users can do nothing with these icons except double-click on them to start them. They cannot move, delete, or alter the icons in any way. The icons are destroyed by the system when the user logs off or shuts down. The new shell is started by the `RUNWORKPLACE` statement in the client's `CONFIG.SYS` file:

```
SET RUNWORKPLACE=Z:\OS2\PMLOGON.EXE
```

`PMLOGON.EXE` manages the construction and operation of the new shell. By default, `PMLOGON` initializes the shell with a basic desktop, then brings up a logon window. Notice that prior to logon, the desktop is hidden behind a special blue window and is not available to the user until a successful logon occurs.

The `PMLOGON` shell provides a number of ways that you can modify its behavior to suit your particular requirements, without the need to replace the shell with a customized shell of your own. You can modify the `PMLOGON` shell using a variety of parameters, user exits, setup strings, and other techniques that are described on the following pages.

### 5.8.5 Modifying the `PMLOGON` shell

You can modify certain options that control the client logon process. These options are controlled in the `CONFIG.SYS` file with the `RUNWORKPLACE=` statement.

If you modify the template for the `CONFIG.SYS` file in the machine template directory, all subsequent clients created using that machine class type have the change. The template for the `CONFIG.SYS` file is located in the `\TDM\MM\CLIENT\RO\OS2\BB20\US\CNFG\<machine template name>` directory on the server.

If you want the change to affect only a single client, modify the `CONFIG.SYS` in that machine's own directory. This `CONFIG.SYS` file is located in the `\TDM\MM\CLIENT\RO\MACHINES\<client_id>\OS2\BB20\US\` directory on the server.

`PMLOGON` accepts a number of parameters, all of which are optional. If you specify `PMLOGON` in the `RUNWORKPLACE=` statement without any parameters,

PMLOGON simply uses its defaults; it displays a logon window with no user ID or password and includes the domain name obtained from the client's IBMLAN.INI file.

You can include the following parameters to specify the user ID, password, and domain in the `PMLOGON` command. They are displayed in the logon window and can be changed by the end user:

**/U:<UserID>**      User ID  
**/P:<Password>**    Password  
**/D:<Domain>**      Domain name

You can include the following parameters to specify the user ID, password, and domain in the `PMLOGON` command. They are displayed in the logon window but cannot be changed by the end user:

**/UF:<UserID>**      Fixed user ID  
**/PF:<Password>**   Fixed password  
**/DF:<Domain>**      Fixed domain name

The following parameters affect the behavior of the PMLOGON program during the logon process:

**/AUTO**            Automatic logon. The logon window is not displayed, and you must specify the user ID and password with the `/PW` parameter.

**/BMP:<filespec>** This specifies a fully-qualified path and file name for the bitmap displayed in the logon window.

**/NOPI**            Turns progress indicators off. The progress indicators are the small dialogs that animate to show that PMLOGON is processing.

**/NOSM1**          Removes system modality from PMLOGON.EXE (this means that PMLOGON will not switch to the foreground while it is processing) after user exit 1 is executed or at any time after a logoff.

**/URX:**            Specifies the name of the user exit program.

You should note the following:

- If you specify multiple instances of the same parameter (for example: `/U:usera /U:userb`), the last parameter is used, and all previous ones are ignored.

- If you specify multiple fixed and non-fixed instances of the same parameter type (for example, /U:usera /UF:userb /U:userc), the last fixed parameter is used (for example, /UF:userb), and all others are ignored.
- You can specify the /AUTO parameter along with /U or /UF, with /P or /PF, and with /D or /DF. If multiple instances of the same parameter type are present, the same rules apply as in the two previous notes.
- If you specify /AUTO, and there is insufficient information to attempt a logon (for example, the user ID is not specified), PMLOGON displays the logon window and the /AUTO parameter is ignored.
- If you specify /AUTO and /PW, and there is insufficient information to attempt a log on (for example, either the user ID or password are not specified), the logon window is displayed, and the /AUTO parameter is ignored. /PW is processed.

You should be cautious about using the /AUTO parameter:

- If you specify /AUTO, and an end user logs on successfully, then subsequently logs off, the user is automatically logged on again.
- If you specify /AUTO, and the user's password has expired, the logon is no longer automatic. PMLOGON displays the logon window and prompts the user to change the password. Note that any user can change the password at this point by completing the information in the Change Password window on the client. This may permit a user to access another user's account.
- If you specify /AUTO, and the user changes the password, the /AUTO parameter is not honored until the next time the client machine is booted. For example, if the user logs on and the password is expired, the user changes the password when prompted. When the user logs off, PMLOGON will attempt to automatically log on again. In this case, the logon will fail because the password specified on the `RUNWORKPLACE` statement and the password at the domain controller no longer match. You must update the `RUNWORKPLACE` statement to the new password.

### 5.8.6 Replacing the logon bitmap

You can set the client logon window to display any bitmap that you wish. For example, you may wish to use your corporate logo here. You can set all client workstations in a specific machine class to use the same bitmap, or you can set any individual client to display its own bitmap image.

The logon window bitmap is controlled by the /BMP parameter in the `RUNWORKPLACE=PMLOGON` statement in the client CONFIG.SYS file. If you include



the /BMP parameter to specify a bitmap, that bitmap is scaled and displayed in the logon window. If you do not specify a bitmap, PMLOGON uses the PMLOGON.BMP file in the TDM\MM\CLIENT\RO\OS2\BB20\US\TREE\OS2\BITMAP directory.

To change the client logon bitmap:

1. Edit the CONFIG.SYS file using an ASCII editor.
2. Modify the `RUNWORKPLACE=Z:\OS2\PMLOGON.EXE` statement to include the parameter /MP:file specification. The file specification parameter is the fully-qualified path and file name of the bitmap to be displayed on the logon window. For example:

```
RUNWORKPLACE=Z:\OS2\PMLOGON.EXE /BMP:Z:\BITMAPS\USER1.BMP
```

3. Save the CONFIG.SYS file.
4. Place the desired bitmap in the directory that is specified in the `RUNWORKPLACE=` statement.

#### Note

If you specify a file name only and not a fully-qualified path, DPATH is used to locate the bitmap. If the specified bitmap file is invalid or cannot be found, PMLOGON uses the PMLOGON.BMP file in the \OS2 directory.

### 5.8.7 Replacing the background bitmap

The background bitmap for the client's desktop is specified in the client's OS2.INI. The client's OS2.INI resides in the TDM\MM\CLIENT\RW\MACHINES\<Machine name>\OS2\BB20\US\OS2 directory. When the client is defined, the OS2.INI is created from a OS2.RC file that resides in the TDM\MM\CLIENT\RO\OS2\BB20\US\CNFG\DEFAULT\OS2 directory. You can edit this template file to change the background bitmap prior to defining your clients, or you can modify individual client background bitmaps after client definition by editing each client's OS2.INI file.

To change the background bitmap, you will need an INI file editor. As always, when manipulating INI files, it is best to make a backup copy of the file first. The name of the bitmap file is stored in the PM\_SystemBackground section under the DefaultDesktopBackground keyname.

### 5.8.8 Locking up the desktop

IBM WorkSpace On-Demand 3.0.1 creates a Lockup program icon on the desktop of all client workstations. Launching this program locks the desktop and presents the end user with a lockup bitmap. Users can unlock the desktop and remove the lockup bitmap by entering their LAN password and pressing the Enter key. If the end user has no LAN password, the lockup bitmap is removed by pressing the space bar and the Enter key.

The default lockup bitmap for all clients is the BIGBLU.BMP file that resides in the TDM\MM\CLIENT\RO\OS2\BB20\US\TREE\OS2\BITMAP directory. The simplest way to change the lockup bitmap file is to replace the BIGBLU.BMP file with a new bitmap.

#### Note

The user's Lockup password is taken from the text entered in the password field on the logon window after a successful logon. However, if a user does not require a password, but nevertheless enters text in the password, the text will be ignored for logon purposes, but will be picked up and used as the Lockup password.

This may result in users being unable to unlock their desktops since they were unaware of, or unable to remember, the text that they entered in the password field on the logon window. In such cases, the user must reboot the client.

Parameters for PMLOGON.EXE relate to the desktop lockup function. Table 2 lists these additional parameters and their meanings.

Table 2. Desktop lockup - PMLOGON parameters

| Parameter    | Description  |
|--------------|--|
| /AL          | Indicates that automatic desktop lockup is enabled. The desktop lockup window will be enabled after 15 minutes of keyboard and mouse inactivity.   |
| /AL: minutes | Indicates that automatic desktop lockup is enabled. The desktop lockup window will be enabled after the specified number of minutes of keyboard and mouse inactivity. Minutes must be a decimal number between 1 and 99. If minutes is not valid, the default time-out value of 15 minutes will be used. |
| /LOS         | Indicates that the desktop lockup window will be displayed when the end user's desktop first appears.  |

| Parameter | Description   |
|-----------|---|
| /NOLOCK   | Indicates that the desktop lockup object should be destroyed when the end user's desktop first appears. |

### 5.8.9 Logoff and shutdown options

The Logoff and Shutdown icons on the user's desktop are used to invoke programs named TLOGOFF.EXE and TSHUTDOWN.EXE, respectively. Under normal circumstances, the PMLOGON shell calls these programs. However, if you are modifying or replacing the PMLOGON shell, you may wish to call these programs directly. For example, you may use the PMLOGON user exits to create a customized logon window and to provide a shutdown option for the user. You can call TSHUTDOWN.EXE directly from your customized logon window.

You can modify the behavior of the Logoff and Shutdown programs with command line parameters. Both TLOGOFF.EXE and TSHUTDOWN.EXE support the following parameters:

- /Q** Prevents the LAN message box from appearing during logoff or shutdown.
- /N** Prevents the "Are you sure" dialog from appearing during logoff or shutdown.

### 5.8.10 Customizing program objects

With a traditional OS/2 client workstation, it is possible for an end user to customize the objects on the Workplace Shell desktop in a wide variety of ways. Most desktop objects have a settings notebook that allows the end user to change the properties for the object. WPS setup strings can also be used to set the initial properties during the creation of an object or to change the properties of an existing object.

Manual methods of modifying the desktop and its objects do not work under IBM WorkSpace On-Demand 3.0.1 since the end user does not have access to system settings, settings notebooks, and so on. Indeed, one of the benefits of IBM WorkSpace On-Demand 3.0.1 is that an end user cannot modify, and potentially corrupt, their desktop environment.

All objects are dynamically created on the desktop for the user during the logon process and are destroyed when the user logs off or shuts down. This effectively prevents any objects on the desktop from being altered in any way by the user.

As a network administrator, however, you may wish to modify the layout of the desktop or the appearance of the program objects on the desktop.

IBM WorkSpace On-Demand 3.0.1 allows you to do this in two ways:

1. As part of the application package you create. Before finishing the application package, you can create folders the application object is in, set environment variables, change object settings, and change the application icon to your needs. This information is stored as part of the application package in the APPPARTM.INI file.
2. By specifying a setup string as part of the public application definition. If both the application package and the setup string contain desktop information, the information stored in the application package is used.

The method of specifying a setup string as part of the public application definition is also discussed in this chapter.

A setup string consists of a keyname followed by an equal (=) sign followed by a value. The value must be encapsulated in single (') or double (") quotes. Figure 155 is an example of a setup string that would set the icon, folder, and folder position of a public application.

```
APPLICATION DEFINE NAME=NetScape OS=OS2 LANG=US SHELL=PMSHELL
AppDrive=P AppPath='\\BRANCH1\APPS\NetScape\Program'
COMMAND='Netscape.EXE -browser /l en_us' ICONTITLE='NetScape'
FOLDER='Common^applications' FOLDERPOS='10,70'
```

*Figure 155. Sample of an OS/2 setup string with desktop settings*

#### **Note**

The title of the icon is taken from the DESCRIPTION field in the network application definition and is unaffected by the object's setup string. Note that if a carriage return is necessary in your title, specify the DESCRIPTION with the caret character (^) as in "WordPro^Startup".

The support for desktop settings of applications has been made available as a setup string in the `APPLICATION DEFINE` command. Table 3 shows the available setup strings.

Table 3. OS/2 application setup strings

| KEYNAME     | VALUE         | DESCRIPTION   |
|-------------|---------------|---|
| DESCRIPTION | Description   | Specifies a resource description. The description must be less than 256 characters.   |
| FOLDER      | Folder title  | Specifies the title of the desktop folder that contains the application icon. FOLDER can be a maximum of 75 bytes in length. If FOLDER is not specified, the application icon is on the desktop.  |
| FOLDERPOS   | X,Y           | Specifies the desktop position of the folder containing the application icon. FOLDERPOS is specified as X,Y. Valid values for X and Y are 0-100 inclusive. These values represent the position in percentage coordinates relative to the lower left corner of the window. FOLDERPOS is valid only when FOLDER is specified. |
| ICONPOS     | X,Y           | Specifies the position of the application icon on the desktop as X,Y. Valid values for X and Y are 0-100 inclusive. These values represent the position in percentage coordinate relative to the lower left corner of the window.   |
| ICONFILE    | Icon filename | Specifies the title displayed for the application icon. ICONTITLE can be a maximum of 75 bytes in length.   |

### 5.8.11 Replacing the PMLOGON shell

The PMLOGON shell does not have to be used as the Workplace Shell for IBM WorkSpace On-Demand 3.0.1. It can be replaced by modifying the `SET RUNWORKPLACE` statement in the `CONFIG.SYS`. This statement indicates what program to load as the Workplace Shell. For example, Netscape Navigator could be used as the shell program to provide the user with a browser interface by modifying the `SET RUNWORKPLACE` statement as shown in Figure 156.

```
SET RUNWORKPLACE=Z:\NETSCAPE\NETSCAPE.EXE -k1
```

Figure 156. Modifying the `SET RUNWORKPLACE` statement

The -k1 option starts Netscape Navigator in kiosk mode with no menu bar. When the workstation is started, it will boot directly to a browser desktop.

This provides a simple way to create a secure kiosk with intranet or Internet access. Note, however, that in this environment, all features provided by PMLOGON are not available. This includes user logon and verification and the ability to use user and application FIT files (since they are only processed at logon and application startup by the default PMLOGON shell and its application launcher).

---

## Chapter 6. Applications

After having defined users and machines, and then defined desktops and assigned them to users, we must define and assign applications to users. The following sections describe the application object and show you how to define applications for each client operating system in IBM WorkSpace On-Demand 3.0.1. How to define and assign several popular applications for each operating system is also described in detail.

---

### 6.1 The general application object structure

An application is an object in IBM WorkSpace On-Demand 3.0.1 just as machines and desktops are objects. The application object has associated actions and a data store representation. The following actions can be performed on the application object:

- List** Displays a list of all applications created on the current server.
- Define** Creates a new application.
- Delete** Removes an application from the server.
- Modify** Changes the attributes of an application.
- Query** Displays the attributes of an application.

When you define an application, you must also specify the operating system, shell, and language for that application. When you define or modify an application, you can specify attributes for that application. The attributes for an application differ depending on the client operating system selected. The following is the generic command to define an application:

```
application define name=<appname> os=<os> lang=<lang> shell=<shell>  
attributes...
```

#### 6.1.1 Windows 2000, NT and 98 application attributes

For Windows 2000 Professional Edition, Windows NT Workstation and Windows 98, the following attributes apply to an application:

- Appdrive** The drive letter on which the application is installed. This is not required if the application is installed locally on the client hard drive. Most applications will be installed on the server, and the Appdrive maps to an alias on the server where the application is installed.
- Apppath** The UNC path to the installed application on the server. This is not required if the application is installed locally on

the client hard drive. The UNC path is set in the format \\<server name>\<share>.

|                        |   |
|------------------------|---|
| <b>Updateatlogon</b>   | This parameter can be set to YES or NO. The default is YES. It specifies that the machine-specific application files, such as DLLs, that must be located on the local machine should be updated when the user first logs on to the machine.   |
| <b>Promptforupdate</b> | If Updateatlogon is set to YES, this parameter specifies whether the user should be prompted before updating the machine-specific application files. If the user chooses not to perform the update, the application will not be executable. The default is NO, forcing the local files to be updated. |
| <b>Folder</b>          | Specifies the name of the folder that will contain the application's icon. By default, no folder is used and the icon is located on the desktop.  |
| <b>Description</b>     | An optional text description of the application.  |

### 6.1.2 OS/2 application attributes

For OS/2, the following attributes apply to an application:

|                  |  |
|------------------|--|
| <b>Appdrive</b>  | Specifies the drive letter (from C to Z) where the application resides. If the Apppath attribute specifies a UNC name, appdrive can be either a drive letter or an asterisk (*) to indicate the next available drive.        |
| <b>Apppath</b>   | Specifies the path to the location where the application resides. This can be either a UNC name to represent an alias on the server or a fully-qualified path without drive specification.                                   |
| <b>Command</b>   | Specifies the executable file name for the application. This must have an extension of .exe, .com, .cmd, or .bat.  |
| <b>Icontitle</b> | Specifies the title for the icon representing this application on the desktop. The default is NONE, which will use the executable file name as the icon title. You can specify a title string up to 75 characters in length. |
| <b>Folder</b>    | Specifies the title of the desktop folder that will contain the application icon. The default is NONE, causing the application to be created on the desktop.   |
| <b>Folderpos</b> | This is set as "X,Y" and sets the position of the folder on the desktop. This parameter is not valid if Folder is not specified.   |



|                    |   |
|--------------------|---|
|                    | Valid values for X and Y are from 0 to 100, indicating the folder's position in percentage coordinates relative to the lower-left corner of the desktop.  |
| <b>Iconpos</b>     | This is set as "X,Y" and sets the position of the icon on the desktop. Valid values for X and Y are from 0 to 100, indicating the icon's position in percentage coordinates relative to the lower-left corner of the desktop. |
| <b>Iconfile</b>    | Specifies the fully-qualified file name of the icon to be used for this application. The default value is NONE, indicating that the program's default icon should be used.  |
| <b>Wrkpath</b>     | Sets the working directory for the application as either a UNC name or fully-qualified path with no drive letter. The default is NONE.  |
| <b>Wrkdrive</b>    | The drive letter (C to Z) for the application's working directory.  |
| <b>Apptype</b>     | Set either of PM, VIO, or FS to indicate that the application is a Presentation Manager application, an OS/2 windowed application, or an OS/2 full screen application, respectively.  |
| <b>Description</b> | An optional text description for the application.   |

### 6.1.3 Application object representation

When an application is created and defined, it is represented in two places in IBM WorkSpace On-Demand 3.0.1. First, the application's files and the changes it makes on a client machine with respect to the client operating system are stored in the TDMPKGS share in IBM WorkSpace On-Demand 3.0.1. This directory structure is known as an application package, and an example of an application package is shown in Figure 157 on page 262. Application packages are discussed in more detail in Section 6.5, "Application packages" on page 272.

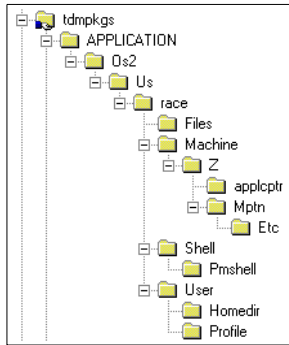


Figure 157. An application package directory

Second, an application is also represented as an object in the data store. The application attributes' as settings when you define or modify an application are recorded here. Figure 158 shows the data store representation of the same application as shown in Figure 157.

```

[/APPLICATION/OS2]

[/APPLICATION/OS2/PMSHELL]

[/APPLICATION/OS2/PMSHELL/US]

[/APPLICATION/OS2/PMSHELL/US/race]
command=race.exe
appdrive=x
apppath=\\atlas2\\apps\\race
icontitle=Race
PackageLocation=\\ATLAS2\\TDMPKG\\APPLICATION\\OS2\\US
AppType=PM
WrkDrive=None
WrkPath=None
Description=None
IconFile=None
IconPos=None
Folder=None
FolderPos=None
associated class=com.ibm.tdm.task.application.PMShellApplication
  
```

Figure 158. A sample application representation in the data store

When an application has been defined, it must be assigned to a user. This is done using the `assign` action on the user object, specifying the application

name as well as the operating system, language, and shell in the same way that desktops are assigned to users. The following is the generic command to assign an application to a user:

```
user assign user=<username> application=<appname> os=<os> lang=<lang>  
shell=<shell>
```

#### 6.1.4 Application sets

In addition to individual application objects, IBM WorkSpace On-Demand 3.0.1 also provides application sets. An application set is referenced by the appset object in IBM WorkSpace On-Demand 3.0.1 and refers to a collection of applications. Using appsets makes it easier to assign several applications to many users. Once you have defined an appset, you can assign the appset to the user, and all applications in the appset will be automatically assigned to that user.

When you define an appset, you must also specify the operating system, language, and shell for the appset. All applications that will be added to the appset must have the same operating system, language, and shell as the appset itself.

The following actions are valid on the appset object:

- Define** Creates a new appset.
- Modify** Changes the attributes of an appset. The only attribute that can be changed is the description.
- Delete** Deletes an appset.
- Query** Lists the contents of an appset.
- List** Lists all currently defined appsets.
- Add** Adds an application to an appset.
- Remove** Removes an application from an appset.

The following are sample appset commands:

```
appset define name=<appsetname> os=<os> lang=<lang> shell=<shell>
```

```
appset add name=<appsetname> application=<appname> os=<os2> lang=<lang>  
shell=<shell>
```

## 6.2 Introduction to defining applications

Making an application available for users of IBM WorkSpace On-Demand 3.0.1 is a complex task. Before applications can be defined on the server and assigned to users, the interaction of the application and the client operating system must be analyzed and captured so that it can be applied on a per-user basis. This section describes the necessary steps to accomplish this for each client operating system type.

Figure 159 illustrates the overall structure of the application support provided by IBM WorkSpace On-Demand 3.0.1.

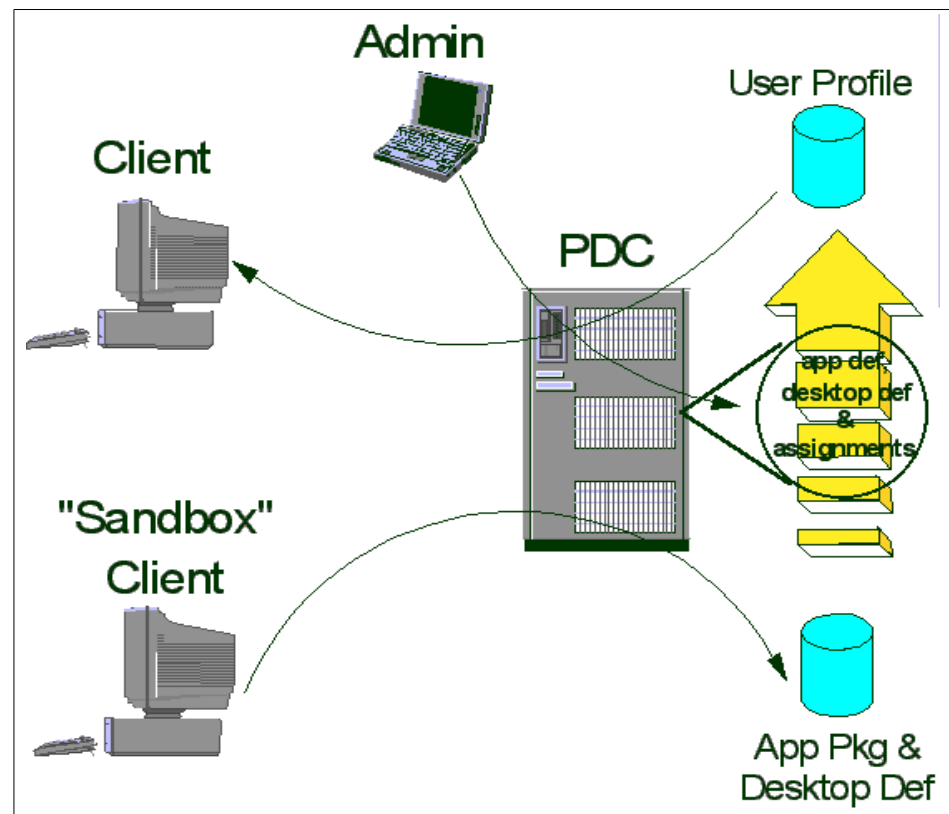


Figure 159. How applications are made available to users

The exact procedure for distributing applications to client machines is different depending on the client OS on which it will deploy. The following outline gives a general idea of how to create and distribute applications. For

specific guidance on applications for each operating system type, see Section 6.6, “Installing Windows 2000 applications” on page 277, Section 6.8, “Installing Windows 98 applications” on page 318, and Section 6.9, “Installing OS/2 applications” on page 335.

1. Create a client with the operating system on which you want your application to run. (For OS/2 applications, this machine must be created as a sandbox as described in Section 6.9, “Installing OS/2 applications” on page 335.
2. Define a sandbox user, taking the following considerations into account:
  - a. The user must be member of the “Domain Admin” group.
  - b. No other application is assigned to this user.
  - c. Instead of the normal restricted desktop, this sandbox user must use a unrestricted *sandbox* desktop. (Note that the sandbox desktop applies only to Windows 2000, Windows NT and Windows 98 clients. OS/2 requires a sandbox machine definition as described in Section 6.9.1, “The OS/2 sandbox” on page 335.)
3. Logon to the newly created client with the sandbox user.
4. Start the CRTPKG tool to take a snapshot of the client before the installation.
5. Install your application with all the necessary adjustments and changes necessary to fit your environment. Reboot the client.
6. Make all changes to the desktop and application that you want to have changed.
7. Start the CRTPKG tool to take a snapshot of the client after the installation.

The result of this snapshot is an application package. Included in this package are all files that were installed on the client during the installation, including all changes to the registry or INI-files. Changes to the desktop, such as new icons, are also saved.

The application package itself only creates a file and directory structure within the installation path of IBM WorkSpace On-Demand 3.0.1. The application is not yet available to users.

8. Define the application within IBM WorkSpace On-Demand 3.0.1.
9. Assign the application to an end user.
10. Logon to a client machine as an end user.

This will cause a transfer of all related files and updates necessary for your application to the user’s client machine. After the update, which might

also include a reboot of the client machine, a second logon of the end user is required. After the second logon, the application is available to the end user.

**Note**

Defined applications within IBM WorkSpace On-Demand 3.0.1 are related to a specific operating system and a language. A user can roam between client machines and even operating systems. This means that applications used on two or more client operating systems and languages have to be defined for each instance of client OS and language.

Figure 160 on page 267 illustrates the process for making application packages.

The process of creating the installation packages depends on how you want the individual applications deployed. You may, for example, choose to run some applications directly from the server (as is a necessity if you run a thin-client environment or for all OS/2 clients) or have them installed locally. In addition, the installation procedure will also change from organization to organization due to the differences in IT policy regarding where home directories and the like are stored.

The examples of application installations given in this chapter will serve as a reference and can be modified to fit the needs of your specific environment.

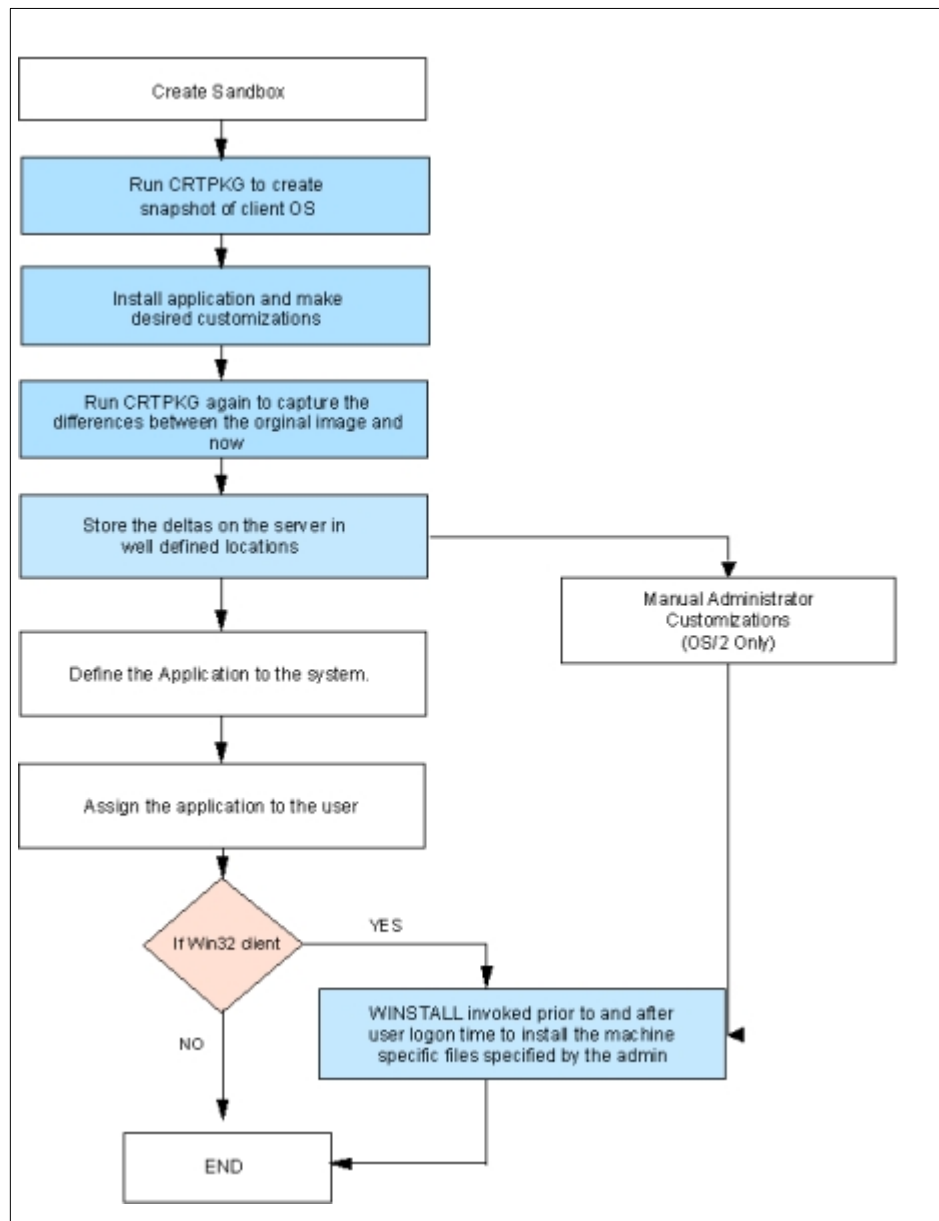


Figure 160. How to make applications available to an end user

---

## 6.3 The CRTPKG tool

The CRTPKG (Create Package) tool is used to create application packages for deployment on network clients. CRTPKG is started with the /start parameter. This takes a snapshot of your client operating system, including all operating system files and entries in the registry (if Win32).

After application installation/configuration, run CRTPKG with the /finish parameter on the sandbox client. The second invocation of CRTPKG analyzes the differences between the initial snapshot and the state of the sandbox after the application has been installed and run. This information is used to create and store the application package on the primary domain controller. The format and contents of application packages are described in Section 6.5, "Application packages" on page 272.

If CRTPKG was used on an OS/2 client, you may need to customize the application package manually on the primary domain controller after the automated part of the package creation has completed.

You can create the application definition and assign the application to end users once application package customization is complete.

### 6.3.1 The configuration file for CRTPKG

The CRTPKG tool initialization file is called CRTPKG.INI. At the point of installation of the sandbox client, it is taken from the WorkSpace On-Demand 3.0.1 file tree on the server and placed in the <Bootdrive>:\CRTPKG subdirectory on the Windows 2000, Windows NT or Windows 98 sandbox client, and in the <<<Bootdrive>:\applcptr subdirectory on the OS/2 sandbox client.

Table 4 shows where the CRTPKG.INI file is located in Windows 2000, Windows NT, Windows 98, and OS/2 file structures.

*Table 4. Location of CRTPKG.INI file on the server (origin)*

| Operating system | Location of CRTPKG.INI file   |
|------------------|---|
| Windows 2000     | <IBM WorkSpace On-Demand 3.0.1 install drive>:\TDM\MM\CLIENT\RO\W2K\W2kimg\US\INST\I386\SOEM\$\C\CRTPKG |
| Windows NT       | <IBM WorkSpace On-Demand 3.0.1 install drive>:\TDM\MM\CLIENT\RO\NT4\SP3\US\INST\I386\SOEM\$\C\CRTPKG    |



| Operating system | Location of CRTPKG.INI file  |
|------------------|--|
| Windows 98       | <IBM WorkSpace On-Demand 3.0.1 install drive>:\TDM\MM\CLIENT\RO\W98\W98SE\US\INST\C\CRTPKG |
| OS/2             | <IBM WorkSpace On-Demand 3.0.1 install drive>:\TDM\MM\CLIENT\RO\OS2\BB20\US\TREE\APPLCPTR  |

This configuration file determines which drives, directories, files, file-extensions, registry keys, and registry values are to be included or excluded for the CRTPKG snapshot. By default, the file should work for most applications you want to install. Nevertheless, it is likely that you have to modify this configuration file for some applications.

By default, the application packages are stored in the following location:

<TDMPKGS>\Application\<platform>\<cc>\<application name>,

where:

**<TDMPKGS>** The TDMPKGS share, which was set up during installation (by default, this would be C:\TDM\TDMPKGS)

**<platform>** The operating system the application is assigned to

**<cc>** The country code of the operating system

**<application name>** The name of the application package you created

#### Note

The applications are specific to the platform that they run on and not the type of application. For example, a DOS-based application that is configured to run on Windows NT 4.0 will be configured under the NT4 tree. Similarly, a DOS-based application that is to be run under on OS/2 will be under the OS2 tree.

The sections shown in Table 5 can be modified in the configuration file CRTPKG.INI.

Table 5. Sections in the configuration file CRTPKG.INI

| Section name              | Description   | Values  |
|---------------------------|---|---|
| [MACHINE_FILENAME_FORMAT] | Specifies the format of the files/folders in the created package. If Use8.3=YES, then 8.3 file names will be used in the package. Otherwise, long folder/folder names will be used.   | USE8.3=YES<br>USE8.3=NO   |
| [DRIVES]                  | Specifies the drive letters which will be included in the snapshots. The CRTPKG tool captures changes made to these drives during the installation of applications. The packaging tool does not capture any changes made to the server. | Example:<br>Drv1=c:<br>Drv2=d:  |
| [EXCLUDE_DIR]             | Specifies directories that should not be included in the snapshots. All subdirectories in these directories are also excluded from the snapshots  | Example:<br>Dir1=c:\exclude\this\dir  |
| [EXCLUDE_EXT]             | Specifies file extensions that the CRTPKG tool should exclude. Files with these extensions are excluded from the snapshots.   | Example:<br>Ext1=.da0   |
| [EXCLUDE_FILE]            | Specifies files that should not be included in the snapshots. If only a file name is specified, the packaging tool ignores that file in every directory. To exclude only one file, specify the fully-qualified path to the file.        | Examples:<br>File1=pagefile.sys<br>(This excludes pagefile.sys everywhere.)<br>File2=c:\winnt\profiles\user\ntuser.dat<br>(This excludes <i>only</i> this ntuser.dat file.) |
| [EXCLUDE_REGISTRY]        | Specifies registry keys that should be excluded from the snapshots. Subkeys of this key are <i>not</i> excluded from the snapshots.   | Example:<br>RegKey1=HKLM\System\Disk<br>RegKey2=HKCU\Software\my key  |

| Section name        | Description   | Values   |
|---------------------|---|--|
| [EXCLUDE_REG_TREE]  | Specifies registry trees that should be excluded from the snapshots. Subkeys of these keys are also excluded from the snapshots.            | Example:<br>RegTree1=HKLM,SAM\SAM<br>RegTree2=HKCU,Software\mykey  |
| [EXCLUDE_REG_VALUE] | Specifies registry values that should be excluded from the snapshots. To exclude the default registry value, include a comma after the key. | Example:<br>RegVal1=HKLM,Software\testkey,newValue<br>RegVal2=HKCU,Software\mykey,test value<br>RegVal3=HKCU,Software\IBM\Network, |

**Note**

The registry specific sections are only valid on Win32 platforms. Also, HKLM represents HKEY\_LOCAL\_MACHINE, and HKCU represents HKEY\_CURRENT\_USER.

### 6.3.2 Specifying the path of the application package

By default, application packages are created in the application's subdirectory on the TDMPKGS share of the server as described in Section 6.3.1, "The configuration file for CRTPKG" on page 268. If you use the optional /PATH parameter with CRTPKG, you can specify where the output from CRTPKG is saved. It is up to the administrator to ensure the client has access to the path specified.

The format of the parameter is:

```
/PATH:{UNC Path | Fully qualified path}
```

CRTPKG will use the input to /PATH and append <platform>\<application name>\ and use this location to save the output.

## 6.4 The WINSTALL tool

The WINSTALL tool was added to the IBM WorkSpace On-Demand 3.0.1 to provide the ability to install machine-specific changes on an installed Windows client. Many applications require changes to the machine, including changes of registry settings or DLLs in the System32 directory. Windows 2000, Windows NT 4.0 and the Windows 98 clients are installed locally on the

individual client hard disks, which means that the changes have to be made there; this is where the WINSTALL tool is very useful.

WINSTALL allows delta machine changes to be applied to an installed Windows client. This capability enables the administrator to apply delta changes to an installed Windows client at remote install time, or to apply application changes to a machine at user logon time.

The tool itself is embedded in the normal application package process and is called whenever a change to a client is necessary. For each application package, a customization file (WINSTALL.CSF) for WINSTALL exists. This specifies the changes that should be applied to the client at install or logon.

---

## 6.5 Application packages

This section describes the structure and content of application packages.

All updates that are necessary to modify a client in order to get an application working are called *application packages*. The necessary updates can be new files or environmental settings that have to be modified or added. Specific to the Windows clients, it can also be changes to, or the addition of, entries in the registry.

When a user logs on to a client the first time after a new application has been assigned to the user account, the relevant files are transferred to the client machine. After the transfer of the data and the modifications, a reboot of the client may be necessary. Independent of the reboot, a second user logon is required.

### 6.5.1 File structure of an application package

Figure 161 on page 273 shows the general file structure created with each application package. Notice that all application packages are created in the TDMPKGS alias, which usually maps to C:\TDM\TDMPKGS.

In the following sections, to avoid repetition of the same directory names, the following short versions of the application package directories will be used:

**<PLATFORM>** This short form stands for \\TDMPKGS\APPS\ <os>\<cc>.

where:

**<OS>** This is the operating system.  
**<CC>** This is the country code.  
**<AppID>** This is the name of the application package.

```
TDMPKGS -- APPS -- NT4 --- <CC> --- <AppID> +- MODFILE.LST
|w2K                                     +- NEWFILE.LST
|                                     +- LINKINFO.DAT
|                                     +- WINSTALL.CSF
|                                     |
|                                     +- USER +- HOMEDIR\
|                                     |       +- *.INF
|                                     |
|                                     +- MACHINE +- C\
|                                     |         +- *.INF
|                                     |
|                                     +- SHELL\EXPLORER\PROFILE +- DESKTOP\*.LNK
|                                     |                         +- START MENU\*.LNK
|
+- W98 --- <CC> --- <AppID> +- MODFILE.LST
|                                     +- NEWFILE.LST
|                                     +- LINKINFO.DAT
|                                     +- WINSTALL.CSF
|                                     |
|                                     +- USER +- HOMEDIR\
|                                     |       +- *.REG
|                                     |
|                                     +- MACHINE +- C\
|                                     |         +- *.INF
|                                     |
|                                     +- SHELL\EXPLORER\PROFILE +- DESKTOP\*.LNK
|                                     |                         +- START MENU\*.LNK
|
+- OS2 --- <CC> --- <AppID> +- APPPARM.INI
|                                     +- APP*.FIT
|                                     +- MODFILE.LST
|                                     +- NEWFILE.LST
|                                     |
|                                     +- USER +- PROFILE\
|                                     |       +- HOMEDIR\
|                                     |       +- USERPARM.INI
|                                     |
|                                     +- MACHINE +- MACHINE.FIT
|                                     |         +- <Boot-drive>\
|                                     |
|                                     +- SHELL\PMShell +- SHELL.INI
|                                     |
|                                     +- FILES +- AUTOEXEC.BAT
|                                     |       +- CONFIG.SYS
|                                     |       +- WIN.INI
|                                     |       +- SYSTEM.INI
|                                     |       +- OS2.INI
|                                     |       +- OS2SYS.INI
```

Figure 161. File structure of application packages

## 6.5.2 Windows 2000, NT and 98 application packages

There are three subdirectories for each application package. These are:

1. The data under <PLATFORM>\<AppID>\USER\ includes the user-specific registry changes made to the HKEY\_CURRENT\_USER tree in the registry as a result of application addition. This data is stored in the windows INF file format (for Windows 2000 and NT) or REG file format (for Windows 98) and the file is named user.inf (for Windows 2000 and NT) or user.reg (for Windows 98).
2. The data under <PLATFORM>\<AppID>\MACHINE\ includes the directory structure of all the files and folders that were modified and/or added and the system-specific (machine-specific) registry changes made to the HKEY\_LOCAL\_MACHINE tree in the registry as a result of the application installation on the client. This data is stored in the windows INF file format (for Windows 2000 and NT) or REG file format (for Windows 98) and the file is named sys.inf (for Windows 2000 and NT) or sys.reg (for Windows 98).
3. The explorer shell information, such as changes to the user's profile or additional links in the **Start** menu, are saved in the <PLATFORM>\<APPID>\SHELL\EXPLORER\ directory.

The additional package information is saved in the <PLATFORM>\<AppID>\ directory. This includes the following files:

- MODFILE.TXT, a simple list file of all *modified* files and directories that were installed on the client as shown in Figure 162 (Adobe Acrobat Reader is used as an example).

```
C:\WINNT
C:\WINNT\Profiles\All Users\Start Menu\Programs
C:\WINNT\system32
C:\WINNT\system32\MFC42.DLL
C:\WINNT\system32\MSVCRT.DLL
C:\WINNT\system32\OLEPRO32.DLL
```

Figure 162. Sample of a MODFILE.TXT (Adobe Acrobat Reader)

- NEWFILE.TXT, a simple list file of all *new* files and directories that were installed on the client as shown in Figure 163 on page 275.

```

C:\WINNT\uninst.exe
C:\WINNT\Profiles\Administrator\Start
Menu\Programs\Accessories\Adobe Acrobat
C:\WINNT\Profiles\Administrator\Start
Menu\Programs\Accessories\Adobe Acrobat\Acrobat Reader 3.02.lnk
C:\WINNT\Profiles\Administrator\Start
Menu\Programs\Accessories\Adobe Acrobat\Acrobat Reader 3.02
ReadMe.lnk
C:\WINNT\Profiles\Administrator\Start
Menu\Programs\Accessories\Adobe Acrobat\Uninstall Acrobat Reader
3.02.lnk
C:\WINNT\Profiles\All Users\Start Menu\Programs\Acrobat Reader
3.02.lnk

```

Figure 163. Sample of a NEWFILE.TXT (Adobe Acrobat Reader)

- LINKINFO.DAT, which contains information about the link files that are included with the application package.
- WINSTALL.CSF, a description file for the update tool WINSTALL. This file specifies which file structure is to be copied when the user logs on the first time after the application was assigned to the user. It also tells WINSTALL which registry keys have to be updated and where they can be found. A sample WINSTALL.CSF is shown in Figure 164.

For more information about WINSTALL, see Section 6.4, “The WINSTALL tool” on page 271.

```

[COPY_DIRECTORY_STRUCTURE]
KEY1="\ATLAS3\TDMPKGS\APPLICATION\NT4\US\AdobeR\MACHINE\C", "C:"
[INSTALL_INF_FILE]
KEY1="\ATLAS3\TDMPKGS\APPLICATION\NT4\US\AdobeR\MACHINE\inifiles.inf"
KEY2="\ATLAS3\TDMPKGS\APPLICATION\NT4\US\AdobeR\MACHINE\sys.inf"

```

Figure 164. Sample of a WINSTALL.CSF (Adobe Acrobat Reader)

### 6.5.3 OS/2 application packages

OS/2 application packages are created and defined in essentially the same way as Windows application packages. The file structure represents the differences that were discovered after the installation of an application.

OS/2 is dynamically loaded over the network and has no local files on the client by default. The created file structure of an application package cannot simply be added to an existing client, since all files of an OS/2 client live on

the server. Therefore, you will have to manually make these files part of the OS/2 boot structure for your client machines. Just as in the IBM WorkSpace On-Demand 3.0.1 environment, you can achieve this by means of FIT files. OS/2 application FIT files are described in more detail in Section 6.9.2, “The OS/2 application FIT file” on page 338.

There are three subdirectories for each OS/2 application package. These are:

1. The data under <PLATFORM>\<AppID>\USER\ includes the user-specific changes that happened as a result of adding the application. For example, Netscape profile information or application INI files can be located here. Normally, the application FIT file is used to ensure that each user gets a separate copy of these files as discussed in Section 6.9.2, “The OS/2 application FIT file” on page 338.
2. The data under <PLATFORM>\<AppID>\MACHINE\ contains the files that are added to the boot drive of the OS/2 image.
3. The data under <PLATFORM>\<AppID>\FILES contains difference files for the following system files:
  - AUTOEXEC.BAT
  - CONFIG.SYS
  - WIN.INI
  - SYSTEM.INI
  - OS2.INI
  - OS2SYS.INI

These changes will be automatically added to the client’s environment when a user who has this application assigned logs on. The summary of all changes is stored in the APPPARM.INI file.

The additional package information is saved in the <PLATFORM>\<AppID>\ directory. This includes the following files:

- MODFILE.LST:  
This file contains a list of all files that have been modified.
- NEWFILE.LST:  
This file contains a list of all files that have been added. The files listed here will be copied to the application profile directory after the application has been created. Then, when the application is assigned to a user, these files will be copied to the user’s profile for the application. The application-specific FIT file will be used to ensure that each user has a separate copy of the files. Application FIT files are discussed in more detail in Section 6.9.2, “The OS/2 application FIT file” on page 338.



- **APPPARM.INI:**

This file is created by running the `CRTPKG /finish` command at an OS/2 sandbox machine after installing, configuring, and running an application. This file is an ASCII flat file that contains a summary of all changes that have to be applied to the system files mentioned under Number 3. on page 276.

---

## 6.6 Installing Windows 2000 applications

This section describes how to create application packages for Windows 2000 client operating systems.

The applications themselves do not have to be Windows 2000 applications; Windows NT, Windows 9x or Windows 3.xx or DOS applications can run under Windows 2000 too.

There are two different approaches to implementing applications in a Windows 2000 environment:

- **Client-based applications**  
In this installation, all application files and components are installed to the client hard drive.
- **Server-based applications**  
In this type of installation, most of the application files and components are installed to a server share and only a few components reside on the client. If there are user-specific configuration files, they will be stored in the user's home directory on the server.

### 6.6.1 Adobe Acrobat Reader as a Server-based Application

For the implementation of Adobe Acrobat Reader, we suggest you implement it as a server-based application. The following description outlines this method:

1. Define a regular Windows 2000 client. (For details, see Section 4.2, "Defining Windows 2000 client machines" on page 105.)
2. Define a sandbox user and assign a sandbox desktop to the user in the CLI console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, logon as the sandbox user (usersand).
4. Create an apps share on the server and make sure that the sandbox user has write access to the share:

```
net share apps=d:\apps
```

5. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```

6. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

7. Start the installation of Adobe Acrobat reader. Install and enter P:\Adobe\Reader as the installation path.

8. Boot your sandbox client after the installation is complete.

9. Log on as the sandbox user (usersand).

10. At a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG Acro4srv /finish
```

This step creates the application package on your server.

11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="acro4srv" os=w2k shell=explorer lang=us  
apppath="\\<server>\APPS" Appdrive="P"
```

12. You can assign the application to other users now with the following command in the command console on your server:

```
user assign application="acro4srv" os=w2k shell=explorer lang=us  
user="User7"
```

Where User7 is the user to whom Adobe Acrobat Reader is assigned.

When you log on with, for example, User7, all related files and necessary updates for Adobe Acrobat Reader will be transferred to the user's client machine. The client machine will reboot automatically, and a second logon is required after the update.

### 6.6.2 Adobe Acrobat Reader as a Local Application

For the implementation of Adobe Acrobat Reader as a local application that uses the local drive for home directory of saving the individual settings. We suggest the following:

1. Define a regular Windows 2000 client. (For details, see Section 4.2, "Defining Windows 2000 client machines" on page 105.)

