Designed for Microsoft® Windows NT® Windows® 95

Microsoft® Professional Editions

The Professional's Companion to Microsoft Internet Explorer Version 4

CD-ROM Included

Microsoft® Internet Explorer 4.0

Includes Microsoft Internet Explorer Administration Kit

TM

The enclosed version of the Internet Explorer Administration Kit (IEAK) is for English language versions of Windows 95 or Windows NT 4.0 only. For other language versions, visit the IEAK Web site at http://ieak.microsoft.com

Microsoft Internet Explorer Resource Kit

Complete Technical Information and Tools for Deploying and Supporting Internet Explorer in Your Organization

COMPUTER DISK(S) THAT ACCOMPANY THIS BOOK CAN BE BORROWED FROM THE CIRCULATION DESK.

SAP AMERICA, INC.
EXHIBIT 1019

Petitioner has added romanettes to pages (i)-(iv). Otherwise, it leaves the original page numbering.

UTAH STATE UNIVERSITY
3 9060 00795 7219

Microsoft®

Internet Explorer Resource Kit

UTAH STATE UNIVERSITY
Merrill Library
DATE DUE

NOV 0 4 1999			
FEB 1 8 2000			
MAR 0 1 2001			

Microsoft Press

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 1998 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data
Microsoft Internet Explorer Resource Kit / Microsoft Corporation.
p. cm.
Includes index.
ISBN 1-57231-842-2
1. Microsoft Internet Explorer. 2. Internet (Computer network)
3. World Wide Web (information retrieval system) I. Microsoft Corporation.
TK5105.883.M53M536 1998
005.7'13769--dc21 97-43135
CIP

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QMQM 3 2 1 0 9 8

Distributed to the book trade in Canada by Macmillan of Canada, a division of Canada Publishing Corporation.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office. Or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at mspress.microsoft.com.

Macintosh, QuickTime, and TrueType fonts are registered trademarks of Apple Computer, Inc. Indeo is a trademark of Intel Corporation. ActiveX, DirectSound, DirectX, FrontPage, IntelliSense, Microsoft, Microsoft Press, MS-DOS, PowerPoint, Visual Basic, Visual C++, WebBot, Win32, Windows, the Windows logo, Windows NT, and the Windows Start logo are registered trademarks and Active Channel, Active Desktop, ActiveMovie, Authenticode, DirectAnimation, JScript, the Microsoft Internet Explorer logo, MSN, NetMeeting, NetShow, Outlook, Visual InterDev, Visual J++, and Visual SourceSafe are trademarks of Microsoft Corporation.

Other product and company names mentioned herein may be the trademarks of their respective owners.

Acquisitions Editor: Casey Doyle
Project Editor: Stuart J. Stuple

Contributors

Project Lead:	Craig Beilinson
Managing Editor:	Helen Kaplow
Lead Technical Writers:	Randy Grandle Bruce Bartram William French III
Technical Contributors:	Kim Akers Joe Belfiore Robert Bennett Scott Berkun Eric Berman Greg Blaum Chris Brown Chris Carper Christine Chang Micheal Dunn Castedo Ellerman Brent Ethington Joe Herman Scott Johnson Timothy Johnson Loren Kohnfelder Luanne LaLonde Ken Ong Srivatsan Parthasarathy Hadi Partovi Brent Ponto Toni Saddler-French Kathryn Schmidt Thad Schwebke Maryanne K. Snyder Robert Stewart Kevin Unangst Michael Wallent Tom Yaryan

Contents

Welcome to the Microsoft Internet Explorer 4 Resource Kit xv
What's New in Microsoft Internet Explorer 4 xvi
Easy Installation with Active Setup xviii
Best Browser xx
True Web Integration xxiv
Webcasting xxxii
Complete Communication and Collaboration xxxvii
A Great Publishing Platform xli
Enhanced Security and Privacy xlii
Conventions Used for This Guide xliv

Part 1 Installation of Internet Explorer 4

Chapter 1 Installation Requirements and Issues 3
Understanding System Requirements 4
Understanding Installation Issues 5

Chapter 2 Phase 1: Active Setup 7
Understanding Active Setup 8
Downloading and Extracting IE4Setup.exe 9
Understanding the Active Setup Log File 15
Using the Setup Wizard 16

Chapter 3 Phase 2: ACME Setup 35
Installing Internet Explorer 4 on Your Computer 36
Understanding the IE4 Setup Log File 42
Understanding SoftBoot 43

Chapter 4 Automated Installation (Hands-Free Mode) 45
Using Automated Installation 46
Using Automatic Configuration 47

Chapter 5 User Profiles 49
Understanding User Profiles 50
Creating User Profiles 54
Specifying the Profile Supervisor 58
Understanding Advanced User Profile Functionality 58