2. Define a sandbox user and assign a sandbox desktop to the user in the CLI console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, logon as the sandbox user (usersand).
4. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

5. Start the installation of Adobe Acrobat reader. Install and enter C:\Adobe\Reader as the installation path.

6. Boot your sandbox client after the installation is complete.

7. Log on as the sandbox user (here usersand).

8. At a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG Acro4loc /finish
```

This step creates the application package on your server.

9. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server. Remember, there are not Apppath and Appdrive parameters for a local application:

```
application define name="acro4loc" os=w2k shell=explorer lang=us
```

10. You can now assign the application to other users with the following command in the command console on your server:

```
user assign application="acro4loc" os=w2k shell=explorer lang=us
user="User7"
```

Where User7 is the user to whom Adobe Acrobat Reader is assigned.

When you log on with, for example, User7, all related files and necessary updates for Adobe Acrobat Reader will be transferred to the user's client machine. A second logon or even a reboot is required after the update.

### 6.6.3 Norton AntiVirus 7.0 as a Local Application

The following steps describe the implementation of Norton AntiVirus 7.0 Corporate Edition as a local application that uses the local drive for the home directory where individual settings are saved:

1. Define a regular Windows 2000 client (for details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).

2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
> user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer user=usersand
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Edit the C:\CRTPKG\CRTPKG.INI file and add the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_REG_VALUE]
RegVal8=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,SelectedSubject

RegVal9=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,SelectedInfectionInformation

RegVal10=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,WarningMessage

RegVal11=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,WarningSubject

RegVal12=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,WarningInfectionInformation

RegVal13=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,SenderMessage

RegVal14=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,SenderSubject

RegVal15=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,SenderInfectionInformation

RegVal16=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,MessageText

RegVal17=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \LocalScans \ClientServerScheduledScan_1,MessageText

RegVal18=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Common,SelectedMessage

RegVal19=HKLM,SOFTWARE \INTEL \LANDesk \VirusProtect6 \
CurrentVersion \Storages \Filesystem \RealTimeScan,MessageText
```

5. From a DOS command prompt on the client, enter:  
`C:\CRTPKG\CRTPKG /start`
6. Start the installation of Norton AntiVirus 7.0 from your source directory or CD.
7. Specify the directory C:\NAVW2K as your installation path.
8. Reboot the client machine.
9. Start Norton AntiVirus, and customize each of the applications
10. From a DOS command prompt on the client, run:  
`c:\crtpkg\crtpkg navw2k /finish`
11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:  
`application define name="navw2k" os=w2k lang=us shell=explorer`
12. Define a non-sandbox user in the command console on your server (here AppTest):  
`user define user="AppTest"`
13. Assign a desktop to the user in the command console on your server:  
`user assign desktop=default os=w2k lang=us shell=explorer  
user="AppTest"`
14. Assign Norton AntiVirus 7.0 to the user in the command console on your server:  
`user assign application="navw2k" os=w2k lang=us shell=explorer  
user="apptest"`
15. Define a regular Windows 2000 client (for details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).

Finally, log on to the non-sandbox client as the non-sandbox user you created (here AppTest). After the first logon, all related files and updates for Norton AntiVirus will be transferred to the user's client machine. A second logon is required after the update.

#### 6.6.4 Host On-Demand as a Server-based Application

The following steps describe the installation of Host On-Demand as a server-based application:

1. Define a regular Windows 2000 client (for details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).

2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:
 

```
user define user="usersand"
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```
3. On the sandbox client, log on as the sandbox user (usersand).
4. Create an apps share on the server and make sure that the sandbox user has write access to the share.
5. From a DOS command prompt on the client, run:
 

```
net use p: \\<server>\apps
```
6. From a DOS command prompt on the client, enter:
 

```
C:\CRTPKG\CRTPKG /start
```
7. Install Host On-Demand and enter P:\HODW2K as your destination directory. Ignore the CreateNTService error at the end of the installation.
8. Reboot the client machine and log on again with your user ID (usersand). Install creates a Program Files directory in addition to HOD in the apps directory.
9. From a DOS command prompt on the client, run:
 

```
net use p: \\<server>\apps
```
10. Start Host On-Demand client 4.0.
11. From a DOS command prompt on the client, run:
 

```
c:\crtpkg\crtpkg HODW2K /finish
```
12. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:
 

```
application define name="HODW2K" os=w2k lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```
13. Define a non-sandbox user in the command console on your server (here AppTest):
 

```
user define user="AppTest"
```
14. Assign a desktop to the user in the command console on your server:
 

```
user assign desktop=default os=w2k lang=us shell=explorer
user="AppTest"
```
15. Assign Host On-Demand to the user in the command console on your server:

```
user assign application="HODW2K" os=w2k lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user you created (here AppTest). After the first logon, all related files and updates for Host On-Demand will be transferred to the user's client machine. A second logon is required after the update.

### 6.6.5 IBM Data Base 2 as a Server-based Application

The following steps describe the installation of IBM Data Base 2 (DB2) run-time client 6.1 or later, as a server-based application:

1. Define a regular Windows 2000 client (for details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Create an apps share on the server and make sure that the sandbox user has write access to the share.

5. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```

6. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

7. Install IBM DB/2 client and enter P:\DB2W2K as your destination directory.
8. Reboot the client machine and log on again with your user ID (usersand).
9. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```

10. Start DB2 Run-Time Client application.

11. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg DB2W2K /finish
```

12. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="DB2W2K" os=w2k lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```

13. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user="AppTest"
```

14. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w2k lang=us shell=explorer
user="AppTest"
```

15. Assign IBM DB/2 Client to the user in the command console on your server:

```
user assign application="DB2W2K" os=w2k lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for IBM DB2 Client will be transferred to the user's client machine. A second logon is required after the update.

### 6.6.6 Microsoft Internet Explorer 5.0 as a Local Application

Microsoft Internet Explorer 5.0 cannot run as a server-based installation; therefore, all files must be installed to the client's hard drive.

Windows 2000 includes Internet Explorer 5.0 as part of the operating system, but the program is disabled by the installation of WorkSpace On-Demand.

The following steps must be executed to enable the application package for Microsoft Internet Explorer 5.0:

1. Define a regular Windows 2000 client (For details, see Section 4.2, "Defining Windows 2000 client machines" on page 105.)
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. At a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```



5. Create a shortcut that points to C:\Program Files\Internet Explorer\IExplore.exe and place the shortcut in the **Start** menu (C:\Documents and Settings\usersand\Start Menu\Programs directory).
6. Run Internet Explorer and complete the Internet Connection Wizard.
7. Reboot the client machine.
8. Start Microsoft Internet Explorer 5.0. Modify all preferences within the Internet options to match your environment. This could include:
  - a. Manual proxy definition
  - b. Security options or restrictions
  - c. Manual socks definition
  - d. Location of the home page you want your users to use
  - e. Size of disk cache used (per user)
9. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Microsoft Internet Explorer 5.0.
  - c. Deleting entries in the **Start** menu if you don't want or need them.
10. From a DOS command prompt on the client, run:
 

```
c:\crtpkg\crtpkg IE5Loc /finish
```
11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:
 

```
application define name="IE5loc" os=w2k lang=us shell=explorer
```
12. Define a non-sandbox user (AppTest) in the command console on your server:
 

```
user define user=AppTest
```
13. Assign a desktop to the user in the command console on your server:
 

```
user assign desktop=default os=w2k lang=us shell=explorer
user="AppTest"
```
14. Assign Microsoft Internet Explorer 5.0 to the user in the command console on your server:
 

```
user assign application="IE5loc" os=w2k lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Microsoft Internet Explorer 5.0

will be transferred to the user's client machine. A second logon is required after the reboot of the client.

### 6.6.7 Paint Shop Pro 5 as a Server-Based Application

The following steps describe the installation of Paint Shop Pro as a server-based application:

1. Define a regular Windows 2000 client (for details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```
3. On the sandbox client, log on as the sandbox user (usersand).
4. Create an apps share on the server and make sure that the sandbox user has write access to the share.
5. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```
6. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```
7. Install Paint Shop Pro 5 and enter P:\Paint Shop Pro 5 as your destination directory.

Select the components of Paint Shop you want to have installed, such as:

  - a. Application files
  - b. Animations shop application Files
  - c. Picture tube brushes
  - d. Paint Shop Pro sample and tutorial files
  - e. Animation shop sample files
8. Reboot the client machine and log on again with your user ID (usersand).
9. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```
10. Start Paint Shop Pro 5.
11. Modify your preferences within Paint Shop Pro to match your environment.

12. Modify your client desktop to your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Paint Shop Pro 5.
  - c. Deleting entries in the **Start** menu if you don't want or need them.

13. From a DOS command prompt on the client, run:

```
net use p: /delete
```

14. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg PaintShop /finish
```

15. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="PaintShop" os=w2k lang=us shell=explorer  
AppDrive="P" AppPath="\\<server>\apps"
```

16. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user="AppTest"
```

17. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w2k lang=us shell=explorer  
user="AppTest"
```

18. Assign Paint Shop Pro 5 to the user in the command console on your server:

```
user assign application="PaintShop" os=w2k lang=us shell=explorer  
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Paint Shop Pro will be transferred to the user's client machine. A second logon is required after the update.

#### 6.6.8 Lotus Notes 5.04 as a Local Application

The following steps describe the installation of Lotus Notes 5.04 as a server-based application.

The following steps must be executed to create the application package for Lotus Notes:

1. Go to the Lotus Notes server.

2. Create an apps share on the server and make sure that the sandbox user has write access to the share.
3. Install the shared installation of Lotus Notes to your <apps>\notes directory on the server. You can perform this installation local on the server.
4. Define a regular Windows 2000 client. (For details, see Section 4.2, "Defining Windows 2000 client machines" on page 105.)
5. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

6. On the sandbox client, log on as the sandbox user (usersand).
7. From a DOS command prompt on the client, run:
 

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```
8. From a DOS command prompt on the client, enter:
 

```
C:\CRTPKG\CRTPKG /start
```
9. Install the Lotus Notes client by entering <apps>\notes\setup.exe at a DOS command prompt on the client.
10. Enter H:\Notes\Data as the Lotus Notes Data directory.
11. Reboot the client and log on with the same user ID.
12. Depending on your Lotus Notes infrastructure, there are many ways to set up the Notes desktop. Here are two possible steps you could take:
  - a. If your Lotus Notes ID files are all named user.id, you can start creating the desktop for your users at this time. When assigning the application Lotus Notes to a user, you only have to copy the user's ID file into the home directory of the user.
  - b. If you have an existing Lotus Notes environment, you can copy the desktop, local address book, and ID file to the new home directory location of the users. In this case, you should not start the Lotus Notes client before finishing the application package.
13. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.

- b. Deleting program links and/or folders that were installed with Lotus Notes.
  - c. Deleting or move entries in the **Start** menu.
14. Disconnect from your network shares by entering the following commands at a DOS command prompt on the client:
- ```
net use p: /delete  
net use h: /delete
```
15. From a DOS command prompt on the client, run:
- ```
c:\crtpkg\crtpkg Notes /finish
```
16. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:
- ```
application define name="Notes" os=w2k lang=us shell=explorer  
AppDrive="P" AppPath="\\<server>\apps"
```
17. Define a non-sandbox user (AppTest) in the command console on your server:
- ```
user define user=AppTest
```
18. Assign a desktop to the user in the command console on your server:
- ```
user assign desktop=default os=w2k lang=us shell=explorer  
user="AppTest"
```
19. Assign a home directory to the user in the command console on your server:
- ```
user modify user="AppTest" homedrive="H"  
homedir="\\<server>\tdmprfls\AppTest\w2k\us\Profiles"
```
20. Assign Lotus Notes to the user in the command console on your server:
- ```
user assign application="Notes" os=w2k lang=us shell=explorer  
user="apptest"
```
21. Copy the \tdmprfls\Notes directory to \tdmprfls\AppTest\w2k\us\Profiles for this user or all relevant Users.
22. The Lotus Notes application needs to write to the file C:\WINNT\NOTES.INI. To allow Notes to write to the NOTES.INI, change the PRTDSK2K.INI file in C:\TDM\TDMPRFLS. See Figure 165 on page 290.

```

[MODE]
Mode="ALLOW"
; Mode can be "ALLOW" or "DENY"

; Specify in "Unrestricted" key of [DRIVES] any local client drives
;   to which access will be fully unrestricted by the driver
; [DRIVES]
;Unrestricted="C"

; Specify below directories that you wish to explicitly
;   "ALLOW" or "DENY" access to
; [DIRECTORIES]
dir1="C:\DOCUMENTS AND SETTINGS\"
dir2="C:\WINNT\SYSTEM32\SPOOL\"

[FILES]
file1="*.TMP"
file2="C:\WINNT\NOTES.INI"

```

Figure 165. Changes for Notes in the PRTDSK2K.INI file

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Lotus Notes client will be transferred to the user's client machine. A second logon is required after the update.

### 6.6.9 Lotus SmartSuite Millennium as a Server-Based Application

Lotus SmartSuite Millennium is also known as Version 9.5; we used an updated Version 9.5.2.

The following steps must be executed to create the application package for Lotus SmartSuite:

1. Create an apps share on the server and make sure that the sandbox user has write access to the share.
2. Define a regular Windows 2000 client. (For details, see Section 4.2, "Defining Windows 2000 client machines" on page 105.)
3. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
```

```
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

4. On the sandbox client, log on as the sandbox user (usersand).
5. From a DOS command prompt on the client, run:  

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```
6. From a DOS command prompt on the client, enter:  

```
C:\CRTPKG\CRTPKG /start
```
7. Start the installation of the Lotus SmartSuite server part on your server. Select **Install to a File Server** and install to P:\Lotus.
8. Start the installation of the Lotus SmartSuite client part by entering P:\Lotus\Install.EXE at the DOS command prompt on the client.
9. Enter H:\Lotus as the installation directory and select the components of Lotus Smart Suite that you want to install.
10. Reboot the client and log on with the same user ID.
11. Start the selected and installed programs of Lotus SmartSuite and modify the preferences to fit your needs.
12. Delete the Lotus Organizer Easy Clip from the StartUp menu by completing the following steps:
  - a. Right-click on an empty part of the Windows Task Bar
  - b. Click **Properties**
  - c. Click the **Advanced** tab
  - d. Click on **Advanced**
  - e. Expand the Programs folder
  - f. Expand the Startup folder and delete the Lotus Organizer Clip icon
  - g. Close the Windows Explorer Window
  - h. Click **Close** and **OK** to close Taskbar Properties
13. Close the SmartCenter bar and all other windows related to customizing your applications.
14. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Lotus Notes.

15. Disconnect from your network shares by entering the following commands in a DOS command prompt on the client:

```
net use p: /delete
net use h: /delete
```

16. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg LotusSS /finish
```

17. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="LotusSS" os=w2k lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```

18. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```

19. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w2k lang=us shell=explorer
user="AppTest"
```

20. Assign a home directory to the user in the command console on your server:

```
user modify user=AppTest homedrive="H"
homedir="\\<server>\tdmprfls\AppTest\w2k\us\Profiles"
```

21. Assign Lotus SmartSuite to the user in the command console on your server:

```
user assign application="LotusSS" os=w2k lang=us shell=explorer
user="apptest"
```

22. Copy the \tdmprfls\Lotus to \tdmprfls\AppTest\w2k\us\Profiles directory.

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Lotus Smart Suite will be transferred to the user's client machine. A second logon is required after the update.

#### 6.6.10 Microsoft Office 2000

The following steps must be executed to install Microsoft Office as a server-based application:

1. Install the Server part of Microsoft Office 2000 to your server (`setup /a data1.msi`). Share the installation directory on the server as "OFFICE" and make sure that the sandbox user has write access to the share.



```
net share office=d:\office2000
```

2. Define a regular Windows 2000 client with at least 1 GB of free space after Base Installation (for details, see Section 4.2, “Defining Windows 2000 client machines” on page 105).
3. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

4. On the sandbox client, log on as the sandbox user (usersand).
5. Edit the file C:\CRTPKG\CRTPKG.INI and add or modify the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_FILE]
File14=c:\windows\tasks\sa.dat

[EXCLUDE_REG_TREE]
RegTree3=HKCU,Software\Microsoft\Windows\CurrentVersion\Uninstall

[EXCLUDE_REG_VALUE]
RegVal5=HKLM,System\CurrentControlSet\Control\SessionManager,PendingFileRenameOperations
RegVal6=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Copyright
RegVal7=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Copyright
```

6. From a DOS command prompt on the client, run:  

```
net use n: \\<server>\office /persistent:yes
```
7. From a DOS command prompt on the client, enter:  

```
C:\CRTPKG\CRTPKG /start
```
8. Start the installation of the Microsoft Office by entering N:\Setup.EXE at a DOS command prompt on the client.
9. In the installer, click **Custom installation** and in the Selecting Features window, click **Run all from Network**.
10. Reboot the client and logon with the same user ID.
11. When the install continues, enter \\<server>\office for the install files location.
12. Modify your client desktop to fit your needs. This could include:

- a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Microsoft Office 2000.
  - c. Deleting or moving entries in the **Start** menu.
13. Outlook 2000 will save its data in the user's profile directory on the client. As IBM WorkSpace On-Demand 3.0.1 uses mandatory user profiles, any data saved in Outlook 2000 will be lost when the user logs off. To prevent this, configure Outlook to save the user's personal folders to the user's home directory using the following steps:
  - a. After configuring Outlook 2000, delete C:\Documents and Settings\usersand\Local Settings\Application Data\Microsoft\Outlook directory on the sandbox client
  - b. Start Outlook 2000. An error message will be displayed stating that the personal folders file could not be found. Following the prompts, create a new personal folder file on drive H: (where H: is the driver letter of the user's home directory)
14. Close all Microsoft Office 2000 application and all other windows related to customizing your applications.
15. Disconnect from your network shares by entering the following commands from a DOS command prompt on the client:  

```
net use n: /delete
```
16. From a DOS command prompt on the client, run:  

```
c:\crtpkg\crtpkg Office /finish
```
17. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:  

```
application define name="Office" os=w2k lang=us shell=explorer  
AppDrive="N" AppPath="//<server>\office"
```
18. Microsoft Office 2000 Standard Edition applications needs to write to the local disk. You must modify the file: PRTDSK2K.INI located in C:\TDM\TDMPRFLS. See Figure 166 on page 295 for an overview of these changes.

```

[MODE]
Mode="ALLOW"
; Mode can be "ALLOW" or "DENY"

; Specify in "Unrestricted" key of [DRIVES] any local client drives
;   to which access will be fully unrestricted by the driver
; [DRIVES]
;Unrestricted="C"

; Specify below directories that you wish to explicitly
;   "ALLOW" or "DENY" access to
[DIRECTORIES]
dir1="C:\DOCUMENTS AND SETTINGS\"
dir2="C:\WINNT\SYSTEM32\SPOOL\"
dir3="C:\config.msi"
dir4="C:\Program Files\Microsoft Office\Office\Shortcut Bar\Office\"
dir5="C:\Program Files\Common Files\Microsoft Shared\"
dir6="C:\WINNT\Installer\"

[FILES]
file1="*.TMP"
file2="C:\WINNT\System32\MSRDO20.DLL"

```

Figure 166. Changes for Microsoft Office 2000 in the PRTDSK2K.INI file

19. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```

20. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w2k lang=us shell=explorer
user="AppTest"
```

21. Assign a home directory to the user in the command console on your sever:

```
user modify user="AppTest" homedrive="H"
homedir="\\<server>\tdmprfls\AppTest\w2k\us\Profiles"
```

22. Assign Microsoft Office 2000 to the user in the command console on your server:

```
user assign application="Office" os=w2k lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client (with at least a 1 GB drive) as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Microsoft Office 2000 will be transferred to the user's client machine. A second logon is required after the update and reboot.

#### 6.6.11 Netscape 4.7 as a Server-based Application

The following steps describe the implementation of Netscape 4.7 as a server-based application that uses the home directory of each user for caching and saving the individual settings:

1. Define a regular Windows 2000 client (for details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer user=usersand
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Edit the C:\CRTPKG\CRTPKG.INI file and add the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_REG_VALUE]
RegVal6=HKLM, SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Pr
eferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Copyright

RegVal7=HKLM, SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Pr
eferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Copyright
```

5. Create an apps share on the server and make sure that the sandbox user has write access to the share.
6. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```

7. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

8. Start the installation of Netscape 4.7 from your source directory or CD.
9. Specify the directory P:\Netscape as your installation path.
10. Reboot the client machine.
11. From a DOS command prompt on the client, run:

```
net use h: \\<server>\tdmprfls /persistent:yes
```

12. Start Netscape. When you are asked to create a new profile, enter  
H:\Netscape\Users\default as the profile location.  
  
All definitions for the user default in Netscape are now stored on your server in the directory \tdmprfls\NetScape\Users\default.
13. Modify all preferences within Netscape to match your environment. This could include:
  - a. Manual proxy definition
  - b. Manual socks definition
  - c. Location of the home page that you want your users to use
  - d. Size of disk cache used (per user)
14. Modify your client desktop to your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Netscape (for example, Netscape SmartUpdate or Palm Tools if not needed or wanted).
  - c. Deleting entries in the **Start** menu if you don't want or need them.
15. Disable RealPlayer G2 SmartStart icon in the system tray, and remove/disable AOL Instant Messenger
16. From a DOS command prompt on the client, run:

```
net use h: /delete  
net use p: /delete
```
17. Empty the Recycle Bin and remove all files from Documents in the Start Menu.
18. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg NetScape /finish
```
19. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="netscape" os=nt4 lang=us shell=explorer  
AppDrive="P" AppPath="\\<server>\apps"
```
20. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user="AppTest"
```
21. Assign a home directory to the user in the command console on your server:

```
user modify user="AppTest" homedrive="h"  
homedir="\\<server>\tdmprfls\AppTest\w2k\us\Profiles"
```

22. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w2k lang=us shell=explorer  
user="AppTest"
```

23. Assign Netscape to the user in the command console on your server:

```
user assign application="netscape" os=w2k lang=us shell=explorer  
user="apptest"
```

24. Define a regular Windows 2000 client (for details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).

25. Copy the \tdmprfls\Netscape directory to \tdmprfls\AppTest\w2k\us\Profiles.

26. The NetScape application needs to write to the local hard disk. To allow Netscape to write to the local hard disk, change the PRTDSK2K.INI file in C:\TDM\TDMPRFLS. See Figure 167.

```
[MODE]  
Mode="ALLOW"  
; Mode can be "ALLOW" or "DENY"  
  
; Specify in "Unrestricted" key of [DRIVES] any local client drives  
; to which access will be fully unrestricted by the driver  
; [DRIVES]  
;Unrestricted="C"  
  
; Specify below directories that you wish to explicitly  
; "ALLOW" or "DENY" access to  
[DIRECTORIES]  
dir1="C:\DOCUMENTS AND SETTINGS\  
dir2="C:\WINNT\SYSTEM32\SPOOL\  
  
[FILES]  
file1="*.TMP"  
file2="C:\WINNT\NSREG.DAT"
```

Figure 167. Changes for Netscape in the PRTDSK2K.INI file

Finally, log on to the non-sandbox client as the non-sandbox user (AppTest) you created. After the first logon, all related files and updates for Netscape will be transferred to the user's client machine. A second logon is required after the update.

**Note**

On Windows 2000 client machines, running Netscape as a user without local administrator rights causes an error message on startup. Close and ignore the error message, and Netscape will run without any problems.

---

## **6.7 Installing Windows NT applications**

This section describes how to create application packages for Windows NT client operating systems.

The applications themselves do not have to be Windows NT applications; Windows 9x or Windows 3.xx or DOS applications can run under Windows NT too.

There are two different approaches to implementing applications in a Windows NT environment:

- **Client-based applications**  
In this installation, all application files and components are installed to the client hard drive.
- **Server-based applications**  
In this type of installation, most of the application files and components are installed to a server share and only a few components reside on the client. If there are user-specific configuration files, they will be stored in the user's home directory on the server.

### **6.7.1 Adobe Acrobat Reader as a Server-based Application**

For the implementation of Adobe Acrobat Reader, we suggest you implement it as a server-based application. The following description outlines this method:

1. Define a regular Windows NT client. (For details, see Chapter 4, "Defining machines" on page 103.)
2. Define a sandbox user and assign a sandbox desktop to the user in the CLI console on your server:

```

user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer
user="usersand"

```

3. On the sandbox client, logon as the sandbox user (usersand).
4. Create an apps share on the server and make sure that the sandbox user has write access to the share.

```
net share apps=d:\apps
```

5. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```

6. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

7. Start the installation of Adobe Acrobat reader. Install and enter P:\Adobe\Reader as the installation path.

8. Boot your sandbox client after the installation is complete.

9. Log on as the sandbox user (usersand).

10. At a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG Acro4rem /finish
```

This step creates the application package on your server.

11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="acro4rem" os=nt4 shell=explorer lang=us
apppath="\\<server>\APPS" Appdrive="P"
```

12. You can assign the application to other users now with the following command in the command console on your server:

```
user assign application="acro4rem" os=nt4 shell=explorer lang=us
user="User7"
```

Where User7 is the user to whom Adobe Acrobat Reader is assigned.

When you log on with, for example, User7, all related files and necessary updates for Adobe Acrobat Reader will be transferred to the user's client machine. A second logon or even a reboot is required after the update.



### 6.7.2 Adobe Acrobat Reader as a Local Application

For the implementation of Adobe Acrobat Reader, you can also implement it as a local application. The following description outlines this method:

1. Define a regular Windows NT client. (For details, see Chapter 4, "Defining machines" on page 103.)
2. Define a sandbox user and assign a sandbox desktop to the user in the CLI console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, logon as the sandbox user (usersand).
4. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

5. Start the installation of Adobe Acrobat reader. Install and enter C:\Adobe\Reader as the installation path.

6. Boot your sandbox client after the installation is complete.
7. Log on as the sandbox user (usersand).
8. At a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG Acro4rem /finish
```

This step creates the application package on your server.

9. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server. Remember, there are no Apppath and Appdrive parameters for a local application:

```
application define name="acro4rem" os=nt4 shell=explorer lang=us
```

10. You can assign the application to other users now with the following command in the command console on your server:

```
user assign application="acro4rem" os=nt4 shell=explorer lang=us
user="User7"
```

Where User7 is the user to whom Adobe Acrobat Reader is assigned.

When you log on with, for example, User7, all related files and necessary updates for Adobe Acrobat Reader will be transferred to the user's client machine. A second logon or even a reboot is required after the update.

### 6.7.3 Netscape 4.7 as a Server-based Application

The following steps describe the implementation of Netscape 4.7 as a server-based application that uses the home directory of each user for the caching and saving of individual settings:

1. Define a regular Windows NT client (for details, see Section 4.3, "Defining Windows NT client machines" on page 124).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
> user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer user=usersand
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Edit the C:\CRTPKG\CRTPKG.INI file and add the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_REG_VALUE]
RegVal6=HKLM, SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Pr
eferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Copyright,
RegVal7=HKLM, SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Pr
eferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Copyright,
```

5. Create an apps share on the server and make sure that the sandbox user has write access to the share.
6. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```

7. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

8. Start the installation of Netscape 4.7 from your source directory or CD.
9. Specify the directory P:\Netscape as your installation path.
10. Reboot the client machine.
11. From a DOS command prompt on the client, run:

```
net use h: \\<server>\tdmprfls /persistent:yes
```

12. Start Netscape. When you are asked to create a new profile, enter H:\Netscape\Users\default as the profile location.

All definitions for the user default in Netscape are now stored on your server in the directory \tdmprfls\NetScape\Users\default.

13. Modify all preferences within Netscape to match your environment. This could include:
  - a. Manual proxy definition
  - b. Manual socks definition
  - c. Location of the home page that you want your users to use
  - d. Size of disk cache used (per user)
14. Modify your client desktop to your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Netscape (for example, Netscape SmartUpdate or Palm Tools, if not needed or wanted).
  - c. Deleting entries in the **Start** menu if you don't want or need them.
15. From a DOS command prompt on the client, run:

```
net use h: /delete
net use p: /delete
```
16. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg NetScape /finish
```
17. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="ns47rem" os=nt4 lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```
18. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user="AppTest"
```
19. Assign a home directory to the user in the command console on your server:

```
user modify user="AppTest" homedrive="h"
homedir="\\<server>\tdmprfls\AppTest\nt4\us\Profiles"
```
20. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=nt4 lang=us shell=explorer
user="AppTest"
```
21. Assign Netscape to the user in the command console on your server:

```
user assign application="ns47rem" os=nt4 lang=us shell=explorer
user="apptest"
```

22. Define a regular Windows NT client (for details, see Section 4.3, “Defining Windows NT client machines” on page 124).
23. Copy the \tdmprfls\Netscape directory to \tdmprfls\AppData\nt4\us\Profiles.
24. The NetScape application needs to write to the local hard disk. To allow Netscape to write to the local hard disk, change the PRTDSKNT.INI file in C:\TDM\TDMPRFLS. See Section Figure 168., “Changes for Netscape in the PRTDSKNT.INI file” on page 304.

```
[MODE]
Mode="ALLOW"
; Mode can be "ALLOW" or "DENY"

; Specify in "Unrestricted" key of [DRIVES] any local client drives
; to which access will be fully unrestricted by the driver
; [DRIVES]
;Unrestricted="C"

; Specify below directories that you wish to explicitly
; "ALLOW" or "DENY" access to
[DIRECTORIES]
dir1="C:\WINNT\SYSTEM32\SPOOL\"

[FILES]
file1="*.TMP"
file2="C:\WINNT\NSREG.DAT"
```

*Figure 168. Changes for Netscape in the PRTDSKNT.INI file*

Finally, log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Netscape will be transferred to the user's client machine. A second logon is required after the update.

#### **6.7.4 Netscape 4.7 as a Local Application**

The following steps describe the implementation of Netscape 4.7 as a local application that uses the local drive for home directory for the caching and saving of individual settings:

1. Define a regular Windows NT client (for details, see Section 4.3, “Defining Windows NT client machines” on page 124).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer user=usersand
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Edit the C:\CRTPKG\CRTPKG.INI file and add the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_REG_VALUE]
RegVal6=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Pr
eferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Copyright,
RegVal7=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Pr
eferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Copyright,
```

5. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

6. Start the installation of Netscape 4.7 from your source directory or CD.
7. Specify the directory C:\Netscape as your installation path.
8. Reboot the client machine.

9. Start Netscape. When you are asked to create a new profile, enter C:\Netscape\Users\default as the profile location.

All definitions for the user default in Netscape are now stored on your local machine in the directory \NetScape\Users\default.

10. Modify all preferences within Netscape to match your environment. This could include:
  - a. Manual proxy definition
  - b. Manual socks definition
  - c. Location of the home page that you want your users to use
  - d. Size of disk cache used (per user)
11. Modify your client desktop to your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Netscape (for example, Netscape SmartUpdate or Palm Tools if not needed or wanted).

- c. Deleting entries in the **Start** menu if you don't want or need them.
12. From a DOS command prompt on the client, run:
- ```
c:\crtpkg\crtpkg NetScape /finish
```
13. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:
- ```
application define name="ns47rem" os=nt4 lang=us shell=explorer
```
14. Define a non-sandbox user (AppTest) in the command console on your server:
- ```
user define user="AppTest"
```
15. Assign a desktop to the user in the command console on your server:
- ```
user assign desktop=default os=nt4 lang=us shell=explorer  
user="AppTest"
```
16. Assign Netscape to the user in the command console on your server:
- ```
user assign application="ns47rem" os=nt4 lang=us shell=explorer  
user="apptest"
```
17. Define a regular Windows NT client (for details, see Section 4.3, "Defining Windows NT client machines" on page 124).

Finally, log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Netscape will be transferred to the user's client machine. A second logon is required after the update.

### 6.7.5 Microsoft Internet Explorer 5.0 as a Local Application

Microsoft Internet Explorer 5.0 cannot run as a server-based installation; therefore, all files must be installed to the client's hard drive.

At least Service Pack 3 is required on the client to run Microsoft Internet Explorer 5.0. See Section 7.2.1, "Windows NT 4.0" on page 373 for more details on how to apply a Service Pack.

The following steps must be executed to create the application package for Microsoft Internet Explorer 5.0:

1. Define a regular Windows NT client (For details, see Chapter 4, "Defining machines" on page 103.)
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"
```

```
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. At a DOS command prompt on the client, enter:  

```
C:\CRTPKG\CRTPKG /start
```
5. Start the installation of Microsoft Internet Explorer 5.0 from your source directory or CD.
6. Specify the local drive C: as your installation drive.
7. Reboot the client machine.
8. Start Microsoft Internet Explorer 5.0. Modify all preferences within the Internet options to match your environment. This could include:
  - a. Manual proxy definition
  - b. Security options or restrictions
  - c. Manual socks definition
  - d. Location of the home page you want your users to use
  - e. Size of disk cache used (per user)
9. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Microsoft Internet Explorer 5.0.
  - c. Deleting entries in the **Start** menu if you don't want or need them.
10. From a DOS command prompt on the client, run:  

```
c:\crtpkg\crtpkg IE5 /finish
```
11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:  

```
application define name="IE5loc" os=nt4 lang=us shell=explorer
```
12. Define a non-sandbox user (AppTest) in the command console on your server:  

```
user define user=AppTest
```
13. Assign a desktop to the user in the command console on your server:  

```
user assign desktop=default os=nt4 lang=us shell=explorer
user="AppTest"
```

14. Assign Microsoft Internet Explorer 5.0 to the user in the command console on your server:

```
user assign application="IE5loc" os=nt4 lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Microsoft Internet Explorer 5.0 will be transferred to the user's client machine. A second logon is required after the reboot of the client.

### 6.7.6 Paint Shop Pro 5 as a Server-Based Application

The following steps describe the installation of Paint Shop Pro as a server-based application:

1. Define a regular Windows NT client (for details, see Section 4.3, "Defining Windows NT client machines" on page 124).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Create an apps share on the server and make sure that the sandbox user has write access to the share.
5. From a DOS command prompt on the client, run:
6. From a DOS command prompt on the client, enter:

```
net use p: \\<server>\apps
```

```
C:\CRTPKG\CRTPKG /start
```

7. Install Paint Shop Pro 5 and enter P:\Paint Shop Pro 5 as your destination directory.

Select the components of Paint Shop you want to have installed, such as:

- a. Application files
- b. Animations shop application Files
- c. Picture tube brushes
- d. Paint Shop Pro sample and tutorial files
- e. Animation shop sample files



8. Reboot the client machine and log on again with your user ID (usersand).
  9. From a DOS command prompt on the client, run:  

```
net use p: \\<server>\apps
```
  10. Start Paint Shop Pro 5.
  11. Modify your preferences within Paint Shop Pro to match your environment.
  12. Modify your client desktop to your needs. This could include:
    - a. Adding new program links to the desktop.
    - b. Deleting program links and/or folders that were installed with Paint Shop Pro 5.
    - c. Deleting entries in the **Start** menu if you don't want or need them.
  13. From a DOS command prompt on the client, run:  

```
net use p: /delete
```
  14. From a DOS command prompt on the client, run:  

```
c:\crtpkg\crtpkg PaintShop /finish
```
  15. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:  

```
application define name="PaintShop" os=nt4 lang=us shell=explorer  
AppDrive="P" AppPath="\\<server>\apps"
```
  16. Define a non-sandbox user (AppTest) in the command console on your server:  

```
user define user="AppTest"
```
  17. Assign a desktop to the user in the command console on your server:  

```
user assign desktop=default os=nt4 lang=us shell=explorer  
user="AppTest"
```
  18. Assign Paint Shop Pro 5 to the user in the command console on your server:  

```
user assign application="PaintShop" os=nt4 lang=us shell=explorer  
user="apptest"
```
- Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Paint Shop Pro will be transferred to the user's client machine. A second logon is required after the update.

### 6.7.7 Lotus Notes 5.04 as a Local Application

At least Service Pack 3 is required on the client to run Lotus Notes 5.0x. See Section 7.2.1, "Windows NT 4.0" on page 373 for more details, on how to apply a Service Pack.

The following steps describe the installation of Lotus Notes 5.04 as a server-based application.

The following steps must be executed to create the application package for Lotus Notes:

1. Go to the Lotus Notes server.
2. Create an apps share on the server and make sure that the sandbox user has write access to the share.
3. Install the shared installation of Lotus Notes to your <apps>\notes directory on the server. You can perform this installation local on the server.
4. Define a regular Windows NT client. (For details, see Chapter 4, "Defining machines" on page 103.)
5. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer
user="usersand"
```

6. On the sandbox client, log on as the sandbox user (usersand).
7. From a DOS command prompt on the client, run:  

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```
8. From a DOS command prompt on the client, enter:  

```
C:\CRTPKG\CRTPKG /start
```
9. Install the Lotus Notes client by entering <apps>\notes\setup.exe at a DOS command prompt on the client.
10. Enter H:\Notes\Data as the Lotus Notes Data directory.
11. Reboot the client and log on with the same user ID.
12. Depending on your Lotus Notes infrastructure, there are many ways to set up the Notes desktop. Here are two possible steps you could take:

- a. If your Lotus Notes ID files are all named user.id, you can start creating the desktop for your users at this time. When finally assigning the application Lotus Notes to a user, you only have to copy the user's ID file into the home directory of the user.
  - b. If you have an existing Lotus Notes environment, you can copy the desktop, local address book, and ID file to the new home directory of the users. In this case, you should not start the Lotus Notes client before finishing the application package.
13. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Lotus Notes.
  - c. Deleting or moving entries in the **Start** menu.
14. Disconnect from your network shares by entering the following commands at a DOS command prompt on the client:

```
net use p: /delete
net use h: /delete
```
15. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg Notes /finish
```
16. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="Notes" os=nt4 lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```
17. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```
18. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=nt4 lang=us shell=explorer
user="AppTest"
```
19. Assign a home directory to the user in the command console on your server:

```
user modify user="AppTest" homedrive="H"
homedir="\\<server>\tdmprfls\AppTest\nt4\us\Profiles"
```
20. Assign Lotus Notes to the user in the command console on your server:

```
user assign application="Notes" os=nt4 lang=us shell=explorer
user="apptest"
```

21. Copy the \tdmprfls\Notes directory to \tdmprfls\AppTest\nt4\us\Profiles for this user or all relevant Users.
22. The Lotus Notes application needs to write to the file C:\WINNT\NOTES.INI. To allow Notes to write to the NOTES.INI, change the PRTDSKNT.INI file in C:\TDM\TDMPRFLS. See Section Figure 169., “Changes for Notes in the PRTDSKNT.INI file” on page 312.

```
[MODE]
Mode="ALLOW"
; Mode can be "ALLOW" or "DENY"

; Specify in "Unrestricted" key of [DRIVES] any local client drives
;   to which access will be fully unrestricted by the driver
; [DRIVES]
;Unrestricted="C"

; Specify below directories that you wish to explicitly
;   "ALLOW" or "DENY" access to
; [DIRECTORIES]
dir1="C:\WINNT\SYSTEM32\SPOOL\"

[FILES]
file1="*.TMP"
file2="C:\WINNT\NOTES.INI"
```

Figure 169. Changes for Notes in the PRTDSKNT.INI file

Log on to the non-sandbox client as the non-sandbox user you created (here AppTest). After the first logon, all related files and updates for the Lotus Notes client will be transferred to the user's client machine. A second logon is required after the update.

### 6.7.8 Lotus SmartSuite Millennium as a Server-Based Application

Service Pack 3 (at the very least) is required on the client to run Lotus SmartSuite Millennium. See Section 7.2.1, “Windows NT 4.0” on page 373 for more details on how to apply a Service Pack.

Lotus SmartSuite Millennium is also known as Version 9.5; we used the updated Version 9.5.2.

The following steps must be executed to create the application package for Lotus SmartSuite:

1. Create an apps share on the server and make sure that the sandbox user has write access to the share.
2. Define a regular Windows NT client. (For details, see Chapter 4, "Defining machines" on page 103.)
3. Define a regular Windows NT client. See Chapter 4, "Defining machines" on page 103 for an example of this process.
4. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer
user="usersand"
```
5. On the sandbox client, log on as the sandbox user (usersand).
6. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```
7. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```
8. Start the installation of the Lotus SmartSuite server part on your server. Click **Install to a File Server** and install to P:\Lotus.
9. Start the installation of the Lotus SmartSuite client part by entering P:\Lotus\Install.EXE at a DOS command prompt on the client.
10. Enter H:\Lotus as the installation directory and select the components of Lotus Smart Suite that you want to install.
11. Reboot the client and log on with the same user ID.
12. Start the selected and installed programs of Lotus SmartSuite and modify the preferences to fit your needs.
13. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Lotus Notes.
  - c. Deleting or moving entries in the Start menu.
14. Disconnect from your network shares by entering the following commands in a DOS command prompt on the client:

```
net use p: /delete
net use h: /delete
```

15. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg SS95rem/finish
```

16. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="SS95rem" os=nt4 lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```

17. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```

18. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=nt4 lang=us shell=explorer
user="AppTest"
```

19. Assign a home directory to the user in the command console on your server:

```
user modify user=AppTest homedrive="H"
homedir="\\<server>\tdmprfls\AppTest\nt4\us\Profiles"
```

20. Assign Lotus SmartSuite to the user in the command console on your server:

```
user assign application="SS95rem" os=nt4 lang=us shell=explorer
user="apptest"
```

21. Copy the \tdmprfls\Lotus directory to the \tdmprfls\AppTest\nt4\us\Profiles directory.

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Lotus Smart Suite will be transferred to the user's client machine. A second logon is required after the update.

### 6.7.9 Lotus SmartSuite Millennium as a Local Application

Service Pack 3 (at the very least) is required on the client to run Lotus Smartsuite Millennium. See Section 7.2.1, "Windows NT 4.0" on page 373 for more details on how to apply a Service Pack.

Lotus Smartsuite Millennium is also known as Version 9.5; we used the updated Version 9.5.2.

The following steps must be executed to create the application package for Lotus SmartSuite:

1. Define a regular Windows NT client. (For details, see Chapter 4, "Defining machines" on page 103.)

2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"  
user modify user="usersand" password="" passwordexpiration=NEVER  
groupmember define user="usersand" group="Domain Admins"  
user assign desktop=sandbox os=nt4 lang=us shell=explorer  
user="usersand"
```

3. On the sandbox client, log on as the sandbox user (usersand).

4. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

5. Start the installation of the Lotus SmartSuite client part on the client.

6. Enter `c:\Lotus\Smartsuite` as the installation directory and select the components of Lotus Smart Suite that you want to install.

7. Reboot the client and log on with the same user ID.

8. Start the selected and installed programs of Lotus SmartSuite and modify the preferences to fit your needs.

9. Modify your client desktop to fit your needs. This could include:

- a. Adding new program links to the desktop.
- b. Deleting program links and/or folders that were installed with Lotus Notes.
- c. Deleting or moving entries in the **Start** menu.

10. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg SS95loc /finish
```

11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="SS95rem" os=nt4 lang=us shell=explorer
```

12. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```

13. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=nt4 lang=us shell=explorer
user="AppTest"
```

14. Assign a home directory to the user in the command console on your server:

```
user modify user="AppTest" homedrive="H"
homedir="//<server>\tdmprfls\AppTest\nt4\us\Profiles"
```

15. Assign Lotus SmartSuite to the user in the command console on your server:

```
user assign application="SS95loc" os=nt4 lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Lotus Smart Suite will be transferred to the user's client machine. A second logon is required after the update.

### 6.7.10 Microsoft Office 2000

Service Pack 3 (at the very least) is required on the client to run Microsoft Office 2000. See Section 7.2.1, "Windows NT 4.0" on page 373 for more detail on how to apply a Service Pack. The following steps must be executed to create the application package for Microsoft Office 2000:

1. Install the Server part of Microsoft Office 2000 to your Server (`setup /a data1.msi`). Share the installation directory on the server as "OFFICE" and make sure that the sandbox user has write access to the share.

```
net share office=d:\office2000
```

2. Define a regular Windows NT client with at least 700 MB of free space after Base Installation (for details, see Chapter 4, "Defining machines" on page 103).
3. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=nt4 lang=us shell=explorer
user="usersand"
```

4. On the sandbox client, log on as the sandbox user (usersand).
5. Edit the file C:\CRTPKG\CRTPKG.INI and add or modify the following lines in the relevant section (noted in brackets):



```
[EXCLUDE_FILE]
File14=c:\windows\tasks\sa.dat

[EXCLUDE_REG_KEY]
;RegKey4=HKLM,Software\Microsoft\Windows\CurrentVersion\Reliability

[EXCLUDE_REG_TREE]
RegTree3=HKCU,Software\Microsoft\Windows\CurrentVersion\Uninstall
;RegTree4=HKCU,Software\Lotus\Components\Fills\is\
;RegTree5=HKCU,Software\Lotus\Components\Fills\ru\
;RegTree16=HKLM,Software\Microsoft\Windows NT\CurrentVersion\perflib

[EXCLUDE_REG_VALUE]
RegVal5=HKLM,System\CurrentControlSet\Control\SessionManager,PendingFileRenameOperations
RegVal6=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Copyright
RegVal7=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Copyright
```

6. From a DOS command prompt on the client, run:
 

```
net use n: \\<server>\office /persistent:yes
```
7. From a DOS command prompt on the client, enter:
 

```
C:\CRTPKG\CRTPKG /start
```
8. Start the installation of the Microsoft Office by entering `N:\Setup.EXE` at a DOS command prompt on the client.
9. In the installer, select Custom installation and in the Selecting Features window click **Run all from Network**.
10. Reboot the client and logon with the same user ID.
11. When the install continues, enter `\\<server>\office` for the install files location.
12. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Microsoft Office 2000.
  - c. Deleting or moving entries in the **Start** menu.
13. Disconnect from your network shares by entering the following commands from a DOS command prompt on the client:
 

```
net use n: /delete
```

14. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg Office /finish
```

15. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="Office" os=nt4 lang=us shell=explorer  
AppDrive="N" AppPath="\\<server>\office"
```

16. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```

17. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=nt4 lang=us shell=explorer user="AppTest"
```

18. Assign Microsoft Office 2000 to the user in the command console on your server:

```
user assign application="Office" os=nt4 lang=us shell=explorer  
user="apptest"
```

Log on to the non-sandbox client (with at least a 700 MB drive) as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Microsoft Office 2000 will be transferred to the user's client machine. A second logon is required after the update.

---

## 6.8 Installing Windows 98 applications

Creating application packages for Windows 98 clients is similar to the process for Windows NT 4.0 Workstation clients.

### 6.8.1 Microsoft Internet Explorer

Microsoft Internet Explorer is already included with Windows 98. Therefore, there is no need to install it.

### 6.8.2 Adobe Acrobat Reader as a Server-based Application

For the implementation of Adobe Acrobat Reader, we suggest you implement it as a server-based application. The following description outlines this method:

1. Define a regular Windows 98 client. (For details, see Chapter 4, "Defining machines" on page 103.)
2. Define a sandbox user and assign a sandbox desktop to the user in the CLI console on your server:

```

user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer
user="usersand"

```

3. On the sandbox client, logon as the sandbox user (usersand).
4. Create an apps share on the server and make sure that the sandbox user has write access to the share.

```
net share apps=d:\apps
```

5. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```

6. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

7. Start the installation of Adobe Acrobat reader. Install and enter P:\Adobe\Reader as the installation path.

8. Boot your sandbox client after the installation is complete.

9. Log on as the sandbox user (usersand).

10. At a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG Acro4rem /finish
```

This step creates the application package on your server.

11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="acro4rem" os=w98 shell=explorer lang=us
apppath="\\<server>\APPS" Appdrive="P"
```

12. You can assign the application to other users now with the following command in the command console on your server:

```
user assign application="acro4rem" os=w98 shell=explorer lang=us
user="User7"
```

Where User7 is the user to whom Adobe Acrobat Reader is assigned.

When you log on with, for example, User7, all related files and necessary updates for Adobe Acrobat Reader will be transferred to the user's client machine. A second logon or even a reboot is required after the update.

### 6.8.3 Adobe Acrobat Reader as a Local Application

For the implementation of Adobe Acrobat Reader, you can also implement it as a local application. The following description outlines this method:

1. Define a regular Windows 98 client. (For details, see Chapter 4, "Defining machines" on page 103.)
2. Define a sandbox user and assign a sandbox desktop to the user in the CLI console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, logon as the sandbox user (usersand).
4. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

5. Start the installation of Adobe Acrobat reader. Install and enter C:\Adobe\Reader as the installation path.

6. Boot your sandbox client after the installation is complete.
7. Log on as the sandbox user (here usersand).
8. At a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG Acro4rem /finish
```

This step creates the application package on your server.

9. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server. Remember, there are no Apppath and Appdrive parameters for a local application:

```
application define name="acro4rem" os=w98 shell=explorer lang=us
```

10. You can assign the application to other users now with the following command in the command console on your server:

```
user assign application="acro4rem" os=w98 shell=explorer lang=us
user="User7"
```

Where User7 is the user to whom Adobe Acrobat Reader is assigned.

When you log on with, for example, User7, all related files and necessary updates for Adobe Acrobat Reader will be transferred to the user's client machine. A second logon or even a reboot is required after the update.

#### 6.8.4 Netscape 4.7 as a Server-based Application

The following steps describe the implementation of Netscape 4.7 as a server-based application that uses the home directory of each user for caching and saving the individual settings:

1. Define a regular Windows 98 client (for details, see Section 4.3, “Defining Windows NT client machines” on page 124).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer user=usersand
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Edit the C:\CRTPKG\CRTPKG.INI file and add the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_FILE]
File14=c:\windows\tasks\sa.dat

[EXCLUDE_REG_TREE]
RegTree1=HKCU,Software\Microsoft\Windows\CurrentVersion\Explorer
RegTree3=HKCU,Software\Microsoft\Windows\CurrentVersion\Uninstall
RegTree17=HKLM,Software\Microsoft\Windows\CurrentVersion\Uninstall

[EXCLUDE_REG_VALUE]
RegVal5=HKLM,System\CurrentControlSet\Control\SessionManager,PendingFileRenameOperations
RegVal6=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Copyright,
RegVal7=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Copyright,
```

5. Create an apps share on the server and make sure that the sandbox user has write access to the share.
6. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```

7. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

8. Start the installation of Netscape 4.7 from your source directory or CD.
9. Specify the directory P:\Netscape as your installation path.

10. Reboot the client machine.
11. From a DOS command prompt on the client, run:  

```
net use h: \\<server>\tdmprfls /persistent:yes
```
12. Start Netscape. When you are asked to create a new profile, enter  
H:\Netscape\Users\default as the profile location.  
All definitions for the user default in Netscape are now stored on your  
server in the directory \tdmprfls\NetScape\Users\default.
13. Modify all preferences within Netscape to match your environment. This  
could include:
  - a. Manual proxy definition
  - b. Manual socks definition
  - c. Location of the home page that you want your users to use
  - d. Size of disk cache used (per user)
14. Modify your client desktop to your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Netscape  
(for example, Netscape SmartUpdate or Palm Tools if not needed or  
wanted).
  - c. Deleting entries in the Start menu if you don't want or need them.
15. From a DOS command prompt on the client, run:  

```
net use h: /delete
net use p: /delete
```
16. From a DOS command prompt on the client, run:  

```
c:\crtpkg\crtpkg NetScape /finish
```
17. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the  
following command in the command console on your server:  

```
application define name="ns47rem" os=w98 lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```
18. Define a non-sandbox user (AppTest) in the command console on your  
server:  

```
user define user="AppTest"
```
19. Assign a home directory to the user in the command console on your  
server:  

```
user modify user="AppTest" homedrive="h"
homedir="\\<server>\tdmprfls\AppTest\w98\us\Profiles"
```

20. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w98 lang=us shell=explorer
user="AppTest"
```

21. Assign Netscape to the user in the command console on your server:

```
user assign application="ns47rem" os=w98 lang=us shell=explorer
user="apptest"
```

22. Define a regular Windows 98 client (for details, see Section 4.3, "Defining Windows NT client machines" on page 124).

23. Copy the \tdmprfls\Netscape directory to  
    \tdmprfls\AppDataTest\w98\us\Profiles.

Finally, log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Netscape will be transferred to the user's client machine. A second logon is required after the update.

### 6.8.5 Netscape 4.7 as a Local Application

The following steps describe the implementation of Netscape 4.7 as a local application that uses the local drive for home directory of caching and saving the individual settings:

1. Define a regular Windows 98 client (for details, see Section 4.3, "Defining Windows NT client machines" on page 124).
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer user=usersand
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. Edit the C:\CRTPKG\CRTPKG.INI file and add the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_FILE]
File14=c:\windows\tasks\sa.dat

[EXCLUDE_REG_TREE]
RegTree1=HKCU,Software\Microsoft\Windows\CurrentVersion\Explorer
RegTree3=HKCU,Software\Microsoft\Windows\CurrentVersion\Uninstall
RegTree17=HKLM,Software\Microsoft\Windows\CurrentVersion\Uninstall

[EXCLUDE_REG_VALUE]
```

```
RegVal5=HKLM,System\CurrentControlSet\Control\SessionManager,PendingFileRenameOperations
RegVal6=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Copyright,
RegVal7=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Copyright,
```

5. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

6. Start the installation of Netscape 4.7 from your source directory or CD.

7. Specify the directory C:\Netscape as your installation path.

8. Reboot the client machine.

9. Start Netscape. When you are asked to create a new profile, enter

C:\Netscape\Users\default as the profile location.

All definitions for the user default in Netscape are now stored on your local machine in the directory \NetScape\Users\default.

10. Modify all preferences within Netscape to match your environment. This could include:

- a. Manual proxy definition
- b. Manual socks definition
- c. Location of the home page that you want your users to use
- d. Size of disk cache used (per user)

11. Modify your client desktop to your needs. This could include:

- a. Adding new program links to the desktop.
- b. Deleting program links and/or folders that were installed with Netscape (for example, Netscape SmartUpdate or Palm Tools if not needed or wanted).
- c. Deleting entries in the **Start** menu if you don't want or need them.

12. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg NetScape /finish
```

13. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="ns47rem" os=w98 lang=us shell=explorer
```

14. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user="AppTest"
```



15. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w98 lang=us shell=explorer
user="AppTest"
```

16. Assign Netscape to the user in the command console on your server:

```
user assign application="ns47rem" os=w98 lang=us shell=explorer
user="apptest"
```

17. Define a regular Windows 98 client (for details, see Section 4.3, "Defining Windows NT client machines" on page 124).

Finally, log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Netscape will be transferred to the user's client machine. A second logon is required after the update.

### 6.8.6 Paint Shop Pro 5 as a Server-Based Application

The following steps describe the installation of Paint Shop Pro as a server-based application:

1. Define a regular Windows 98 client (for details, see Section 4.3, "Defining Windows NT client machines" on page 124).

2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, log on as the sandbox user (usersand).

4. Create an apps share on the server and make sure that the sandbox user has write access to the share.

5. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```

6. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

7. Install Paint Shop Pro 5 and enter `P:\Paint Shop Pro 5` as your destination directory. Select the components of Paint Shop you want to have installed, such as:
  - a. Application files
  - b. Animations shop application Files
  - c. Picture tube brushes
  - d. Paint Shop Pro sample and tutorial files
  - e. Animation shop sample files
8. Reboot the client machine and log on again with your user ID (usersand).
9. From a DOS command prompt on the client, run:

```
net use p: \\<server>\apps
```
10. Start Paint Shop Pro 5.
11. Modify your preferences within Paint Shop Pro to match your environment.
12. Modify your client desktop to your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Paint Shop Pro 5.
  - c. Deleting entries in the **Start** menu if you don't want or need them.
13. From a DOS command prompt on the client, run:

```
net use p: /delete
```
14. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg PaintShop /finish
```
15. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="PaintShop" os=w98 lang=us shell=explorer  
AppDrive="P" AppPath="\\<server>\apps"
```
16. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user="AppTest"
```
17. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w98 lang=us shell=explorer  
user="AppTest"
```
18. Assign Paint Shop Pro 5 to the user in the command console on your server:

```
user assign application="PaintShop" os=w98 lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for Paint Shop Pro will be transferred to the user's client machine. A second logon is required after the update.

### 6.8.7 Lotus Notes 5.04 as a Local Application

The following steps describe the installation of Lotus Notes 5.04 as a server-based application.

The following steps must be executed to create the application package for Lotus Notes:

1. Go to the Lotus Notes server.
2. Create an apps share on the server and make sure that the sandbox user has write access to the share.
3. Install the shared installation of Lotus Notes to your <apps>\notes directory on the Server. You can perform this installation local on the Server.
4. Define a regular Windows 98 client. (For details, see Chapter 4, "Defining machines" on page 103.)
5. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer
user="usersand"
```

6. On the sandbox client, log on as the sandbox user (usersand).
7. From a DOS command prompt on the client, run:  

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```
8. From a DOS command prompt on the client, enter:  

```
C:\CRTPKG\CRTPKG /start
```
9. Install the Lotus Notes client by entering <apps>\notes\setup.exe at a DOS command prompt on the client.
10. Enter H:\Notes\Data as the Lotus Notes Data directory.

11. Reboot the client and log on with the same user ID.
12. Depending on your Lotus Notes infrastructure, there are many ways to set up the Notes desktop. Here are two possible steps you could take:
  - a. If your Lotus Notes ID files are all named user.id, you can start creating the desktop for your users at this time. When finally assigning the application Lotus Notes to a user, you only have to copy the user's ID file into the home directory of the user.
  - b. If you have an existing Lotus Notes environment, you can copy the desktop, local address book, and ID file to the new home directory of the users. In this case, you should not start the Lotus Notes client before finishing the application package.
13. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Lotus Notes.
  - c. Deleting or moving entries in the Start menu.
14. Disconnect from your network shares by entering the following commands at a DOS command prompt on the client:

```
net use p: /delete
net use h: /delete
```
15. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg Notes /finish
```
16. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="Notes" os=w98 lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```
17. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```
18. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w98 lang=us shell=explorer
user="AppTest"
```
19. Assign a home directory to the user in the command console on your server:

```
user modify user="AppTest" homedrive="H"
homedir="\\<server>\tdmprfls\AppTest\w98\us\Profiles"
```

20. Assign Lotus Notes to the user in the command console on your server:

```
user assign application="Notes" os=w98 lang=us shell=explorer
user="apptest"
```

21. Copy the \tdmprfls\Notes directory to \tdmprfls\AppTest\w98\us\Profiles for this user or all relevant users.

22. The Lotus Notes application needs to write to the file C:\WINNT\NOTES.INI. To allow Notes to write to the NOTES.INI, change the PRTDSK9X.INI file in C:\TDM\TDMPRFLS. See Section Figure 169., "Changes for Notes in the PRTDSKNT.INI file" on page 312.

```
[MODE]
Mode="ALLOW"
; Mode can be "ALLOW" or "DENY"

; Specify in "Unrestricted" key of [DRIVES] any local client drives
; to which access will be fully unrestricted by the driver
; [DRIVES]
;Unrestricted="C"

; Specify below directories that you wish to explicitly
; "ALLOW" or "DENY" access to
[DIRECTORIES]
dir1="C:\WINDOWS\PROFILES\"
dir2="C:\WINDOWS\TEMPORARY INTERNET FILES\"
dir3="C:\WINDOWS\TEMP\"
dir4="C:\WINDOWS\SPOOL\"

[FILES]
file1="C:\WINDOWS\WIN.INI"
file2="C:\WINDOWS\SYSTEM.INI"
file3="C:\WINDOWS\*.DAT"
file4="C:\WINDOWS\SYBCKUP\*.CAB"
file5="C:\WINDOWS\NOTES.INI"
```

Figure 170. Changes for Notes in the PRTDSKNT.INI file

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Lotus Notes client will be transferred to the user's client machine. A second logon is required after the update.

### 6.8.8 Lotus SmartSuite Millennium as a Server-Based Application

Lotus SmartSuite Millennium is also known as Version 9.5; we used the updated Version 9.5.2.

The following steps must be executed to create the application package for Lotus SmartSuite:

1. Create an apps share on the server and make sure that the sandbox user has write access to the share.
2. Define a regular Windows 98 client. (For details, see Chapter 4, "Defining machines" on page 103.)
3. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer
user="usersand"
```

4. On the sandbox client, log on as the sandbox user (usersand).
5. From a DOS command prompt on the client, run:  

```
net use p: \\<server>\apps /persistent:yes
net use h: \\<server>\tdmprfls /persistent:yes
```
6. From a DOS command prompt on the client, enter:  

```
C:\CRTPKG\CRTPKG /start
```
7. Start the installation of the Lotus SmartSuite server part on your Server. Click **Install to a File Server** and install to P:\Lotus.
8. Start the installation of the Lotus SmartSuite client part by entering P:\Lotus\Install.EXE at a DOS command prompt on the client.
9. Enter H:\Lotus as the installation directory and select the components of Lotus Smart Suite that you want to install.
10. Reboot the client and log on with the same user ID.
11. Start the selected and installed programs of Lotus SmartSuite and modify the preferences to fit your needs.
12. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Lotus Notes.

- c. Deleting or moving entries in the **Start** menu.
13. Disconnect from your network shares by entering the following commands in a DOS command prompt on the client:
- ```
net use p: /delete
net use h: /delete
```
14. From a DOS command prompt on the client, run:
- ```
c:\crtpkg\crtpkg SS95rem/finish
```
15. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:
- ```
application define name="SS95rem" os=w98 lang=us shell=explorer
AppDrive="P" AppPath="\\<server>\apps"
```
16. Define a non-sandbox user (AppTest) in the command console on your server:
- ```
user define user=AppTest
```
17. Assign a desktop to the user in the command console on your server:
- ```
user assign desktop=default os=w98 lang=us shell=explorer
user="AppTest"
```
18. Assign a home directory to the user in the command console on your server:
- ```
user modify user=AppTest homedrive="H"
homedir="\\<server>\tdmprfls\AppTest\w98\us\Profiles"
```
19. Assign Lotus SmartSuite to the user in the command console on your server:
- ```
user assign application="SS95rem" os=w98 lang=us shell=explorer
user="apptest"
```
20. Copy the \tdmprfls\Lotus directory to the  
\\tdmprfls\AppTest\w98\us\Profiles directory.

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Lotus Smart Suite will be transferred to the user's client machine. A second logon is required after the update.

### 6.8.9 Lotus SmartSuite Millennium as a Local Application

Lotus Smartsuite Millennium is also known as Version 9.5; we used the updated Version 9.5.2.

The following steps must be executed to create the application package for Lotus SmartSuite:

1. Define a regular Windows 98 client. (For details, see Chapter 4, “Defining machines” on page 103.)
2. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user="usersand"  
user modify user="usersand" password="" passwordexpiration=NEVER  
groupmember define user="usersand" group="Domain Admins"  
user assign desktop=sandbox os=w98 lang=us shell=explorer  
user="usersand"
```

3. On the sandbox client, log on as the sandbox user (usersand).
4. From a DOS command prompt on the client, enter:  
  
C:\CRTPKG\CRTPKG /start
5. Start the installation of the Lotus SmartSuite client part on the client.
6. Enter c:\Lotus\Smartsuite as the installation directory and select the components of Lotus Smart Suite that you want to install.
7. Reboot the client and log on with the same user ID.
8. Start the selected and installed programs of Lotus SmartSuite and modify the preferences to fit your needs.
9. Modify your client desktop to fit your needs. This could include:
  - a. Adding new program links to the desktop.
  - b. Deleting program links and/or folders that were installed with Lotus Notes.
  - c. Deleting or moving entries in the Start menu.
10. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg SS95loc /finish
```

11. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="SS95rem" os=w98 lang=us shell=explorer
```

12. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```

13. Assign a desktop to the user in the command console on your server:



```
user assign desktop=default os=w98 lang=us shell=explorer
user="AppTest"
```

14. Assign a home directory to the user in the command console on your server:

```
user modify user="AppTest" homedrive="H"
homedir="\\<server>\tdmprfls\AppTest\w98\us\Profiles"
```

15. Assign Lotus SmartSuite to the user in the command console on your server:

```
user assign application="SS95loc" os=w98 lang=us shell=explorer
user="apptest"
```

Log on to the non-sandbox client as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Lotus Smart Suite will be transferred to the user's client machine. A second logon is required after the update.

### 6.8.10 Microsoft Office 2000

The following steps must be executed to create the application package for Microsoft Office 2000:

1. Install the Server part of Microsoft Office 2000 to your Server (`setup /a data1.msi`). Share the installation directory on the server as "OFFICE" and make sure that the sandbox user has write access to the share.

```
net share office=d:\office2000
```

2. Define a regular Windows 98 client with at least 700 MB of free space after Base Installation (for details, see Chapter 4, "Defining machines" on page 103).
3. Define a sandbox user and assign a sandbox desktop to the user in the command console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w98 lang=us shell=explorer
user="usersand"
```

4. On the sandbox client, log on as the sandbox user (usersand).
5. Edit the file `C:\CRTPKG\CRTPKG.INI` and add or modify the following lines in the relevant section (noted in brackets):

```
[EXCLUDE_FILE]
File14=c:\windows\tasks\sa.dat
```

```
[EXCLUDE_REG_KEY]
;RegKey4=HKLM,Software\Microsoft\Windows\CurrentVersion\Reliability
```

```
[EXCLUDE_REG_TREE]
RegTree3=HKCU,Software\Microsoft\Windows\CurrentVersion\Uninstall
;RegTree4=HKCU,Software\Lotus\Components\Fills\is\
;RegTree5=HKCU,Software\Lotus\Components\Fills\ru\
;RegTree16=HKLM,Software\Microsoft\Windows NT\CurrentVersion\perflib
```

```
[EXCLUDE_REG_VALUE]
RegVal5=HKLM,System\CurrentControlSet\Control\SessionManager,PendingFileRenameOperations
RegVal6=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rare3260.dll-0\Coyright
RegVal7=HKLM,SOFTWARE\Classes\Software\RealNetworks\RealMediaSDK\6.0\Preferences\PluginHandlerData\PluginInfo\rvre3260.dll-0\Coyright
```

6. From a DOS command prompt on the client, run:

```
net use n: \\<server>\office /persistent:yes
```

7. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

8. Start the installation of the Microsoft Office by entering N:\Setup.EXE at a DOS command prompt on the client.

9. In the installer, select Custom installation and in the Selecting Features window click **Run all from Network**.

10. Reboot the client and logon with the same user ID.

11. When the install continues, enter \\<server>\office for the install files location.

12. Modify your client desktop to fit your needs. This could include:

- a. Adding new program links to the desktop.
- b. Deleting program links and/or folders that were installed with Microsoft Office 2000.
- c. Deleting or moving entries in the **Start** menu.

13. Disconnect from your network shares by entering the following commands from a DOS command prompt on the client:

```
net use n: /delete
```

14. From a DOS command prompt on the client, run:

```
c:\crtpkg\crtpkg Office /finish
```

15. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server:

```
application define name="Office" os=w98 lang=us shell=explorer  
AppDrive="N" AppPath="\\<server>\office"
```

16. Define a non-sandbox user (AppTest) in the command console on your server:

```
user define user=AppTest
```

17. Assign a desktop to the user in the command console on your server:

```
user assign desktop=default os=w98 lang=us shell=explorer  
user="AppTest"
```

18. Assign Microsoft Office 2000 to the user in the command console on your server:

```
user assign application="Office" os=w98 lang=us shell=explorer  
user="apptest"
```

Log on to the non-sandbox client (with at least a 700 MB drive) as the non-sandbox user (AppTest). After the first logon, all related files and updates for the Microsoft Office 2000 will be transferred to the user's client machine. A second logon is required after the update

---

## 6.9 Installing OS/2 applications

The following sections describe the installation and management of OS/2 applications. There are some differences between OS/2 and Windows when managing applications, because the OS/2 client does a true network boot instead of a remote installation with local boot. The utilities for managing application packages, however, remain the same.

### 6.9.1 The OS/2 sandbox

With Windows NT and Windows 98 clients, we created a sandbox by assigning a sandbox desktop to a user ID with administrator authority. This user ID could log on to any client machine and create an application package. OS/2 sandbox management is different in that we must create an OS/2 sandbox machine first and log on to this machine with an administrator user ID and a default desktop. Creating a sandbox machine is identical to creating an OS/2 client machine, except that we add the additional parameter `SANDBOX='YES'` to the `machine define` command. Figure 171 on page 336 shows the definition of an OS/2 token-ring sandbox machine in the IBM WorkSpace On-Demand 3.0.1 console CLI.

```
Machine define OS='os2' NETADAPTER='IBM Token-Ring PCI Family Adapter
(IBMTRP.OS2)' LANG='us' IMAGENAME='bb20' VIDEO='s3t64vp'
RESOLUTION='1024x768x256@43' MONITOR='default' MOUSE='PS/2 Mouse'
SANDBOX='YES' DHCP='YES' CACHE='NO' Name='machine2' MAC='000629844293'
SERVER='atlas2'
```

Figure 171. Defining an OS/2 sandbox machine

The JavaScript file `machines.js` encapsulates the definition of an OS/2 sandbox using the function `CreateOS2Sandboxt (name, MAC, server)`.

The OS/2 sandbox differs slightly from normal OS/2 client machines. Remember that all OS/2 client machines boot from the same OS/2 image installed on the IBM WorkSpace On-Demand 3.0.1 deployment server. Because the sandbox is used to capture differences in the operating system when an application is installed and configured, we need to ensure that these differences for each application are preserved independently of the default operating system client image. Therefore, when you create an OS/2 sandbox machine, a duplicate operating system tree is created in the RPLFILES alias on your deployment server, as shown in Figure 172.

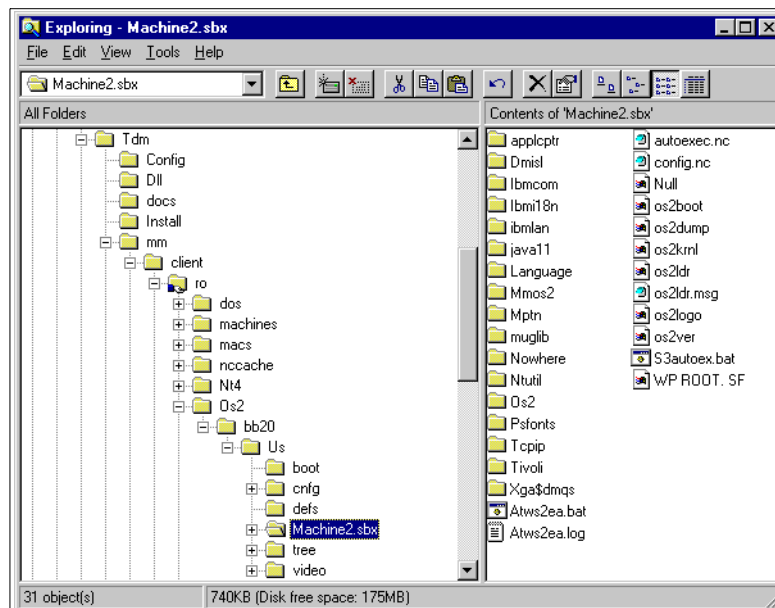


Figure 172. The OS/2 sandbox duplicate operating system tree

When you define an OS/2 sandbox, you will notice that it takes much longer for IBM WorkSpace On-Demand 3.0.1 to perform this definition than to define a normal OS/2 client machine. The reason for the length of time is that IBM WorkSpace On-Demand 3.0.1 has to copy the entire OS/2 operating system tree to the <machine name>.sbx directory.

This duplicate tree allows changes to be captured safely, without affecting the default operating system image used for normal clients. The machine FIT file for the OS/2 sandbox reflects this duplicate OS/2 image as well, as shown in Figure 173.

```

\\ATLAS2\RPLFILES
; The first line of this file MUST be UNC name

; Per-workstation read-only configuration files.
Z:\CONFIG.SYS           MACHINES\MACHINE2\OS2\BB20\US\CONFIG.SYS
Z:\STARTUP.CMD          MACHINES\MACHINE2\OS2\BB20\US\STARTUP.CMD
Z:\IBMLAN\IBMLAN.INI
MACHINES\MACHINE2\OS2\BB20\US\IBMLAN\IBMLAN.INI

; These OS/2 files must be writeable.
Z:\OS2\BOOT\ARCHBASE.*
\\ATLAS2\WRKFILES\MACHINES\MACHINE2\OS2\BB20\US\OS2

; Shared OS2 and network files are read-only.
Z:\OS2                  OS2\BB20\US\MACHINE2.SBX\OS2
Z:\OS2\DLL\IBMNULL.DRV
OS2\BB20\US\MACHINE2.SBX\OS2\DLL\IBMNULL\IBMNULL.DRV
Z:\XGA$DMQS             OS2\BB20\US\MACHINE2.SBX\XGA$DMQS
Z:\PSFONTS              OS2\BB20\US\MACHINE2.SBX\PSFONTS
Z:\OS2KRNL              OS2\BB20\US\MACHINE2.SBX\OS2KRNL
Z:\OS2LOGO              OS2\BB20\US\MACHINE2.SBX\OS2LOGO
Z:\OS2VER               OS2\BB20\US\MACHINE2.SBX\OS2VER
Z:\OS2LDR.MSG           OS2\BB20\US\MACHINE2.SBX\OS2LDR.MSG
Z:\OSO001.MSG           OS2\BB20\US\MACHINE2.SBX\OS2\SYSTEM\OSO001.MSG
Z:\IBMVESA              OS2\BB20\US\MACHINE2.SBX\IBMVESA
Z:\LANGUAGE             OS2\BB20\US\MACHINE2.SBX\LANGUAGE
Z:\MMOS2                OS2\BB20\US\MACHINE2.SBX\MMOS2
Z:\OS2INST              OS2INST

```

Figure 173. The operating system section of the OS/2 sandbox FIT

After defining the sandbox, we need to create an administrator user ID and assign the OS/2 default desktop. Use the following commands:

```
user define user=os2sand type=USER
```

```
user modify user="os2sand" password="" passwordexpiration=NEVER
user assign user="os2sand" desktop=default os=os2 shell=pmsHELL lang=us
groupmember define group="Domain Admins" user="os2sand"
```

Two further steps are necessary to give the administrator full control over the sandbox machine. First, we need to give the administrator access to a command prompt on the OS/2 sandbox machine in order to install applications. We can do this by copying the OS/2 command prompt program, cmd.exe, to the TDMPKGS alias and then defining a simple application for the command prompt and assigning it to the sandbox user. In a DOS command prompt on your IBM WorkSpace On-Demand 3.0.1 server, type:

```
copy c:\tdm\mm\client\ro\os2\bb20\us\tree\os2\cmd.exe c:\tdm\tdmpkgs
```

Then, in the IBM WorkSpace On-Demand 3.0.1 console CLI, enter the following commands:

```
application define name=prompt os=os2 lang=us shell=pmsHELL appdrive="p"
apppath="\\<server>\tdmpkgs" command="cmd.exe" icontitle="Command Prompt"
```

```
user assign application=prompt user="os2sand" os=os2 lang=us shell=pmsHELL
```

Second, we need to edit the config.sys file for our sandbox in order to disable the protdisk driver. If this driver is enabled, we will not have write access to the C: drive of the OS/2 sandbox client. For the sandbox client defined in Figure 171 on page 336, we can find its config.sys file in the RPLFILES alias in the machines\machine2\os2\bb20\us directory. Edit the config.sys in this directory and change the line:

```
DEVICE=Z:\OS2\BOOT\PROTDISK.SYS
```

to:

```
REM DEVICE=Z:\OS2\BOOT\PROTDISK.SYS
```

You are now ready to boot the OS/2 sandbox, log on, and install an application.

### 6.9.2 The OS/2 application FIT file

When you install an application through the OS/2 sandbox, you will probably find that some application files will differ per user. For example, the Netscape profile information contains settings, such as home page and bookmarks, that will be different for each user. Also, general INI file settings, such as fonts and colors, should be customizable per user. When you install an application on the OS/2 sandbox, these files will often be placed somewhere on the Z: drive for the sandbox or, in some cases, on the mapped network drive where you installed the application. Each user should get their own copy of these files.

To enable this, the application FIT file was introduced in IBM WorkSpace On-Demand 2.0. Application FIT files have the same syntax as those for users and applications, but are loaded into memory only when the application's primary executable is loaded, and are discarded when the application is terminated.

Application FIT files were introduced in order to circumvent the 64 K limit imposed on FIT entries in memory. In older RIPL and IBM WorkSpace On-Demand environments, this limit could cause problems, because all FIT information was loaded at machine boot or user logon. Using application FIT files reduces the amount of FIT information in memory to only the machine FIT entries, user FIT entries and application FIT entries for those applications that are currently active.

The application FIT file resides in the same directory as the primary executable of the application. You can create up to two FIT files per application. These are, by default, called APPPRE.FIT and APPPOST.FIT:

- APPPRE.FIT is loaded into memory before the entries from the user FIT file. This means that the FIT entries in APPPRE.FIT will override any conflicting entries in the user FIT.
- APPPOST.FIT is loaded into memory after the entries from the user FIT. Therefore, the user FIT entries will override any conflicting entries in the application FIT file.

Application FIT files, like user FIT files, allow you to use variables to make the FIT file more generic. This way, for example, one single application FIT file can be used to handle the application customizations for all users of that application, since a variable is used to represent the user ID.

Table 6 shows the variables that are permitted in the application FIT file.

*Table 6. Variables in the application FIT file*

| Variable        | Meaning                                         |
|-----------------|-------------------------------------------------|
| ?               | The boot drive, typically Z                     |
| <DCSERVER>      | The server name of the domain controller        |
| <USER>          | The current user ID                             |
| <RPLBOOTSERVER> | The server name of the deployment server        |
| <MACHINE>       | The machine name the user is logged on to       |
| <OSVERSION>     | The operating system version, for example, BB20 |

| Variable | Meaning                                |
|----------|----------------------------------------|
| <LANG>   | The language selected, for example, US |

These variables will be substituted with the appropriate values at the time that the user launches the application and the FIT entries are loaded into memory. For example, the following line taken from the appost.fit file for Netscape provides variable-based redirection of the netscape INI file:

```
?:\OS2\NSCP.INI \\<DCSERVER>\TDMPRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NSCP.INI
```

In the case of a standard implementation of Netscape for a user named USER3 on an IBM WorkSpace On-Demand 3.0.1 server named ATLAS2, this FIT entry will redirect Z:\OS2\NSCP.INI to \\ATLAS2\TDMPRFLS\USER3\OS2\US\PROFILES\NETSCAPE\NSCP.INI.

Generally, an application FIT file will have to be created each time you define an OS/2 application in IBM WorkSpace On-Demand 3.0.1. To determine which files need to be mapped using the application FIT, begin by looking at the newfile.lst file, which is created in the application package's directory on the server when you finish the application package using the `crtpkg /finish` command. The files listed in newfile.lst are the new files that were added as part of the application install, and these will need to be added to the operating system image for each user that will be using the application. Figure 174 shows a selection of the entries in newfile.lst for a Lotus Notes client installation as described in Section 6.9.5, "Lotus Notes client 4.5 / 4.6" on page 349.

```
Z:\NOTES
Z:\NOTES\alog4.ntf
Z:\NOTES\binary.gif
Z:\NOTES\desktop.dsk
Z:\NOTES\lotusen1.dic
Z:\NOTES\mab45.ntf
Z:\NOTES\mail.box
Z:\NOTES\mail45.ntf
Z:\NOTES\mailbox.ntf
Z:\NOTES\names.nsf
Z:\NOTES\notes.ini
Z:\NOTES\pernames.ntf
```

Figure 174. Selections from newfiles.lst for a Lotus Notes installation

After running `crtpkg /finish` on the sandbox, you will find a subdirectory named user\profile in the application's directory on the TDMPKGS alias. For



example, creating the Lotus Notes package caused the `\application\os2\us\notes\user\profile` directory to be created in TDMPKGS as described in Section 6.9.5, “Lotus Notes client 4.5 / 4.6” on page 349. Every file listed in `newfile.lst` should be copied to this directory. The easiest way to do this is from the sandbox, if you first use the `net use` command to map a drive letter to the TDMPKGS alias.

When you have copied the files to the application’s `profile\users` directory, the files will be copied again to each user’s profile when the application is assigned to the specific user. For example, assigning the Lotus Notes application to the user named `user2`, as described in Section 6.9.5, “Lotus Notes client 4.5 / 4.6” on page 349, causes the contents of the `notes\user\profile` directory to be copied to the `user2\os2\us\profiles\notes` directory in the TDMPRFLS alias.

Finally, the application FIT file can be created to map the new files for each user. When you create the FIT file, remember to use variables for the boot drive, server, and user so that the FIT file can be used in multiple domains with little or no changes. Also, map directories as much as possible in order to reduce the size of the FIT file. For example, in the Lotus Notes application install, all the files listed in `newfile.lst` were in the `Z:\NOTES` directory. Therefore, the application FIT file needs only one entry mapping `Z:\NOTES` to the user profile as shown in Figure 178 on page 352.

### 6.9.3 Netscape Communicator 4.61

Follow these steps to install Netscape Communicator as an OS/2 application:

1. Set up an OS/2 sandbox machine and an OS/2 sandbox user as described in Section 6.9.1, “The OS/2 sandbox” on page 335. Remember to edit the sandbox `CONFIG.SYS` file to disable the `PROTDISK.SYS` device driver.
2. Create an apps share on your IBM WorkSpace On-Demand 3.0.1 server. Make sure that the sandbox user has write access to the share. In our example, we use the alias APPS, which will represent the directory `C:\APPS`.
3. Log on as the sandbox user to the sandbox client machine.
4. Open the command prompt window on the sandbox machine, and execute the `net use` command to map to the APPS alias on the server. We use the following command for our example:

```
net use x: \\atlas2\apps
```

5. On the sandbox machine, run the `crtpkg` utility in order to begin capturing changes to the operating system. From the command prompt, execute the following:

```
z:
cd\applcptr
crtpkg /start
```

6. Now, install Netscape. The Netscape installation files must be either on a CD-ROM or on a drive on the server to which your sandbox client has access. From the drive or directory where the Netscape files reside, run `install.exe`.
7. When the Netscape installer prompts you for a target directory to which it will install, make sure you select `X:\NETSCAPE`. Use defaults for all other Netscape installation prompts.
8. When the installation completes, reboot the sandbox machine.
9. Log on to the sandbox machine with your sandbox user ID.
10. Use the `net use` command again to restore your connection to the APPS alias:

```
net use x: \\atlas2\apps
```

11. Launch Netscape from the icon on your sandbox desktop.
12. When Netscape asks you to create a new profile, enter `default` as the profile name and `Z:\NETSCAPE\USERS\DEFAULT` as the profile location. (The default profile location that Netscape will install to is on the APPS alias, in our case, the X: drive. Make sure you change to Z:, so that profile data is stored local to the sandbox machine and not elsewhere on the server.) Later, we will copy these files and use the FIT file to ensure that each user to which Netscape is assigned will have a separate profile location.
13. After building the profile, Netscape will start for the first time. From the menu bar, click **Edit**, then **Preferences**, and change any of the Netscape preferences you want. You may, for example, want to change the home page or proxy information for your users.
14. Exit from Netscape.
15. From a command prompt on the sandbox, use `crtpkg` again to finish the application package:

```
z:
cd\applcptr
crtpkg netscape /finish
```

16. Now, we need to copy user-specific files to a different location in our TDMPKGS alias so that they can be mapped for each user via the FIT file. The files we need to copy are the following:

- a. z:\os2\nscp.ini
- b. z:\os2\nsreg.dat
- c. x:\netscape\program\dynfonts\fonts.cat
- d. z:\netscape\users (the entire directory and subdirectories)

These files all need to be copied to the Netscape application location in our TDMPKGS share on the IBM WorkSpace On-Demand 3.0.1 deployment server. For our example, this location is:

```
\\atlas2\tdmpkgs\application\os2\us\netscape\user\profile
```

The easiest way to copy these files is to work from the sandbox, using the `net use` command to map a drive letter to the TDMPKGS alias. For example, from the sandbox command prompt, type:

```
net use p: \\atlas2\tdmpkgs
```

Then, copy all the files to `p:\application\os2\us\netscape\user\profile` as indicated. When the applications are assigned to users, this profile information will be copied to each user's directory in the TDMPKGS alias. We will later use the FIT file to map all this profile information on a per-user basis so that each user can maintain their own distinct Netscape profile.

17. On the IBM WorkSpace On-Demand 3.0.1 server, define the Netscape application. For our example, from the administration console, we can enter the following command:

```
application define name=Netscape os=os2 lang=US shell=pmshell  
appdrive="X" apppath="\\atlas2\apps\netscape\program"  
command="netscape.exe -browser -l en_us" icontitle="Netscape"
```

18. Now we can assign the Netscape application to any non-sandbox OS/2 user. From the administration console, in our example, we can enter the following command:

```
user assign user="user3" application="Netscape" os=os2 lang=us  
shell=pmshell
```

19. Finally, we need to use an application FIT file for Netscape in order for each user to get access to the individual Netscape profile. The application FIT file must reside in the same directory as the Netscape executable program. In our example, this is in the `\netscape\program` directory on the APPS alias of our server. We used a post-FIT file, named `apppost.fit`, as shown in Figure 175 on page 344. Notice that this FIT file uses variables

to be as generic as possible. This way, one FIT file can redirect the user-specific profiles for all users of Netscape.

```
;APPFIT      ; required for application FITs
;
; Default FIT file for Netscape Communicator 4.04
;   ?:\ is replaced with the client's boot drive
;   <DCSERVER> is replaced by the server name of Domain Controller
;   <USER> is replaced by the user id
;   <RPLBOOTSERVER> is replaced by the name of the RPL Boot Server
;   <MACHINE> is replaced by the machine name
;   <OSVERSION> is replaced by the Operating System version. (bb10.us, bb20.us etc)
;   <LANG> is replaced by the language code.
;
?:\NSCPDIR                      \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE
?:\OS2\NSCP.INI                 \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NSCP.INI
?:\OS2\NSCP.!!!                 \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NSCP.!!!
?:\OS2\NSCP.###                 \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NSCP.###
?:\OS2\NSREG.DAT                \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NSREG.DAT
?:\NETSCAPE\PROGRAM\NETSCAPE.INI \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NETSCAPE.INI
?:\NETSCAPE\PROGRAM\NETSCAPE.!!! \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NETSCAPE.!!!
?:\NETSCAPE\PROGRAM\NETSCAPE.### \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\NETSCAPE.###
?:\NETSCAPE\PROGRAM\DYNFONTS\FONTS.CAT \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\FONTS.CAT
?:\NETSCAPE\USERS                \\<DCSERVER>\TDMRFLS\<USER>\OS2\<LANG>\PROFILES\NETSCAPE\USERS
```

Figure 175. The apppost.fit file for Netscape Communicator

The user can now log on to any OS/2 client machine and launch Netscape from the Netscape icon on the desktop.

#### 6.9.4 Lotus SmartSuite 1.5

This section describes the installation of Lotus SmartSuite. We install the entire SmartSuite package, which includes the following five applications:

- Lotus Word Pro
- Lotus 123
- Lotus Freelance Graphics
- Lotus Organizer
- Lotus Approach

We can then use the appset object to collect subsets of these applications in order to more easily assign them to our users.

Use the following steps to install Lotus SmartSuite as an OS/2 application:

1. Set up an OS/2 sandbox machine and an OS/2 sandbox user as described in Section 6.9.1, “The OS/2 sandbox” on page 335. Remember to edit the sandbox CONFIG.SYS file to disable the PROTDISK.SYS device driver.

2. Create an apps share on your IBM WorkSpace On-Demand 3.0.1 server. Make sure that the sandbox user has write access to the share. In our example, we use the alias APPS, which will represent the directory C:\APPS.
3. Log on as the sandbox user to the sandbox client machine.
4. Open the command prompt window on the sandbox machine, and execute the `net use` command to map to the APPS alias on the server. We use the following command for our example:  

```
net use x: \\atlas2\apps
```
5. On the sandbox machine, run the `crtpkg` utility in order to begin capturing changes to the operating system. From the command prompt, execute the following:  

```
z:  
cd\applcptr  
crtpkg /start
```
6. Install Lotus SmartSuite. The installation files must be either on a CD-ROM or on a drive on the server to which your sandbox client has access. From the drive or directory where the SmartSuite files reside, run `install.cmd`.
7. In the Lotus SmartSuite installer, click **File Server Install**, and when prompted for a target, install to your network drive. In our case, this is X:. Use defaults for everything else.
8. When the installation has finished, you will find a folder named Lotus SmartSuite for OS/2 Warp 4 on the sandbox desktop. In this folder is an icon labelled Lotus SmartSuite for OS/2 Warp 4 Node Install. Launch the Node Install program in order to install the Lotus applications for a network client.
9. When prompted in the Node installation, ensure that you install to Z:\LOTUSW4. This will set up the Client Enablement of Lotus SmartSuite on the redirected Z: drive for all client machines.
10. When the Node Install program has finished, reboot the sandbox machine.
11. Log on to the sandbox again with your OS/2 sandbox user ID. When you have logged on, restore your network connection to the APPS alias on your server by running:  

```
net use x: \\atlas2\apps
```
12. Run each of the SmartSuite applications in turn. This will create an .INI file for each application. If you want, modify any settings or preferences for the applications. Close each application in turn.

13. From a command prompt on the sandbox, use `crtpkg` again to finish the application package:

```
z:
cd\applcptr
crtpkg netscape /finish
```

14. We will now make separate application packages for each of the five component applications in Lotus SmartSuite. Lotus Word Pro will use the package we just created, called Lotusw4. In the TDMPKGS share on the IBM WorkSpace On-Demand 3.0.1 server, copy the entire LOTUSW4 subdirectory in Application\OS2\US four times, and name the copies as follows:

- a. 123
- b. flg
- c. approach
- d. organizer

The resulting directory structure should look like similar to the one shown in Figure 176.

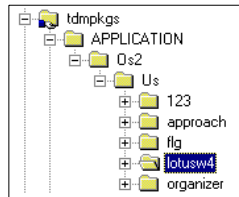


Figure 176. Application packages for the SmartSuite components

15. Now, we need to copy user-specific files to a different location in our TDMPKGS alias so that they can be mapped for each user via the FIT file. The files we need to copy are the following:

- a. `z:\os2\system.dat`
- b. `z:\os2\user.dat`
- c. `z:\lotusw4\*` (the entire directory contents including subdirectories)

These files all need to be copied to the Lotusw4 application location in our TDMPKGS share on the IBM WorkSpace On-Demand 3.0.1 deployment server. For our example, this location is:

```
\\atlas2\tdmpkgs\application\os2\us\lotusw4\user\profile
```

The easiest way to copy these files is to work from the sandbox, using the `net use` command to map a drive letter to the TDMPKGS alias. For example, from the sandbox command prompt, type:

```
net use p: \\atlas2\tdmpkgs
```

Then, copy all the files to `p:\application\os2\us\lotusw4\user\profile` as indicated. When the applications are assigned to users, this profile information will be copied to each user's directory in the TDMPKGS alias. We will later use the FIT file to map all this profile information on a per-user basis so that each user can maintain their own distinct SmartSuite profile.

16. On the IBM WorkSpace On-Demand 3.0.1 server, define an application for each of the Lotus SmartSuite component applications. For our example, from the administration console, we can enter the following commands:

```
application define name=lotusw4 os=os2 lang=US shell=pmshell  
appdrive="X" apppath="\\atlas2\apps\lotusw4\wordpro"  
command="wordpro.exe -i=z:\lotusw4\wordpro" icontitle="Lotus Wordpro"
```

```
application define name=123 os=os2 lang=us shell=pmshell appdrive="X"  
apppath="\\atlas2\apps\lotusw4\123" command="123w.exe" icontitle="Lotus  
123"
```

```
application define name=flg os=os2 lang=us shell=pmshell appdrive="X"  
apppath="\\atlas2\apps\lotusw4\flg" command="flg.exe" icontitle="Lotus  
Freelance Graphics"
```

```
application define name=approach os=os2 lang=us shell=pmshell  
appdrive="X" apppath="\\atlas2\apps\lotusw4\approach"  
command="approach.exe" icontitle="Lotus Approach"
```

```
application define name=organizer os=os2 lang=us shell=pmshell  
appdrive="X" apppath="\\atlas2\apps\lotusw4\organize"  
command="org32.exe" icontitle="Lotus Organizer"
```

17. We can now assign the SmartSuite applications to any non-sandbox OS/2 user. From the administration console, in our example, we can enter the following commands:

```
user assign user=user3 application=lotusw4 os=os2 lang=us shell=pmshell  
user assign user=user3 application=123 os=os2 lang=us shell=pmshell  
user assign user=user3 application=flg os=os2 lang=us shell=pmshell  
user assign user=user3 application=approach os=os2 lang=us  
shell=pmshell  
user assign user=user3 application=organizer os=os2 lang=us  
shell=pmshell
```

18. Finally, we need to use an application FIT file for each of the Lotus SmartSuite components in order for each user to get access to their individual SmartSuite profiles. The application FIT file must reside in the same directory as the executable program for each component. In our example, these are the following directories on the APPS alias of our server:

- a. lotusw4\wordpro
- b. lotusw4\123
- c. lotusw4\flg
- d. lotusw4\approach
- e. lotusw4\organizer

We used a post-FIT file, named apppost.fit, as shown in Figure 177. Notice that this FIT file uses variables to be as generic as possible. This way, one FIT file can redirect user-specific profiles for all users of the SmartSuite applications.

```
;APPFIT      ; required for application FITs
;
; Default FIT file for SmartSuite
;   ?:\ is replaced with the client's boot drive
;   <DCSERVER> is replaced by the server name of Domain Controller
;   <USER> is replaced by the user id
;   <RPLBOOTSERVER> is replaced by the name of the RPL Boot Server
;   <MACHINE> is replaced by the machine name
;   <OSVERSION> is replaced by the Operating System version. (bb10.us, bb20.us etc)
;   <LANG> is replaced by the language code.
;
?:\LOTUSW4                \\<DCSERVER>\TDMPRFLS\<USER>\OS2\<LANG>\PROFILES\LOTUSW4
?:\OS2\SYSTEM\SYSTEM.DAT  \\<DCSERVER>\TDMPRFLS\<USER>\OS2\<LANG>\PROFILES\LOTUSW4\SYSTEM.DAT
?:\OS2\SYSTEM\SYSTEM.~~~  \\<DCSERVER>\TDMPRFLS\<USER>\OS2\<LANG>\PROFILES\LOTUSW4\SYSTEM.~~~
?:\OS2\SYSTEM\USER.DAT    \\<DCSERVER>\TDMPRFLS\<USER>\OS2\<LANG>\PROFILES\LOTUSW4\USER.DAT
?:\OS2\SYSTEM\USER.~~~    \\<DCSERVER>\TDMPRFLS\<USER>\OS2\<LANG>\PROFILES\LOTUSW4\USER.~~~
```

Figure 177. The apppost.fit file for Lotus SmartSuite

19. Before booting the OS/2 client machine, edit its config.sys (in the RPLFILES alias in the machines\<name>\os2\bb20\us directory) and add the following line:

```
set somdthreadpeer=1
```

20. The user can now log on to the OS/2 client machine and run the SmartSuite applications.

SmartSuite is a collection of five applications and could be managed as an application set. If you create an appset containing all the SmartSuite applications, you can assign the entire appset to a user with only one



command instead of using multiple assign commands. We can use the following commands to define and assign the SmartSuite appset object:

```
appset define user assign user=user2 appset=smartsuite os=os2 lang=us
shell=pmshell
user assign user=user2 appset=smartsuite os=os2 lang=us shell=pmshell
appset add name=smartsuite application=lotus4w os=os2 lang=us shell=pmshell
appset add name=smartsuite application=123 os=os2 lang=us shell=pmshell
appset add name=smartsuite application=flg os=os2 lang=us shell=pmshell
appset add name=smartsuite application=approach os=os2 lang=us
shell=pmshell
appset add name=smartsuite application=organizer os=os2 lang=us
shell=pmshell
user assign user=user3 appset=smartsuite os=os2 lang=us shell=pmshell
```

We can also create appsets for subsets of the SmartSuite applications that can be assigned to users that do not need access to the entire set of SmartSuite.

### 6.9.5 Lotus Notes client 4.5 / 4.6

Use the following instructions to install Lotus Notes on an OS/2 client. We only install the Notes client code, assuming that the Notes server has been installed elsewhere in the network.

1. Set up an OS/2 sandbox machine and an OS/2 sandbox user as described in Section 6.9.1, “The OS/2 sandbox” on page 335. Remember to edit the sandbox CONFIG.SYS file to disable the PROTDISK.SYS device driver.
2. Create an apps share on your IBM WorkSpace On-Demand 3.0.1 server. Make sure that the sandbox user has write access to the share. In our example, we use the alias APPS, which will represent the C:\APPS directory.
3. Log on as the sandbox user to the sandbox client machine.
4. Open the command prompt window on the sandbox machine, and execute the `net use` command to map to the APPS alias on the server. We use the following command for our example:

```
net use x: \\atlas2\apps
```

5. On the sandbox machine, run the `crtpkg` utility in order to begin capturing changes to the operating system. From the command prompt, execute the following:

```
z:
cd\app\cptr
crtpkg /start
```

6. Install Lotus Notes. The Notes installation files must be either on a CD-ROM or on a drive on the server to which your sandbox client has access. From the drive or directory where the Notes files reside, run `instpm.exe`.

**Note**

In order to install Lotus Notes 4.6 as a client, you have to change one file in the installation source manually, namely P32WKS.PKG. Change the line:

```
DISPLAY='NO'
```

to

```
DISPLAY='YES'
```

After the installation you need to add the following statement to your NOTES.INI:

```
unsupportedclient=1
```

As this parameter indicates, this environment is unsupported!

7. Follow the Lotus Notes installation prompts. Click **Shared Installation**, and when prompted for a target directory, select `x:\NOTES`. Use defaults for the rest of the prompts.
8. When the installation has finished, you will find a folder named Lotus Applications on the sandbox desktop. In this folder is an icon labelled Lotus Notes Node Installer. Launch the Node Installer program in order to install Lotus Notes for a network client.
9. When prompted in the Node installation, ensure that you install to `Z:\NOTES`. This will set up the Client Enablement of Lotus Notes on the redirected Z: drive for all client machines.
10. When the Node Install program has finished, reboot the sandbox machine.
11. Log on to the sandbox again with your OS/2 sandbox user ID. When you have logged on, restore your network connection to the APPS alias on your server by running:  

```
net use x: \\atlas2\apps
```
12. Run Lotus Notes. Create an .INI file for the application. If you want, modify any settings or preferences for Lotus Notes, then close the application.

13. From a command prompt on the sandbox, use `crtpkg` again to finish the application package by running:

```
z:
cd\applcptr
crtpkg notes /finish
```

14. Now, we need to copy user-specific files to a different location in our TDMPKGS alias so that they can be mapped for each user via the FIT file. The file we need to copy is the following:

```
z:\notes\*.*
```

This file needs to be copied to the Notes application location in our TDMPKGS share on the IBM WorkSpace On-Demand 3.0.1 deployment server. For our example, this location is:

```
\\atlas2\tdmpkgs\application\os2\us\notes\user\profile
```

The easiest way to copy these files is to work from the sandbox, using the `net use` command to map a drive letter to the TDMPKGS alias. For example, from the sandbox command prompt, type:

```
net use p: \\atlas2\tdmpkgs
```

Copy all the files to `p:\application\os2\us\notes\user\profile` as indicated. When the applications are assigned to users, this profile information will be copied to each user's directory in the TDMPKGS alias. We will later use the FIT file to map all this profile information on a per-user basis so that each user can maintain their own distinct Notes profile.

15. On the IBM WorkSpace On-Demand 3.0.1 server, define the Notes application. For our example, from the administration console, we enter the following command:

```
application define name=notes os=os2 lang=US shell=pmshell appdrive="X"
apppath="\\atlas2\apps\notes" command="ilnotes.exe" icontitle="Lotus
Notes"
```

16. Now, we assign the Notes application to any non-sandbox OS/2 user. From the administration console, in our example, we enter the following command:

```
user assign user=user3 application=Notes os=os2 lang=us shell=pmshell
```

17. Finally, we need to use an application FIT file for Notes in order for each user to get access to the individual Notes profile. The application FIT file must reside in the same directory as the Notes executable program. In our example, this is in the Notes directory on the APPS alias of our server. We used a post-FIT file, named `apppost.fit`, as shown in Figure 178 on page 352. Notice that this FIT file uses variables to be as generic as possible.

This way, one FIT file can redirect the user-specific profiles for all users of Netscape.

```
;APPFIT      ; required for application FITs
;
; Default FIT file for Lotus Notes
;   ?:\ is replaced with the client's boot drive
;   <DCSERVER> is replaced by the server name of Domain Controller
;   <USER> is replaced by the user id
;   <RPLBOOTSERVER> is replaced by the name of the RPL Boot Server
;   <MACHINE> is replaced by the machine name
;   <OSVERSION> is replaced by the Operating System version. (bb10.us, bb20.us etc)
;   <LANG> is replaced by the language code.
;
?:\NOTES      \\<DCSERVER>\TDMPRFLS\<USER>\OS2\<LANG>\PROFILES\NOTES457
```

Figure 178. The apppost.fit file for Lotus Notes

18. The user can now log on to the OS/2 client machine and run Lotus Notes.

---

## 6.10 Making System Changes to the Win32 Clients

There is always to a need to customize your clients in a corporate environment. In the following example, we will make changes to the Startup timer for the client, the time format (from 12 hour to 24 hour) and the short date format.

In the following section, we will be using a Windows 2000 client. Follow these steps to make the required system changes:

1. Define a regular Windows 2000 client. (For details, see Section 4.2, "Defining Windows 2000 client machines" on page 105).
2. Define a sandbox user and assign a sandbox desktop to the user in the CLI console on your server:

```
user define user=usersand
user modify user="usersand" password="" passwordexpiration=NEVER
groupmember define user="usersand" group="Domain Admins"
user assign desktop=sandbox os=w2k lang=us shell=explorer
user="usersand"
```

3. On the sandbox client, logon as the sandbox user (usersand).
4. From a DOS command prompt on the client, enter:

```
C:\CRTPKG\CRTPKG /start
```

5. On the sandbox client go to **Control Panel** and click **System**.
6. At the Advanced Tab, click on **Startup and Recovery**.
7. Change the 30 seconds to 5 seconds at “Display list of Operating Systems for 30 seconds”.
8. Click on **OK** to close the System folder.
9. In **Control Panel**, select **Regional Settings**.
10. At the **Time** tab, change the Time Format from **h:mm:ss tt** to **H:mm:ss tt**.
11. At the **Date** tab, change the Short Date from **m/d/yyyy** to **yyyy-mm-dd**. Also change the Date Separator from **“/”** to **“-”**.
12. Click on **Apply**, then click **OK** and close all open windows and reboot the sandbox client.
13. Log on as the sandbox user (usersand).
14. At a DOS command prompt on the client, enter:  

```
C:\CRTPKG\CRTPKG system1 /finish
```

This step creates the application package on your server.
15. Define the application to IBM WorkSpace On-Demand 3.0.1. Enter the following command in the command console on your server.  

```
application define name="system1" os=w2k shell=explorer lang=us
```
16. You can assign the application to other users now with the following command in the command console on your server:  

```
user assign application="system1" os=w2k shell=explorer lang=us  
user="User7"
```

When you log on with, for example, User7, all related files and necessary updates will be transferred to the user's client machine. A second logon or even a reboot is required after the update.

**Note**

When making System Changes to your client, always ensure that the changes are not hardware specific. Changes that can be made is Time, Date, Icons, Start Menu, Screen Saver, Fronts, and so on.

Due to different machine types (like Sound Drivers, Video Drivres, IDE Drivers, Scic Drivers, and so on), system changes to hardware should be avoided.

## 6.11 Defining applications in IBM WorkSpace On-Demand 3.0.1 GUI

After you have installed your applications through the sandbox machine and made any necessary modifications on the server, you can use the IBM WorkSpace On-Demand 3.0.1 GUI instead of the CLI to create the application objects and assign the applications to users.

To create an application through the GUI, log on to the server and select **Define an application** from the **Tasks** menu. You will be presented with a drop-down list to select the operating system for the application you are creating. Select the operating system, then click **Next**. You will be presented with two tabs at the bottom of the window. On the Identity tab, you can enter the application name and description, then select the shell and language you want to use.

Figure 179 shows the application Identity tab for Netscape for OS/2 exactly as it was defined in the CLI in Section 6.9.3, “Netscape Communicator 4.61” on page 341.

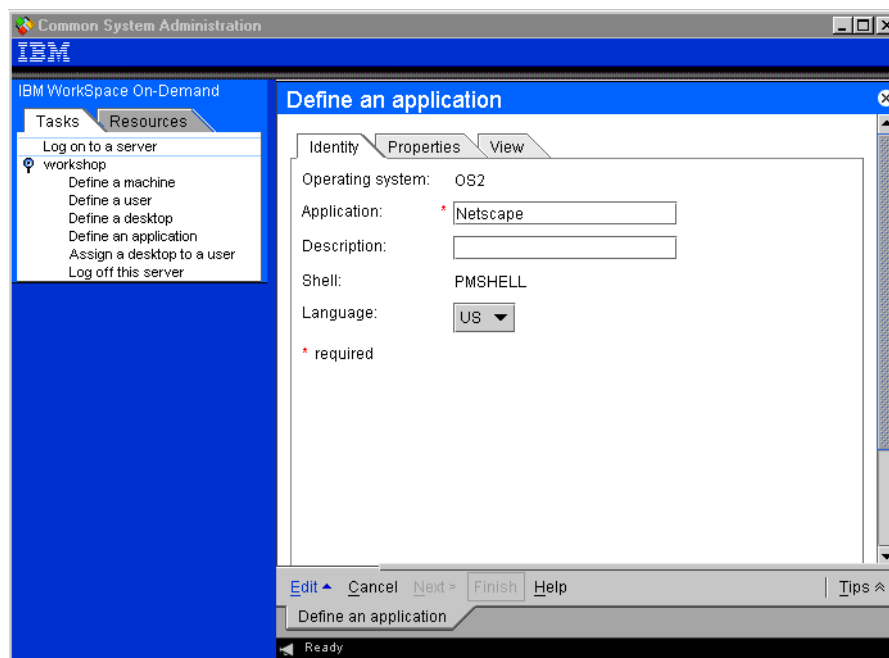


Figure 179. Application Identity tab for Netscape

Click the **Properties** tab, and fill in all the application properties exactly as they were entered in the application define command. The entry fields for

**Command** path and **Command** drive refer to the Apppath and Appdrive parameters for machine define. You can also set the icon and location on the window.

Figure 180 shows the application Properties tab for the Netscape application for OS/2 exactly as it was defined in Section 6.9.3, “Netscape Communicator 4.61” on page 341.

Click **Finish** to define the application.

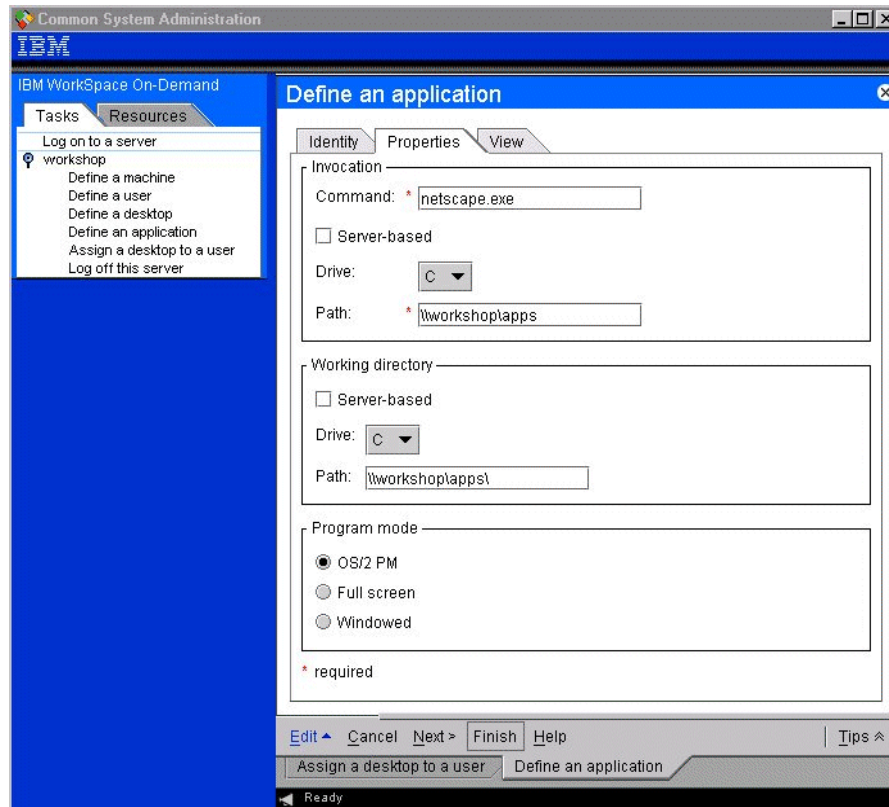


Figure 180. Application Properties tab for Netscape

When you have defined your applications, you can assign them to users via the GUI as well. However, to assign many applications or manage many users, it will probably be easier to perform these tasks via JavaScript in the CLI.

To assign an application, select **Assign a desktop to a user** from the **Tasks** menu in the GUI. You will assign desktops and applications at the same time for the same user. After selecting the operating system, you will be presented with a menu allowing you to select the desktop as well as the applications you want to assign, as shown in Figure 181.

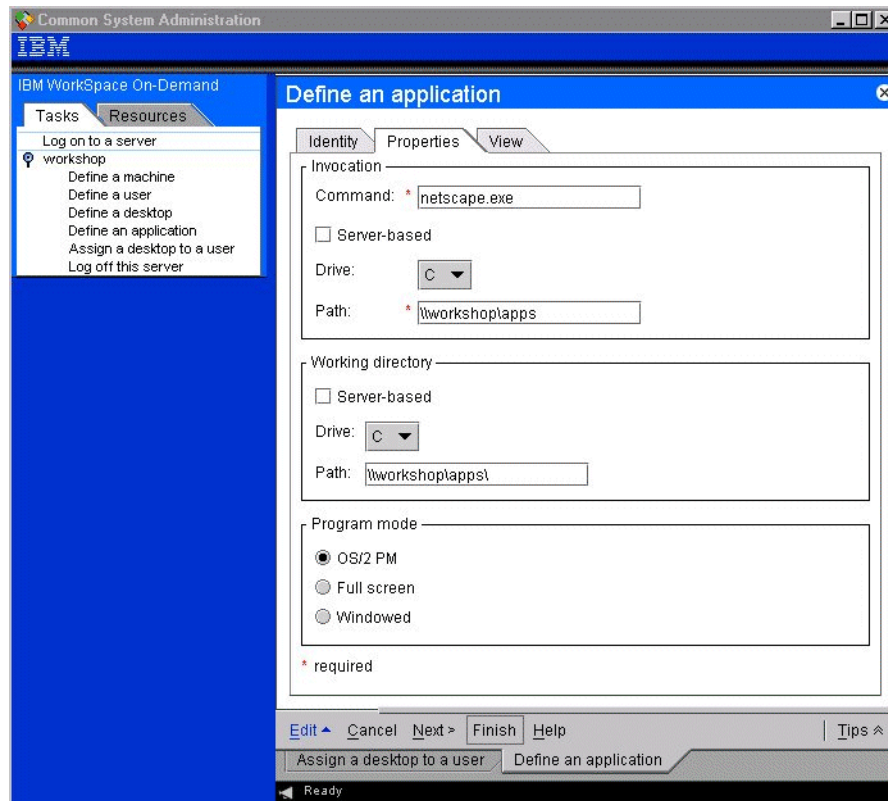


Figure 181. Assigning applications to a user in the GUI

## 6.12 Problem determination for Win32 Application and Desktop

### 6.12.1 Application

The most common thing that can go wrong with applications is that they do not work on a test client when they worked on the sandbox. This can happen for various reasons (such as an error in the application package or incorrect access to the server).



- Problem:
  - Application doesn't work properly on a regular client when it worked on the sandbox
- Possible Cause:
  - Winstall failed
  - Update.bat failed
  - Incorrect permissions set on server for application share or home directory
  - Incorrect desktop policy file
  - Protodisk not configured properly
  - Something missing from the application package
- Solutions:
  - Winstall failure: see Section 6.13.2, "Winstall" on page 360.
  - Update.bat failure: Log in as an administrator and try to install the user.inf manually to see if there are any errors. See Winstall tips on debugging sys.inf for debugging user.inf in Section 6.13.2, "Winstall" on page 360.
  - Permissions: Make sure the user has read access to the application share and write access to their home directory. Make the user an admin to see if this might be the problem.
  - Policy file: Make sure the desktop's policy file doesn't restrict any part of the application. Give the user the sandbox policy file to see if this might be the problem.
  - Protodisk: Make sure that the protodisk.ini file allows access to all files that need to be writable by the application. Edit the protodisk.ini file and enable the entire C drive to see if this might be the problem.
  - application packages: Try to compare the sandbox and the test client to find registry or file differences. You can do this by exporting registry trees from the two machines in regedit and comparing them with windiff. Look for things that might have been excluded by crtpkg.ini.
  
- Problem:
  - User assigned fails or times out
- Possible Causes:

- Home directory package may be too large to be copied before server times out.
- Application package, user profile, or user's home directory may be locked
- Solutions:
  - Home directory packages: Set the server's time-out parameters higher (see `server modify` command).
  - Files locked: Make sure no command lines or Explorer windows have any of the file locked. Make sure that the user is logged off from the client.

### 6.12.2 Desktop

The problems that can occur with the desktop include missing icons, missing **Start** menu items, incorrect policies and extra things that should not be there.

- Problem
  - Desktop doesn't appear or function correctly
- Possible Cause:
  - Desktop package did not get copied to user's profile.
  - Desktop package is not correct.
- Solution:
  - Check the user's profile directory to see if all the expected files are present. If no files are present, unassign and reassign the desktop. You can also add or edit files manually to see if that's will resolve the problem.
  - Make sure that the desktop package has all the correct components: [nt]user.dat, [nt]config.pol, or any additions to the Explorer directory that are necessary. Replace files with the originals (if necessary).

---

## 6.13 Win32 Client Tools Problem determination

### 6.13.1 CRTPKG

With the `crtpkg` utility, most problems can be traced to a file or registry access. The start phase of `crtpkg` should normally not fail, but the finish phase might fail for various reasons. During the finish phase, `crtpkg` copies files and registry keys that have changed. When this part fails, either the file that the `crtpkg` is trying to access can not be read or the

registry key that crtpkg is trying to access can not be read. Other problem include trying to create a package that already exists, and errors copying files to the tdmpkgs share of the server.

- Problem:
  - Crtpkg fails because a file can not be read or registry key can not be read.
- Possible Cause:
  - Crtpkg is trying to read a file or registry key that it does not have read access to.
- Solution:
  - If a file is the problem, add the file to C:\CRTPKG\CRTPKG.INI under the [EXCLUDE\_FILE] section. If the file isn't listed in the command line output, check C:\CRTPKG\CRTPKG.LOG for the file path. If a registry key or value is the problem, add the key or value to C:\CRTPKG\CRTPKG.INI under [EXCLUDE\_REG\_KEY] or [EXCLUDE\_REG\_VALUE] section.
- Problem:
  - Crtpkg fails because a package already exists with that app id.
- Solution:
  - Delete the existing package on the server or choose a different app id.
- Problem:
  - Crtpkg fails because it can't write to the application package path.
- Possible Cause:
  - The application package path is incorrect or the user does not have write access to the path.
- Solution:
  - If the application package path written to the command line output looks correct, make sure that the user you're logged on as has write access to the tdmpkgs share on the server. If the path isn't correct, specify the path to the tdmpkgs share manually on the `crtpkg` command with the `/path:\\<server>\tdmpkgs` option.

### 6.13.2 Winstall

Winstall can fail because something in the application package that it is trying to install is incorrect. For example, the sys.inf file may have a registry key that is restricted, so the installation fails.

- Problem:
  - Winstall reinstalls a package every time you logon to a machine, winstall reboots the machine every time you logon, or winstall hangs the logon window.
- Possible Cause:
  - One of the components that is being installed fails.
- Solution:
  - Logon to the machine as an administrator without any applications assigned. Check C:\WINNT\WINSTALL.LOG for Windows 2000 and NT, and C:\WINDOWS\WINSTALL.LOG for Windows 98 to see what failed during the installation.
  - If the file copy fails, try copying the files manually to see if there's a problem with locked files or access to the drive. If there are problem files, delete them from the package.
  - If the sys.inf install fails, try looking in the file for registry string values that have new lines (that is, a string value continues on multiple lines). Note that it is okay for binary values to continue on multiple lines. If there are any strings that have new lines, then delete them from the sys.inf.
  - Some of the registry keys in the sys.inf may be restricted. In this case you can use regmon (a free download from shareware sites like [www.os2ss.com](http://www.os2ss.com) and [hobbes.nmsu.edu](http://hobbes.nmsu.edu)) to find out if any of the registry keys are restricted.
  - Install the sys.inf manually by right-clicking on it and choosing install while regmon is running. Look for "access denied" errors and delete any keys that show "access denied" from sys.inf
  - If the inifiles.inf or renfile.bat fails, check the paths in the files to make sure they are correct and that there are no access problems with the files.

---

## 6.14 Moving entire application packages

This section gives a short overview of how you can easily define all your application packages on one server and then copy them to other IBM WorkSpace On-Demand 3.0.1 configuration servers in your environment.

As mentioned in Section 6.5, “Application packages” on page 272, an application package is represented by a file structure on the configuration server (see Figure 161 on page 273).

You only have to copy this file structure to your new server and define the application in the IBM WorkSpace On-Demand 3.0.1 console. However, you also might have to modify the Winstall.csf file of each application. This file tells the WINSTALL tool, during the logon of the user, which changes have to be applied to the client in order to make an application available. This configuration file also contains the server name, where the sources of these changes are located.

You might also consider having your application packages on one central server only. In this case, you would not have to modify the Winstall.csf file.

The following list gives you a detailed procedure of how to copy a sample application (in our case, we chose Acrobat Reader; “AdobeR” is the defined application name) from the server “MASTER” to a newly-created IBM WorkSpace On-Demand 3.0.1 server (in our example, BRANCH15).

1. Copy the directory structure  
\\MASTER\TDMPKGS\Application\Nt4\Us\AdobeR to the new server  
\\BRANCH15\TDMPKGS\Application\Nt4\Us\AdobeR.
2. Copy the directory structure \\MASTER\APPS\Adobe to  
\\BRANCH15\APPS\Adobe.

If you plan to store all your application packages in one central location, skip step 3 and go directly to step 4.

3. Modify the  
\\BRANCH15\TDMPKGS\Application\Nt4\Us\AdobeR\Winstall.CSF file to point to your new server name, as shown in Figure 182 on page 362.

```
[COPY_DIRECTORY_STRUCTURE]
KEY1="//BRANCH15\TMPKGS\APPLICATION\NT4\US\Adobe\MACHINE\C", "C:\"
[INSTALL_INF_FILE]
KEY1="//BRANCH15\TMPKGS\APPLICATION\NT4\US\Adobe\MACHINE\inifiles.inf"
KEY2="//BRANCH15\TMPKGS\APPLICATION\NT4\US\Adobe\MACHINE\sys.inf"
```

*Figure 182. Modified WINSTALL.CSF on the new server*

4. Define the application on the server BRANCH15 in the IBM WorkSpace On-Demand 3.0.1 console with the following command:

```
application define name=AdobeR os=nt4 shell=explorer lang=us
(apppath='\\BRANCH15\APPS\Adobe\Reader' Appdrive=P)
```

---

## Chapter 7. Updating the client images

This chapter describes the client update and customization process—specifically, how to update the Windows NT, 98, 2000 and OS/2 client images, add device drivers, and apply fixpacks, Service Packs, or bug fixes. Information about tools and utilities that aid the installation and configuration process is also addressed.

### Note

The examples in this chapter are based on the ITSO lab environment. The image names created are the same as in Chapter 2, “Installation” on page 9. The OSs and their corresponding image names can be found in Table 7

Table 7. OSs and respective image names

| OS           | Image Name |
|--------------|------------|
| Windows 98   | W98SE      |
| Windows NT   | SP3        |
| Windows 2000 | GA         |
| OS/2         | BB20       |

---

### 7.1 The install scripts

The install process for all three flavors of Windows (98, NT and 2000) makes use of the Microsoft unattended install process. These processes are not identical for all three operating systems. The response files look somewhat similar, but differ greatly in most respects. See Figure 183 on page 364 for an outline of the various Windows install processes. Since Windows 2000 is the successor to NT, the install process is nearly identical for both versions.

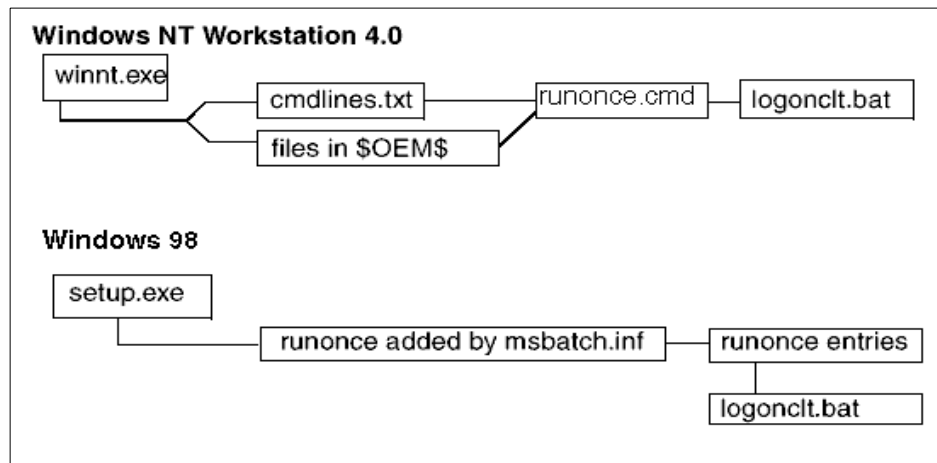


Figure 183. Windows install process

### 7.1.1 Windows NT Workstation and 2000 Professional install files

Windows NT and 2000 both use the `winnt` command for installation. In IBM WorkSpace On-Demand 3.0.1, this command uses the network connection to access the installation files on the boot server. The `winnt` command copies all the files needed to complete the installation over the network into a temporary directory and then continues by performing the installation from the local hard disk (first going through the text mode setup and then GUI mode setup).

The syntax of the `winnt` command is as follows:

```
winnt [/s:sourcepath] [/i:inf_file] [/t:drive_letter] [/x] [/b] [/o[x]]
[/u:answer_file] [/udf:id, [UDF_file]]
```

Table 8 describes the parameters of the `winnt` command.

Table 8. Parameters of the `winnt` command

| Parameter       | Description                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------|
| /s:sourcepath   | Specifies the location of the Windows NT files.                                             |
| /i:inf_file     | Specifies the file name (no path) of the setup information file. The default is DOSNET.INF. |
| /t:drive_letter | Forces setup to place temporary files on the specified drive.                               |



| Parameter           | Description                                                                                                                                                                                                                                                                                                                                     |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /x                  | Prevents setup from creating setup boot floppies. Use this when you already have setup boot floppies (from your administrator, for example).                                                                                                                                                                                                    |
| /b                  | Causes the boot files to be loaded on the system's hard drive rather than on floppy disks so that floppy disks do not need to be loaded or removed by the user.                                                                                                                                                                                 |
| /o                  | Specifies that setup only create boot floppies.                                                                                                                                                                                                                                                                                                 |
| /ox                 | Specifies that setup create boot floppies for CD-ROM or floppy-based installation.                                                                                                                                                                                                                                                              |
| /u:answer_file      | Specifies the location of an answer file that provides answers the user would otherwise be prompted for during setup.                                                                                                                                                                                                                           |
| /udf:id [,UDF_file] | Specifies the identifier that is to be used by the setup program to apply sections of the UDF_file in place of the same section in the answer file. If no UDF is specified, the setup program prompts the user to insert a disk that contains a file called \$UNIQUE\$.UDF. If a UDF is specified, setup looks for the identifier in that file. |

Because of the way IBM WorkSpace On-Demand 3.0.1 operates, many of the command line options are not used. The command issued by a IBM WorkSpace On-Demand 3.0.1 client provides the response file (answer file) and the location of the source. The following command is issued in state.ini:

```
winnt.exe /u:c:\IBMWIN32\Unattend.txt /s:x:\nt40\i386
```

### 7.1.2 Unattend.txt

The unattend.txt response file or answer file is a file containing the information required to automate the install process. The format of the file consists of section headers, parameters, and values for those parameters. The section headers are predefined and some may be user-defined. It is not necessary to specify all the parameters and keys. The file format is as follows:

```
[section1]
; Section contains keys and the corresponding
; values for those keys/parameters.
; keys and values are separated by "=" signs
; Values usually require double quotes "" around them
;
key = value
.
```

```
.  
[section2]  
key = value  
.
```

Information about the sections and keys can be found in the Microsoft Resource Kit, or at:

<http://support.microsoft.com/support/kb/articles/Q155/1/97.asp>

### 7.1.3 Copying the \$OEM\$ directory and its subdirectories

Once the operating system is installed, any additional components, files, or applications that you might need and that are not included in the Windows NT/2000 retail products must be added to subdirectories of the \$OEM\$ directory on the server.

The \$OEM\$ directory includes the following subdirectories:

- \$OEM\$\Textmode

The \$OEM\$\Textmode directory contains all of the files that the setup program needs to load and that text mode setup needs to copy to the target computer so that the system can boot into a GUI setup.

If you are installing SCSI, keyboard, video, or pointer device drivers, or HALs, that are not included or have been updated since the release of the Windows NT Workstation 4.0 retail version, this directory should also contain a txtsetup.oem file. This file contains pointers to all the files required by the setup program and the text mode setup to load and install these components. Txtsetup.oem and all files listed in it must also be listed in the [OEMBootFiles] section of the answer file.

- \$OEM\$\\$\$

This directory contains system files (either new files or replacements to files included in the retail product) that need to be copied to the various subdirectories in the Windows NT system directory. You need to maintain the directory structure used in the retail product and place in each subdirectory the files that need to be copied to the corresponding system directory on the destination computer. If some of the files use long file names, add the file \$OEM\$\\$\$\\$\$Rename.txt. This file lists all files that have long names and their corresponding short names.

- \$OEM\$\NET

This directory contains only subdirectories. You need to place in each subdirectory the files for a particular network component, such as network

cards, network services, and network protocols. Files in this directory are used by the network setup module.

- **\$OEM\$\drive\_letter**

The \$OEM\$\drive\_letter directory holds files that are to be copied by the setup program to corresponding drives on the destination computers. For example, files in the \$OEM\$\C directory will be copied to C: on the destination computer during text mode setup.

Place files for applications that you want to install along with the Windows NT Workstation in subdirectories of the \$OEM\$\drive\_letter directories on the distribution sharepoint. The files in these subdirectories must have short file names. To rename them with long file names after they have been copied to the destination computer, list them in a file named \$\$rename.txt in the same directory.

Applications included in the \$OEM\$\drive\_letter directories must support a scripted (silent) installation; to include applications that require interactive installation, use the sysdiff utility.

Setup uses commands specified in the \$OEM\$\Cmdlines.txt file to install applications or copy files that are available in subdirectories of \$OEM\$\drive\_letter on the distribution sharepoint. You can use these commands to invoke a Win95-style INF file, run the `sysdiff` command, or perform other actions.

For example: to install MS Office on C: in the \MSOffice directory, place the files in \$OEM\$\C\MSOffice. Then specify the command to install the application in the \$OEM\$\Cmdlines.txt file.

You can also place files that you only want copied (but not installed) in subdirectories of \$OEM\$\drive\_letter. For example, if you have written help files that cover policies and procedures used in your organization, you can place them in a subdirectory of \$OEM\$\C. Then use a command in the \$OEM\$\cmdlines.txt file to copy them to the destination computers.

#### **7.1.4 The cmdlines.txt and other files**

Once the `winnt` command completes, and the files are copied across, the system processes the cmdlines.txt file. You can install files located in the subdirectories of \$OEM\$ by listing the installation commands in a text file named cmdlines.txt. The cmdlines.txt file is located in the C:\TDM\MM\CLIENT\RO\NT4\SP3\US\INST\I386\OEM\$ directory for Windows NT and in the C:\TDM\MM\CLIENT\RO\W2K\GA\US\INST\I386\OEM\$ directory for Windows 2000.

All workstations share the same cmdlines.txt file. The syntax is as follows:

```
[Commands]
"command 1"
"command 2"
"command 3"
```

Enter the entire command line in double quotation marks. The following is an example of a cmdlines.txt file for Windows NT Workstation:

```
[Commands]
"rundll32 setupapi,InstallHinfSection DefaultInstall 128 .\delicons.inf"
"rundll32 setupapi,InstallHinfSection DefaultInstall 128 c:\logon\ntclean.inf"
"cmd /c copy c:\logon\state.ini \winnt"
"cmd /c copy c:\logon\custom.inf \winnt"
"c:\logon\RunOnce.cmd"
```

The cmdlines.txt file has all the comments removed. The first thing it does is remove the icons on the desktop. Then it copies the state.ini and custom.inf to change the clients NT Workstation configuration. Finally, it runs the file Runonce.cmd, which applies some entries to the registry. State.ini is discussed in Chapter 4, "Defining machines" on page 103.

Runonce.cmd contains the following command:

```
\WINNT\REGEDIT /s c:\logon\runonce.reg
```

Runonce.reg contains the following registry entries:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce]
"BootApps"="C:\\LOGON\\BOOTAPPS.EXE"
```

```
[HKEY_CURRENT_USER\Control Panel\Desktop]
"Wallpaper"="wsoddtbg.bmp"
```

```
[HKEY_CURRENT_USER\Control Panel\Colors]
"Background"="0 0 128"
```

Runonce.reg runs Bootapps.exe, then changes the wallpaper and background color.

Cmdlines.txt also runs ntclean.inf, which is listed below:

```
[version]
signature="$Windows NT$"
[DefaultInstall]
```

```

AddReg=InstallRunOnce
[InstallRunOnce]
;KLM,%RunOnceKey%,%RunOnceEvent%,,%RunOnceProg%
HKLM,%AutoLogon%,%AutoAdmin%,,%Val1%
HKLM,%AutoLogon%,%Pass%,,%Val2%

[DeleteAutoAdminLogon]
DelReg=DelAutoLogon
[DelAutoLogon]
HKLM,%AutoLogon%,%AutoAdmin%

[Strings]
RunOnceKey="SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce"
RunOnceEvent="LogonClnt"
RunOnceProg="c:\logon\logonclnt.bat"
AutoLogon="SOFTWARE\Microsoft\Windows NT\CurrentVersion\WinLogon"
AutoAdmin="AutoAdminLogon"
Val1="1"
Pass="DefaultPassword"
Val2=""

```

Ntclean.inf adds registry entries that allow the workstation to autologon and then runs the batch file logonclnt.bat.

### 7.1.5 The \$\$Rename.txt files

To map short file names (used in subdirectories of \$OEM\$) to long file names that should be assigned to those files after they are copied to the destination computers, list them in a \$\$Rename.txt file. Each subdirectory of \$OEM\$ requires its own \$\$Rename.txt file if the files in that directory need to be assigned long file names on the destination computer. For example, if some of the system files in \$OEM\$\\$\$ use long file names, they must be mapped in the \$OEM\$\\$\$\\$\$Rename.txt file. If some of the application files in \$OEM\$\C use long file names, they must be mapped in the \$OEM\$\C\\$\$Rename.txt file.

The format for the \$\$Rename.txt file is as follows:

```

[section name]
short name 1 = "long name 1"
short name 2 = "long name 2"

```

Table 9 outlines the various cmdlines.txt parameters that can be used.

Table 9. Cmdlines.txt parameters

| Parameter    | Description                                                                                                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Section name | This parameter is the path to the directory that contains the files. To indicate that the section contains the names of files or subdirectories that are on the root of the drive, use a backslash [\] as the section name. |
| Short name   | This parameter is the name of the file or subdirectory to be renamed in this directory.                                                                                                                                     |
| “Long name”  | This parameter is the new name of the file or subdirectory. This name must be in double quotes.                                                                                                                             |

### 7.1.6 Windows 98 installation flow

Windows 98 works differently in that it will copy all files into a temporary directory and then run setup.exe to start the installation locally. The copying of the files to the local drive is performed by the IBM WorkSpace On-Demand 3.0.1 installation process and not by the operating system install process, as is the case with Windows NT Workstation.

The setup command is:

```
setup.exe /is /iw MsBatch.inf
```

It is not very important to know all the switches. However, for completeness sake, they are described in Table 10.

Table 10. Windows 98 setup switches

| Switch | Description                                                                                                                                                                                                                                                                                             |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /m     | This switch bypasses the playing of the setup sound (.wav) files.                                                                                                                                                                                                                                       |
| /na    | This switch bypasses the program check and can use the following values:<br>0: default.<br>1: No Windows-based program check, but MS-DOS-based programs are blocked.<br>2: No MS-DOS-based program check, but Windows-based programs are blocked.<br>3: No Windows-based or MS-DOS-based program check. |
| /nd    | This switch ignores the presence of a Migration.dll file and is used to force Windows 98 to overwrite newer files. Note that files that use the “,,,32” flag in the .INF file still force Windows 98 setup to keep the newer files.                                                                     |

| Switch   | Description                                                                                                                                                                                                     |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /nf      | Does not prompt to remove the floppy disk from the drive (for bootable CD-ROMs); same as if there is a file named BOOTCD in the cabinet folder.<BR/>, or if there is a "BootCD=1" line in the Msbatch.inf file. |
| /nh      | This switch bypasses running the Hwinfo.exe program at zero percent files and RunOnce.                                                                                                                          |
| /nx      | Does not check the version of setup that is running.                                                                                                                                                            |
| /ie      | This switch bypasses the Windows 98 Startup Disk Wizard windows. If this switch is used, the Windows\Command\EBD folder is not created.                                                                         |
| /iv      | This switch bypasses displaying the Setup windows during an upgrade within Windows.                                                                                                                             |
| /?       | This switch provides a brief summary of the available setup switches and the correct command line syntax to use them.                                                                                           |
| /c       | This switch bypasses running SMARTDrive.                                                                                                                                                                        |
| /d       | This switch bypasses using your existing Windows configuration (such as your current Win.ini and System.ini files).                                                                                             |
| /l       | Use this switch if you have a Logitech <sup>TM</sup> mouse and want it enabled during setup.                                                                                                                    |
| /n       | This switch causes setup to run without a mouse.                                                                                                                                                                |
| -s       | Use this switch to use an alternate Setup.inf file.                                                                                                                                                             |
| /t:<dir> | This switch lets you specify where setup copies its temporary files. Note that any existing files in this folder are deleted.                                                                                   |
| /ig      | Allows setup to run on some older Gateway <sup>TM</sup> and Micron <sup>TM</sup> computers with an early BIOS.                                                                                                  |
| /ih      | This switch causes setup to run ScanDisk in the foreground.                                                                                                                                                     |
| /im      | Causes setup to ignore the conventional memory check.                                                                                                                                                           |
| /iq      | If you use the /is switch to bypass ScanDisk, or ScanDisk fails, setup checks your drive for cross-linked files. The /iq switch prevents setup from doing this.                                                 |
| /is      | This switch causes setup to not run ScanDisk.                                                                                                                                                                   |
| /iw      | This switch causes setup to bypass licensing.                                                                                                                                                                   |
| /it      | This switch bypasses checking for the presence of <i>dirty</i> or <i>deadly</i> terminate-and-stay-resident programs (TSRs) that are known to cause problems with Windows setup.                                |

| Switch | Description                                                                                                                                                                                           |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /p     | This switch causes setup to pass string(s) directly to Detection Manager (or Sysdetmg.dll). Setup does not interpret the content of the string. The string can contain one or more detection options. |

The /p switch is not used by itself. The string can contain one or more detection switches separated by a semicolon (;). For example, if you want to use /p f and /p i, you type: setup /p f;i

Some switches are simply on/off switches. The absence of the switch implies off; the presence of the switch turns it on. A minus sign (-) appended immediately after a switch turns it off.

Some switches take parameters in the form of <c>=<params>. If there is more than one parameter to a switch, the parameters are separated by a comma (,). There must not be any spaces in the detection option string.

Valid detection switches are available from the Microsoft Web site (<http://support.microsoft.com/support/kb/articles/Q186/1/11.ASP>).

### 7.1.7 The msbatch.inf file

Once the operating systems are installed, additional components, such as Service Packs, the TMA client, and other software can be installed. For the different versions of Windows, they are all started from different places. For Windows 98 clients, it is best to start the install by adding a line to the msbatch.inf file that contains all the install responses.

The msbatch.inf file is the response file for Windows 98 clients. This is the equivalent of the unattend.txt file for Windows NT Workstation. Unlike the Windows NT Workstation install and configuration process, Windows 98 does not have as many additional files.

A part of the msbatch.inf file is reproduced below. The install section shows the additional items to be executed after setup. The product registration and the logonclt.reg section run after the install.

The logonclt.reg section has a number of RunOnce items. It does an automatic logon, installs the Tivoli client, installs the JVM, and finally runs a file called LOGONCLT.BAT. If you look for the LOGONCLT.BAT file, it does not contain anything. It is made available for you to add commands you want executed.



```
[LOGONCLT.REG]
HKLM,"Software\Microsoft\Windows\CurrentVersion\Network\Real Mode Net",AutoLogon,1,0
;HKLM,"Software\Microsoft\Windows\CurrentVersion\RunOnce",TMA,, "c:\$TIVOLI\TIVOLI.bat"
HKLM,"Software\Microsoft\Windows\CurrentVersion\RunOnce",JVM,, "c:\$IBM JVM\IBM JVM.bat"
HKLM,"Software\Microsoft\Windows\CurrentVersion\RunOnce",Logonclt,, "C:\LOGON\LOGONCLT.BAT"
```

## 7.2 Updating Service Packs, fixpacks, and device drivers

Service Packs, fixpacks, patches, or bug fixes are released regularly by the companies that make operating systems. The procedures described in the following sections allows the installation of Service Packs, fixpacks, and device drivers. These procedures also allow automatic installation of Service Packs on deployment.

### 7.2.1 Windows NT 4.0

IBM WorkSpace On-Demand 3.0.1 requires Service Pack 4 or higher. In earlier chapters, we made use of the original Windows NT CD to create the operating system image. This image is the one we will be using to create client machines. If we do not update it to a current Service Pack, we risk the possibility that we may encounter bugs and unsupported devices. Since the Windows NT 4.0 release, many new peripherals and devices have been invented. To remove bugs and provide support for new devices, Microsoft has been releasing Service Packs. As of the writing this book, Service Pack 6a was the most current one. The steps are identical for other Service Packs (4 or higher) as well.

To install a Service Pack to the Windows NT image, follow these steps:

1. Under i386, make sure you have a \$OEM\$ directory. This directory is copied during installation. It contains a new driver, files, and directory. Create a new folder named SERVICE under \$OEM\$.
2. Copy the Service Pack from the Microsoft Download Center web page [www.microsoft.com/downloads/default.asp](http://www.microsoft.com/downloads/default.asp) to:  
C:\TDM\MM\CLIENT\RO\NT4\IMAGENAME\LANG\INST\i386\\$OEM\$\SERVICE  
where:
  - a. Imagename represents the directory name you chose when you defined the Windows NT client machine.
  - b. Lang represents the two character country code.

- c. Service represents a directory name you chose to contain the Service Pack.
3. Add the following line to the end of the base CMDLINES.TXT file.

```
.\service\servicepackname.exe -U -O -N -Z
```

This file resides in the TDM\MM\CLIENT\RO\NT4\imagename\lang\DEFS directory. You must include the quotation marks! For a description of the parameters, see Table 11.

Servicepackname represents the name of the Service Pack executable file that you are installing. Your CMDLINES.TXT should look similar to the following:

```
[Commands]
"rundll32 setupapi,InstallHinfSection DefaultInstall 128 .\delicons.inf"
"rundll32 setupapi,InstallHinfSection DefaultInstall 128 c:\logon\ntclean.inf"
"cmd /c copy c:\logon\state.ini \winnt"
"cmd /c copy c:\logon\custom.inf \winnt"
"c:\logon\RunOnce.cmd"
" .\service\sp6128.exe -U -O -N -Z"
```

4. Save and close the file.
5. To enable the Service Pack on new client machines, refer to Section 7.2.1.1, "Enabling Service Packs on new client machines" on page 374. To enable the Service Pack on existing IBM WorkSpace On-Demand 3.0.1 client machines, refer to Chapter 7.2.1.2, "Enabling Service Packs on existing clients machines" on page 375.

Table 11. Install switches for Windows NT Service Packs

| Parameter | Description                                                          |
|-----------|----------------------------------------------------------------------|
| -U        | Force unattended installation                                        |
| -N        | Do not create uninstall directory on target                          |
| -F        | Force other application to close while installing Service Pack       |
| -Z        | Do not restart after applying Service Pack                           |
| -Q        | Quiet Mode. Do not show User Interface for Service Pack installation |
| -Y        | Perform uninstall of Service Pack                                    |

#### 7.2.1.1 Enabling Service Packs on new client machines

After you installed the Service Pack to the Windows image according to the steps above, define the client machine as a IBM WorkSpace On-Demand

3.0.1 client machine with the `machine define` command or using the GUI and specify `imagename` to be the name of the image where you installed the Service Pack. The Service Pack will be applied automatically after base installation.

#### 7.2.1.2 Enabling Service Packs on existing clients machines

IBM WorkSpace On-Demand 3.0.1 provides two methods for enabling the Service Pack on existing client machines. You can either change the `CMDLINES.TXT` file for an individual client machine, or you can modify the base `CMDFILES.TXT` for all machines, as discussed before.

To enable the Service Pack on an individual machine, follow these steps:

1. Add the following line to the `CMDLINES.TXT` file for that machine:

```
. \service\servicepackname.exe -U -O -N -Z
```

The `CMDLINES.TXT` file for a specific machine is located in:

`C:\TDM\MM\CLIENT\RO\MACHINES\clientname\NT4\imagename\lang`

where:

- a. `Clientname` represents the client machine name.
  - b. `Imagename` represents the directory name you chose when you defined the Windows NT client machine.
  - c. `Lang` represents the two character country code.
2. Reinstall the client machine with the `machine modify` command or using the GUI.

To enable the Service Pack on multiple client machines:

1. Edit the base `CMDLINES.TXT` file according to step 2 in Section 7.2.1, "Windows NT 4.0" on page 373.
2. Delete each machine definition using `machine delete` command.
3. Redefine each client machine using the `machine define` command.

#### 7.2.2 Windows 2000

When you install a Windows 2000 Service Pack, you might consider increasing the default partition size for the client machine. Typically, an addition of 400 MB is sufficient. The preferred method is to add one Service Pack per boot image.

To install a Service Pack to the Windows 2000 image, follow these steps:

1. Copy the Service Pack from the Microsoft Download Center web page ([www.microsoft.com/downloads/default.asp](http://www.microsoft.com/downloads/default.asp)) to:

C:\TDM\MM\CLIENT\RO\W2K\imagename\lang\INST\i386\OEM\C.

where:

- a. Imagename represents the directory name you chose when you defined the Windows 2000 client machine.
  - b. Lang represents the two character country code.
2. Rename the Service Pack to the 8.3 file name format. For example: SP1NETWORK.EXE to SP1NET.EXE.
  3. For an individual client machine, copy RUNONCE.REG to:

TDM\MM\client\RO\MACHINES\machine\_name\W2K\imagename\US.

For all client machines that use the same imagename, copy the RUNONCE.REG to:

TDM\MM\client\RO\W2K\imagename\US\DEFS.

The RUNONCE.REG file resides in the directory:

TDM\MM\CLIENT\RO\W2K\LOGON directory

where:

- a. Imagename represents the directory name you chose when you defined the Windows 2000 client machine.
  - b. Machine\_name represents the name of the client machine.
4. Add the following lines to the RUNONCE.REG file:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce]
"ServicePack1"="C:\\servicepackname.exe -u -o -n -z"
```

Remember that:

- a. You must include the quotation marks.
- b. Servicepackname represents the name (eight characters or less) of the Service Pack executable file that you are installing.
- c. The new lines are directly after the REGEDIT4 line and before the lines:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce]
"BootApps"="C:\\ OGON\\BOOTAPPS.EXE"
```

The following is an example of a RUNONCE.REG file for Windows 2000 Service Pack 1:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce]
"ServicePack1"="C:\\SPINET.EXE -u -o -n -z"
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce]
"BootApps"="C:\\LOGON\\BOOTAPPS.EXE"
[HKEY_CURRENT_USER\Control Panel\Desktop]
"Wallpaper"="wsoddtbg.bmp"
[HKEY_CURRENT_USER\Control Panel\Colors]
"Background"="0 0 128"
```

5. Save and close the file.

6. For an individual client machine, add:

```
Z:\MACHINES\machine_name\W2K\imagename\US\RUNONCE.REG C:\ LOGON
```

to LOGON.LST after the following line

```
Z:\W2K\LOGON\*. * C:\LOGON.
```

After the machine modify command executes, the new line is removed.

For all client machines that use the same imagename, add:

```
Z:\<OS>\<imagename>\<lang>\DEFS\RUNONCE.REG C:\LOGON to LOGON.LST
```

after the following line

```
Z:\W2K\ LOGON\*. * C:\ LOGON
```

where:

- a. Imagename represents the directory name you chose when you defined the Windows 2000 client machine.
- b. Machine\_name represents the name of the client machine.

7. Save and close the file.

8. To enable the Service Pack on new client machines, refer to Section 7.2.2.1, “Enabling Service Packs on new client machines” on page 377. To enable the Service Pack on existing WorkSpace On-Demand client machines, refer to Section 7.2.2.2, “Enabling Service Packs on existing clients machines” on page 377.

#### 7.2.2.1 Enabling Service Packs on new client machines

After you installed the Service Pack to the Windows 2000 image according to the steps above, define the client machine as a IBM WorkSpace On-Demand 3.0.1 client machine with the `machine define` command or using the GUI, specify imagename to be the name of the image where you installed the Service Pack. Service Pack will be automatically applied after base installation.

#### 7.2.2.2 Enabling Service Packs on existing clients machines

IBM WorkSpace On-Demand 3.0.1 provides methods for enabling the Service Pack on existing client machines.

To enable the Service Pack on client machines for Service Packs installed on an existing Windows 2000 setup:

- Reinstall the client machine with the `machine modify` command.
- The installation will occur on the first logon from the client machine.

To enable the Service Pack on client machines for Service Packs installed on a new Windows 2000 setup:

- Delete each machine definition using `machine delete` command.
- Redefine each client machine using the `machine define` command and specify imagename to be the name of the image where you installed the Service Pack.

### 7.2.3 Windows 98

Unless you already have Windows 98 Second Edition, it will be necessary to install Windows 98 Service Pack 1. The procedure to install the Service Pack with the client installation is as follows:

1. Go to the `\TDM\MM\CLIENT\RO\W98\W98SE\US\INST\C` directory.
2. Create a directory called `$Service`.
3. Copy the Service Pack file to this directory.
4. Go to `\TDM\MM\CLIENT\RO\W98\W98SE\US\DEFS\RESP` and edit `MSBATCH.INF`.
5. Find `[LOGONCLT.REG]` and insert the following line:

```
HKLM,"Software\Microsoft\Windows\CurrentVersion\RunOnce",MGA,, "c:\$service\wucsp.exe -q"
```

### 7.2.4 OS/2

All OS/2 clients defined on one Deployment Server share the OS/2 image on this server. Therefore, we need to service each product only one time on each Deployment Server.

OS/2 FixPaks are applied using the Corrective Service Facility (CSF, also known as the FixTool). The FixTool provides two ways to apply service to the products: manual installations using a graphical tool (`SERVICE.EXE`), or using the command line tool `FSERVICE.EXE` for unattended installations (using response files).

OS/2 FixPaks and the FixTool are available on the web at <http://service5.boulder.ibm.com/pspfixpk.nsf/>

As the FixTool only works on OS/2, it can't be used to fix the OS/2 client tree directly on the server. When running on an OS/2 client, it won't accept the virtual local boot drive of the client as valid local partition, so we need to copy the OS/2 client tree to the local hard drive, fix it there, and copy it back.

The process to apply a FixPak to the OS/2 client tree consists of these steps:

1. Define an OS/2 client or select an existing one.
2. Select an empty partition on the client hard disk or create one using FDISK. For these examples we selected D:
3. Edit the client CONFIG.SYS located in the directory  
    \TDM\MM\CLIENT\RO\MACHINES\<client\_id>\OS2\BB20\US\  
    change the line Z:\OS2\BOOT\PROTDISK.SYS to REM  
    Z:\OS2\BOOT\PROTDISK.SYS, and reboot the client to get access to the local hard disk.
4. Copy the OS/2 client image tree from the Deployment Server to the partition selected in Step 2 using the commands:
  - a. NET USE X: \\<server>\C\$
  - b. XCOPY X:\TDM\CLIENT\RO\OS2\BB20\US\TREE\\* D:\TREE\ /H/O/T/S/E/R/V
5. Apply the FixPak to the selected partition.
  - a. Using SERVICE.EXE:
    1. Set the path where the FixPak is located by typing SET  
    CSFCDROMDIR=X:\FIXES\<fixpak path>.
    2. Change the directory to the path where the FixTool is located and type SERVICE.
    3. After a scan of the machine, the FixTool displays a list of serviceable products. Select only the products with SYSLEVEL files in your client image tree and click the **Service** button.
    4. On the next window do not enter an archive or backup path and click the **OK** button (to disable archiving for the OS/2 Base FixPak, see Section 7.2.4.1, "FixPak specific steps" on page 380).
    5. If you get a popup for newer files see Section 7.2.4.1, "FixPak specific steps" on page 380 for a list files which need to be kept for each FixPak.
  - b. Using FSERVICE.EXE:
    1. Prepare a response file for the FixPak (see Section 7.2.4.2, "FixPak sample response files for FSERVICE.EXE" on page 382)

2. Change the directory to the path where the FixTool is located and type `FSERVICE /R:<response file>`
3. If you get a pop-up for newer files, see Section 7.2.4.1, “FixPak specific steps” on page 380 for a list of files which need to be kept for each FixPak.
6. Make a backup of the client image tree on the Deployment Server. Copy the image tree from the client back to the server using the command  
`XCOPY D:\TREE\* X:\TDM\CLIENT\RO\OS2\BB20\US\TREE\ /H/O/T/S/E/R/V`

#### 7.2.4.1 FixPak specific steps

##### ***OS/2 Base FixPak XR\_M014***

To avoid problems with applying future FixPaks, we need to disable archiving of the OS/2 Base FixPaks. Insert this step after Step 4. on page 379:

- Change the directory to the path where the FixTool is located and type  
`ARCHCTL OFF <fixpak path>`

When applying Base FixPak XR\_M014, do not overwrite the following files with older versions from the FixPak:

- `d:\TREE\OS2\NCAPPUTL.EXE`
- `d:\TREE\OS2\APPSTART.EXE`

##### ***OS/2 Base Device Driver FixPak XR\_D001***

In order to apply the Base Device Driver FixPak, there needs to be a `SYSLEVEL.BDD` file in the `\OS2\INSTALL` path. If it is the first install of this FixPak, this file does not yet exist, and needs to be created using the following steps, which are inserted after Step 4. on page 379:

1. Create the directory `D:\OS2\INSTALL`
2. Copy the files `SYSLEVEL.OS2` and `SYSLEVEL.FPK` from `D:\TREE\OS2\INSTALL` to `D:\OS2\INSTALL`
3. Type `X:\FIXES\XR_D001\FIX\FIXTOOL.EXT\FIXTPREP`
4. Copy the file `SYSLEVEL.BDD` from `D:\OS2\INSTALL` to `D:\TREE\OS2\INSTALL`
5. Remove the directory `D:\OS2\INSTALL`

When applying Device Driver FixPak XR\_D001, do not overwrite the following files with the older versions from the FixPak

- `D:\TREE\OS2\BOOT\AIC7870.ADD`



- D:\TREE\OS2\BOOT\IPSRAID.ADD
- D:\TREE\OS2\BOOT\VKBD.SYS
- D:\TREE\OS2\BOOT\7870PRES.EXE
- D:\TREE\OS2\BOOT\CLOCK01.SYS
- D:\TREE\OS2\BOOT\CLOCK02.SYS
- D:\TREE\OS2\BOOT\PNP.SYS
- D:\TREE\OS2\BOOT\POINTDD.SYS
- D:\TREE\OS2\BOOT\MOUSE.SYS

#### ***MPTS/TCP/IP FixPak UN\_2200***

This FixPak is only available on the web with subscription to Software Choice for IBM WorkSpace On-Demand 3.0.1 using the following link:

[http://techsupport.services.ibm.com/asd-bin/doc/en\\_us/tcpwsod/f-feat.htm](http://techsupport.services.ibm.com/asd-bin/doc/en_us/tcpwsod/f-feat.htm)

At the time of the writing of this redbook, this FixPak only supports the client image tree of WorkSpace On-Demand 2.0, so we needed a little trick to apply it to the client image path. Change the local path d:\TREE in Step 4. on page 379 and in Step 6. on page 380 to D:\IBMLAN\RPL\BB20.US.

#### **Note**

Do not apply FixPak UN\_2200 when you need TCPBEUI support! At the time of the writing of this redbook, IBM WorkSpace On-Demand 3.0.1 OS/2 clients do not start when using TCPBEUI with MPTS 6.0 and TCP/IP 4.3!

#### ***LAN requester / IBMPEER FixPak IP08413***

When applying LAN FixPak IP08413, do not overwrite the following files with older versions from the FixPak:

- D:\TREE\IBMLAN\NETPROG\NETWKSTA.200
- D:\TREE\IBMLAN\NETLIB\UPMLAN.DLL

#### 7.2.4.2 FixPak sample response files for FSERVICE.EXE

Below are some sample response files for FSERVICE.exe:

##### **XR\_M014.RSP**

```
:LOGFILE D:\XR_M014.LOG
:FLAGS REPLACE_PROTECTED EXIT_WHEN_DONE
:SOURCE X:\FIXES\XR_M014
:SERVICE
:SYSLEVEL D:\TREE\OS2\INSTALL\SYSLEVEL.OS2
```

##### **XR\_D001.RSP**

```
:LOGFILE D:\XR_D001.LOG
:FLAGS REPLACE_PROTECTED EXIT_WHEN_DONE
:SOURCE X:\FIXES\XR_D001
:SERVICE
:SYSLEVEL D:\TREE\OS2\INSTALL\SYSLEVEL.BDD
```

##### **UN\_2200.RSP**

```
:LOGFILE D:\UN_2200.LOG
:FLAGS REPLACE_PROTECTED EXIT_WHEN_DONE
:SOURCE X:\FIXES\UN_2200
:SERVICE
:SYSLEVEL D:\IBMLAN\RPL\BB20.US\TCPIP\BIN\SYSLEVEL.TCP
```

##### **IP08413.RSP**

```
:LOGFILE D:\IP08413.LOG
:FLAGS REPLACE_PROTECTED EXIT_WHEN_DONE
:SOURCE X:\FIXES\IP08413
:SERVICE
:SYSLEVEL D:\TREE\IBMLAN\SYSLEVEL.REQ
:SERVICE
:SYSLEVEL D:\TREE\MUGLIB\SYSLEVEL.MUG
:SERVICE
:SYSLEVEL D:\TREE\MUGLIB\SYSLEVEL.UPE
```

---

## 7.3 Updating drivers

IBM WorkSpace On-Demand 3.0.1 includes support for a number of different types of client workstations and devices. However, it is likely that you will need to provide support for clients and devices that are not supported by the standard IBM WorkSpace On-Demand 3.0.1 product. This section describes how you can add support for different Networking and Video adapters to support new types of client workstation under IBM WorkSpace On-Demand 3.0.1. A step-by-step process is described for each of the different client Operating Systems.

### 7.3.1 Installing support for video adapters

Some video adapters may have become available after the operating system was shipped, so they are not included in the operating system image. To add support for video adapters, you have to perform a few steps for each operating system:

1. Ensure that the driver works with a stand-alone version of the operating system.
2. Add the video device driver files to the client image.
3. Update the response files to include the new driver.

#### 7.3.1.1 Windows NT

There are a few ways of installing a video driver for Windows NT; this example is just one of them. This example deals with installing an S3 Trio 3D card, which is found on some newer PC300GLs. It is an AGP video card. AGP is not supported by Windows NT out of the box, as the S33d video driver requires that the client already have Service Pack 4. During client installation, Windows NT does not have any Service Packs installed, so a workaround should be done to address this.

To set up the workaround, complete the following steps:

1. Go to \TDM\MM\CLIENT\RO\NT4\SP3\US\INST\I386.
2. On the same directory, issue the `DEL HAL.*` command.
3. If your Windows NT Server is already on the same Service Pack level, copy the HAL.DLL file.

After doing the workaround, we can begin installing the S33D video drivers:

1. Create a new directory named DISPLAY under  
  \TDM\MM\CLIENT\RO\NT4\SP3\INST\I386\SOEM\$.

2. Copy the contents of the S3 Trio 3D video drivers into the DISPLAY directory.
3. Create the machine on the console.
4. Edit the UNATTEND.TXT file, located at  
\TDM\MM\CLIENT\RO\NT4\SP3\US\DEFS\RESP\UNATTEND.TXT, to reflect the following changes:

```
[Display]
ConfigureAtLogon=0
BitsPerPel=16
XResolution=800
YResolution=600
VRefresh=60
AutoConfirm=1
InstallDriver =1
InfFile=S3TRIO3D.INF ; S3 Trio 3D VIDEO
InfOption="S3 Incorporated Trio3D Display Driver Version 3.26.28
Production Release"
```

For more information about the unattended install of video drivers, visit:

<http://support.microsoft.com/support/kb/articles/q166/0/28.asp>

### 7.3.1.2 Windows 2000 OEM plug-and-play adapters

Since Windows 2000 incorporates some of the plug-and-play technologies already found in Windows 98, the video setup procedure has changed significantly. In most cases, it is no longer necessary to specify the video adapter for the client. Instead, Windows 2000 will try to autodetect the adapter for you and automatically install the correct driver. If a suitable driver is not part of the Windows 2000 client image, you can add it by following these instructions:

1. Create an \$oem\$\\$1\Drivers\Video directory under the i386 directory of your client image (the \Drivers\Video name is suggested to keep the different types of drivers separate. It is possible, though not recommended, to put all drivers directly into the \Drivers directory). The "\$1" resolves to %System Drive%, e.g. \WINNT. During the text mode setup, its content gets copied to the \WINNT\Drivers directory).
2. Copy the OEM-supplied driver files for the device into the directory created in the previous step.
3. Add the following entry to the [Unattended] section of the response file:

```
OemPnPDriversPath = Drivers\Video
```

#### Note

You can list multiple paths in this key by separating them with a semicolon character (;). For example:

```
[Unattended]
OemPnPDriversPath = Drivers\Network, Drivers\Modem, Drivers\Video
```

The %SystemDrive% environment variable is automatically inserted before each of the search paths, so it does not have to be specified here.

4. If the OEM-supplied drivers are not digitally signed, you will receive a warning message during setup. To disable this message, add the following key to the [Unattended] section of the response file:

```
DriverSigningPolicy = Ignore
```

This procedure is valid for other plug-and-play device classes as well, such as modems, printers, and so on.

#### 7.3.1.3 Windows 98

Most of the time, Windows 98 can autodetect your video adapter. The only time you should do the following procedure is if your video adapter was not detected and your Windows 98 client keeps reverting back to VGA mode.

Windows 98 is a plug-and-play operating system, meaning it can automatically detect newly-attached devices since its last boot up. If Windows 98 finds the driver for the device on its database, Windows 98 will install it without much bother to the end user. However, there will be cases where the device is new and the driver is not yet included with the Windows 98 package. Windows 98 will then prompt the user for the device driver. The following procedure will allow automatic install of the video driver. For this example, we are making use of Matrox Millennium.

1. Go to the \TDM\MM\CLIENT\ROW98\W98SE\US\INST\C directory.
2. Create a directory called \$Display.
3. Copy the Matrox drivers to this directory.
4. Go to the \$Display directory and edit MGA.INI.
5. Find the following lines and change them to:

```
Reboot = no
Final_Box = no
Install_Type = Typical [AVAILLANG]
```

6. Remove the language files you do not need.
7. Close and save the document.
8. Go to \TDM\MM\CLIENTS\RO\W98\W98SE\US\DEFS\RESP and edit MSBATCH.INF.
9. Find [LOGONCLT.REG] and insert the following line:
 

```
HKLM, "Software\Microsoft\Windows\CurrentVersion\RunOnce", MGA, , "c:\$display\setup.exe"
```
10. Use the `machine define` command to create the client with the following parameters:
 

```
srcrespfile=MSBATCH.INF
```

#### 7.3.1.4 OS/2

The process for supporting additional video adapters in IBM WorkSpace On-Demand 3.0.1 is very similar to the WorkSpace On-Demand 2.0 process. If you are familiar with WorkSpace On-Demand 2.0, you will note that the only differences are in the directory structure on the server.

The following text describes how to support a Cirrus Logic 5434 video adapter on a Compaq Prolinea™ 5120e client machine. You can use this information as a guideline for installing similar video adapters.

To confirm that a particular client machine is an appropriate IBM WorkSpace On-Demand 3.0.1 candidate, define the IBM WorkSpace On-Demand 3.0.1 client using `video=VGA` and verify that you can start the machine from the deployment server. After you successfully create a machine definition, the GUI will show that adapter as a supported selection.

Determine which files are added or modified when you install the video adapter as follows:

1. To establish a baseline for determining which client-configuration files are changed or added, start with a Compaq Prolinea 5120e client machine installed with OS/2 Warp 4 and VGA support.
2. Verify that the Compaq Prolinea 5120e client machine boots.
3. Create a snapshot file that captures the name, size, and date for every file on the boot drive. For example:

```
ATTRIB -R -H */S
DIR C:\*.* /S /O:GN >PREVGA.DAT
```

4. Run the following command to remove the archive bit from all files on the deployment server:

```
ATTRIB -A C:\*.* /S
```

Removing the archive bit enables you to verify the files that are modified during installation.

5. To recreate the changes to the OS/2 system files, save a backup copy of the following files:

```
CONFIG.SYS  
AUTOEXEC.BAT  
OS2.INI  
OS2SYS.INI  
WIN.INI  
SYSTEM.INI
```

6. Use the Selective Install program, provided by the OS/2 Warp 4 installation facility, to install support for the Cirrus Logic 5434 video adapter.

**Note:**

To enable maximum video resolution flexibility, do not change the video resolution after installing SVGA support. This enables you to select the desired resolution from the list when you use the IBM WorkSpace On-Demand 3.0.1 administration GUI to define a client machine.

7. Create a second snapshot file to capture the name, size and date of each file on the boot drive after installing SVGA support. For example:

```
ATTRIB -R -H * /S  
DIR C:\*.* /S /O:GN >POSTVGA.DAT
```

8. To recreate the changes to the OS/2 system files, recapture the following files:

```
CONFIG.SYS  
AUTOEXEC.BAT  
OS2.INI  
OS2SYS.INI  
WIN.INI  
SYSTEM.INI
```

This enables you to determine the changes made during the installation of SVGA support. When you save these files into the same directory as the pre-SVGA files, rename the post-SVGA files to differentiate them.

9. Capture the names of all the other files that are added or changed by checking files for the changed archive bit as follows:

```
DIR C:\*.* /S /AA >ATTRIB.DAT
```

Now that you have identified the files affected by the Cirrus Logic 5434 video-adaptor installation, you can create a video definition that supports this configuration on the client machine:

1. Copy the VIDEO.CFG and SVGADATA.PMI files to the following location:

```
C:\TDM \MM \CLIENT \RO \OS2 \imagename \lang \TREE \OS2 \VIDEO
```

2. Create a video subdirectory for the video driver files under the \VIDEO directory:

```
cd C:\TDM \MM \CLIENT \RO \OS2 \imagename \lang \TREE \OS2  
\VIDEO  
md CL5434
```

3. Copy the video driver files CIRRUS.DLL, DISPLAY.DLL, BVHSVG.A.DLL, and VIDEOPMI.DLL into this directory. You might want to rename CIRRUS.DLL to CL5434.DLL if you are installing several versions of video drivers to support multiple video definitions.

4. Create a subdirectory for the video configuration files:

```
cd C:\TDM \MM \CLIENT \RO \OS2 \imagename \lang \VIDEO  
md CL5434  
cd C:\TDM \MM \CLIENT \RW \OS2 \imagename \lang \VIDEO  
md CL5434
```

The list of video files that must reside in this directory varies depending on the adapter. Typically, the list includes the following files:

```
CL5434.FIT (RO)  
CL5434.INF (RO)  
CONFIG.HW (RO)  
INI.RCH (RW)  
INISYS.RCH (RW)  
WIN.INH (RW)  
SYSTEM.INH (RW)
```

You append the information in these files to the existing system files when you define the machine (you are creating a new video definition). For example, the information in CONFIG.HW is added to the CONFIG.SYS file, and the information in OS2.RCH is added to the OS2.INI file.

5. Create FIT file extensions to point to the new video driver location. For example, the FIT file CL5434.FIT remaps the system as follows:



```
;video support for the Cirrus Logic 5434 driver

;writeable files
Z:\OS2 \PRIVATE.*\BBSRV01 \WRKFILES \DEFAULT \OS2 \IMAGENAME \US \OS2
Z:\OS2 \SVGADATA.*\BBSRV01 \WRKFILES \DEFAULT \OS2 \IMAGENAME \US \OS2
Z:\OS2 \VIDEO.*\BBSRV01 \WRKFILES \DEFAULT \OS2 \IMAGENAME \US \OS2

;readonly files
Z:\OS2 \DLL \BVHSVGA.DLL OS2 \IMAGENAME \US \OS2 \VIDEO \CL5434
\BVHSVGA.DLL
Z:\OS2 \DLL \CIRRUS.DLL OS2 \IMAGENAME \US \OS2 \VIDEO \CL5434
\CL5434.DLL
Z:\OS2 \DLL \DISPLAY.DLL OS2 \IMAGENAME \US \OS2 \VIDEO \CL5434
\DISPLAY.DLL
Z:\OS2 \DLL \VIDEOPMI.DLL OS2 \IMAGENAME \US \OS2 \VIDEO \CL5434
\VIDEOPMI.DLL
Z:\OS2 \DLL \IBMGPMI.DLL OS2 \IMAGENAME \US \OS2 \VIDEO \CL5434
\IBMGPMI.DLL
```

6. Create an .INF file that lists the video files required to complete the first phase of the client DHCP boot process. For example:

```
OS2 \IMAGENAME \US \OS2 \BOOT \SCREEN01.SYS
OS2 \IMAGENAME \US \OS2 \BOOT \SCREEN02.SYS
OS2 \IMAGENAME \US \OS2 \BOOT \VIOTBL.DCP
OS2 \IMAGENAME \US \OS2 \DLL \BVSCALLS.DLL
OS2 \IMAGENAME \US \OS2 \DLL \VIOCALLS.DLL
OS2 \IMAGENAME \US \OS2 \DLL \BVHSVGA.DLL
OS2 \IMAGENAME \US \OS2 \DLL \BVHVGA.DLL
```

7. To determine which CONFIG.SYS lines changed during installation, use a file comparison tool. Create a file named CONFIG.HW with the changed lines. The following segment shows the changed lines for this example:

```
REM *****NCVIDEO BEGIN *****
DEVINFO=SCR,VGA,Z:\OS2 \BOOT \VIOTBL.DCP
SET VIDEO_DEVICES=VIO_SVGA
SET VIO_SVGA=DEVICE (BVHVGA,BVHSVGA)
DEVICE=Z:\OS2 \MDOS \VVGA.SYS
REM *****NCVIDEO END *****
```

8. Use an .INI file editor to check for changes in the OS2.INI file. If there are changes, create an INI.RCH file (refer to the INI.RCH in the video subdirectory for an example). The INI.RCH file is used to update the OS2.INI file.
  - a. Use the DELTA4.CMD and COPYAPP.CMD files, in the following examples, to create a DELTA.INI file. DELTA4.CMD requires the OS2VGA.INI (renamed OS2.INI file from step 5 on page 274) and the

OS2SVGA.INI (renamed OS2.INI file from step 8 on page 274) file names as the input to the procedure. The DELTA4.CMD procedure searches for new or changed application data, keywords or data entries in the OS2SVGA.INI and places them into the DELTA.INI file. DELTA4.CMD calls COPYAPP.CMD.

You will find both CMD files on the redbook CD-ROM.

- b. Use an .INI file editor to edit the DELTA.INI file. Ensure that only the changes are placed in the new file. In our example, the following two application entries are required:

```
PM_DISPLAYDRIVERS
```

Cirrus Logic device information requires this application. Check for keywords, such as DEFAULTSYSTEMRESOLUTION. If keywords exist, delete them and then manually add RESOLUTION\_CHANGED with a data entry of 1.

The WIN-OS/2 ® environment requires similar entries to change to another video resolution.

9. Use an .INI file editor to check for changes in OS2SYS.INI file. For this example, the installation did not change any entries in the file, so update procedures are not necessary. If there were changes in OS2SYS.INI, you would create an INISYS.RCH file to update the OS2SYS.INI file.
10. Create a WIN.INH file. This is an ASCII text file that updates WIN.INI. You can use an ASCII text editor to determine changes to the WIN.INI file. Add only changes to WIN.INH. In this example, the following changes are added to WIN.INH file:

```
[Desktop ]  
IconSpacing=100
```

11. Create a SYSTEM.INH file. This is an ASCII text file that updates SYSTEM.INI. You can use an ASCII text editor to determine the changes to SYSTEM.INI. Add only changes to SYSTEM.INH. In this example, the following changes are added:

```
[boot ]  
display.drv=256_1280.drv  
sdisplay.drv=256s1280.drv  
  
[boot.description ]  
display.drv=Cirrus 54xx Accelerated 640x480x256  
sdisplay.drv=Cirrus 54xx Accelerated 640x480x256  
  
[256s1280.drv ]
```

```
fontSize=96
dacdepth=6
Enable216=0
WidthxHeight=640x480

[clvga ]
videomode=0x5f
logo=0
```

At this point, you can select the new CL5434 video type from the IBM WorkSpace On-Demand 3.0.1 administration GUI when you define an OS/2 client machine.

### 7.3.2 Installing support for network interface cards

Operating system designers have tried to integrate support for every known network card available on the market before their general announcements. OS and software developers test different types of network cards with their product prior to release. But due to time constraints, there are always network cards that do not make the list of supported network cards. This is why operating systems and software have their own list for supported network cards. IBM WorkSpace On-Demand 3.0.1 has its own list of supported network cards. Unlisted network card manufacturers bundle their device drivers along with their product. This section discusses the automated installation of these device drivers.

The following procedure describes how to add a device driver for an unsupported network card. The adapter's device drivers must support unattended installation. For details on tested network adapters, see Chapter 2, "Installation" on page 9.

This procedure assumes that you are adding a new adapter, called TRP.

#### 7.3.2.1 Adding network card drivers for Windows NT

Complete the following steps to add network card drivers:

1. Create a directory for the network adapter files. For example:  
C:\TDM\MM\CLIENT\RO\NT4\SP3\US\INST\I386\OEM\$\NET\TRP  
Apply access controls to any directory you create.
2. From the Windows NT adapter diskette, locate the device ID for the driver (most network interface card manufacturers use the OEMSETUP.INF file).
3. Copy the files to the  
C:\TDM\MM\CLIENT\RO\NT4\SP3\US\INST\I386\OEM\$\NET\TRP  
directory.

4. Copy the response file, UNATTEND.TXT, located in the C:\TDM\MM\CLIENT\RO\NT4\SP3\US\DEFS\RESP directory to IBMTRP.TXT.
5. Edit the IBMTRP.TXT file as follows:
  - a. In the [Network] section, change DetectAdapters="" to InstallAdapters = SelectedAdaptersSection.
  - b. Add the following sections after the [Network] section:
 

```
[SelectedAdaptersSection]
TRP= TRPPParamSection,\$OEM$\NET\TRP
[TRPPParamSection]
```

Using this method, you can prevent duplicate file names when adapters share a file name.
6. Use the machine define command to create the client with the following parameters:
 

```
srcrespfile=IBMTRP.TXT
netadapter=TRP
```

### 7.3.2.2 Configuring unsupported Windows 2000 network adapters

Since Windows 2000 incorporates some of the plug-and-play technologies already found in Windows 98, the setup procedure has changed significantly. In most cases, it is no longer necessary to specify the network adapter for the client. Instead, Windows 2000 will try to autodetect the adapter for you and automatically install the correct driver. If a suitable driver is not part of the Windows 2000 client image, you can add it by following these instructions:

1. Create an \$oem\$\\$1\Drivers\Network directory under the i386 directory of your client image (the \Drivers\Network name is suggested to keep the different types of drivers separate. It is possible, though not recommended, to put all drivers directly into the \Drivers directory.)
 

(The "\$1" resolves to %System Drive%, that is, \WINNT. During text mode setup, its content gets copied to the \WINNT\Drivers directory.)
2. Copy the OEM-supplied driver files for the device into the directory created in the previous step
3. Add the following entry to the [Unattended] section of the response file:
 

```
OemPnPDriversPath = Drivers\Network
```

#### Note

You can list multiple paths in this key by separating them with a semicolon character (;). For example:

```
[Unattended]
OemPnPDriversPath = Drivers\Network, Drivers\Modem, Drivers\Video
```

The %SystemDrive% environment variable is automatically inserted before each of the search paths, so it does not have to be specified here.

4. If the OEM-supplied drivers are not digitally signed, you receive a warning message during setup. To disable this message, add the following key to the [Unattended] section of the response file:

```
DriverSigningPolicy = Ignore
```

This procedure is valid for other plug-and-play device classes as well, such as modems, printers, and so on.

#### 7.3.2.3 Configuring unsupported Windows 98 network adapters

Video and network cards can be treated the same in the sense that they are both adapters. The IBM WorkSpace On-Demand 3.0.1 Administrator Guide recommendation is as follows:

“Do not specify a netadapter when you define a Windows 98 client. The plug-and-play functions of Windows 98 should detect the card and install the drivers automatically or prompt you for the appropriate drivers during the client boot.”

The only downside to this recommendation is that you still need to be at the user workstation to insert the necessary drivers. Most of the time, the drivers are autodetected. If the device is already supported by Windows 98, it will install the drivers automatically. But for cards that are unsupported, there is a procedure to automate network card installation. Although the IBM PCI Token-Ring card driver was shipped pre-installed with the product, we use it here as an example.

The following instructions are for enabling the IBM Turbo 16/4 Token-Ring ISA network adapter for Windows 98. These instructions distribute the driver within their own directories. In our lab setup, we did the following:

1. Locate the newest Windows 98 device driver diskettes from the adapter manufacturer. The adapter must support unattended installation. Refer to the manufacturer documentation for details about unattended installation.
2. Create a separate directory for each adapter under the \TDM\MM\CLIENT\RO\W98\W98SE\US\INST\NET directory.
3. Copy the diskettes for each adapter to the separate subdirectory you created for each adapter in step 1.
4. Next, update the response file. List the files in the device driver diskette and open the .INF file to obtain the value for the unique device ID of the driver. Write down the name so that you can use it in a later step.
5. Using any ASCII editor, open the MSBATCH.INF file in the \TDM\MM\CLIENT\RO\W98\W98SE\US\DEFS\RESP directory.
6. Under the [Network] section, fill in the value for Netcards= with the device ID you found in the .INF file. For example, in the case of the IBM Auto 16/4 Token-Ring ISA Adapter, the device ID is \*IBM1080. For this example, the second device ID is \*XYZ123.
7. Because you are installing more than one network adapter, you need to add a section, [NetCardsDirs], to the MSBATCH.INF file to indicate the corresponding directory where you installed each driver. For example:

```
[Network]
netcards=*IBM1080,*XYZ123
IgnoreDetectedCards=1
[NetCardsDirs]
*IBM1080=NET\IBMTRSR
*XYZ123=NET\XYZ
```

8. Save the MSBATCH.INF file in the \TDM\MM\CLIENT\RO\W98\W98SE\DEFS\RESP directory. The file you saved is the response file you use when you define clients. The file name is the value for the Srcrespfile parameter for the machine define command.

#### 7.3.2.4 Configuring unsupported OS/2 adapters

To configure unsupported OS/2 adapters, complete the following steps:

1. From the appropriate subdirectory on the OS/2 adapter diskette, copy the OS/2 device driver (for example, NSIDADAP.OS2) and the OS/2 NIF files (for example, NSIDADAP.NIF) to the d:\TDM\MM\CLIENT\RO\OS2\image\cc\TREE\IBMCOM\MACS subdirectory on the server hard disk, where image represents the directory image name chosen during the Windows NT client installation,

cc represents the two-character country code, and ndisadap represents the directory name you created for the adapter files.

From the appropriate subdirectory, copy the OS/2 message file (for example, NSIDADAP.MSG) to the d:\TDM\MM\CLIENT\RO\OS2\BB20\US\TREE\IBMCOM subdirectory.

**Note**

An .NIF file is required for IBM WorkSpace On-Demand 3.0.1 support. Some adapters might not have .MSG files or .NIF files. If .MSG and .NIF files are present, they might have different names than the device driver

2. Add an entry for the NDISADAP adapter to the NDISDD.PTO control file in the \TDM\MM\CLIENT\RO directory. The entry then appears in the listbox when you create a requester using the LAN Server Administration GUI. A sample entry follows:

```
~ ~ NDISADAP.NIF ~ DHCP
```

where

- a. Field one represents a reserved field. The tilde ( ~ ) is a field position holder.
- b. Field two represents a reserved field. The tilde ( ~ ) is a field position holder.
- c. Field three represents the name of the .NIF file associated with the OS/2 device driver.
- d. Field four represents a reserved field. The tilde ( ~ ) is a field position holder.
- e. Field five specifies the remote IPL protocol that the card supports. The only valid option is DHCP.

You now can select the new server records when defining remote IPL client machines. The server record selected determines the files that are used to configure the remote IPL client machines.

---

## 7.4 Modifying a preboot image

The default preboot image is TDMW32UT.IMG and is located in C:\TDM\MM\CLIENT\RO\NT4\<IMG>\US\BOOT for NT4 clients, in C:\TDM\MM\CLIENT\RO\W2K\<IMG>\US\BOOT for Windows 2000 clients and in C:\TDM\MM\CLIENT\RO\W98\<IMG>\US\BOOT for Windows 98 clients.

The preboot image is a DOS 2.88MB boot diskette image. This image is used during the boot process to start a small DOS environment for STATE.EXE (see Section 4.2.3.4, “The state.ini file” on page 116 for Windows NT or Section 4.3.3.4, “The state.ini file” on page 135 for Windows 98).

The default preboot image TDMW32UT.IMG consists of the following files (as shown in Figure 184):

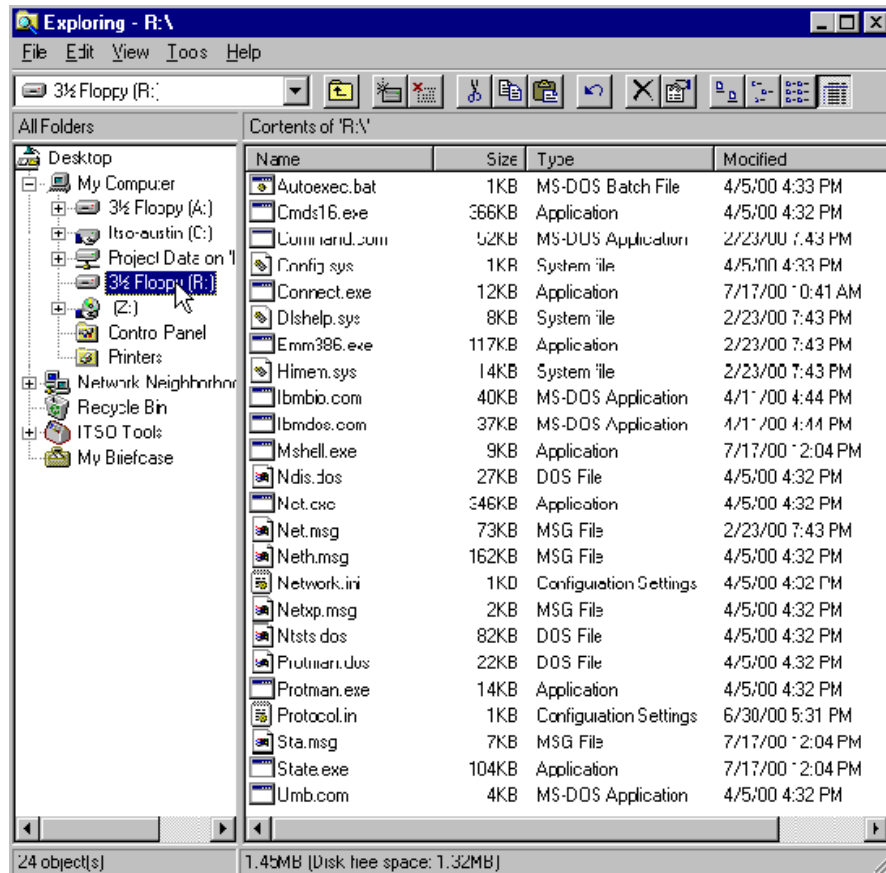


Figure 184. Contents of preboot image TDMW32UT.IMG

The easiest way to modify this image is to use a virtual floppy disk driver like RamDiskNT, which is available as shareware from John Lajoie Consulting (<http://www.jlajoie.com/ramdiskNT>) or VFDDISK for OS/2. To unpack the image file to the virtual floppy, you need an image tool like DSK4WIN or DSK4PM from Daniel Valot (<http://perso.libertysurf.fr/dvalot/emtcopy.htm>). The



image tools, virtual floppy drivers and installation instructions are provided on the redbook CD.

The process to modify a preboot image consists of these steps:

1. Install a virtual floppy disk driver.
2. Unpack the default preboot image to the virtual floppy using an image tool.
3. Modify the contents of the virtual floppy.
4. Use an image tool to create a new preboot image from the virtual floppy.
5. Copy the new preboot image to the \BOOT subdirectory of the client image tree installed in \TDM\MM\CLIENT\RO
6. Specify the new preboot image name as parameter PrebootImage during machine creation.



---

## Chapter 8. Migration and product services

The installation and deployment of IBM WorkSpace On-Demand 3.0.1 and associated client operating systems on new servers and desktops has been covered in the preceding chapters. This chapter will cover some of the considerations that have to be made when planning to migrate from an existing Server Managed Clients environment (such as IBM WorkSpace On-Demand 2.0) to 3.0.1. We will also outline the services available from IBM which can help you make the move.

---

### 8.1 Migration planning

The first step of any project is to manage the effort properly so that there is assurance that the right amount of time, resources and skills are available for a successful migration.

With the help of the project process outlined, as well as a number of example questions we have found useful, you should be able to manage the migration project with confidence.

We recommend that you start out with a pilot involving just a single or a few servers with roughly 10 to 15 clients. The lessons learned doing this pilot will come in handy as you prepare for the bigger roll-out, and thus help you make a smoother transition.

#### 8.1.1 Requirements phase

The requirements phase is perhaps the most important phase of the entire project plan. Too often, though, the project team does not spend adequate time gathering requirements. There seems to be a need to hurry forward and begin the implementation of the application as soon as possible. It is very important that sufficient time be devoted to collecting, discussing, documenting and verifying the requirements. Since this phase is the foundation upon which all the other phases are built and developed, problems in later phases can be avoided by spending adequate time in this phase. In the end, gathering requirements early in the project is much more productive than correcting fundamental flaws near the deadline.

##### 8.1.1.1 Functional requirements

Functional requirements refer to all the business requirements that you expect to be met in the final production environment. This process is an iterative one. Therefore, multiple meetings and interviews may be necessary to ensure all details are documented and agreed upon before proceeding to

the next step. The process involves breaking down the highest level functionality of the platform into more discrete functional units. The final goal is for these units to be grouped into a logical hierarchy that completely defines the managed environments combined functionality.

The following questions will help you gather the functional requirements:

1. Describe your current managed environment:
  - a. Are you migrating from an older version of IBM WorkSpace On-Demand to version 3.0.1?
  - b. Are you changing client operating systems? If so, are all your applications compatible with the new client OS?
  - c. Outline the key areas of functionality you want the new environment to have. Are these similar to what you have in your existing environment?
2. Describe your understanding of IBM WorkSpace On-Demand 3.0.1 with respect to functionality:
  - a. Is this your first deployment? If so, we suggest that you keep this book handy as well as attend the IBM WorkSpace On-Demand 3.0.1 Workshop prior to starting the project.
  - b. Do you believe 1a can be achieved? If not, now is the time to either find out if it can be done or to reset expectations.
3. Is there a requirement for custom development?
  - a. Do you have JavaScript experience? Even if you do, you may find the scripts on the CD that accompanies this book useful.
  - b. Can you write batch files?
  - c. Describe your IBM WorkSpace On-Demand skill level and any IBM WorkSpace On-Demand training that the programmers may have attended.
4. What are the applications that will be deployed?
  - a. List the business critical applications to be migrated from the existing managed environment. Also, include a list of new applications to be added to the new environment.
  - b. Will the applications be server-based or installed locally on the client systems?
  - c. Provide an architectural overview of each application
  - d. Describe both the server and client components of the application and function provided, especially related to managing users, groups, domains, and so on.

The time it will take to complete a IBM WorkSpace On-Demand migration project is highly dependent on the amount of custom code that needs to be created and implemented for the project. In some cases, all the functionality required for the project is available out-of-the-box. In many cases, it is in your interest to create custom scripts, as that will help you save time in the long run. It is also possible to streamline the administration of the IBM WorkSpace On-Demand 3.0.1 environment. The time required to implement custom code is a function of the number of scripts and other coding required, the amount of out-of-the-box code that can be leveraged, the complexity of the scripts and commands, and the skill level of the developers.

#### **8.1.1.2 Technical requirements**

Technical requirements consist of the low-level requirements that are needed for a IBM WorkSpace On-Demand 3.0.1 deployment. The process of gathering technical requirements can be performed either in parallel to the functional requirements gathering or afterwards. The decision rests on the logic of the process. As with functional requirements gathering, it is important to go through the findings several times in order to refine the findings.

Please see the product documentation for the minimum technical requirements for IBM WorkSpace On-Demand 3.0.1 itself, and use this documentation to base your decisions on whether to upgrade or keep existing equipment. Some of the questions in the Architecture Design section below can be used as a helpful reminder of the areas you should be considering.

### **8.1.2 Design phase**

Like all types of projects, the design phase is usually the most critical phase after the requirements have been gathered. It is often stated that for every hour spent in design, you save several hours during the implementation phase. As a result, the importance of the design phase cannot be understated.

#### **8.1.2.1 IT architecture design**

You should now take all the information gathered and use it to further define and document the key system components, such as physical servers, network configuration, database and software packages to be added. This will ultimately make up your design plan.

These questions will help you work through the design phase:

1. What type of systems will be used as the IBM WorkSpace On-Demand 3.0.1 server?
  - a. Make and model of system(s) (for example: 300GL and 300PL).

- b. Type of network adapter.
  - c. Will a separate DHCP server be used, or will you use the one shipped with IBM WorkSpace On-Demand 3.0.1?
  - d. Are there any security issues that need to be taken into, that is, can the migration team create Admin IDs on the domain?
2. What type of system(s) will be used as IBM WorkSpace On-Demand 3.0.1 clients?
- a. Make and model of client system.
  - b. Type of video card.
  - c. Type of network adapter.
  - d. Is the adapter PXE enabled? If so, what level of PXE?
  - e. What size are the hard drives?
  - f. Will any of the clients need to access servers outside of the environment/domain?
  - g. Is it possible to flash-upgrade system BIOS and adapters on the clients?
3. What does/will the network topology look like?
- a. Will the environment be connected to or separate from the site backbone?
  - b. Is it Token-Ring or Ethernet?
  - c. Type of bridges.
  - d. Type of routers.
  - e. How many IP segments will be used?
  - f. Will the client access a backbone, ATM, or T1?
  - g. Are any switches on the network? If so, what type?
  - h. What speed will the network be running?

It is important to note that a IBM WorkSpace On-Demand 3.0.1 design should also include a script control system that tracks changes in the scripts used.

---

## 8.2 Testing phase

As mentioned in the beginning of this chapter, we recommend that you start out with a pilot deployment. Testing on a small scale will help you iron out any problems before the final roll-out.

It should be no surprise that not all areas of the system will be as clearly specified as possible before the testing phase. So, it is necessary to have an effective plan for handling enhancements and gray areas to contain the scope and time frame of the project.

Enhancements generally arise from the learning process that all team members undergo as the project develops. As plans are put into requirements and designs become implemented modules, better ways to make the system work become apparent. Often, just the addition of testers to the team as a whole will bring in new perspectives that have not been considered in earlier phases of the project. In addition, some initial designs will be found too awkward to implement.

You need to establish a testing infrastructure from the start so that you can ensure that you have the capability to perform stress and performance profiling before and after launch. We suggest that you do the following tests as a minimum:

#### **Unit tests**

Unit tests should be performed in conjunction with development. This allows you to test specific functions, such as network connectivity.

#### **System tests**

The system test environment allows multiple clients to be connected to one or more servers and then run as they are expected to in the production environment.

#### **Stress test**

Establish the stress limits for your implementation and document these limits. If these do not meet your expectations, go back to the IT Architecture Design Phase.

#### **Acceptance test**

This type of testing involves end-user tests. For example: when clicking on an application icon on the desktop, it loads without errors.

### **8.2.1 Deployment phase**

The deployment phase is the true test of how well you have done the preparatory work. One managed environment has to be migrated to a new

one and, ideally, as painlessly as possible. This phase includes installation, configuration, and the final set up of the production environment.

---

## **8.3 Migration services**

A number of services are available from the IBM e-Business Operating Systems Solutions (e-Boss) Services team (which is based in the US but can be called on-site worldwide). We will briefly cover some of these offerings which can help you take a more hands-off approach when migrating to the IBM WorkSpace On-Demand 3.0.1 environment.

### **8.3.1 Independent project review**

It is beneficial to have an independent project review conducted prior to a major roll-out in order to ensure that the project will indeed be a success. Specifically, you can take advantage of the vast experience of software implementation projects that consultants from the Rapid Deployment Team (RDT) have.

A consultant from the RDT can review your project plan and documentation remotely, or work directly with your team on-site. In either case, a comprehensive analysis can be carried out. Here are just some of the variables the consultant will explore to make sure that possible risks and pitfalls are identified:

- The project plan itself
- Network topology
- Hardware environment
- Software environment
- Change management process

The result: a deliverable which could be a presentation, or a document outlining project strengths as well as potential gaps in the project plan. In case of the latter, a number of recommendations on how to improve the plan would also be included.

### **8.3.2 Product services**

As mentioned in Chapter 1 of this book, there are a number of services available for customers requiring more than out-of-the-box functionality. This includes support for additional client operating systems, such as Windows 95 and DOS, support for additional network adapters, and so forth.



### **8.3.3 Customized product enhancements**

There are a number of services offerings available which can extend the functionality of IBM WorkSpace On-Demand 3.0.1, including:

- Support for additional network adapters
- Customized deployment tools
- Remote discovery of MAC address
- Remote modification of client BIOS to enable network boot

### **8.3.4 Find out more about migration services**

You can learn more about the migration services that are available by looking at <http://www-4.ibm.com/software/os/services/> or by writing to [ncsdsvcs@us.ibm.com](mailto:ncsdsvcs@us.ibm.com).

In Europe, you can also contact The IBM EMEA Warp Services team at [emeawarp@de.ibm.com](mailto:emeawarp@de.ibm.com) for assistance in planning and performing any type of IBM WorkSpace On-Demand 3.0.1 deployment.



---

## Chapter 9. Comparisons

IBM WorkSpace On-Demand 3.0.1 and the Tivoli management agent (TMA) represent two different approaches to software distribution. The two products integrate to provide a set of unparallel client management features. Although it may appear as if there is some overlap between the two products, they are, in fact, distinctively different. Figure 185 outlines some of these areas of perceived overlap.

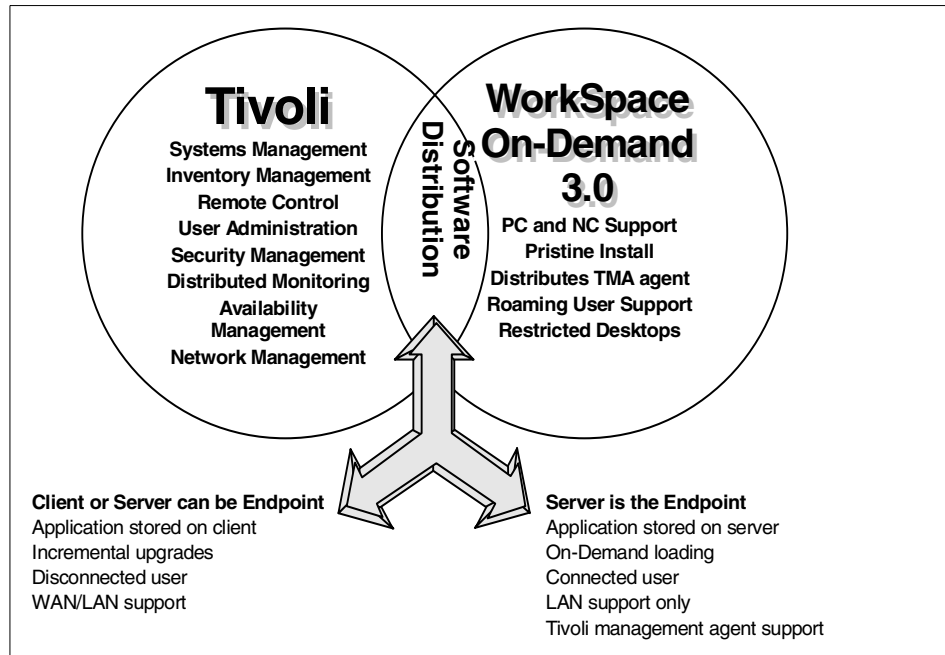


Figure 185. IBM WorkSpace On-Demand 3.0.1 and TMA

This chapter provides an overview of the differences between the two software distribution models, briefly describes how the Tivoli management agent works and how it integrates with IBM WorkSpace On-Demand 3.0.1, and also outlines the benefits of deploying both products.

---

### 9.1 Differences between IBM WorkSpace On-Demand 3.0.1 and TMA

The IBM WorkSpace On-Demand 3.0.1 software distribution model is covered in detail in Chapter 1, "Introduction" on page 1, but here is a quick summary of the main features:

IBM WorkSpace On-Demand 3.0.1 provides an infrastructure by which both NCs and PCs (Thin Clients and Better Managed Clients) can be managed. A pristine install of a client operating system, as well as applications and the Tivoli management agent, is provided from the IBM WorkSpace On-Demand 3.0.1 deployment server. This enables an organization to easily roll out transaction-oriented clients with restricted desktops and on-demand loading of operating systems and applications. This allows users to roam and always have access to the same applications. Security is enhanced and training costs are lower than on traditional fat client machines, since users will have a consistent Graphical User Interface.

Tivoli, on the other hand, is focused on general software distribution and hardware monitoring. By using Tivoli in conjunction with IBM WorkSpace On-Demand 3.0.1, it is possible to:

- Shield administrators from platform-specific details of day-to-day operations. Common operations, such as deploying server OS updates to the IBM WorkSpace On-Demand 3.0.1 servers, as well as routine network maintenance, can be performed with a single action.
- Administrators are no longer required to repeat the same operation for each platform in the enterprise.
- Integrate with IBM WorkSpace On-Demand 3.0.1 as well as third-party applications, because the Tivoli Management Framework is an open solution. The Tivoli Partner Association provides many of these third-party solutions to enterprise management challenges, such as job scheduling, intrusion detection, and backup and restore, all of which snap into the Tivoli Management Framework.
- By providing a truly open and comprehensive foundation for network computing management, the Tivoli Management Framework simplifies today's most complex network computing challenges.

---

## 9.2 The IBM WorkSpace On-Demand 3.0.1 and Tivoli combination

When combining IBM WorkSpace On-Demand 3.0.1 and Tivoli, it is possible to:

- Manage IBM WorkSpace On-Demand 3.0.1 servers with Tivoli.
- Integrate IBM WorkSpace On-Demand 3.0.1 into the Tivoli Console via the Plus Module.
- Install the IBM WorkSpace On-Demand 3.0.1 to servers via Tivoli Software Distribution.
- Manage both 'fat' and 'thin' client machines in a consistent way.

### 9.2.1 Introducing the TMA

The most visible new feature of Version 3.6 of the Tivoli Management Framework is the Tivoli management agent (TMA), previously called the Lightweight Client Framework (LCF) endpoint. The TMA is an extension of the classic TME 10 Framework that increases scalability of TMRs, while reducing the hardware and software requirements on the managed systems.

The TMA-related extensions to the framework introduce three object types that represent system roles in a TMR:

1. Endpoint (TMA)
2. Endpoint Gateway
3. Endpoint Manager

Although each of the above systems logically represents a different system's role in the Tivoli environment, it should be noted that a single physical system can contain more than one of the above object types. That is, one system could contain an endpoint manager, an endpoint gateway, and an endpoint. (In most environments, endpoints gateways reside on different systems than endpoint managers.)

### 9.2.2 The endpoint (TMA)

The endpoint is installed on the systems to be managed. The endpoint does not include any capability to perform management operations on other systems. These systems are managed and are not involved in the management of other nodes. The endpoint function resides in the node to be managed. It runs as a small daemon or background task. This daemon is called `lcfd`. It is responsible for executing methods at the request of a managing system. Its only connection to the rest of the Tivoli world is through an endpoint gateway.

When an endpoint is installed, a minimal number of files are installed on the managed system. Functionally, the only thing that is installed is the `lcfd` itself. When an application invokes a method to be executed on the managed system (endpoint), the method is automatically downloaded to the endpoint and the `lcfd` is executed.

### 9.2.3 The endpoint gateway

The endpoint gateway is a software component that runs on a full Tivoli managed node, enabling the managed node to operate as a gateway between a cluster of endpoints and the rest of the TMR. Each TMR can have

multiple endpoint gateways. Currently, one TMR server can handle up to a approximately 200 endpoint gateways.

#### **9.2.4 The endpoint manager**

The endpoint manager stores the association between the endpoint gateways and endpoints. It establishes and maintains the relationship between an endpoint and a gateway and is automatically created on the TMR server when you install the server.

An endpoint manager's primary role is to assign an endpoint to a gateway when the endpoint first logs in. If an endpoint's assigned gateway stops responding to the endpoint for some reason, the endpoint manager might again be involved in assigning the endpoint to a new gateway.

#### **9.2.5 The role of the TMA**

The Tivoli management agent (TMA), in conjunction with the Tivoli Framework and Management applications, provides the customer with the framework necessary to perform management operations, such as software distribution, inventory, user administration, and distribution monitoring. TMA is already provided with IBM WorkSpace On-Demand 3.0.1. The preloaded TMA allows the customer to immediately use the management features provided by Tivoli. Windows clients with the Feature for Windows Clients installed have the option of having the TMA within their software package (it is not installed by default). This means the lcf daemon could already exist on the user's hard disk. However, by default, the preloaded TMA engine (the lcf daemon) does not start automatically, since it will only be of value if the Tivoli Framework is installed in the customer's environment. Therefore, when the client is installed, the TMA (lcf daemon) exists under the \Tivoli\lcf directory in an inactive state.

#### **9.2.6 TMA installation prerequisites**

The following is a list of the management objects required to install and use endpoints:

- TMR server - The Tivoli Management server component includes the libraries, binaries, data files, and graphical user interfaces needed to install and manage the Tivoli environment.
- Endpoint manager - The endpoint manager software runs on the TMR server. The endpoint manager maintains the information related to known endpoints and endpoint gateways.

- Endpoint gateway - The endpoint gateway provides the primary interface between a set of endpoints and the rest of the TMR. A single endpoint gateway can support communications with thousands of endpoints.
- Endpoints - The endpoints are managed systems taking advantage of the Lightweight Client Framework. You can gather required management information from thousands of endpoint machines and remotely manage those machines with very little overhead. The endpoint is officially referred to as the Tivoli management agent (TMA) in Version 3.6 of Tivoli.

### 9.2.7 Enabling the TMA

This section describes the steps that need to be taken to enable the TMA. Figure 186 shows the various components within a Tivoli environment and their interaction.

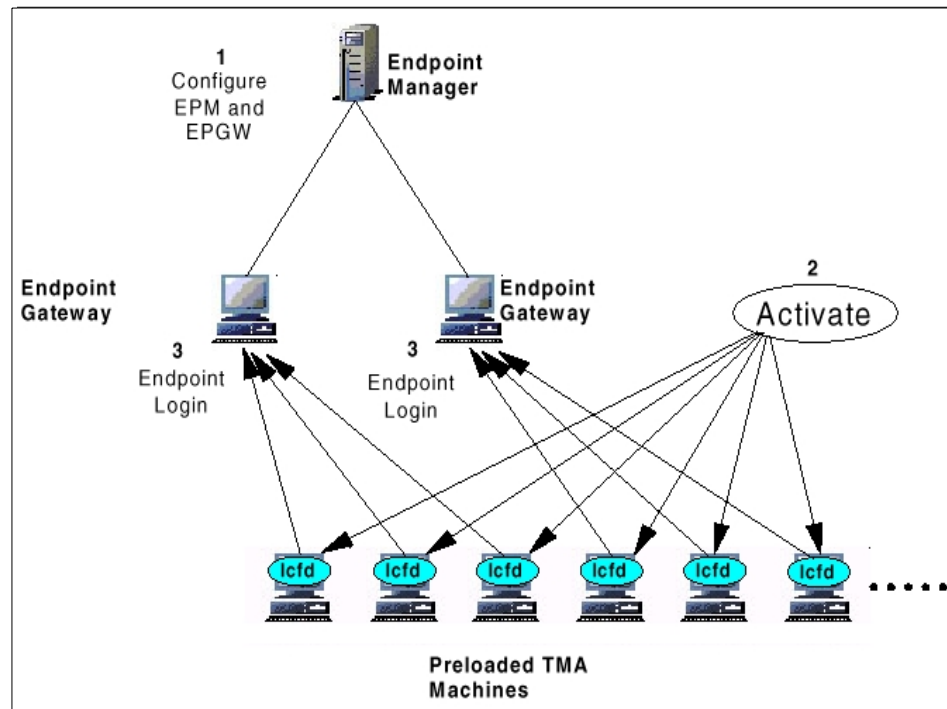


Figure 186. Steps to enable the TMA

To enable the TMA, you need to complete the following steps in your environment:

1. Create and configure the endpoint manager and endpoint gateways.

2. Activate the preloaded TMA.
3. Endpoint login—The TMA logs in to the appropriate endpoint gateway that you configured during the activation process. Then, you can activate the preloaded TMA with the appropriate options. The active process of the preloaded TMA allows you to specify options for the lcf daemon, so you can configure and control the endpoint login for all of the endpoints.

When the endpoint is started, it performs the endpoint login process to participate in the TMR. This is illustrated in Figure 187 on page 413. The login process is as follows:

1. The endpoint establishes communications with the TMR.
2. The endpoint manager selects the endpoint gateway for the endpoint.
3. The endpoint receives its gateway assignment information and performs the normal login to the assigned gateway.



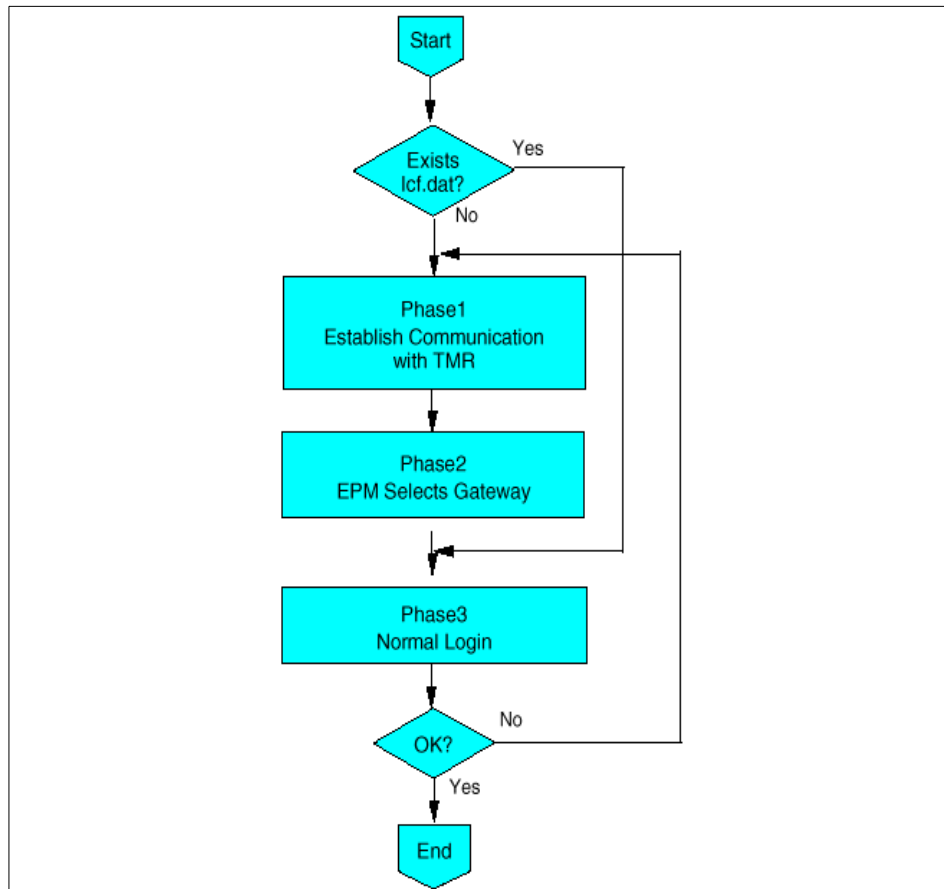


Figure 187. The endpoint login process

In the classic Tivoli management environment, the installation process implicitly determined the TMR to which a manager system belonged; there was no requirement to explicitly configure the managed nodes to communicate with their TMR server.

In the TMA environment, the endpoint must be configured to communicate with an appropriate endpoint gateway. To allow for both flexibility and availability, the association between the endpoint and the endpoint gateway can be dynamically determined and may change if a particular endpoint gateway is not available.

After receiving its gateway assignment from the intercepting gateway, the endpoint performs the normal login to its assigned gateway. At this time, the

login policy is run for the first time, and the endpoint codeset is downloaded to the endpoint. The endpoint is now ready to participate in Tivoli management operations.

### 9.2.8 Installing the TMA

The Tivoli management agent (TMA) is included as part of IBM WorkSpace On-Demand 3.0.1.

The IBM WorkSpace On-Demand 3.0.1 install program installs the TMA files in the \$TIVOLI directory for each operating system type on the deployment server. The TMA install program is added to the list of OEM software to be installed for each operating system response file provided. (TMA is installed on clients through commands in the default response files MSBATCH.INF for Win9X clients or CMDLINES.TXT for Windows NT clients.)

There are three sets of client agents:

- Client agent for Windows NT 4.0 and Windows 2000
- Client agent for Windows 9x (95 and 98)
- Client agent for OS/2

The OS/2 TMA client agent, like the rest of the OS, is run from the server; thus, no special installation is required.

Each of the Windows client agents have their own installation program, setup.exe. These agents can be manually installed on the client by typing setup. Alternatively, they can be installed in an unattended mode by using the setup program with the -s option:

```
Setup -s <path to response file>
```

This command invokes an unattended install.

By default, IBM WorkSpace On-Demand 3.0.1 installs the Tivoli management agent Version 3.6. The directory trees for the Windows client TMA on the deployment server are as follows:

```
IBMLAN\RPL\W95\C\TIVOLI for Windows 95
IBMLAN\RPL\W98\C\TIVOLI for Windows 98
IBMLAN\RPL\NT40\I386\SOEM\C\TIVOLI for Windows NT
IBMLAN\RPL\BB20\TMA for OS/2
```

By default, the TMA software is not installed on clients. If you want to use TME 10 products, you need to modify the default response file and

uncomment a line on the MSBATCH.INF file on Windows 9X clients or the CMDLINES.TXT on Windows NT clients, and the TMA will be installed.

To install the TMA on Windows NT clients:

1. Change to X:\BMLAN\RPL\NT40\I386\SOEM\$. (X represents the drive letter on the boot server where you installed your Windows NT operating system.)
2. Open the CMDLINES.TXT file.
3. At the bottom of the file, uncomment (remove the ; from) the following line under the [Commands] section:

"C:\\$TIVOLI\SETIVOLI.CMD"

The directory of the TMA agent for Windows NT is C:\Program Files\Tivoli\lcf, and the location of the lcf daemon is C:\Program Files\Tivoli\lcf\bin\w32-ix86\mrt

## 9.2.9 Configuring the clients

### 9.2.9.1 Windows 98/NT/2000 clients

To configure clients with the correct port and address information to connect to Tivoli gateways, update the InstallShield setup files for the Tivoli management agent (TMA). To configure the Tivoli management agent:

1. Log on to the deployment server as an administrator.
2. Open the file SETUP.ISS in any ASCII text editor. This file can be found in the places noted in Table 12:

Table 12. Location of the SETUP.ISS file

| Operating System | Path to SETUP.ISS                   |
|------------------|-------------------------------------|
| Windows NT       | \TDM\MM\CLIENT\RO\NT4\TMA\SETUP.ISS |
| Windows 2000     | \TDM\MM\CLIENT\RO\W2K\TMA\SETUP.ISS |
| Windows 98       | \TDM\MM\CLIENT\RO\W98\TMA\SETUP.ISS |

3. Find the following section:

```
[SdShowDIgEdit3-0]
szEdit1=9494
szEdit2=9494
szEdit3=d1
Result=1
```

Change the settings to represent the port and IP address of your Tivoli gateway. For example, to configure a client with port 9999 and an IP address of 9.3.169.88, type:

```
[SdShowDIgEdit3-0]
szEdit1=9999
szEdit2=9999
szEdit3=-g 9.3.169.88+9999
Result=1
```

See the documentation for Tivoli Framework 3.6 for more information.

#### 9.2.9.2 OS/2 clients

To configure the TMA endpoint for OS/2 clients, execute the following commands from a CMD file:

```
x:
cd \TIVOLI\LCF\DAT\1
X:\TIVOLI\LCF\BIN\OS2-IX86\MRT\LCFD.EXE -g<ip-address>+<port>
```

where <ip-address> and <port> describe the Tivoli gateway machine that the client should connect to.

#### Note

A suitable place for starting the TMA client is the user defined logon script for the PMLOGON shell. See the IBM WorkSpace On-Demand 3.0.1 Administrative Guide for a description of this feature.

### 9.2.10 Stopping and starting the TMA

Occasionally, you may need to manually stop or restart an endpoint. These are the manual procedures.

#### 9.2.10.1 For Windows NT/2000 clients:

- From the command line:

Use the `net start lcf` or `net stop lcf` command to start or stop NT endpoints.

- From the desktop:

Select **Control Panel** -> **Services**, then start or stop the Tivoli endpoint service.

To connect the TMA to a specific gateway:

- From the command line:

Use the `lcf start -g Gateway_IP_Address` command. For example: `lcf start -g 9.12.12.42`

To remove an endpoint on a Windows NT machine:

1. Stop and remove the Endpoint service by issuing the following command:

```
C:\Program Files\Tivoli\lcf\unist
```

2. Follow the instructions on the screen.

#### **9.2.10.2 For OS/2 and Windows 98 clients**

To stop the TMA client, simply Stop the LCFD.EXE process by killing it from the task list.

To start the client again, run LCFD.EXE with the appropriate parameters.



### A.1 IBM WorkSpace On-Demand 3.0

#### At a Glance

An IBM cross-platform, server-based client management solution

#### Features:

- Restricted user access to client desktops
- Support for user roaming within the network
- Software and maintenance updates delivered and managed from the server
- Support for legacy client platforms — Windows 98, Windows, NT™ 4.0, OS/2®
- Support for new client platforms — Windows 2000
- Support for a broad range of applications
- Support for the new, Java™-based application model

#### Benefits:

- Centralized and remote client management
- Easy maintenance and deployment of applications
- Low cost of ownership
- Broad access to servers
- Helps protect and extend investments
- Smooth migration to Java-based applications and solutions

#### For ordering, contact:

- Your IBM representative, an IBM Business Partner, or IBM.
- Americas Call Centers at 800-IBM-CALL (Reference: SE001).

---

## A.2 Overview

### What It Is

IBM WorkSpace On-Demand 3.0 is a cross-platform middleware solution that provides server-based client management for a broad range of client hardware, platforms, and applications.

### What It Does

IBM WorkSpace On-Demand 3.0 enables you to manage a variety of heterogeneous clients — Windows NT 4.0, Windows 2000, Windows 98, and OS/2 Warp from Windows NT 4.0 or Windows 2000 servers.

### How It Benefits You

IBM WorkSpace On-Demand 3.0 combines the best of two worlds — the compelling value proposition of thin-client computing and the local processing power of Intel®-compatible clients.

### Statement of Direction

IBM will entertain requests for quotations for Linux client support on IBM WorkSpace On-Demand 3.0 pursuant to a services engagement. IBM will consider making Linux client support available in the future as a product.

### Key Prerequisites

You must obtain the appropriate licenses for:

- All server operating systems
- Connection or attachment to Microsoft Windows NT 4.0 or Windows 2000 servers
- All Microsoft™ Windows™ client operating systems to be managed

### Planned Availability Dates

July 31, 2000: IBM WorkSpace On-Demand 3.0 (with support for NT Server 4.0)

October 27, 2000: IBM WorkSpace On-Demand 3.0 (with support for Windows 2000 server and clients)



---

## **A.3 Description**

### **Laying the Groundwork**

Today's network computing enterprise requires an open, scalable, integrated cross-platform approach. The Application Framework for e-business is a multi-tier distributed information technology environment, based on open-industry standards, that integrates Internet technologies with traditional information technology. The Managing Technology segment of the Application Framework for e-business includes integrated solutions for managing a secure, reliable, and scalable e-business infrastructure. It includes Tivoli management solutions that address performance and availability, security, storage; change, asset, operations and service management, and e-business application management. IBM WorkSpace On-Demand 3.0 is a key component of the Managing Technology segment.

### **A Fresh Approach**

The traditional computing model is called a "fat-client" model because the client data, applications and operating system reside locally on the client hard drive. The so-called "thin-client" computing model is a new computing paradigm. The most common form of this network-based model centralizes a client's data, applications and operating system on the server and provides access to them over the network through "thin" devices, such as Network Computers (NCs). While this form of computing leverages the power of the network, IBM WorkSpace On-Demand 3.0 takes a slightly different approach by leveraging the local processing power of Intel-based clients.

### **New for IBM WorkSpace On-Demand 3.0**

IBM WorkSpace On-Demand 3.0 is designed to allow various resources to be managed independently. For example: the management of a client machine is separate from that of users or applications. This allows an administrator to define the operating system to be deployed on a client machine regardless of who will use it. IBM WorkSpace On-Demand 3.0 supports pure remote boot and remote boot/install. Both can be done on a pristine machine, that is, a new machine with nothing on the hard drive. “Pristine install” refers to remote boot/install. The pristine client boots from the server, IBM WorkSpace On-Demand 3.0 brings a complete operating system down to the client, and installs it locally. Both methods use a server-managed image of the client operating system and employ the standard DHCP/PXE boot protocol.

### **Thin and Lean Clients**

Server-managed clients include thin and lean clients, both of which can be managed by IBM WorkSpace On-Demand 3.0. While pure remote boot is used on thin clients, remote boot/install is used on lean clients. The distinction between a thin client and a lean client lies in whether the operating system is stored on the client, regardless of what the underlying client hardware is. In a thin client implementation, the user's operating system, data, and applications all reside on the server and are loaded into memory and run on demand on the client.

For a lean-client user, IBM WorkSpace On-Demand 3.0 stores the server-based operating system image on the client hard drive, while maintaining the user's data and applications on the server. Whenever the client is turned on, it sends a remote boot request to the server. If the server-based operating system image remains the same, the client will boot locally. If the server-based operating system image has been updated, the client will boot remotely, and a copy of the new operating system image will be downloaded and replace the old one on the client. The user's data and applications are still loaded and run on demand on the client.

Both types of users can log on from any client in the network (albeit application-dependent) and have immediate access to their personalized desktops.

### **PC and NC Roles**

In general, NCs are used more as thin clients, while PCs are better candidates for lean clients. However, their roles can be switched. If necessary, a PC can function as a thin client, and an NC can act as a lean client. Although NCs do not usually come with a hard drive, a hard drive can be optionally installed on an NC, or a non-volatile memory card can be

installed, effectively taking the place of a hard drive. In this sense, the rationale for categorizing server-managed clients into “thin” and “lean” is clearly software-related rather than hardware-related.

---

## **A.4 Key Requirements of Network Computing and e-business**

### **Centralized Management and Control**

IBM WorkSpace On-Demand 3.0 is designed so that little or no data resides on the client desktop, and applications are delivered from the server. Placing control at the server can significantly reduce the risks and support costs associated with traditional client PCs.

Storing data on a centralized server can simplify data management and synchronization, and also protect it from either subversive or unintentional loss or damage. A server-resident file is much less likely to be deleted inadvertently. Also, there's less chance of unauthorized access to or tampering with proprietary and confidential information. If a network computing workstation is stolen, the hardware asset is lost, but the sensitive data is not.

IBM WorkSpace On-Demand 3.0 features a “restricted” user desktop that allows administrators to select and control which applications are available to users and to add new applications when they are needed. Keeping data off individual workstations simplifies the management of network security, making the implementation of security policies and monitoring for compliance easier. Also, by limiting user access to the hard drive and removable media, IBM WorkSpace On-Demand 3.0 helps reduce the chance of corruption and damage to both the hardware and software.

For Tivoli users, administrators can monitor IBM WorkSpace On-Demand 3.0 from the Tivoli Enterprise console. These features can help your network managers spend their time more efficiently, transforming their job descriptions from fire fighters to masters of proactive network planning.

### **Easy Maintenance and Deployment of Applications**

For a network with a few thousand clients, it may take from six months to a year to upgrade software throughout the network. IBM WorkSpace On-Demand 3.0 lets network managers make software upgrades for many users at one time from the server or from a remote administration client.

Users no longer have to wait to benefit from new and improved applications. Once your network staff updates the servers, users can access new applications immediately. Users can also access their business applications from any PC or network computer connected to their server. With a solution written in and embracing Java, your company can begin the transition to Internet-based collaboration and transaction processing that network computing makes possible.

### **Low Total Cost of Ownership**

While Internet technologies are taking the world by storm, businesses need to leverage their large investments in current hardware and software technologies. IBM WorkSpace On-Demand 3.0 is an IBM solution designed to help lower the cost of ownership, while protecting investments in current technologies. IBM WorkSpace On-Demand 3.0 offers you flexibility in your choice of client platforms. You can choose to manage either Microsoft Windows clients or OS/2 clients or combine the two.

IBM WorkSpace On-Demand 3.0 provides two Registered User options. The first option provides client access and enabling that allows Windows client platforms to be managed by IBM WorkSpace On-Demand 3.0. The second option provides client access and enabling that allows the OS/2 client platform to be managed by IBM WorkSpace On-Demand 3.0. If needed, the second option also includes entitlement for a special version of the OS/2 client operating system.

Customers with licenses for IBM WorkSpace On-Demand 3.0 can use IBM WorkSpace On-Demand 2.0 or 3.0 either alone or in combination as long as:

- The total number of server installs does not exceed the number of authorized server installs acquired.
- The total number of clients to be managed does not exceed the total number of registered user authorizations acquired.

### **Support for Java Applications**

In mid-1999, IBM commissioned several case studies of thin client installations in retail stores and banking. Based on these studies, the ongoing operational maintenance costs of thin client solutions can provide substantial savings for customers, compared with typical “loosely-managed” PC networks. The savings for thin client solutions are likely to be greatest for task-oriented applications. For task-oriented systems, the annual hard savings (including categories such as depreciated capital, technical support,

help desk, and administration) may be approximately \$2,000 for a thin client computing solution versus a typical PC installation. In addition, it may be possible to obtain a further \$1,900 per year in soft savings (including categories such as user self help, peer support, and downtime).

Application models that require homogeneity, or favor a single client type, or require simultaneous deployment of both the client and server are doomed to niche roles in an Internet economy. With IBM WorkSpace On-Demand 3.0, you can continue to run a wide variety of your legacy applications. IBM WorkSpace On-Demand 3.0 includes a Java Virtual Machine for each supported client operating system and is compatible with popular browsers. You can run new Web-based applications alongside your legacy applications. In time, you can easily migrate your legacy applications to Java and browser-based solutions. WorkSpace On-Demand works with most of your current hardware, so there's no need to upgrade unless your business needs change.

### **Broad Access to Servers**

With IBM WorkSpace On-Demand 3.0, user data and applications reside safely on the server. If the client hardware fails, simply plug in a new one. If the client software fails, simply download a new copy from the server. Restoring the state of the system is easy, and there's a minimum of user downtime.

Another benefit is that users can log on and use their desktops, which you can tailor uniquely for your organization, from any client in the network. A user's "desktop" becomes any place that has a connection to the network.

Growing a Network Computing foundation into an Internet and intranet solution, requires scalable, integrated software products that can be easily expanded or added as business needs grow and change. IBM WorkSpace On-Demand 3.0 is a flexible component of a network computing installation that can be extended with IBM's e-business servers — Lotus® Go, Lotus Domino™, Lotus Domino Mail, DB2 Universal Database®, and the IBM Transaction Series.

### **Year 2000**

This product is Year 2000 ready. When used in accordance with its associated documentation, it is capable of correctly processing, providing, or receiving date data within and between the twentieth and twenty-first centuries, provided that all products (for example, hardware, software, and firmware) used with the product properly exchange accurate date data with it.

The service end date for this Year 2000 ready product is October 27, 2003.

### **Product Positioning**

IBM WorkSpace On-Demand 3.0 is an extension of the IBM WorkSpace On-Demand 2.0 technology and is positioned as a scalable family of cross-platform middleware. WorkSpace On-Demand unites the best of two worlds — the compelling value proposition of thin-client computing and the local processing power of Intel-compatible clients.

### **Statement of Direction**

Support for the Linux client platform will not be available in the third quarter of 2000.

IBM will entertain requests for quotations for Linux client support on IBM WorkSpace On-Demand 3.0 pursuant to a services engagement.

IBM will consider making Linux client support available in the future as a product.

### **Trademarks**

OS/2 and DB2 Universal Database are registered trademarks of International Business Machines Corporation in the United States or other countries or both.

Intel is a registered trademark of Intel Corporation.

Microsoft, Windows, and Windows NT are trademarks of

Microsoft Corporation.

Java is a trademark of Sun Microsystems, Inc.

Tivoli is a registered trademark of Tivoli Systems, Inc. in the United States or other countries or both. In Denmark, Tivoli is a trademark licensed from Kjobenhavns Sommer -- Tivoli A/S.

Domino is a trademark of Lotus Development Corporation.

Lotus is a registered trademark of Lotus Development Corporation.

Other company, product, and service names may be trademarks or service marks of others.

## **Education Support**

Technical workshops for IBM WorkSpace On-Demand 3.0 are available through the IBM International Technical Support Organization (ITSO). See the following Web pages for current schedules and enrollment information:

IBM La Hulpe:

<http://w3.lahulpe.ibm.com/itso>

IBM Global Campus:

<http://w3.education.ibm.com>

Curriculum supporting the NT 4.0 Server will be updated to include supplemental information for Windows™ 2000. Call IBM Education and Training at 800-IBM-TEACH (426-8322) for catalogs schedules, and enrollments.

## **Offering Information**

Product information is available through Offering Information (OITool) at:

<http://www.ibm.com/wwoi>

## **Publications**

No publications are shipped with this program. Displayable Softcopy Publications: IBM WorkSpace On-Demand 3.0 includes displayable manuals online as .PDF and as .PS files. The softcopy publication files are shipped on the same media as the basic machine-readable material.

---

## **A.5 Technical Information**

### **Hardware Requirements:**

#### **Server**

- Processor: Minimum 200 MHz
- Memory: Minimum 64 MB (recommended 128MB)

#### **Client**

- Memory: Minimum 32 MB
- Hardfile: Minimum 600 MB (optional for remote boot clients)

- Network Interface Card (NIC) Adapters: PXE 1.1 or 2.0

**Software Requirements:**

Customers are responsible for obtaining the appropriate licenses for all server operating systems, for connection or attachment to Microsoft™ Windows NT™ 4.0 or Windows 2000 servers, and for all Microsoft Windows client operating systems that are to be managed.

**Planning Information:**

**Packaging:** The IBM WorkSpace On-Demand 3.0 media pack includes a CD-ROM containing the server management and client enabling software. The package also includes a separate CD-ROM that contains the IBM WorkSpace On-Demand 3.0 OS/2® Client operating system, which can be installed and used only with WorkSpace On-Demand 3.0.

The IBM WorkSpace On-Demand 3.0 program package includes a CD-ROM containing the server management and client-enabling software. The package also includes a separate CD-ROM that contains the IBM WorkSpace On-Demand 3.0 OS/2 Client operating system, which can be installed and used only with IBM WorkSpace On-Demand. The program package also includes the International Program License Agreement (IPLA) and IPLA pointer sheet, the Service and Support Statement, and the Proof of Entitlement.

**Security, Auditability, and Control**

User management is responsible for evaluation, selection, and implementation of security features, administrative procedures, and appropriate controls in application systems and communication facilities. The customer is responsible for evaluation, selection, and implementation of security features, administrative procedures, and appropriate controls in application systems and communication facilities.

**Ordering Information**

IBM WorkSpace On-Demand 3.0 has two chargeable units:

- **Server Install** — Each Server Install for WorkSpace On-Demand includes one Registered User authorization for administering the IBM WorkSpace On-Demand 3.0 server. Customers must acquire an additional Registered User authorization for each additional client that accesses the WorkSpace On-Demand server.



- Registered User — IBM WorkSpace On-Demand 3.0 provides two Registered User options. The first option provides client access and enabling that allows Windows client platforms to be managed by IBM WorkSpace On-Demand 3.0. The second option provides client access and enabling that allows the OS/2 client platform to be managed by IBM WorkSpace On-Demand 3.0. The second option also includes entitlement for a special version of the OS/2 client operating system. WorkSpace On-Demand offers customers flexibility in their choice of client platforms; they can choose to manage either Microsoft Windows clients or OS/2 clients or combine the two to best meet their needs.

In all cases, customers are responsible for obtaining the appropriate licenses for all server operating systems, for connection or attachment to Microsoft Windows NT 4.0 or Windows 2000 servers, and for all Microsoft Windows client operating systems that are to be managed.

Client Accesses are based on the Registered User concept.

Server/Manager installs are based on the Installs concept.

Customers Licensed for IBM WorkSpace On-Demand 3.0

Customers licensed for IBM WorkSpace On-Demand 2.0 with active Passport Advantage subscriptions will receive the WorkSpace On-Demand 3.0 media package.

The media pack for IBM WorkSpace On-Demand 2.0 is intended to be used with OS/2 Warp Server for e-business. The WorkSpace On-Demand registered user authorization also includes, when required, an IBM Warp Server for e-business registered user authorization.

The media pack for IBM WorkSpace On-Demand 3.0 is intended to be used with Windows NT 4.0 and/or Windows 2000 servers, but also includes a special version of the OS/2 Client operating system, called the IBM WorkSpace On-Demand 3.0 OS/2 Client, for use when managing OS/2 clients from Windows NT 4.0 and/or Windows 2000 servers. Customers with licenses for IBM WorkSpace On-Demand 3.0 can use IBM WorkSpace On-Demand 2.0 or 3.0 either alone or in combination in their enterprises as long as the total number of Server Installs does not exceed the number of authorized Server Installs that have been acquired and the total number of clients that are to be managed does not exceed the total number of registered user authorizations that have been acquired.

Customers without an active Passport Advantage subscription for IBM WorkSpace On-Demand 3.0 may purchase an IBM WorkSpace On-Demand 3.0 subscription after license.

#### **Customers Licensed for OS/2 Warp 4**

There is a special migration path for customers who are licensed for OS/2 Warp 4. Customers with an active Passport Advantage subscription for OS/2 Warp 4 are entitled to trade in their OS/2 Warp 4 licenses for an equal number of WorkSpace On-Demand registered user authorizations.

Customers without an active Passport Advantage subscription for OS/2 Warp 4 may purchase an IBM WorkSpace On-Demand 3.0 registered user subscription after license.

This special migration pricing for OS/2 Warp 4 clients is available only under Passport Advantage.

#### **Customers Licensed for Earlier Versions of OS/2**

There is a special migration path for customers who are using with OS/2 1.x, OS/2 2.x, OS/2 Warp 3.0, or OS/2 Warp Connect 3.0. Customers may trade in their OS/2 1.x, 2.x, or 3.0 licenses for an equal number of IBM WorkSpace On-Demand 3.0 registered user authorizations for a reduced fee.

This special migration pricing for IBM OS/2 clients is available only under Passport Advantage. IBM WorkSpace On-Demand 3.0 is available for order as a media pack through IBM Passport Advantage. Program packages are also available.

For Passport Advantage ordering information and charges, contact your Lotus® representative or authorized Lotus Business Partner. Additional information is also available on the Passport Advantage Web site:

<http://www.lotus.com/passportadvantage>

The use authorizations described in this announcement are included in Passport Advantage.

| <b>Program Name</b>         | <b>Number</b> |
|-----------------------------|---------------|
| IBM WorkSpace On-Demand 3.0 |               |
| International English       | BC75GIE       |

|                 |         |
|-----------------|---------|
| Spanish         | BC75GES |
| Canadian French | BC75GCF |
| Program Package |         |

| Program Name                | Number  |
|-----------------------------|---------|
| IBM WorkSpace On-Demand 3.0 |         |
| International English       | 0781913 |
| Spanish                     | 0781917 |
| Canadian French             | 00P7710 |

Orders for the Canadian French National Language Version (NLV) will be fulfilled with the International English version of the media pack starting on July 31, 2000. You will receive a letter containing instructions on how to obtain the requested NLV within 30 days of IBM's receipt of your order.

#### **Additional Licenses**

| Program Name                | Number  |
|-----------------------------|---------|
| IBM WorkSpace On-Demand 3.0 |         |
| Server Install              | 00P7711 |
| Registered User             | 00P7712 |

#### **Customer Financing**

IBM Global Financing offers attractive financing to credit-qualified commercial and government Customers and Business Partners to assist them in acquiring IT solutions. Offerings, rates, terms, and availability can vary by country. Contact your local Global Financing organization. Country contacts are on the Web at:

<http://www.financing.ibm.com>

#### **Terms and Conditions**

Licensing: IBM International Program License Agreement. Proofs of Entitlement (PoE) are required for all authorized use.

Limited Warranty Applies: Yes

Program Services: Available until October 27, 2003.

Money-Back Guarantee: 30-day, money-back guarantee

Copy and Use on Home/Portable Computer: No

Volume Orders (IVO): No

Passport Advantage Applies: Yes

Passport Advantage Subscription Applies: Yes

Usage Restriction: Yes

Support Line: Yes

AIX®/UNIX® Upgrade Protection Applies: Not Entitled Upgrade for Current  
AIX/UNIX Upgrade Protection

Licensees: No

AS/400® Software Subscription Applies: No

Variable Charges Apply: No

Educational Allowance Available: Not applicable

### **Charges**

The charges provided in this announcement are suggested retail prices for the U.S. only and are provided for your information only. Dealer prices may vary, and prices may also vary by country. Prices are subject to change without notice. For additional information and current prices, contact your local IBM representative.

### **Upgrades**

Contact your sales channel for Support Line pricing information.

### **Passport Advantage**

For Passport Advantage and charges, contact your Lotus representative or authorized Lotus Business Partner. Additional information is also available on the Passport Advantage Web site:

<http://www.lotus.com/passportadvantage>

## Customer Financing

IBM Global Financing offers attractive financing to credit-qualified commercial and government customers and Business Partners in more than 40 countries around the world. IBM Global Financing is provided by the IBM Credit Corporation in the United States. Offerings, rates, terms and availability may vary by country. Contact your local IBM Global Financing organization. Country organizations are listed on the Web at:

<http://www.financing.ibm.com>

## Order Now

|                              |                                                                                    |
|------------------------------|------------------------------------------------------------------------------------|
| Use Priority/Reference Code: | SE001                                                                              |
| Phone:                       | 800-IBM-CALL                                                                       |
| Fax:                         | 800-2IBM-FAX                                                                       |
| Internet:                    | ibm_direct@us.ibm.com                                                              |
| Mail:                        | IBM Atlanta Sales Center<br>Dept. SE001<br>P.O. Box 2690<br>Atlanta, GA 30301-2690 |

You can also contact your local IBM Business Partner or IBM representative. To identify them, call 800-IBM-4YOU. Note: Shipments will begin after the planned availability date.

## Trademarks

OS/2, AIX, and AS/400 are registered trademarks of International Business Machines Corporation in the United States or other countries or both.

Windows, Microsoft, and Windows NT are trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries exclusively through X/Open Company Limited.

Lotus is a registered trademark of Lotus Development Corporation.

Other company, product, and service names may be trademarks or service marks of others.

---

## Appendix B. WorkSpace On-Demand 3.0 Performance White Paper

Authors: Khoa Huynh, Boyd Jasperson, Telford Knox

July 2000

---

### B.1 Abstract

IBM WorkSpace On-Demand 3.0 is a cross-platform IT solution that provides customers with the capability to manage a broad range of client hardware, platforms, and applications from one or multiple servers. It offers two approaches to thin-client computing for managing client machines: the remote-boot (“thin”) approach for OS/2 Warp clients and the remote boot/install (“lean”) approach for Windows NT 4.0 Workstation and Windows 98 Second Edition (SE) clients. For OS/2 Warp clients, the OS/2 client operating system is remote-booted from the server; the client machines do not need local disks (if they do, the local disks can be used for caching). For Windows NT 4.0 and Windows 98 SE clients, the client operating systems are first installed on the clients' local disks. This installation is managed remotely from the server using customized response files. On subsequent boots, the client operating systems are booted from the clients' local disks, rather than from the server across the network.

In this paper, we present some performance data collected by IBM for the client boot and log-on processes. We found that, for OS/2 Warp clients, many more clients can be booted simultaneously by using the local caching option, provided that the initial cache-initializing boots are “staggered” properly. In most cases, the network bandwidth is the key performance factor, but on a 100-Mbps Ethernet, it is likely that a server processor equivalent to an Intel 400-MHz Pentium II Xeon will become a performance bottleneck during “boot storms.” For Windows clients, the initial installation boots demand very high network bandwidth as files need to be copied from the server to the clients' local disks. However, once the installation boots are done, subsequent boots are very fast and require very little network bandwidth, allowing many more users to boot their client machines simultaneously. With no new applications assigned, the user log-on process was found to be relatively quick and deterministic. We also found that the server memory usage stayed well below the minimum requirements specified for IBM WorkSpace On-Demand 3.0.

Detailed data, analysis, and important performance recommendations are provided in Sections 3 and 4 of this paper. The data contained in this paper can also be used to support capacity planning activities.

---

## B.2 Introduction

Today's network computing enterprise requires an open, scalable, cross-platform approach that is also integrated. The Application Framework for e-business is a multi-tiered, distributed information technology environment which is based on open industry standards and integrates Internet technologies with traditional information technology. The Managing Technology segment of the Application Framework for e-business includes integrated solutions for managing a secure, reliable, scalable e-business infrastructure. IBM WorkSpace On-Demand 3.0 is a key component of this Managing Technology segment. It is a cross-platform IT solution that provides customers the capability to manage a broad range of client hardware, platforms, and applications from one or multiple servers. With the core logic and data structures implemented in the Java™ programming language, IBM WorkSpace On-Demand 3.0 can run across a number of server platforms.

The initial release of IBM WorkSpace On-Demand, which starts shipping at the end of July 2000, will support servers running Windows NT 4.0 Server. As far as client support is concerned, IBM WorkSpace On-Demand 3.0 supports both personal computers (PCs) and network computers (NCs) as clients. The client platforms supported by IBM WorkSpace On-Demand 3.0 include Windows NT 4.0 Workstation, Windows 98 Second Edition (SE), and OS/2 Warp. Virtually all application types on the supported platforms are supported by IBM WorkSpace On-Demand 3.0.

From the start, IBM WorkSpace On-Demand 3.0 has been designed to help customers address the following critical information technology management needs arising from the industry's rapid adoption of network computing and e-business.

### **Centralized management and control of clients**

IBM WorkSpace On-Demand 3.0 allows administrators to select and control which applications are available to a user and add new applications when needed. Applications that are not available to a user will not be visible on that user's desktop at the client system. The restricted user desktop does not allow the user access to the local hard drive and removable media. This prevents the deletion of key operating system files as well as the addition of unauthorized applications and, in turn, helps minimize the threat of client corruption and introduction of viruses.



### **Easy maintenance and deployment of applications**

With IBM WorkSpace On-Demand 3.0, administrators can deploy and update operating system and application software on the server and then download them to client systems remotely. This eliminates the need to physically install software and software updates on the clients, thus making applications available to users much more quickly.

### **Broad access to servers**

With IBM WorkSpace On-Demand 3.0, user data and applications reside safely on the server. This server-based architecture allows users to log on and use their personalized, restricted desktop from any client system in the network domain (user roaming). Furthermore, it allows the state of a client system to be quickly restored from the server with minimal downtime if something goes wrong at that client.

### **Helps protect and extend existing hardware and software investments**

As mentioned previously, IBM WorkSpace On-Demand 3.0 provides cross-platform client and server support, so customers can use and get all of the benefits of IBM WorkSpace On-Demand 3.0 with their existing hardware, operating systems, and applications. In addition, IBM WorkSpace On-Demand 3.0 includes a Java Virtual Machine and run-time environment for each supported client platform and is compatible with most popular browsers. This enables customers to run newer Web-based applications along with legacy applications, giving them time to gradually migrate their legacy applications to Java and browser-based solutions. In addition, the inclusion of Tivoli Management Agent (TMA) in the IBM WorkSpace On-Demand 3.0 package enables customers to take advantage of the additional management capabilities provided by Tivoli Enterprise.

### **Low total cost of ownership**

As a server-based client management solution, IBM WorkSpace On-Demand 3.0 helps reduce costs associated with technical support, help desk, and administration, by providing the benefits just discussed, including easy maintenance and deployment of operating system and application software, and leveraging current investments in hardware and software technologies.

## **B.2.1 Functional Overview**

The traditional network computing model is called a “fat-client” model because the clients have their own local disks where the operating system, all applications and data reside. Clients are booted from their own disks and

applications are executed on the clients using the clients' own processing power, with the servers providing remote file or print services. The so-called "thin-client" computing model is an alternative and increasingly popular computing paradigm. There are several different approaches to thin-client computing. IBM WorkSpace On-Demand 3.0 supports the following two approaches:

### **Thin (Remote Boot)**

In this approach, the operating system, applications, and data are stored and managed from one or more central servers. User access to applications and data is provided over the network through "thin" clients, such as network computers. These client systems typically do not have disks. At every boot, the operating system is remote-booted from the server. Applications are downloaded on demand (i.e., upon being invoked) from the server into the memory of the client and executed on the client. If the client has a local disk or a non-volatile memory card, it would be used for caching read-only files to reduce network traffic during boots and application execution. IBM WorkSpace On-Demand 3.0 supports this pure remote boot approach for OS/2 Warp clients.

### **Lean (Remote Boot/Install)**

In this approach, the client must have its own local disk. The operating system is installed on the local disk of the client on the very first boot. This installation is managed from the server using remote boot technologies and customized response files. Once installed, on subsequent boots, the operating system is booted from the client's local disk after a brief handshake with the server. The applications and data are still stored and managed from a central server, but downloaded on demand to the client for execution. IBM WorkSpace On-Demand 3.0 supports this remote boot/install approach for Windows 98 SE and Windows NT 4.0 Workstation clients.

In both approaches, application execution is done at the client, taking advantage of the local processing power of client machines. As a result, IBM WorkSpace On-Demand 3.0 combines the best of both worlds: the compelling value proposition of thin-client computing and leveraging the local processing power of clients in a manner similar to the fat-client computing model.

In order to provide a robust client management solution, IBM WorkSpace On-Demand 3.0 contains several main components:

## **Machine management**

This component supports both the remote boot (for OS/2 Warp clients) and the remote boot/install (for Windows clients) capabilities as described previously. In addition, it also offers pre-boot services, such as disk partitioning, which can be performed on target client machines before the primary operating system for the clients are run. These services can be customized by OEMs and business partners through a flexible architecture. This machine management component also allows administrators to define machine templates as a basis for client machine definitions as well as define/delete/list client machine definitions.

## **User management**

This component allows administrators to manage the users. After the operating system setup is complete, the client machines boot to a log-on window. The user is then required to enter his or her user ID and password. Once entered, the user ID and password are validated against the native server. In the case of a Windows NT server, the user ID is a regular Windows NT domain user ID, which can be created using existing product function like the Windows NT User Manager for Domains, Tivoli User Admin, or the IBM WorkSpace On-Demand 3.0 console. IBM WorkSpace On-Demand 3.0 extends these user definitions to provide roaming access. To support roaming access, the administrator needs to set up a home directory for each user on the server where user-specific data are stored. This allows the user to access his or her data from any client in the network domain.

## **Desktop management**

This component allows the administrator to define/delete/list desktop definitions. A desktop definition provides the contents and “look and feel” of the user's desktop. When the user logs on, his or her desktop definition is downloaded from the user home directory on the server to the client machine where the user logs on. IBM WorkSpace On-Demand 3.0 supports multiple pluggable shells and desktops, which can be enabled on a per-user basis. The desktop can be a fully server-managed, simplified desktop or a full-function desktop based on configuration choices made by the administrator. A special PROTDISK filter driver (provided with the product) can be customized by the administrator to provide a restricted user desktop which prevents the user from writing files to the client's local disk or modifying existing system settings.

## **Application management**

This component allows the administrator to manage applications and control user access to the applications. Application executables are stored on a shared server file system that is visible to the clients. Although they are resident on the server, some applications may require specific system files in the local Windows system directories. IBM WorkSpace On-Demand 3.0 provides a unique “update-at-logon” technology to allow necessary system files to be downloaded from the server to a client machine in order to enable that client machine to run all of the applications assigned to the user. The application management component allows the administrator to assign, list, and un-assign applications to users. The administrator can also group applications into “application sets”, which can then be assigned to users with a single administrative action.

## **Administration console**

Provides graphical and command-line interfaces which the administrator uses to configure and manage IBM WorkSpace On-Demand 3.0 resources (client machines, users, desktops, and applications). The administration console can be installed on a server or on an independent Windows NT client that has a TCP/IP connection to the server. The administration console is implemented in the Java programming language for cross-platform portability.

### **B.2.2 Architectural Overview**

The major components of IBM WorkSpace On-Demand 3.0 include the configuration server, deployment server, a cache management service, and corequisite IP services, such as the Dynamic Host Configuration Protocol (DHCP) server, Pre-boot eXecution Environment (or PXE, as defined in the Intel Wired for Management specification) Proxy Server, Boot Image Negotiation Layer (BINL) server, and the Trivial File Transfer Protocol (TFTP) server. Please refer to Figure 188 on page 442 for an overview of these components.

The configuration server is responsible for storing information about client machines, users, and applications that are managed by IBM WorkSpace On-Demand 3.0. It is implemented in the Java programming language and supports configuration tasks whose role is to store the configuration information at the right level in the datastore schema and to call the required transform tasks on the deployment servers. When writing into the datastore, the configuration tasks use the data services interface, which encapsulates the actual implementation of the datastore, thereby facilitating a different implementation of the datastore in the future.

The deployment server is responsible for manipulating the actual operating system and application files and for running the transform tasks that create client-specific files based on administrative action. There are machine, user, and application transform tasks, corresponding to the machine, user, and application management components of IBM WorkSpace On-Demand 3.0. When the administrator creates a client machine definition, a machine transform task is executed to analyze the configuration information stored in the datastore and set up the appropriate values in a customized installation response file. This response file is then used to drive the boot process on that particular client machine. The deployment server also manages the entire client boot process.

The cache management service (caching daemon), which runs on the deployment server, is used for caching read-only files on the client's local disk or flash memory card to reduce network traffic during remote boots and application execution on OS/2 Warp clients. It makes sure that the files in the client cache is synchronized with the master version on the (deployment) server.

With IBM WorkSpace On-Demand 3.0, all clients are booted using the DHCP PXE standard. Please refer to Section B.3, "Boot Process Overview" on page 444 for more details on the boot process.

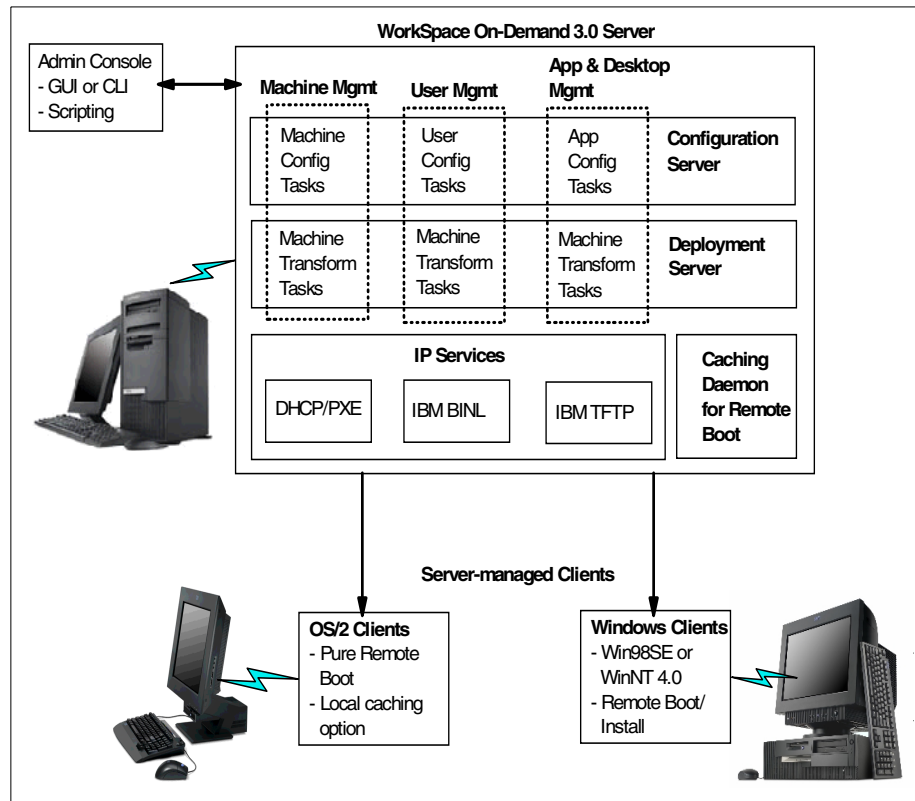


Figure 188. Architectural overview of IBM WorkSpace On-Demand 3.0

IBM WorkSpace On-Demand 3.0 has been designed to be more scalable than previous versions of IBM WorkSpace On-Demand: it supports a split-server model that manages a large number of client machines. In the initial release, however, only the machine management and boot portion of the deployment server can be spread across multiple physical servers. The rest of the deployment server, as well as the configuration server, must remain on the main server (the domain controller). In addition to the machine management and boot functions, the cache management service (caching daemon) can also be spread across multiple servers, as can the IP services (DHCP, BINL, and TFTP). These multiple servers are then referred to as the machine deployment servers or boot servers. The split-server model is not considered in this paper. Please refer to Section B.6, "Additional information sources" on page 472 for more details.

### **B.2.3 Server requirements**

#### ***Hardware***

As mentioned above, with IBM WorkSpace On-Demand 3.0, the configuration server and the deployment server(s) can be on different physical server machines. It is recommended that the physical server running the configuration server has the following minimum hardware:

- 256 MB of physical memory
- 200-MHz Pentium processor
- 150 MB of hard disk space
- SVGA graphics (256 colors or more)

The recommended hardware requirement for a deployment server is as follows:

- 256 MB of physical memory (additional)
- 200-MHz Pentium processor
- 200 MB of hard disk space available for each operating system image
- SVGA graphics (256 colors or more)

#### ***Software***

The following is a list of the minimum software requirements

- Windows NT 4.0 Server with Service Pack 4 or above
- New Technology File System (NTFS) on the partition used for IBM WorkSpace On-Demand 3.0
- TCP/IP configured on every server with access to a DNS or host file that can resolve IP addresses to fully qualified host names

### **B.2.4 Client requirements**

#### ***Hardware***

Each client machine supported by IBM WorkSpace On-Demand 3.0 must have the following minimum hardware requirements:

- 32 MB of physical memory
- 600 MB of hard disk space (for Windows NT 4.0 and 98 SE clients)
- PXE-compliant, DHCP-based boot ROM or a network card supported by a third party boot diskette (with PXE emulation)

For OS/2 Warp clients, there is no local disk required. However, if caching read-only files in order to reduce the network traffic is desired, the client must have one of the following:

- A partitioned hard disk (partition C must be empty, formatted with the FAT16 file system, and have at least 48 MB of free space), or
- A compact flash card

### **Software**

The initial release of IBM WorkSpace On-Demand 3.0, which begins shipping at the end of July 2000, supports the following client operating systems:

- OS/2 Warp
- Windows 98 Second Edition
- Windows NT 4.0 Workstation with Service Pack 4 or above

### **B.2.5 Scope of this paper**

In a server-based client management environment for which IBM WorkSpace On-Demand 3.0 is targeted, the client boot and log-on processes are very important to system administrators, because it is through these processes that the clients' operating systems, middleware, and applications are distributed to the clients. They can consume a lot of network bandwidth. As a result, the primary purpose of this paper is to present some performance and server memory usage data collected by IBM for the client boot/install/log-on processes, some data analysis, and a number of performance recommendations derived from this data. This paper will not cover in detail the performance of applications executing on the clients because it is dependent on the client hardware and the nature of the applications themselves; furthermore, it is of secondary concern to this paper's target audience (the system administrators). However, factors affecting the application performance will be discussed.

---

## **B.3 Boot Process Overview**

Since the primary focus of this white paper is on the boot performance, a high-level overview of the boot process is given here. For more details, please refer to Section B.6, "Additional information sources" on page 472.

The general flow of the boot process is as follows:



- Upon power on, the client uses the DHCP PXE protocol to obtain the IP address of the boot server and the name of the initial bootstrap. The same initial bootstrap is used for all clients.
- After getting downloaded to the memory of the client, this initial bootstrap obtains the client configuration information from a client-specific configuration file.
- Based on this information, it then determines which second bootstrap to download. The second bootstrap then does what is required to:
  - Initiate the remote boot process (for OS/2 Warp clients).
  - Start the remote boot/install process (for Windows clients).
  - Boot the local hard disk in the case where a local boot of an installed operating system is required.

The following descriptions are simply overviews of the initial boot and of each operating system boot.

### **B.3.1 Initial Boot (for all clients)**

1. The client machine initializes and passes control to the client's PXE network adapter or boots from the PXE Boot Diskette, which then emulates the PXE network adapter function for the following steps.
2. The PXE network adapter broadcasts DHCP request.
3. The DHCP/PXE server (or DHCP server and DHCP/PXE Proxy) responds.
4. The client's PXE network adapter receives DHCP/PXE packet(s).
5. The client's PXE network adapter requests the initial bootstrap's name and location from the BINL server. This initial bootstrap is the same for all client operating system types.
6. The client's PXE network adapter receives the initial bootstrap's name and location from the BINL server.
7. The client's PXE network adapter downloads the initial bootstrap from the deployment (boot) server using Trivial File Transfer Protocol (TFTP).
8. The client's PXE network adapter passes control to the initial bootstrap.
9. The initial bootstrap downloads the TDMCLNT.INF file from the boot server using TFTP.
10. Based on information in the TDMCLNT.INF file, the initial bootstrap determines the second bootstrap's name and location. This second bootstrap is different for each operating system type.

11. The initial bootstrap relocates itself in the client's memory.
12. The initial bootstrap downloads the second bootstrap from the boot server using TFTP.
13. The initial bootstrap passes control to the second bootstrap.

Since each operating system type has its own second bootstrap, let's assume for now that the client is a Windows client (running either Windows 98 Second Edition or Windows NT 4.0 Workstation).

### **B.3.2 Remote Boot/Install (Windows NT 4.0 and 98 SE clients only)**

After the initial boot, new Windows NT 4.0 and 98 SE clients always start with an installation boot where the client operating system is downloaded and installed on the client's local disk.

#### **Installation Boot**

1. The second bootstrap gains control of the client.
2. The second bootstrap downloads the TDMDOS.INF file from the boot server using TFTP.
3. Based on information in the TDMDOS.INF file, the second bootstrap determines the boot image's name and location. This boot image is a DOS boot image.
4. The second bootstrap downloads the DOS boot image from the boot server using TFTP.
5. The second bootstrap applies required fix-ups to the DOS boot image.
6. The second bootstrap intercepts the BIOS disk I/O to redirect floppy access to the DOS boot image.
7. The second bootstrap "boots" the DOS image.
8. A network driver and network support code are loaded.
9. A DOS executable, called CONNECT.EXE, establishes two connections to the boot server: one to the Read-Only share, and the other to the Read-Write share.
10. An installation shell is started.
11. The installation shell repeatedly calls another DOS executable, called STATE.EXE, and processes commands sent to it by STATE.EXE.
12. STATE.EXE uses a file called STATE.INI to sequence through all necessary phases in the boot process. STATE.INI is really a script file, containing all of the commands to be executed by STATE.EXE to set up

the Windows operating system, the Win32 Java Virtual Machine (JVM), the Tivoli Management Agent (TMA), and the WorkSpace Log-on Client (which facilitates the user log-on process to the domain controller). STATE.EXE also updates the COUNTER.INI file to record what it has completed, so it knows what to do the next time it gains control. Here are the phases which STATE.EXE goes through in the boot sequence (in the order of execution):

- a. Deletes the partition table, creates new partition on the client's local disk, and reboots.
- b. Formats the installation partition on the client's local disk.
- c. Executes the `copy` commands, specified in STATE.INI, to copy the Windows operating system installation files, device drivers, response files, and all other files required to set up the WorkSpace Log-on Client, the Win32 JVM, and the TMA.
- d. Executes commands to release the redirected floppy drive A, loads the disk caching program to speed up the copying of Windows files, writes a trigger file to the boot server, and calls the Windows setup program to start the installation of the Windows operating system on the client's local disk in unattended mode. The trigger file is used to notify a daemon running on the boot server to switch the bootstrap program for the client to the hybrid boot image, so that the next time the client boots, it will go through the hybrid boot process instead (to be discussed later). After the server daemon has switched the bootstrap image for the client, it deletes the trigger file.

There may be cases where the network drivers consume a significant amount of conventional memory such that the memory requirement of the Windows setup program cannot be satisfied. In such cases, STATE.EXE will install DOS (without the network support) on the client's local disk, write a trigger file to the boot server (so a daemon running on the server can switch the bootstrap program to the hybrid boot image), and copy the following from the boot server to the client's local disk: the Windows setup files, the STATE.INI (script) file, and other support files and executables. The client is then rebooted. After the client is rebooted, STATE.EXE resumes the execution of commands in STATE.INI locally on the client and starts the Windows setup program without network connections.

After the Windows operating system has been set up successfully, executables required to set up the WorkSpace Log-on Client, the Win32 JVM, and the TMA are then executed (also in unattended mode).

## Hybrid Boot

Once the daemon running on the boot server detects that a trigger file has been placed on the server by a client, it will modify the TDMCLNT.INF file on the server to point to a hybrid boot image, instead of the second bootstrap. The hybrid boot image is a small bootstrap executable which, once downloaded to the client, will allow the client to boot directly from the client's local disk.

The hybrid boot process begins with the initial boot sequence as described at the very beginning of this section. During this initial boot sequence, the initial bootstrap will download the TDMCLNT.INF file from the boot server using TFTP (please see Step 9 in Section B.3.1, "Initial Boot (for all clients)" on page 445). Based on the information in the TDMCLNT.INF file, which now points to the hybrid boot image, the initial bootstrap will next download the hybrid boot image from the boot server using TFTP, and pass control to the hybrid boot image. The hybrid boot image will then switch to the Windows bootstrap program on the client's local disk, and the Windows operating system is booted from there.

Hybrid boots are very fast and do not require much network bandwidth because the Windows operating system is booted from the client's local disk. All subsequent boots after the installation boot use this very fast hybrid boot process.

## Remote Boot (OS/2 Warp clients only)

After the initial boot, the boot process for OS/2 Warp clients continues as follows:

1. The second bootstrap gains control of the client.
2. The second bootstrap downloads the TDMOS2.INF file from the boot server using TFTP.
3. Using TFTP, the second bootstrap downloads all files specified in TDMOS2.INF from the boot server to a RAM DISK (including the BPCOMMON file, which contains a list of additional files that are needed).
4. The second bootstrap downloads all files specified in BPCOMMON from the boot server to the RAM DISK using TFTP.
5. The second bootstrap executes the OS/2 Warp loader (OS2LDR) with itself as the mini-Installation File System (min-IFS). It fulfills all disk requests using the RAM DISK.
6. The OS/2 operating system, network, and redirected file system are loaded.

7. The redirected file system replaces the RAM DISK. The OS/2 boot process continues with the redirected file system using the System Message Block (SMB) protocol. SMB connections are made to RPLFILES and WRKFILES shares on the boot server.
8. Boot continues until OS/2 Warp is up and the log-on window is displayed.

If the caching option is used, the client is required to have a local disk. Non-writable, common OS/2 files will be copied from the boot server to this disk, so that the next time the client boots, these files will be loaded from the client's local disk, instead of loaded from the boot server across the network. As a result, the first boot to initialize the cache takes more time than subsequent cached boots. During the first boot with caching, the following takes place:

- A device driver, called NCCACHED.SYS, is loaded and executed early in the boot sequence, just after the network drivers and support are loaded. This driver creates a trigger file and places it on the boot server.
- At a later time in the boot sequence, an executable, called NCCACHEE.EXE, runs and waits for an action file to be created by the caching daemon running on the boot server. This action file contains instructions on which files to cache. NCCACHEE.EXE then copies these files from the boot server to the client's local disk. The client is then rebooted.

During the first remote boot with caching, the action file will denote that every file listed should be copied from the boot server to the client's local disk. Subsequently, it will denote that there is either nothing to do or that files need to be refreshed. New or refreshed files will be copied to the cache (client's local disk) and the client will reboot. If there is nothing to do, the boot just continues on to completion.

---

## **B.4 Performance data & analysis**

In this section, a description of the performance measurement environment will be given, followed by the presentation of data that was collected. This data will then be analyzed to determine several performance recommendations to the customers. Note that the data given in this section was collected in the controlled environment described below; the actual system performance may vary and is dependent on many factors, including hardware and software configuration.

## **B.4.1 Measurement environment**

The test environment consists of two parts: the hardware environment and the software environment. We shall discuss both environments in turn below.

### **B.4.1.1 Hardware environment**

Our test environment consists of a single server and up to 32 clients, all connected on a single local area network. The following descriptions defines each of the machine configurations.

#### **Server**

- Netfinity 7000 (running as a 1-way SMP system with a uni-kernel operating system)
- 400-MHz Pentium II Xeon processor
- 3-GB physical memory
- 2 x 4-GB hard drives
- 1 x 8-GB hard drive
- 1 x IBM 16/4 PCI Token Ring
- 1 x IBM 10/100-Mbps PCI Etherjet

#### **Clients**

- IBM 6589 (32 machines)
- 200-MHz Pentium Pro processor
- 48-MB physical memory
- 3Com Etherlink PCI Ethernet NIC
- IBM 16/4 ISA Turbo Token Ring NIC (for IBM Workspace On-Demand 2.0)
- IBM 16/4 PCI Token Ring Adapter II (for IBM Workspace On-Demand 3.0)

#### **Local Area Network** (only one of the following is in use at any given time)

- 16-Mbps Token Ring (private - no other use)
- 10-Mbps Ethernet (private - no other use)
- 100-Mbps Ethernet (private - no other use)

#### **Hubs**

- 100-Mbps Ethernet
  - 2 x IBM 8225-008 (8 ports each)
  - 3 x IBM 8223-008 (8 ports each)

- 10-Mbps Ethernet
  - 1 x IBM 8722-016 (16 ports each)
  - 1 x IBM 8242-008 (8 ports each)
  - 1 x IBM 8224-016 (16 ports each)

#### **B.4.1.2 Software environment**

Two software environments are measured for performance comparisons: the previously released IBM WorkSpace On-Demand 2.0 and the new IBM WorkSpace On-Demand 3.0. For IBM WorkSpace On-Demand 3.0, all server services were run on a single server machine, including the configuration service, deployment services, and all IP services (DHCP/PXE, BINL, TFTP, etc.). The split-server model is not considered in this paper. IBM WorkSpace On-Demand 2.0 does not support the split-server model.

### **Server**

#### **IBM WorkSpace On-Demand 2.0 environment**

- OS/2 Warp Server for e-business
- IBM WorkSpace On-Demand 2.0 (with Feature for Windows Clients)
- HPFS386 File System with 150-MB cache (for Windows clients)
- Journal File System (JFS) with 65-MB cache (for OS/2 clients)
- IBM Etherjet and IBM Token Ring network drivers

#### **IBM WorkSpace On-Demand 3.0 environment**

- Windows NT 4.0 Server with Service Pack 4
- IBM WorkSpace On-Demand 3.0
- NTFS with default cache
- IBM Etherjet and IBM Token Ring network drivers

### **Client install images**

Client operating system:

- Windows NT clients:
  - Windows NT 4.0 Workstation
  - Base (for IBM WorkSpace On-Demand 2.0 and 3.0)
  - Service Pack 4 (for IBM WorkSpace On-Demand 3.0)
- Windows 98 clients:

- Windows 98 First Edition (for IBM WorkSpace On-Demand 2.0)
- Windows 98 Second Edition (for IBM WorkSpace On-Demand 3.0)
- OS/2 clients:
  - OS/2 Warp 4 with FixPak 6
  - VGA video driver
  - 3Com Ethernet or IBM Token Ring network driver
  - No application packages

#### **B.4.2 Boot performance for Windows NT 4.0 clients**

Figure 189 on page 453 shows the installation boot times for Windows NT 4.0 clients on both the new IBM WorkSpace On-Demand 3.0 and the previously released IBM WorkSpace On-Demand 2.0. With IBM WorkSpace On-Demand 3.0, the installation boot time is about 20 minutes with a single base Windows NT 4.0 client and just under an hour for 32 simultaneous clients. With Service Pack 4, the installation boot time takes quite a bit longer because of the additional time it takes to download Service Pack 4 files to the client and install it. The total amount of data going across the network, including the downloaded files, headers, and so on, during the installation boot is 143 MB for a base Windows NT 4.0 client, and 260 MB for a client with Service Pack 4. Please keep in mind that IBM WorkSpace On-Demand 3.0 runs on Windows NT 4.0 Server while IBM WorkSpace On-Demand 2.0 runs on OS/2 Warp Server for e-business. In addition, IBM WorkSpace On-Demand 3.0 only supports PXE boot technology with multiple bootstraps, as well as TCPBeui transport protocol for Windows clients, while IBM WorkSpace On-Demand 2.0 only supports RIPL and NetBeui for Windows clients. NetBeui is generally faster, but less flexible, than TCPBeui. These differences are largely responsible for the minor boot time differences between IBM WorkSpace On-Demand 3.0 and IBM WorkSpace On-Demand 2.0.



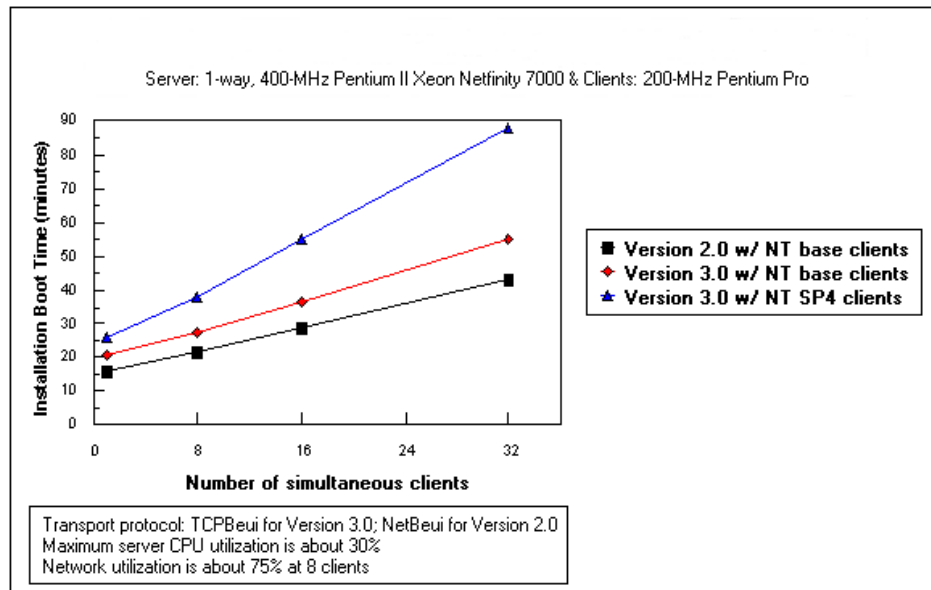


Figure 189. Installation boot performance for NT 4.0 on Token Ring

The 16-Mbps token-ring network quickly becomes the performance bottleneck. The network utilization gets to 75 percent during “peak” periods with only eight clients while the maximum server processor utilization is only about 30 percent. Now what is meant by the “peak” periods? “Peak” periods are those periods during the installation boot sequence which require heavy network bandwidth. In our case, they are the periods during which files are copied from the boot server to a Windows client. Please refer to Section B.3, “Boot Process Overview” on page 444 for a description of the boot process. In particular, in the installation boot sequence, the file copying during Steps 2, 4, 12(c), and 12(d) are the peak periods. The rest of the installation boot sequence is mostly done locally on the client without involving the network or the boot server. In fact, of the 20-minute installation boot time for a base Windows NT 4.0 client, most of the file copying is done in less than eight minutes, so the network is not used much during the remaining 12 minutes. Of course, as more clients are added, the file copying periods (that is, the peak periods) will get longer and longer as the network is fully utilized and becomes the bottleneck. This is reflected in Figure 189 which shows the installation boot time increasing in an almost linear fashion as more clients are added. Let's see what happens on a faster network. Figure 190 on page 454 shows the installation boot performance on a much faster 100-Mbps Ethernet.

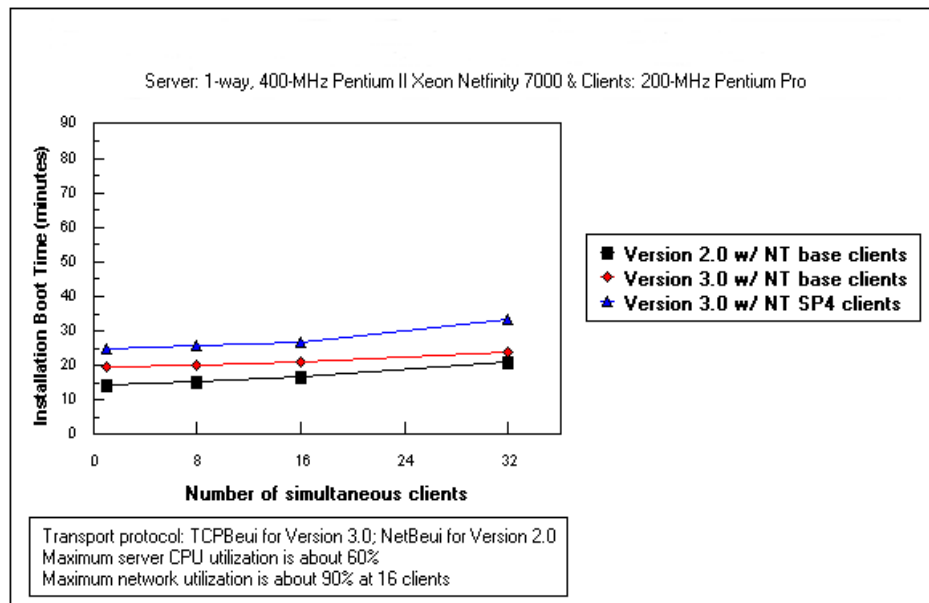


Figure 190. Installation Boot Performance for NT 4.0 - 100 - Mbps Ethernet

On a faster 100-Mbps Ethernet network, the installation boot time improves quite a bit (as expected). With one base Windows NT 4.0 client, the installation boot time is about 20 minutes, as on the token ring, but as more clients are added, the installation time stays almost flat (under 25 minutes) through 32 clients. The boot time for a client with Service Pack 4 stays at just under 40 minutes (with 32 simultaneous clients) compared to the 1.5 hours on the token ring. The server processor is better utilized, but the 100-Mbps Ethernet is still the performance bottleneck as it gets to 90 percent utilized (with 16 clients) during the peak periods. As a result, in an environment similar to our test environment (please refer to Section B.4.1, "Measurement environment" on page 450, for a description of our test environment), administrators should not perform simultaneous installation boots on more than 16 clients. In other words, if the boot server and all clients are connected through a single-segment LAN without sophisticated switches, the installation boots should be "staggered" among groups of, at most, 16 clients. Each group should be allowed to complete the installation boot process before the next group is started. This helps ensure that all clients complete their installation boots properly.

With a 10-Mbps Ethernet, the network is 40 percent utilized even with only one client, so it is almost saturated as soon as a second client is added. As a

result, the installation boot time increases linearly with the number of clients, as shown in Figure 191.

By the time the installation boot sequence is completed, Windows NT 4.0 and any middleware specified by the administrator have already been installed to the client's local disk. Subsequent boots will then go through the much faster hybrid boot process (please refer to Section B.3, "Boot Process Overview" on page 444 for more details). These boots are very fast and deterministic. Figure 192 on page 456 shows how the hybrid (subsequent) boots compared to the initial installation boot.

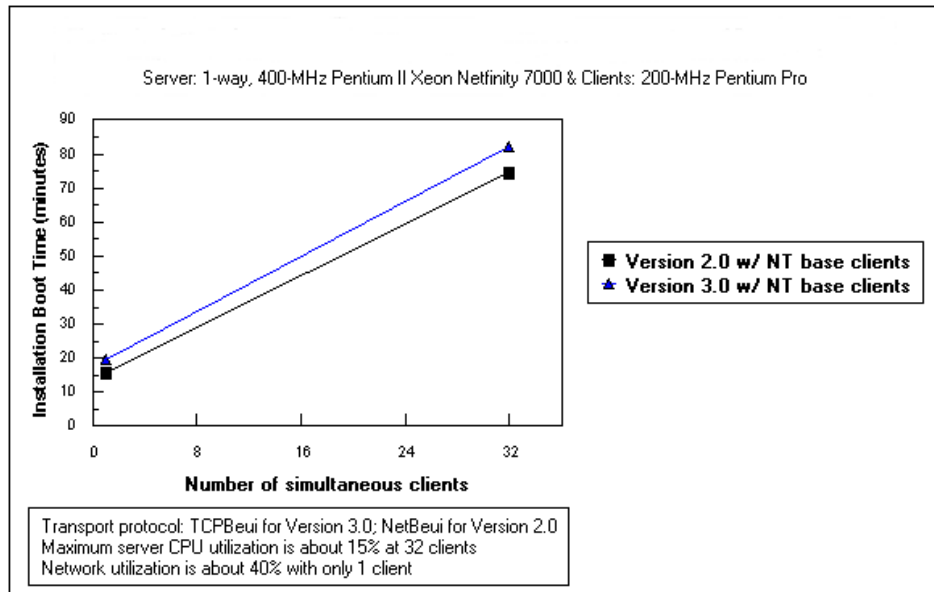


Figure 191. Installation Boot Performance for NT 4.0 Clients - 10-Mbps Ethernet

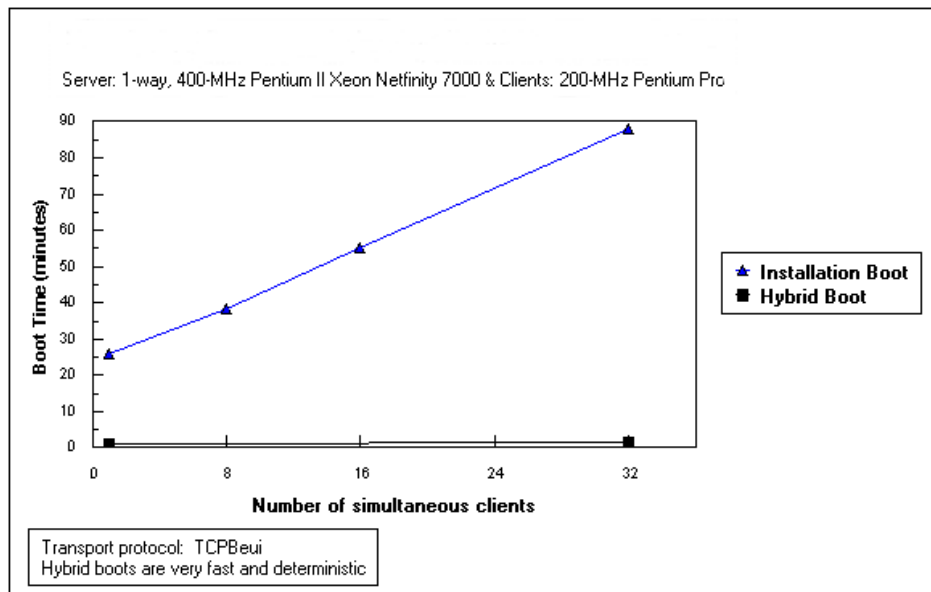


Figure 192. Boot performance for NT 4.0 clients on token ring

### B.4.3 Boot performance for Windows 98 clients

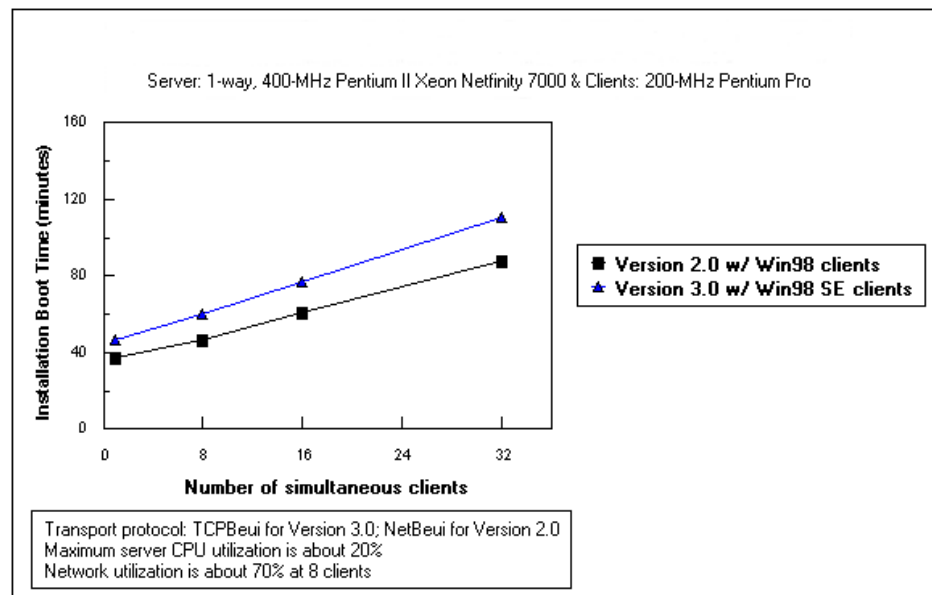


Figure 193. Installation boot performance for Win98 clients on token ring

Figure 193 shows the installation boot performance for Windows 98 clients on a token ring. As with Windows NT clients, the 16-Mbps token-ring network is the performance bottleneck; with eight clients, it is already 70 percent utilized during peak periods. Consequently, the server processor remains under utilized and the installation boot time increases almost linearly with the number of clients booting simultaneously. The boot times for IBM WorkSpace On-Demand 3.0 are also slightly larger than IBM WorkSpace On-Demand 2.0. This can be attributed to the following reasons:

- IBM WorkSpace On-Demand 3.0 only supports Windows 98 Second Edition (SE) whereas its predecessor supports the earlier Windows 98 version. The Windows 98 SE image is slightly larger than the original Windows 98 image. This translates into more time required to copy files from the boot server to the client.
- Unlike its predecessor, IBM WorkSpace On-Demand 3.0 does not support RIPL technology. IBM WorkSpace On-Demand 3.0 only supports PXE technology, which is more flexible and portable, but its design constraints dictate the use of multiple bootstraps, which add to the total boot time

(please refer to Section B.3, “Boot Process Overview” on page 444 for more details).

- For Windows clients, IBM WorkSpace On-Demand 3.0 only supports the TCPBeui transport protocol, which is relatively slower than the NetBeui protocol used in IBM WorkSpace On-Demand 2.0. The total amount of data going across the network using TCPBeui on IBM WorkSpace On-Demand 3.0 is 251 MB, compared to only 197 MB for IBM WorkSpace On-Demand 2.0 with NetBeui.
- IBM WorkSpace On-Demand 3.0 runs on Windows NT Server 4.0 environment while IBM WorkSpace On-Demand 2.0 runs on OS/2 Warp Server for e-business.

Compared to Windows NT 4.0 clients, the installation boot time for a Windows 98 SE client is longer: 46 minutes compared to just 25 minutes for Windows NT 4.0 SP4. This is mainly because of the time required to set up and install Windows 98 SE on the client: it takes almost 20 minutes to set up and install Windows 98 SE after files have already been downloaded, compared to the eight minutes for Windows NT 4.0 SP4. However, it should be noted that the process of setting up and installing the Windows operating systems is done locally on the clients, and the network is not involved much at all.

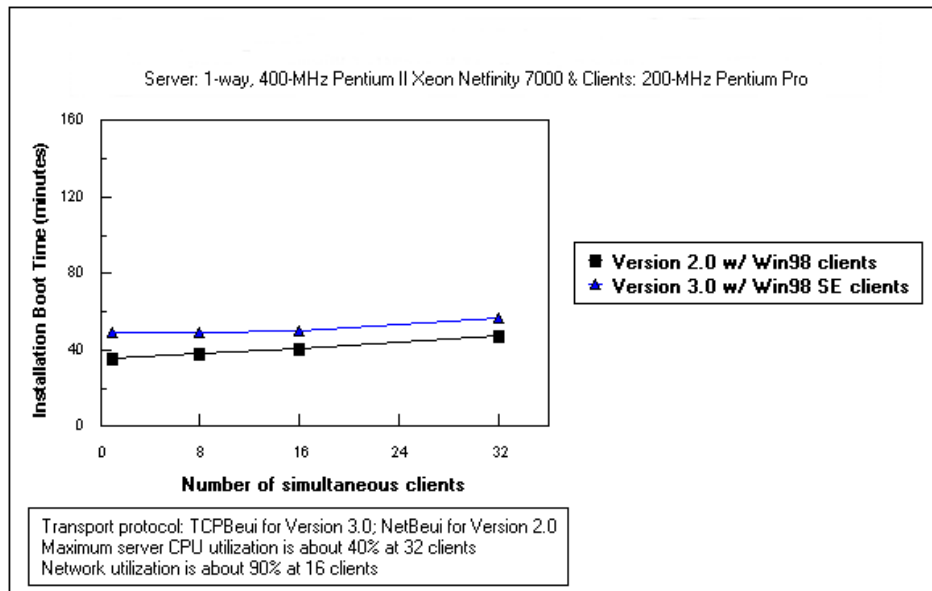


Figure 194. Installation boot performance for Win98 clients - 100-Mbps Ethernet

Figure 194 on page 458 shows the installation boot performance for Windows 98 clients on a 100-Mbps Ethernet. With this faster network, the boot time remains flat as the number of simultaneous clients increases. The server processor is better utilized, but the network is still the performance gating factor. Figure 195 on page 460 shows what happens on a much slower 10-Mbps Ethernet. The network becomes a severe bottleneck, so the installation boot time increases significantly. In fact, in our test environment, some clients fail to continue the boot process. As a result, it is highly recommended that, in an environment similar to our test environment (please refer to Section B.4.1, “Measurement environment” on page 450 for a description of our test environment), administrators should not perform simultaneous installation boots on more than 16 clients. In other words, if the boot server and all clients are connected through a single-segment LAN without any sophisticated switches, installation boots should be “staggered” among groups of at most 16 clients. Each group should be allowed to complete the installation boot process before the next group is started. This helps ensure that all clients complete their installation boots properly. Even though the installation boots take much time, they are only used rarely. After everything has been set up on the client's local disk, subsequent boots will go through the much faster and more predictable hybrid boots. Figure 196 on page 461 shows how fast and predictable booting Windows 98 SE from the client's local disk is.

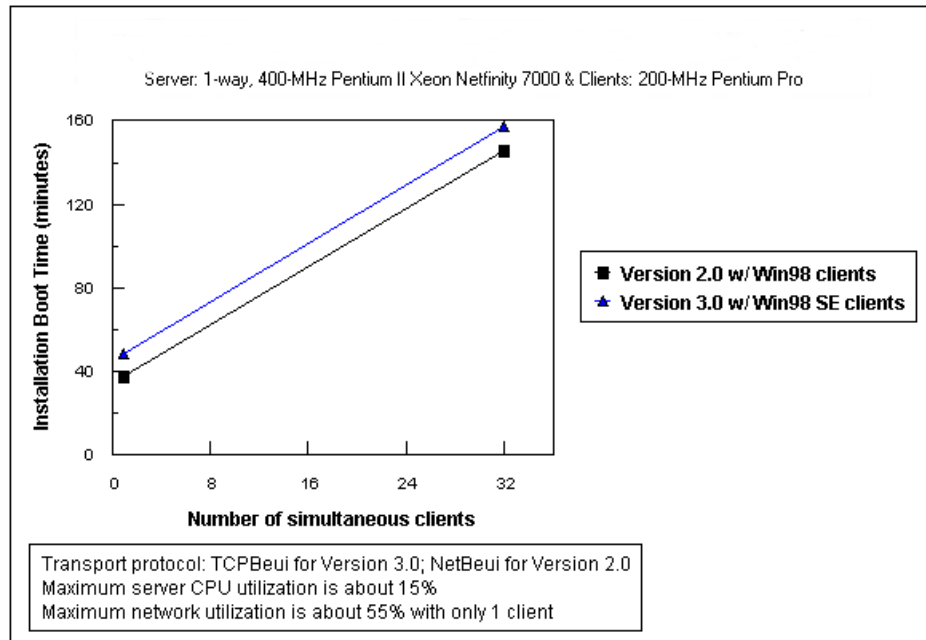


Figure 195. Installation boot performance for Win98 clients - 10-Mbps Ethernet



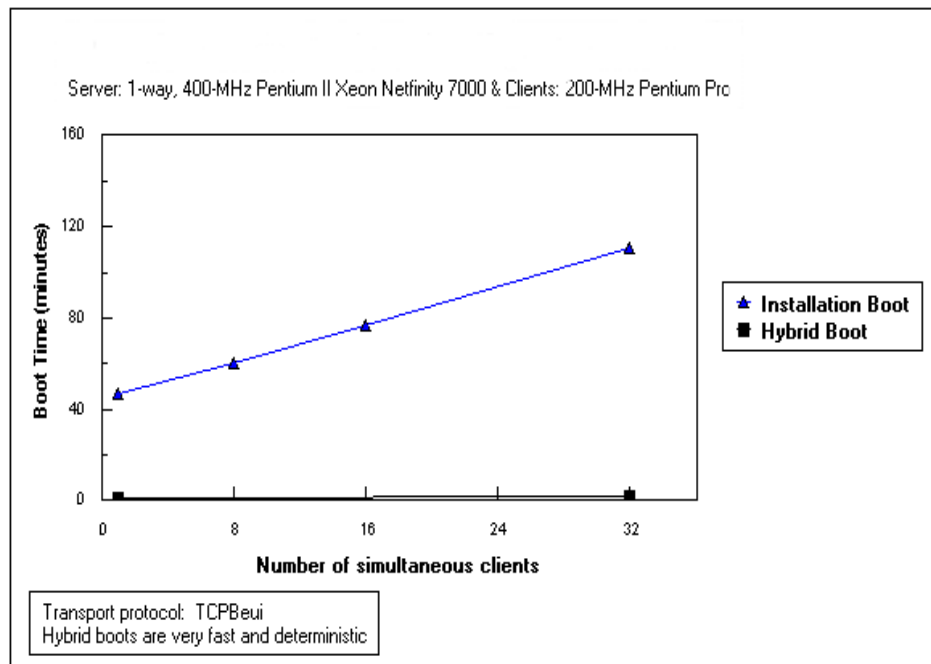


Figure 196. Boot performance for Win98 clients on token ring

#### B.4.4 Boot performance for OS/2 Warp clients

Unlike the remote boot/install approach used for Windows clients, IBM WorkSpace On-Demand 3.0 uses a pure remote boot process for OS/2 Warp clients. The OS/2 operating system is not downloaded to the client's local disk; rather, it is remote-booted from the server. Please refer to Section B.3, "Boot Process Overview" on page 444 for a more detailed description of the remote boot process. Figure 197 on page 462 shows the remote boot performance for OS/2 clients on the token ring.

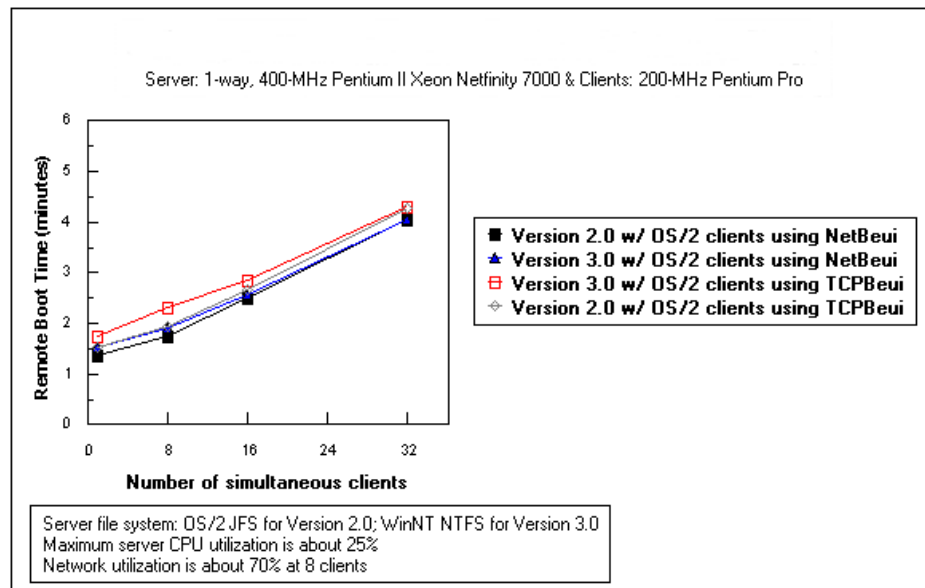


Figure 197. Remote boot performance for OS/2 clients on token ring

The OS/2 remote boot time increases as more clients are added. The 16-Mbps token ring is about 70 percent utilized with eight clients and becomes a performance bottleneck shortly thereafter, leaving the server processor under-utilized at 25 percent (maximum). Compared to the installation boot times for Windows clients, the OS/2 remote boot times are much quicker. However, they are slower and more dependent on network utilization than the hybrid boot times for Windows clients. IBM WorkSpace On-Demand 3.0 supports both TCPBeui and NetBeui transport protocols for OS/2 clients, but the TCPBeui protocol is recommended, especially since there is no significant performance difference in remote boot times.

Figure 198 on page 463 shows the OS/2 remote boot performance on a faster 100-Mbps Ethernet. Here the network is no longer a bottleneck, so the OS/2 remote boot time stays essentially flat under two minutes with up to 32 clients. With 32 clients simultaneously booting, the server processor (a 400-MHz Pentium II Xeon), not the network, becomes the performance bottleneck, as it is 90 percent utilized (the 100-Mbps Ethernet is less utilized at 75 percent).

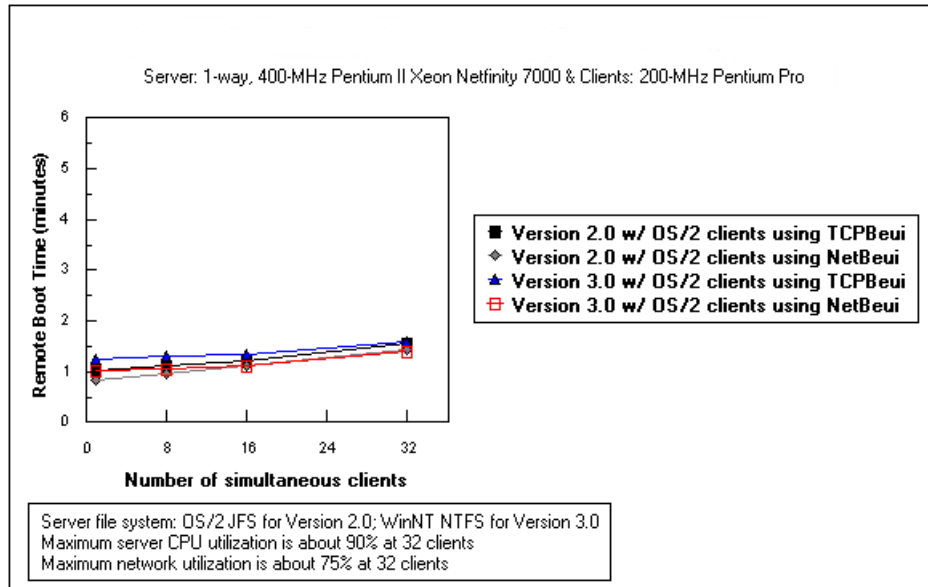


Figure 198. Remote boot performance for OS/2 clients - 100-Mbps Ethernet

Figure 199 on page 464 shows a completely different story with a 10-Mbps Ethernet. This slower Ethernet becomes the performance bottleneck very quickly, causing the OS/2 remote boot time to increase to more than six minutes with 32 clients and leaving the server processor sitting mostly idle. IBM WorkSpace On-Demand 3.0 also supports local boot caching for OS/2 clients. Please refer to Section B.3, “Boot Process Overview” on page 444 for more details on how this caching is done. With the local boot caching option, the first boot takes a much longer time since the cache (the client's local disk) must be initialized; files that are cached must first be copied from the boot server to the cache. Subsequent boots can then take advantage of this cache to boot much more quickly.

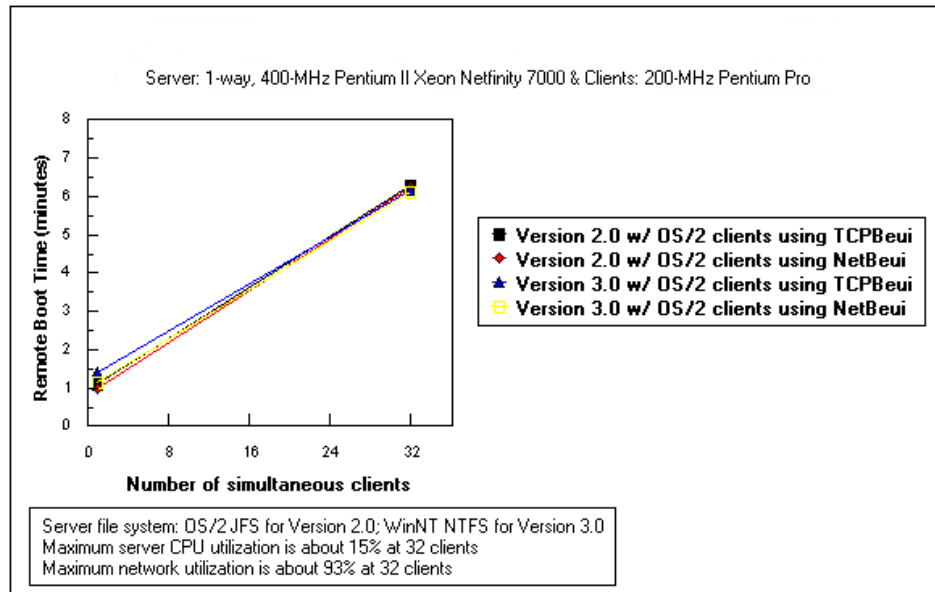


Figure 199. Remote boot performance for OS/2 clients - 10-Mbps Ethernet

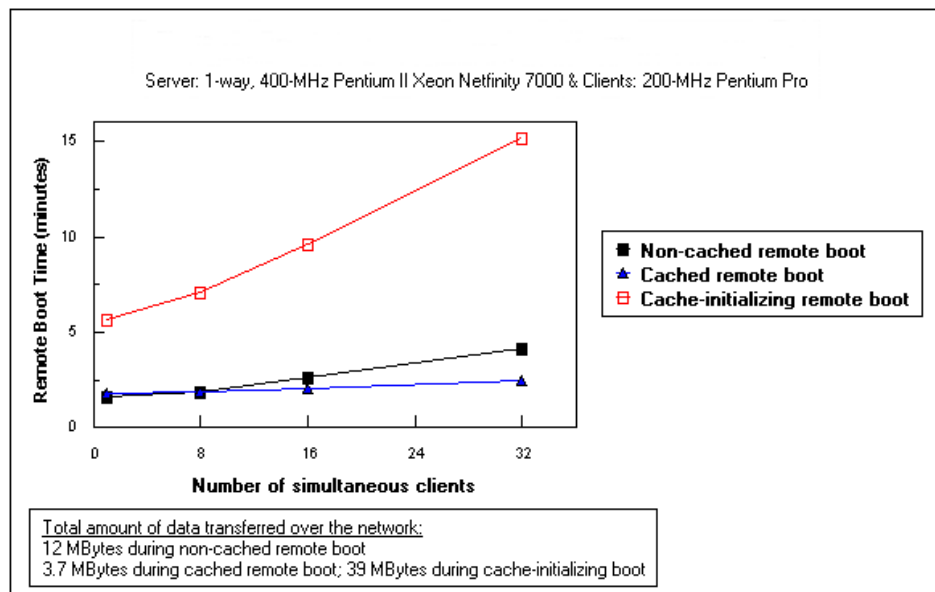


Figure 200. Remote boot performance for OS/2 clients on token ring

Figure 200 on page 464 shows the remote boot performance for OS/2 clients with and without the local caching option. As expected, the cache-initializing boot takes much longer compared to a normal, non-cached remote boot; with one client, it takes over five minutes, compared to just under two minutes for the non-cached boot. Since the token ring quickly becomes the performance bottleneck, the cache-initializing boot time increases dramatically as more and more clients are doing the same thing. However, once the cache has been initialized, subsequent remote boots can take advantage of the cache; the cached remote boot time stays essentially flat at under two minutes through 32 clients. The total amount of data going across the network during a cached boot is only 3.7 MB, compared to 12 MB during a non-cached boot, and 39 MB during a cache-initializing boot. Consequently, the network utilization is much smaller during cached boots, allowing more clients to be supported simultaneously on the network. Thus, with the local caching option, more clients can be supported on a given network, provided the initial cache-initializing boots are properly staggered.

#### **B.4.5 Log-on performance**

To many users sitting at the clients, the time required to log on to the main server (the primary domain controller) can be very important. Log-on time denotes the time interval from the moment the user enters his or her user ID, password, and presses Enter until the user desktop is completely displayed and ready to receive user inputs. After the user ID and password are entered, the Workspace Log-on Client validates them against the native server (in this case, the Windows NT Server). As mentioned in Section B.2, "Introduction" on page 436, IBM Workspace On-Demand 3.0 supports multiple pluggable shells and desktops which can be enabled on a per-user basis by the system administrator. All of this per-user desktop information is stored in a user profile that is downloaded from the server during the log-on process, after the user ID and password have been validated. After the user profile is downloaded, the desktop information is applied to the client machine to personalize the machine for the user for the duration of the entire log-on session. In addition, if the applications assigned to the user require specific system files to be placed in the local Windows system directories on the client, those files are downloaded from the server to the client during the log-on process to enable the client machine to run all applications assigned to the user.

In our tests, the administrator user ID is used and the user desktop is the default desktop for the client operating system under test. There are no applications assigned to the user. All log-on measurements were obtained after the client system had "settled down" after the log-on window appears;

otherwise, the log-on times could be as long as 40 seconds for a Windows NT 4.0 client. Two types of log-on measurements were obtained: the first log-on and subsequent log-ons. The first log-on refers to the first log-on after the client operating system (and any other middleware specified by the administrator) has been set up completely on the client. This is the first real user log-on, not the intermediate log-ons sometimes required during the Windows set-up. For example, Windows 98 SE asks the user to log on and verify his or her password before the installation completes, but this is an intermediate log-on, not the real user log-on. Subsequent log-ons were measured after the user has logged off the domain controller at least once after the client operating system has been set up completely.

Table 13 shows the IBM WorkSpace On-Demand 3.0 log-on times for OS/2, Windows NT 4.0, and Windows 98 SE clients. The log-on times are essentially the same whether there is only one user logging on or 32 users logging on simultaneously to the domain controller. This indicates that the user log-on process itself does not require much network bandwidth or processor power. However, there is a little difference between the first and subsequent log-ons. For OS/2 clients, the small difference between first and subsequent log-on times can be attributed to the various caching effects which help the subsequent log-on times (most of the log-on client's code could have been cached during the first log-on). For Windows clients, the larger difference between first and subsequent log-on times can be attributed to the fact that, after the user profile has been downloaded from the server, the WorkSpace Log-on Client has to apply the desktop information in the user profile to the client machine (that is, "build" the user desktop) during the first log-on process. Once the desktop has been built on a client machine, there is no need to do it again if the user logs on to the same client machine, thus reducing the subsequent log-on times.

*Table 13. IBM WorkSpace On-Demand 3.0 log-on performance*

| <b>Client Type</b> | <b>Number of Clients</b> | <b>First Log-on</b> | <b>Subsequent log-on</b> |
|--------------------|--------------------------|---------------------|--------------------------|
| OS2                | 1                        | 4                   | 3                        |
|                    | 32                       | 4                   | 3                        |
| Windows NT 4.0     | 1                        | 10                  | 7                        |
|                    | 32                       | 10                  | 7                        |
| Windows 98SE       | 1                        | 9                   | 3                        |
|                    | 1                        | 10                  | 3                        |

## B.4.6 Server memory usage

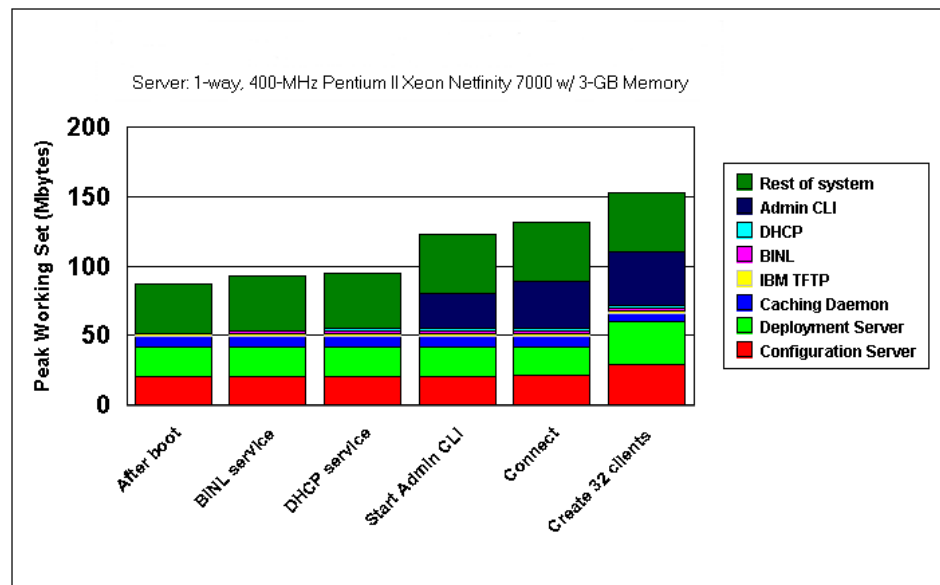


Figure 201. Total peak memory working set for setting up clients

As mentioned in Section B.4.1, “Measurement environment” on page 450, our test environment uses a single physical server machine. This server runs on Windows NT 4.0 Server and is the primary domain controller. All of IBM WorkSpace On-Demand 3.0 server components run on this server. In addition, we did not use a separate machine for the administration console, so the Administration GUI and Command Line Interface (CLI) also run on the server.

Figure 201 and Figure 202 on page 469 show the total peak memory working set of all processes running on the server as reported by the Windows NT Performance Monitor. The peak memory working set of a process is the maximum number of memory bytes in the working set of that process at any point in time. In other words, the peak working set of a process represents the total amount of memory that has been accessed by all threads in that process, regardless of access frequency. As a result, this data represents the high water-mark in memory usage. The true memory working set, which really matters in the overall system performance, is probably much smaller. The peak memory working set data, being the high water-mark in memory usage, is still useful when we do capacity planning for server hardware. In Figure 201 and Figure 202 on page 469, the peak working set data are

broken down into different IBM WorkSpace On-Demand 3.0 server components to make it easier for capacity planning activities.

After IBM WorkSpace On-Demand 3.0 has been installed and set up on the server, the Windows NT Performance Monitor reports that the total peak working set of the whole system is just under 100 MB. As noted in Section B.2, “Introduction” on page 436, the configuration and deployment servers are implemented in Java for cross-platform server support. Each requires a separate Java Virtual Machine (JVM), so they can consume quite a bit of memory (each has about 20 MB of peak working set). The core IP Services, which include DHCP, BINL, and TFTP, do not use much memory; in fact, their peak working set data are barely visible in Figure 201 on page 467. However, the GUIs used to configure the IP Services do use a lot of memory, so it is recommended that these IP Services Configuration GUIs should be terminated as soon as they are no longer needed to conserve memory resources.

After booting up the server and starting the IP Services, the administrator needs to bring up the administration GUI or CLI to set up the users, client machines, or applications. Both the GUI and CLI are implemented in Java, so invoking either of them would start another JVM, causing more memory to be used. As a result, it is highly recommended that the administrator should terminate the administration GUI and CLI as soon as they are no longer needed. Executing commands to create or delete client machine definitions through the CLI increases the peak working set of the configuration server, deployment server, as well as the CLI, by about 20 MB (total). One key observation here is that the total peak memory working set of the whole system does not exceed 200 MB in Figure 201 on page 467. This is well within the minimum server memory requirements specified for IBM WorkSpace On-Demand 3.0 (please refer to Section B.2.3, “Server requirements” on page 443 for more details).

After all client machines, users, and applications have been set up properly, it is time now to boot up the clients. Figure 202 on page 469 shows the total peak memory working set of all processes running on the server as the server is booted up, the IP Services are started (not the configuration GUIs!), and OS/2 clients are remote-booted from the server. The total peak memory working set of the server does not seem to increase much from the time the server boots up until all 32 OS/2 clients come up. The key thing here is that the total peak memory working set of the whole system never exceeds 100 MB; again, well below the minimum server memory requirements specified for IBM WorkSpace On-Demand 3.0 (please refer to Section B.2.3, “Server requirements” on page 443 for more details).



Since the real memory working set is probably much smaller than the peak working set reported by the Windows NT Performance Monitor, it can be safely concluded from the data in Figure 201 on page 467 and Figure 202 that the minimum server memory requirements specified for IBM WorkSpace On-Demand 3.0 should be quite sufficient as long as the Administration GUI, CLI, and the configuration GUIs for IP Services are terminated when they are not needed. It would be even better if the administrator's console ran on a separate machine.

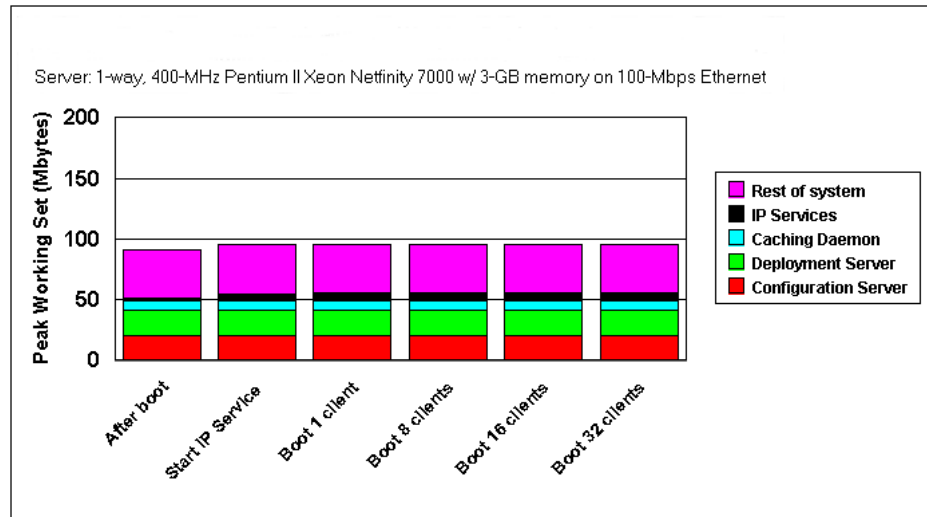


Figure 202. Total peak memory working set for client boot on NT 4.0, OS/2 clients

#### B.4.7 Factors affecting application performance

When using IBM WorkSpace On-Demand 3.0, there are several factors that should be taken into account by the administrator to maximize the performance of applications. The administrator should consider the following:

- Applications run locally on the client, not on the server, so their performance depends in large part on the processing speed of the clients.
- Most Windows application files are split between the client and the server, so the Windows application performance also depends on the nature of the applications themselves:

For those Windows applications that can have the majority of their files reside on the server, their performance depends on:

- Processing speed of the client (for application execution)

- Speed of the server, server load, and file system on the server (for file accesses)
- Available network bandwidth

For those Windows applications which require most of their files to reside locally on the client, their performance primarily depends on the speed of the client system, not on the server and the network, but with the additional costs of disk space on the client and installation time.

OS/2 application files reside on the server (the OS/2 client systems can be diskless), so their performance depends on:

- Processing speed of the client (for application execution)
- Speed of the server, server load, and file system on the server (for file accesses)
- Available network bandwidth

---

## **B.5 Performance recommendations**

Based on the performance analysis in Section B.4, “Performance data & analysis” on page 449, the system administrator should follow the following key recommendations to get the maximum performance and effectiveness when using IBM WorkSpace On-Demand 3.0. Please refer to Section B.4, “Performance data & analysis” on page 449 for more details.

### **General Recommendations**

- Since applications run locally on the client, the client should have adequate processing power to run applications. Servers are used primarily as file servers by the applications, so the server's file system performance is also important.
- As with other server-managed network computing environments, the network bandwidth is the key performance factor.
- The Administration GUI and CLI, as well as the configuration GUIs for IP Services, should be terminated as soon as they are no longer needed, to conserve memory resources.

### **For Windows NT 4.0 Workstation and Windows 98 SE clients**

- If the boot server and all clients are connected through a single-segment 16-Mbps token ring, or even a 100-Mbps Ethernet, without any sophisticated network switches, it is recommended that the system administrator should not have more than 16 clients doing installation boots

simultaneously. In other words, installation boots should be “staggered” among groups of at most 16 clients. Each group should be allowed to complete the installation boot process before the next group is started. This helps ensure that all clients complete their installation boots properly. Please refer to Section B.4.2, “Boot performance for Windows NT 4.0 clients” on page 452, and Section B.4.3, “Boot performance for Windows 98 clients” on page 457 for more information on how much time it would take to boot 16 clients simultaneously. Note that one of the ways to alleviate this restriction is to use a multi-segment LAN with switches or hubs. Upgrading the server or client processors and/or memory would not help here because the network is the bottleneck.

- Installation boots should be performed in off-peak hours, such as at night.
- The installation boot time for Windows NT 4.0 clients is generally faster than that for Windows 98 SE.
- After the installation boot, the user is free to reboot his or her client system as frequently as he or she wants, and at any time he or she prefers, because these boots go through the hybrid boot sequence. Hybrid boots are very fast and do not require much network bandwidth.
- The administrator should refrain from changing the client install image, which includes the client's operating system, the Win32 JVM, TMA, and Workspace Log-on Client too often, because doing so would prevent that client from taking advantage of the hybrid boot sequence.

#### **For OS/2 Warp clients**

- A single-segment 16-Mbps token ring without switches would become the performance bottleneck with eight clients booting simultaneously. On a 100-Mbps Ethernet, the network becomes less of a bottleneck; with 32 clients booting at the same time, a server processor equivalent to a 400-MHz Pentium II Xeon would become the bottleneck first.
- Compared to Windows clients, the remote boot time for OS/2 clients is much faster than the installation boot time, but it is slower and much more dependent on the network than the hybrid boot time.
- More OS/2 clients can be supported on a given network with the local caching option, provided that the initial cache-initializing boots are “staggered” properly. These cache-initializing boots consume a lot of network bandwidth because of the need to copy files from the boot server to the cache (client's local disk). The administrator should refrain from modifying the cached image, but if he or she must do it, then it should be done during the off-peak hours. It should also be noted that, with the local

caching option, all files on the client's local disk will be deleted before the cached files are copied from the boot server to the client's local disk.

### **Log-on performance**

- For Windows users, the system administrator should refrain from adding and/or deleting user access to applications frequently. Every time the administrator assigns a new application to (or removes one from) a user, that user's next log-on will be longer.
- If a Windows user roams to another client system which has not been enabled to run all of the applications assigned to him or her, the first log-on time will be longer because that client system would have to be updated during the log-on process.

---

## **B.6 Additional information sources**

For more information, the reader is referred to the following documentation:

"Technical Overview of IBM Workspace On-Demand 3.0" which can be found on the Web at <http://www.ibm.com/software/network/workspace/3.0/library>

"Administration Guide for IBM WorkSpace On-Demand 3.0" that is shipped with the product. This documentation is also available in PDF format on the Web at <http://www.ibm.com/software/network/workspace/3.0/library>

"Pre-boot eXecution Environment Specifications", Version 2.1, Intel Corporation, September 1999.

Additional documentation and technical guidance can also be provided by the IBM e-business operating system solution's Rapid Deployment Team.

---

## **B.7 Acknowledgments**

This paper has been extensively reviewed by many of our colleagues in Architecture, Planning, Development, Performance, Legal, and Marketing teams. In particular, we are very grateful for the valuable comments from CJ Paul, Steven French, Kerry Fehlis, Glenn Brew, Marc-Arthur Pierre-Louis, Rajiv Arora, Herb Hintze, Larry Thorn, and Linda Magnuson. We would also like to thank our management team, Patricia Orlosky, Farhad Fatahalian, and Jerry Kilpatrick, for their input and support without which this work would not be possible.

---

## **B.8 Disclaimer**

The performance values shown in this paper were derived using particular, well-configured computer systems. All performance data and analysis are provided “AS IS” and no warranties or guarantees are expressed or implied by IBM. Actual system performance may vary and is dependent upon many factors including system hardware configuration, software design and configuration. Customers should consult other sources of information to evaluate the performance of systems they are considering buying and should consider conducting application oriented testing. For additional information about the performance data, analysis, and systems tested, please contact your IBM local Branch Office or IBM Authorized Reseller.

IBM may have patents or pending patent applications covering subject matter in this paper. The furnishing of this paper does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

---

## **B.9 Trademarks & Copyrights**

IBM, IBM WorkSpace On-Demand, AIX, OS/2 and PC DOS 2000 are registered trademarks of International Business Machines Corporation.

Microsoft and Windows are trademarks of Microsoft Corporation.

Java is a trademark of Sun Microsystems, Incorporated.

Other company, product and service names may be trademarks or service marks of others.

(c) Copyright International Business Machines Corporation, 2000. All rights reserved.



---

## Appendix C. Special notices

This publication is intended to help IBM technical personnel, Business Partners, and customers to plan, install, and configure IBM WorkSpace On-Demand 3.0.1. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM WorkSpace On-Demand 3.0.1. See the PUBLICATIONS section of the IBM Programming Announcement for IBM WorkSpace On-Demand 3.0.1 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.


Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee

that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

|                      |                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------|
| IBM                  | Domino                                                                                            |
| Netfinity            | Network Station                                                                                   |
| OS/2                 | Passport Advantage                                                                                |
| Presentation Manager | PS/2                                                                                              |
| Redbooks             | Redbooks Logo  |
| SP                   | System/390                                                                                        |
| Wake on LAN          | WIN-OS/2                                                                                          |
| Wizard               | XT                                                                                                |

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Københavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.



UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Lotus, Approach, Freelance Graphics, Lotus Notes, Lotus Organizer, Word Pro, and Lotus SmartSuite are registered trademarks of Lotus Development Corporation.

Other company, product, and service names may be trademarks or service marks of others.



---

## Appendix D. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

---

### D.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 481.

- *Beyond DHCP - Work your TCP/IP Internetwork with Dynamic IP*, SG24-5280
- *WorkSpace On-Demand Handbook*, SG24-2028
- *IBM WorkSpace On-Demand Handbook Release 2.0*, SG24-5117
- *WorkSpace On-Demand 2.0 Feature for Windows Clients*, SG24-5396
- *IBM WorkSpace On-Demand Customer Scenarios*, SG24-5107

---

### D.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at [ibm.com/redbooks](http://ibm.com/redbooks) for information about all the CD-ROMs offered, updates and formats.

| CD-ROM Title                                                       | Collection Kit Number |
|--------------------------------------------------------------------|-----------------------|
| IBM System/390 Redbooks Collection                                 | SK2T-2177             |
| IBM Networking Redbooks Collection                                 | SK2T-6022             |
| IBM Transaction Processing and Data Management Redbooks Collection | SK2T-8038             |
| IBM Lotus Redbooks Collection                                      | SK2T-8039             |
| Tivoli Redbooks Collection                                         | SK2T-8044             |
| IBM AS/400 Redbooks Collection                                     | SK2T-2849             |
| IBM Netfinity Hardware and Software Redbooks Collection            | SK2T-8046             |
| IBM RS/6000 Redbooks Collection                                    | SK2T-8043             |
| IBM Application Development Redbooks Collection                    | SK2T-8037             |
| IBM Enterprise Storage and Systems Management Solutions            | SK3T-3694             |

---

### D.3 Referenced Web sites

These Web sites are also relevant as further information sources:

- <http://developer.intel.com/ial/WfM/wfm20/design/pxedt>
- <http://www.ibm.com/software/workspace>

- <http://www.fatbrain.com>
- <http://www.mozilla.org/rhino>
- <http://support.microsoft.com/support/kb/articles/q166/0/28.asp>
- <http://support.microsoft.com/support/kb/articles/Q155/1/97.asp>
- <http://www.os2ss.com>
- <http://hobbes.nmsu.edu>

---

## How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** [ibm.com/redbooks](http://ibm.com/redbooks)

Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

Send orders by e-mail including information from the IBM Redbooks fax order form to:

|                            |                                                                                                                                                                         |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            | <b>e-mail address</b>                                                                                                                                                   |
| In United States or Canada | <a href="mailto:pubscan@us.ibm.com">pubscan@us.ibm.com</a>                                                                                                              |
| Outside North America      | Contact information is in the "How to Order" section at this site:<br><a href="http://www.elink.ibm.link.ibm.com/pbl/pbl">http://www.elink.ibm.link.ibm.com/pbl/pbl</a> |

- **Telephone Orders**

|                           |                                                                                                                                                                                      |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| United States (toll free) | 1-800-879-2755                                                                                                                                                                       |
| Canada (toll free)        | 1-800-IBM-4YOU                                                                                                                                                                       |
| Outside North America     | Country coordinator phone number is in the "How to Order" section at this site:<br><a href="http://www.elink.ibm.link.ibm.com/pbl/pbl">http://www.elink.ibm.link.ibm.com/pbl/pbl</a> |

- **Fax Orders**

|                           |                                                                                                                                                                      |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| United States (toll free) | 1-800-445-9269                                                                                                                                                       |
| Canada                    | 1-403-267-4455                                                                                                                                                       |
| Outside North America     | Fax phone number is in the "How to Order" section at this site:<br><a href="http://www.elink.ibm.link.ibm.com/pbl/pbl">http://www.elink.ibm.link.ibm.com/pbl/pbl</a> |

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

### IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at <http://w3.itso.ibm.com/> and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at <http://w3.ibm.com/> for redbook, residency, and workshop announcements.

---

## IBM Redbooks fax order form

Please send me the following:

| Title | Order Number | Quantity |
|-------|--------------|----------|
|       |              |          |
|       |              |          |
|       |              |          |
|       |              |          |
|       |              |          |
|       |              |          |
|       |              |          |
|       |              |          |
|       |              |          |

---

|            |           |
|------------|-----------|
| First name | Last name |
|------------|-----------|

---

|         |
|---------|
| Company |
|---------|

---

|         |
|---------|
| Address |
|---------|

---

|      |             |         |
|------|-------------|---------|
| City | Postal code | Country |
|------|-------------|---------|

---

|                  |                |            |
|------------------|----------------|------------|
| Telephone number | Telefax number | VAT number |
|------------------|----------------|------------|

---

|                                                     |  |
|-----------------------------------------------------|--|
| <input type="checkbox"/> Invoice to customer number |  |
|-----------------------------------------------------|--|

---

|                                             |  |
|---------------------------------------------|--|
| <input type="checkbox"/> Credit card number |  |
|---------------------------------------------|--|

---

|                             |                |           |
|-----------------------------|----------------|-----------|
| Credit card expiration date | Card issued to | Signature |
|-----------------------------|----------------|-----------|

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**

---

## Abbreviations and acronyms

|                      |                                                  |
|----------------------|--------------------------------------------------|
| <b><i>BINL</i></b>   | Boot Information<br>Negotiation Layer            |
| <b><i>CLI</i></b>    | Command Line<br>Interface                        |
| <b><i>CSA</i></b>    | Common Systems<br>Administration                 |
| <b><i>DHCP</i></b>   | Dynamic Host<br>Configuration Protocol           |
| <b><i>e-Boss</i></b> | IBM e-Business<br>Operating Systems<br>Solutions |
| <b><i>GUI</i></b>    | Graphical User<br>Interface                      |
| <b><i>IBM</i></b>    | International Business<br>Machines Corporation   |
| <b><i>ITSO</i></b>   | International Technical<br>Support Organization  |
| <b><i>JRE</i></b>    | Java Runtime<br>Environment                      |
| <b><i>JVM</i></b>    | Java Virtual Machine                             |
| <b><i>LCF</i></b>    | Lightweight Client<br>Framework                  |
| <b><i>NFS</i></b>    | Network File System                              |
| <b><i>POST</i></b>   | Power-On Self Test                               |
| <b><i>PXE</i></b>    | Preboot eXecution<br>Environment                 |
| <b><i>RMI</i></b>    | Remote Method<br>Invocation                      |
| <b><i>TFTP</i></b>   | Trivial File Transfer<br>Protocol                |
| <b><i>TMA</i></b>    | Tivoli Management<br>Agent                       |
| <b><i>TSR</i></b>    | Terminate and Stay<br>Resident                   |
| <b><i>WPS</i></b>    | Workplace Shell                                  |





---

## Index

### Symbols

\$\$Rename.txt 369  
\$Display 385  
\$OEM 373  
\$OEM\$ 369  
\$OEM\$ ET 366  
\$OEM\$Textmode 366  
\$TIVOLI 414

### A

access.cmd 120, 139  
Administering the configuration server 95  
Administering the deployment server 96  
Administering users 76  
Administration 57  
administration CLI 6  
Administration Console 5  
administration console 4, 57, 78  
administration GUI 5  
Administrative GUI 29  
administrator privilege 24  
Adobe Acrobat Reader 277, 278, 299, 301, 318, 320  
AIX Server 3  
AL 254  
    minutes 254  
aliases 105  
Appearance 233  
Appearance tab 209, 222  
APPLICATION DEFINE 257  
application FIT 173  
application package 265  
application packages 272  
application roaming 3  
Application sets 263  
Application Support 3  
APPPARTM.INI 256  
ARCHCTL 380  
Architecture 4  
ASCII 253  
Assign 59  
assign 83, 95  
assign profile 84  
ATALK 109, 127  
ATNEXTLOGON 84

attended 24  
ATTRIB 387  
authserver 96  
Auto-Detected 125  
automated uninstallation 54

### B

Background 206, 208, 219, 230, 232, 245  
background 201, 205, 218  
Background Bitmap 253  
background bitmap 244  
background bitmaps 205, 218  
background color 241  
background colors 245  
backgrounds 85  
batch file 84  
batch files 75  
batch script 75  
BB20 177  
BIGBLU.BMP 254  
BINL 16, 18, 47  
BINL Server 52  
BINL server 17, 19  
BINL\_TFTP\_BOOT 96  
BINLSD.CFG 47  
BIOS 10, 16  
Bitmap 206, 219, 230, 245  
bitmap 241  
BitmapDisplay 206, 219, 230  
Bitmapdisplay 245  
bitmaps 205, 218  
BMP 251  
Boot 22  
    Hybrid 22  
boot drive 164  
boot history 105  
Boot Image Negotiation Layer 47  
Boot Information Negotiation Layer 18  
Boot PROM 16, 19  
boot sequence 199  
Boot Server 17  
Bootapps.exe 368  
Bootdrive 164  
BootImageApps 117, 136  
bootstrap 199  
bootstrap routine 17  
bpcommon.isa 173

Broadcast Address 46  
broadcast node 163

## C

Cache 165  
canceltask 97  
ccrtpk 158  
CD key 121, 140, 144, 159, 185  
CDKey 106, 125, 144  
CDROM 165  
CLI 5, 57, 70  
Client 7  
Client Enablement 33  
client hardware 12  
client images 363  
client installation 112, 131  
CLIENT\_NAME 199  
CMDLINES.TXT 374, 375, 415  
cmdlines.txt 123, 142, 197, 367, 368  
color scheme 209, 222  
color schemes 205, 218  
colors 85  
command handler 57  
Command Line Interface 5, 57, 82  
Common Systems Administration 5, 59  
complex commands 66  
conditional branching 194  
conditional error handling 74  
CONFIG 41  
CONFIG.HW 388  
CONFIG.SYS 250, 253, 257  
config.sys 173  
configfile 111, 130  
Configsrc 167  
configuration server 4, 5, 6, 30, 61, 95  
Configuration Service 34, 52  
configuration task 57  
configure 97  
Connect 65  
CONNECT.EXE 21  
Connecting to 95  
    Configuration server 95  
ConsoleManager 61, 62  
control characters 76  
Control Panel 54, 77, 205, 218  
Copy 116, 135, 154  
copy.out 123, 142  
COPYAPP.CMD 389

Corporate environment 352  
Corrective Service Facility 378  
counter.ini 118, 123, 137, 142, 199  
CPU architecture 20  
CreateMachinet 75, 112, 131  
CreateMachinet() 68, 71, 149  
CreateMultiple() 71, 73, 97  
CRTPKG 265, 268, 269, 271  
crtpkg.exe 158  
CSA 5, 59  
custom.inf 120, 139, 197, 368  
customize 248  
cwindows 158

## D

Data Management 2  
datafile 111, 130  
DATASTOR.INI 41  
datastor.ini 84, 112, 131  
datastore 6, 112, 131, 179, 180  
datastore entry 104, 150, 168  
datastore representation 151, 169  
DDNS 163  
DEC40 146  
DEC50T 146  
default password 86  
default.fit 172  
DefaultDesktopBackground 253  
Define 59  
define 83, 95, 103, 202  
Define a user 78  
define action 83  
define machine 105, 124  
Defining Machines 103  
Delete 59  
delete 83, 95, 103, 202  
DeleteAllMachines() 73  
Deleting a user 89  
DELTA.INI 389  
DELTA4.CMD 389  
deployment server 4, 6, 96, 103, 112, 131, 149,  
180, 199, 386  
Deployment Service 34, 52  
Description 110, 129, 148, 166, 245  
Desktop 85  
desktop 201, 239  
desktop background 206, 219, 230  
desktop bitmap 230

- desktop define 215, 226, 238, 245
- desktop object 201, 202
- desktop representation 202
- desktopw98us directory 227
- device 383
- device driver 385, 391, 394
- DHCP 16, 96, 108, 127, 145, 162, 164, 389, 395
- DHCP configuration 43
- DHCP PXE 17, 18, 42
- DHCP PXE Boot 16
- DHCP Server 52
- DHCP server 17, 18, 189
- DHCP service 17, 18, 42
- DHCPACKNOWLEDGE 20
- DHCPDISCOVER 19
- DHCPOFFER 19
- DHCPREQUEST 19
- DHCP.D.CFG 42
- Disconnect 65
- Disconnecting from 95
  - Configuration server 95
- disk partition 191
- Diskette 165
- diskette image 396
- Display Properties 209, 211, 222, 223, 234
- display resolution 121, 140, 159
- DLC 108, 127
- domain 3
- domain name 251
- Domain Name Server 46
- domain prompts 242
- Domain RPLGroup 104
- Domain TDMGroup 77
- DOS 3, 7, 119, 121, 138, 140, 147, 188
- DOS batch files 117, 136, 156
- DOS boot image 109, 115, 128, 134, 153
- DOS bootable image 114, 133, 152
- DOS external commands 117, 136, 156
- DOS FDISK 195
- DOS image 115, 134
- DOS internal commands 117, 136, 155
- DOS shell 194
- dos2kuser 115, 134
- driver 383
- drivers 107, 126
- Dynamic Domain Name Service 163

## E

- end-of-file character 76
- Error 116, 135, 154
- EtherJet 9
- execute() 68
- Exit 66
- exitjs() 70, 75
- Explorer 202

## F

- FAT 197
- FDISK 195, 379
- File Index Table 171, 173
- file input 75
- file object 75
- file operations 75
- file output 75
- FIT 171, 173, 337
- FIT entries 174
- FIT file 171, 247, 258
- fixpack 373
- fixpacks 363
- FixPak 14, 378, 379
- FixTool 378, 380
- FIXTPREP 380
- fixup1.lst 123, 142
- flexibility 94
- FOLDERPOS 257
- Format 116, 135, 154
- Format.inp 123, 142
- format.out 123, 142
- FSERVICE.EXE 378, 379, 382

## G

- games 148
- global group 104
- global variables 86
- Graphical User Interface 57
- group 82
- group define 94
- group delete 94
- groupmember 83, 94
- groupmember delete 94
- groupmember list 94
- GUI 5, 24, 57, 59, 81, 181, 184, 187
- GUID 20

## H

HAL.DLL 383  
hard drive size 191  
Hardfiletype 165  
hardware considerations 10  
Hardware tab 185  
hdboot.com 121, 140  
Help 66  
help 74  
holdqueue 97, 98  
holdtask 97  
home directory 86  
HOMEDIR 91  
homedir 84  
HOMEDRIVE 91  
homedrive 84  
horizontal resolution 166  
Host On-Demand 281  
Hybrid 116, 135, 154, 163  
hybrid 121, 140

## I

I/O control interface 51  
IBM 3  
IBM 10/100 PCI Ethernet adapter 107, 126, 187  
IBM 16/4 PCI Token-Ring adapter 107, 126  
IBM Data Base 2 283  
IBM LAN Server 171, 173  
IBM WorkSpace On-Demand 3.0 250  
IBM WorkSpace On-Demand 3.01 1, 2, 3  
IBMCOM directory 174  
IBMCOM ACS directory 162  
IBMFE 107, 126, 187  
IBMLAN.INI 251  
IBMTDMCNFG 77  
IBMTDMDPLY 77  
icon background 247  
icon layout 85  
icon text 247  
Iconbackground 245  
Iconfont 245  
Iconfontsize 245  
icons 245  
Iconcontext 245  
IDE 165  
IFS 51  
IMAGE\_FILE 115, 134, 153  
IMAGENAME 176

ImageName 106, 125, 144, 161  
Imagename 178  
Import from Windows NT users 81  
Importing users from Windows NT 93  
indexOf() 73  
INF file 9  
INI.RCH 389  
INISYS.RCH 390  
Init 116, 135, 154  
initdisk 119, 138, 156, 195  
Install 116, 135, 154  
Install shell batch files 117, 136, 156  
InstallDOS 117, 119, 120, 136, 138, 139, 156, 157  
Installdos 109, 121, 128, 140, 147, 158  
Installshield 54  
Intel 1, 10, 11  
Intermediate Support Driver 32  
Intermediate Support Drivers 55  
Internal commands 117, 136, 155  
Internet 66  
IOCTL 51  
IP 108, 127, 145, 163  
IP address 17, 63, 145, 163  
IP configuration 16, 17, 18, 20  
IP domain name 108, 127  
IP router 108, 127, 145, 163  
IP Services 42  
IP services 50  
IPXODI 146

## J

Jaaslogon 53  
Java 3, 6, 15, 57, 62, 175  
Java Runtime Environment 15  
Java support 185  
Java Virtual Machine 3, 9, 15, 109, 120, 128, 139, 146, 157  
JavaScript 6, 57, 65, 66, 69, 82, 86, 90, 99, 103, 112, 131, 149, 179  
JavaScript automation 75  
JRE 15, 41  
jsextend.js 69, 70  
JVM 15, 61, 109, 128, 146, 158

## K

Kbdtype 111, 130, 148, 165  
keyboard type 185  
key-value pairs 110, 129, 148

## L

LAN password 254  
Lang 85, 106, 125, 144, 161, 178, 245  
lcmd 410  
Lexmark 164, 166  
License Agreement 26  
Lightweight Client Framework (LCF) 409  
List 59, 66  
list 83, 95, 103  
Load 66  
loadfile() 70  
local boot 8, 105  
Local Cache 13  
Local Cache Server 34  
local group 105  
local groups 77  
local printer support 166  
Local queue 164  
Localadmpw 110, 129, 148  
lookup bitmap 254  
lookup password 254  
log files 104  
logical boot image 174  
Logon 109, 128, 146  
logon assignments 85  
logon bitmap 242  
logon script 84  
LOGON.LST 377  
Logonassignment 85  
LOGONCLT.BAT 372  
logonclt.reg 372  
logonscript 84  
loop control counter 90  
LOS 254  
Lotus Notes 5.04 287, 327  
Lotus SmartSuite 1.5 344, 356, 358, 360  
Lotus SmartSuite Millennium 290  
Lotus Smartsuite Millennium 312, 314, 330, 331  
Louts Notes 5.0x 310  
LPDSRV 51

## M

Mac 143, 161  
MAC address 67, 71, 75, 103, 104, 106, 112, 113, 124, 131, 133, 143, 149, 152, 169, 179, 185, 199  
Machine actions 181, 190  
machine define 66, 97, 105, 111, 121, 124, 130, 140, 143, 149, 161, 191, 197, 199, 245

machine definition 103, 113, 120, 132  
machine delete 179, 180  
Machine Deployment Server 29  
machine FIT 173  
machine list 179  
Machine Management 38  
machine modify 113, 132, 180  
machine monitor 98  
machine object 103  
machine parmlist 162, 165, 167, 180  
machine query 179  
machine representation 103  
machine template 177  
machines directory 115, 134, 153  
machines resource list 189  
machines.js 149  
machinesmachine1 t4sp3us 115, 134  
machinesmachine1  
t4sp3us 120, 123, 139, 142  
machinesmachine2w98w98seus 157, 160  
machinesmachine3os2bb20us 172  
maclist.txt 75  
MACS 199  
MACS directory 113, 133  
macs002035fe4a6f 161  
Managing Client Desktops 201  
Managing client machines 181  
Managing users 78  
managing users 82  
MCOPY 196  
Microsoft Internet Explorer 318  
Microsoft Internet Explorer 5.0 284, 285, 306  
Microsoft Office 2000 292, 316, 318, 333  
misc.lst 120, 139  
MM 41  
ModelName 111, 130, 148  
modify 83, 95, 103, 202  
Monitor 165  
monitor 97  
Mouse 166  
mouse pointers 148  
Mozilla 66  
MSBATCH.INF 378, 386, 394  
msbatch.inf 372  
MSDLC 146  
MSGs 41  
MTFTP 20  
multi.js 97

## N

Name 143, 161  
name 96  
naming convention 143  
nativegroup 82, 94  
nativegroup delete 94  
nativeuser 82, 83, 93, 94  
nativeuser delete 89  
Nbddaddr 163  
NBF 108, 127  
NBSScope 109, 127, 146  
Nbsscope 163  
NCCACHED.SYS 24  
NCCACHEE.EXE 24  
NDISBAN 146  
NDISDD.PTO 395  
NDTOKBAN 146  
Netadapter 107, 126, 145, 162  
Netadapterdir 145  
NETBEUI 146  
NetBEUI 20, 164  
NetBIOS 109, 127  
NetBIOS Datagram Distribution Server 163  
NetBIOS Remote IPL (RIPL) 171  
Netmask 108, 127, 145, 164  
Netscape 175  
Netscape 4.7 296, 302, 304, 321, 323  
Netscape Communicator 341  
Network Adapter 383  
network adapter 10  
network adapter cards 104  
network adapter drivers 147  
Network Adapters 9  
network boot 159  
network boot specification 107, 126  
network card 391  
network drivers 172  
Network Interface Cards 391  
Network Interface Identifier 19  
network parameters 189  
network protocol 159  
network protocol information 121, 140  
Network Station 1  
network-based installation 121, 140  
NETWORKTIMEOUT 101  
networktimeout 96  
New 116, 135, 154  
new users 78  
new() 71

NFS 51  
NFS Server 52  
NFS\_FILE 96  
NFSLINK 146  
NLV 7  
Nodetype 163  
NOLOCK 255  
NOPI 251  
Norton AntiVirus 7.0 Corporate Edition 279  
NOSM1 251  
nt4sp3usnst386\$soem\$ 126, 186  
nt4sp3usnst386\$soem\$  
et 187  
nt4sp3usboot 188  
nt4sp3us  
esp 184  
ntclean.inf 368  
ntconfig.pol 203, 216  
NTFS 14, 197  
NTTempl 179  
ntuser.dat 211, 223  
NWLINK 146  
NWLNKIPX 108, 127  
NWNLINK 146

## O

OEMSETUP.INF 391  
  
omachines 180  
Options 148  
OS 85, 106, 124, 143, 161  
OS/2 2, 3, 4, 6, 14  
OS/2 desktop 245  
OS/2 remote boot 170  
OS/2 Warp 18  
OS/2 Warp 4.0 7  
OS/2 Warp Server for e-Business 3  
OS2.INI 6, 253, 389  
OS2.RCH 388  
os2bb20usreeos2 246  
OS2BB20USCNFGDEFAULT 172  
os2bb20uscnfgdefault 176  
OS2BB20USTREEOS2INSTALL 164, 166, 167  
OS2BB20USVIDEO 166  
os2bitmaps 246  
OS2SYS.INI 390  
os2sys.ini 173

## P

PackageLocation 204, 217, 228  
Paint Shop Pro 286, 308, 325  
Parallel 166  
parallel devices 166  
parmlist 103  
Partition 110, 129, 147, 154  
PartitionSize 117, 136, 192, 195  
password 74, 84, 110, 129, 148, 242, 251  
passwordexpiration 84  
PC300GL 383  
PCNFSD Protocol Server 51, 53  
PCOMM 176  
Pentium 11  
Pentium II 10  
pervasive 109, 128, 146  
Planning your desktop 201  
Plug&Play 384, 392  
PM\_SystemBackground 253  
PMLOGON 249, 250, 251  
PMLOGON Shell 257  
PMLOGON.EXE 250, 251, 254  
PMSHELL 245  
pmshell.exe 202  
Point-to-Point 163  
poedit.exe 207, 213, 225, 229, 232  
policydefaultconfig.pol 227  
PolicyLocation 204, 217, 228  
policysandboxconfig.pol 228  
policysandbox  
tconfig.pol 203, 217  
Portmapper 51  
PortUNCName 111, 130, 148  
POST 16, 19  
Power-On Self Test 16, 19  
prdesc.lst 164  
Preboot Execution Environment 16  
PrebootImage 109, 128, 147  
prebootimage 115, 134, 153  
pre-install 198  
Primary Domain Controller 4, 10  
Primary Domain Name Serve 145  
Primary Domain Name Server 108, 127  
primary partition 147, 191  
print queue 164  
Print() 70  
printer data type 164  
Printer driver 164  
Printer port 164

printer port 164  
Printern 111, 130, 148, 166  
printers 85  
println() 70  
PrintResult() 75  
printResult() 86  
privilege 84  
profile 207, 212, 220, 224, 232  
ProfileLocation 228  
profilesdefault  
tuser.dat 203, 216  
profilesdefaultuser.dat 227  
Program Objects 255  
PROM 19  
Prot disk 51, 52  
protect mode redirector 171  
PROTOCOL.INI 174  
Protocol1 164  
Protocol2 164  
Protocols 108, 127, 146  
PRTDSKCL.EXE 51, 52  
PS/2 Mouse 166  
Publications 29  
PXE 7, 16, 107, 126  
PXE boot 10, 171  
PXE Proxy 20  
PXE Proxy Server 29  
PXEDHCP 19

## Q

Query 59  
query 83, 95, 97, 103, 202  
query action 84  
queued 96

## R

RAM 8, 10  
RASPPPT 108, 127  
real-time 97  
reboot locally 116, 135, 154  
redirection 74  
REGEDIT 243  
REGEDIT4 376  
registry 52, 123, 142  
registry entries 6, 159  
Reguser 107, 110, 129, 147  
releasequeue 97, 98  
reloadcrt 97

- Remote
  - Boot 23
- Remote boot 10
- remote boot 8
- Remote boot capability 10
- remote installation 105
- Remote IPL support 173
- remote management 78
- Remote Method Invocation 57
- remote method invocation 6
- Remote Queue 164
- remote queue 164
- Resolution 166
- Resource tab 60
- Resources 81
- Resources tab 81, 181
- response file 24, 147, 184
- Response files 363
- restrict 84
- resultCode 69
- RGB 206, 219, 230
- Rhino engine 66
- R IPL 171
- RISC 1
- RMI 6, 61
- RMI (Remote Method Invocation) 57
- RMI port 99
- RMI port number 97
- RO 41
- ro directory 104
- roam 203
- rodos2kusree 153
- romachines 104
- romachinesmachine2w98w98seus 153
- romacs 104, 180
- row98w98seusnst 158
- RPC Portmapper 51, 53
- RPL.MAP 171
- RPLFILES 23, 41, 105, 115, 120, 134, 139, 157, 160, 166, 171, 176, 184, 186, 188, 197, 208, 233, 246, 336
- RPLFILES alias 164, 171, 172
- RPLGROUP 77, 105, 176, 197
- RPLGroup 77
- RPLSERVER 177
- RunOnce 372
- Runonce.cmd 368
- RUNONCE.REG 376
- RUNWORKPLACE 250, 252, 253, 257

- RW 41
- rw directory 104
- rwmachines 104, 179

## S

- S3 Trio 3D 383
- Sandbox 167
- sandbox 52, 207, 220, 265
- sandbox shell 202
- sandbox user 202
- schedule 98
- Screen Saver 210, 222
- screen savers 205, 218
- script 74, 97
- SCSI 165, 167
- Section 110, 129, 148
- Security 2
- security 1
  - enhanced 1
- Serial 167
- serial devices 167
- serial ports 85
- Server 106, 124, 143, 161, 164
- server 74, 94
- server assign 96
- server delete 96
- server list 96
- server unassign 96
- SERVERTIMEOUT 101
- servvertimeout 96
- Service Pack 373, 374, 375, 377
- Service Pack 3 306, 310, 312, 314, 316
- Service Pack 4 8
- service packs 372
- SERVICE.EXE 378, 379
- services 34, 52
- setup.exe 370
- shared directories 86
- Shell 85, 245
- shutdown 97
- Shutdown Options 255
- Simplifying commands 66
- SMB\_AUTHENTICATION 96
- software distribution 24
- Software Licence Agreement 25
- Software prerequisites 14
- space-delimited 85
- Srcrespfile 110, 129, 147



- standardize 86
- Start menu 201, 211, 213, 223, 235
- state section 117, 136
- STATE.EXE 21, 396
- State.exe 116, 119, 121, 135, 138, 140, 154, 156, 194
- state.exe 115, 117, 119, 123, 136, 138, 142, 158
- STATE.INI 21
- state.ini 115, 116, 117, 120, 121, 135, 136, 139, 140, 153, 155, 156, 158, 159, 191, 368
- state.ini file 154
- state.ini.bigdrive 192
- state.ini.multidisk 193
- StateNames 116, 135, 154
- Status 109, 128, 146, 167
- status 114
- STREAMS 108, 127
- string concatenation 194
- subnet 45
- substring() 73
- SVGADATA.PMI 388
- Swapper 167
- swapper.dat 167
- syntax 64, 74
- sysdiff 367
- SYSLEVEL 379
- System policies 205, 218, 229
- System Policy Editor 205, 213, 218, 236
- SYSTEM.INH 390

## T

- task 96
- Task flow 60
- task number 98
- Taskbar 211
- task-based users 201
- Tasks menu 354
- TCP Domain Server 163
- TCP/IP 1, 7, 14, 16, 41, 60, 108, 127, 163
- TCP/IP host name 108, 127, 145
- TCP/IP Print Server 51, 53
- TCPBEUI 20, 163, 381
- TCPBeui 164
- Tcpdomain 108, 127, 146, 163
- Tcpname 108, 127, 145, 163
- Tcpnamesrv 108, 127, 145, 163
- Tcprouter 108, 127, 145, 163
- TDM 41

- TDMCLI 41
- tdmcli.bat 74
- TDMCLNT.INF 22
- tdmclnt.inf 121, 140, 152, 199
- TDMConsoleResult 62, 68, 69
- tdmdos.inf 115
- tdmdos.sys 121, 140, 152, 199
- TDMGROUP 77
- TDMGUI 41
- TDMLOGON.BMP 242
- TDMOS2.INF 23
- tdmos2.inf 170, 173
- tdmos2.sys 170
- tdmpkg 202
- TDMPKGS 42, 203, 216, 227, 244, 261, 271, 272, 338
- TDMPRFLS 42
- tdmserver 94, 96, 97
- tdmserver monitor 97
- TDMW32UT.IMG 395
- tdmw32ut.img 109, 128, 147
- Template 107, 125, 144, 162
- template 179, 181
- Templatename 144, 162
- Terminate 66, 97
- TFTP 18, 20, 49, 50, 114, 115, 133, 134, 152
- TFTP protocol 20
- TFTP server 20
- TFTPD 18
- The administration CLI 62
- time zone 121, 140, 159
- Tivoli 24, 120, 139
- Tivoli gateway 416
- Tivoli Management Agent 109, 120, 128, 139, 185, 407
- Tivoli Management Framework 408
- TLOGOFF.EXE 255
- TMA 21, 109, 120, 128, 146, 157, 167, 372, 407
- Token-Ring 9
- token-ring 112, 131, 149
- TPFTP Server 53
- Trace 50
- transform task 57
- Trivial File Transfer Protocol 49
- TRP 107, 126
- true thin-client 7
- TSHUTDOWN.EXE 255
- txtsetup.oem 366
- type 96

TZ 107, 125

## U

Uconfigsys 168  
UN\_2200 381  
unassign 83, 95  
unattend.tx 365  
UNATTEND.TXT 392  
unattend.txt 110, 121, 123, 129, 140, 142, 147, 372  
unattended installation 24, 129, 147  
UNC name 80  
uninstall feature 52  
Uninstalling 52  
Universal Naming Convention 80  
user 82  
User actions 78  
user assign 83, 85, 94, 97, 230, 238, 246  
user define 93  
user delete 89  
user FIT 173  
user interface 205, 218  
user list 93  
user modify 94  
user shell 202  
user.dat 235  
UserDesk 235  
userdesk  
t4usprofilesdesktop 212, 224  
userid 74  
UUID 20

## V

variable substitution 194  
variables 86  
vertical resolution 166  
VFDISK 396  
VGA 10, 125, 165  
Video 125, 165  
video 121, 140  
Video Adapter 383  
video adapter 165, 186, 383, 384  
video driver 385, 388  
Video parameter 165  
Video resolution 126, 145  
video resolution 180  
VIDEO.CFG 388  
Videoinf 126

virtual floppy 396

## W

w98se 144  
w98w98seus 158  
w98w98seusnstcwindows 233  
Wake on LAN 107, 126, 187, 198  
wallpaper 206, 219, 230  
wild cards 175  
WIN.INH 390  
Win32 7  
WIN98\_SESETUPTOOLSRESKIT ETADMINPOL  
EDIT 229  
Windows 2000 1, 203, 204, 205, 207, 211, 265, 277, 278, 279, 281, 283, 284, 286, 288, 290, 293, 352  
Windows 2000 Professional desktops 203  
Windows 2000 Professional Edition 4, 202, 259  
Windows 2000 Server 3, 205  
Windows 98 2, 3, 7, 202, 227, 259, 265, 318, 320, 321, 323, 325, 327, 330, 332, 333  
Windows 98 Second Edition 4, 7  
Windows N 305  
Windows NT 1, 2, 3, 7, 265, 296, 299, 301, 302, 306, 308, 310, 313, 315, 316  
Windows NT 4.0 Workstation 7  
Windows NT Client Feature 12  
Windows NT Server 3  
Windows NT Workstation 202, 259  
Windows NT Workstation 4.0 4  
winnt 364  
winntpoedit.exe 205, 218  
WINS 108, 127, 146  
WINS servers 108, 127, 146  
WINSTALL 271, 361  
wininstall.exe 158  
Workplace Shell 249, 255  
WorkSpace On-Demand 1, 168, 171, 173, 249  
workstation 84  
WPS 249, 255  
wrapper function 71  
WRKFILES 23, 41, 105, 115, 123, 134, 142, 153, 159, 160, 165, 200  
WRKFILES alias 172, 199  
WRKFILES, 176

## X

XML 59

XR\_D001 380  
XR\_M014 380

**Y**  
yes.txt 198



## IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at [ibm.com/redbooks](http://ibm.com/redbooks)
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to [redbook@us.ibm.com](mailto:redbook@us.ibm.com)

|                                                                                                                                                                                   |                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Document Number</b>                                                                                                                                                            | SG24-6006-00                                                                                                                                                                                                     |
| <b>Redbook Title</b>                                                                                                                                                              | IBM WorkSpace On-Demand 3.0.1                                                                                                                                                                                    |
| <b>Review</b>                                                                                                                                                                     | <div></div> <div></div> <div></div> <div></div> <div></div> <div></div>                                                                                                                                          |
| <b>What other subjects would you like to see IBM Redbooks address?</b>                                                                                                            | <div></div> <div></div> <div></div>                                                                                                                                                                              |
| <b>Please rate your overall satisfaction:</b>                                                                                                                                     | <input type="radio"/> Very Good <input type="radio"/> Good <input type="radio"/> Average <input type="radio"/> Poor                                                                                              |
| <b>Please identify yourself as belonging to one of the following groups:</b>                                                                                                      | <input type="radio"/> Customer <input type="radio"/> Business Partner <input type="radio"/> Solution Developer<br><input type="radio"/> IBM, Lotus or Tivoli Employee<br><input type="radio"/> None of the above |
| <b>Your email address:</b><br>The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities. | <input type="radio"/> Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.                                |
| <b>Questions about IBM's privacy policy?</b>                                                                                                                                      | The following link explains how we protect your personal information.<br><a href="http://ibm.com/privacy/yourprivacy/">ibm.com/privacy/yourprivacy/</a>                                                          |





# IBM WorkSpace On-Demand 3.0.1









# IBM WorkSpace On-Demand 3.0.1



**Cost effectively  
manage multiple  
clients from a single  
server**

**Custom desktop  
environments allow  
application roaming**

**Enhanced systems  
security**

IBM WorkSpace On-Demand 3.0.1, a new program offering, addresses key requirements created by the rapid adoption of network computing and e-business.

IBM WorkSpace On-Demand 3.0.1, a cross-platform middleware solution, brings the value of server-based client management to a broad range of client hardware, including both Personal Computers (PCs), Network Computers (NCs), and, potentially, other specialized thin-client hardware, such as point-of-sale devices.

IBM WorkSpace On-Demand 3.0.1 is intended to deliver centralized, server-based management for a broad range of client platforms (including Windows 98, Windows NT 4.0, IBM OS/2, and Windows 2000 clients). It also allows for rapid deployment of clients and installation of supported client operating systems without user intervention. It provides support for user roaming, which allows user access to tailored desktops from anywhere in the network. Accelerated deployment of browser- and Java-based e-business clients is also featured.

## **INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

### **BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)

SG24-6006-00

ISBN 0738419419