Chapter 6 Adding and Removing Components After Setup 61
Understanding the Add/Remove Dialog Boxes 62
Removing Components 63
Adding Components 64
Reinstalling Components 67
Adding and Removing the Windows Desktop Update 69
Updating Internet Explorer 4 70

Chapter 7 Uninstalling Internet Explorer 73
Removing Internet Explorer Core Files 74
Understanding the Uninstall Process 76
Understanding the Uninstall Log 82
Understanding Emergency Uninstall: IERemove.exe 82

Chapter 8 Configuring Connection Settings 85
Using the Internet Connection Wizard 86
Using Manual Configuration 93

Part 2 The Internet Explorer Browser

Chapter 9 Browser Features and Functionality 99
Ease of Use and Personalization Features 100
Using Explorer Bars 108
Browsing Offline 119
Setting Advanced Browser Options 121
Integrating Third-Party Programs 126

Chapter 10 Browser Performance 129
Optimizing Browser Performance 130
Using HTTP 1.1 130
Using the Java Just In Time Compiler 131
Using Cached Content 132
Configuring Multimedia Settings 134

Chapter 11 The WebBrowser Control 135
Understanding Browser Architecture 136
Reusing the WebBrowser Control 139
Distributing Browser Controls with Applications 142

Part 3 True Web Integration: Windows Desktop Update

True Web Integration Architecture 145

Chapter 12 Start Menu and Taskbar 147
Start Menu and Taskbar Overview 148
Configuring the Start, Favorites, and Programs Menus 155
Configuring the Taskbar and Toolbars 157

Chapter 13 Web View 161
Understanding Web View 162
Web View Architecture 164
Customizing Web View 165
Selecting Thumbnail View in a Folder 171

Chapter 14 Active Desktop 173
Understanding the Active Desktop 174
Customizing the Active Desktop 177
Developing Active Desktop Items 182

Part 4 Information Delivery: Webcasting

Chapter 15 Basic Webcasting 189
Understanding Basic Webcasting 190
Subscribing to Sites 193
Customizing Subscriptions 196
Locking Down Subscriptions 203
Beyond Basic Webcasting: Addressing Site-Crawling Issues 206

Chapter 16 Managed Webcasting 209
Active Channels Overview 210
Understanding Active Channels 212
Understanding CDF Files 215
Subscribing to Active Channels and Desktop Items 229
Customizing Active Channels and Desktop Items 230
Deploying Custom Packages with Predefined Channels for User Groups 234

Chapter 17 True Webcasting 241

Part 5 Communication and Collaboration

Chapter 18 Outlook Express 247
Outlook Express Overview 248
Understanding Outlook Express 258
Configuring and Customizing Outlook Express 261
Understanding Security in Outlook Express 274
Using Security Features in Outlook Express 275

Chapter 19 NetShow 279
NetShow Overview 281
Components of NetShow 288
Webcasting with NetShow 291
Understanding NetShow Technology 293
Using HTTP Streaming 311
Developing Content for NetShow 313
Using the NetShow Player 323

Chapter 20 NetMeeting 327
NetMeeting Overview 329
What's New in NetMeeting 2.1 336
NetMeeting Application Scenarios 337
NetMeeting Architecture Components 339
Understanding Conferencing Standards 343
Understanding Product Interoperability 353
Firewall Configuration 359
System Policies and Restrictions 363
Using NetMeeting on Intranet Web Pages 368

Chapter 21 FrontPage Express 375
FrontPage Express Overview 376
Using FrontPage Express 379

Chapter 22 Personal Web Server 383
Personal Web Server Overview 384
Understanding Personal Web Server 387
Using Personal Web Server 388
Personal Web Server Scenarios 391
Understanding the Web Publishing Wizard 393

Part 6 Authoring for Internet Explorer 4

Chapter 23 Dynamic HTML 397
Dynamic HTML Overview 398
Dynamic HTML and Internet Standards 399
Understanding the Document Object Model 400
Using Dynamic HTML 401

Chapter 24 ActiveX 413
ActiveX Overview 414
Using ActiveX Controls 414
Using Multimedia Controls 419

Chapter 25 Java 421
Java Overview 422
Java Improvements for Internet Explorer 4 423
How Java Works 424
Microsoft Virtual Machine for Java 425
Java Security 428
Using Java Applets in Web Pages 428

Chapter 26 Active Scripting 429
Active Scripting Overview 430
ECMAScript: The Internet Standard 431
JScript 3.0 in Internet Explorer 4 432

Part 7 Security and Privacy

Overview: Understanding Trust Management 440

Chapter 27 Security Zones 447
Security Zones Overview 448
Setting Up Security Zones 451
Customizing Security Zones 456
Deploying Security Zones in User Groups 463

Chapter 28 Digital Certificates 467
Digital Certificates Overview 468
Understanding Commercial Certificate Authorities 469
Understanding Certificate Servers 471
Understanding Authenticode 471
Using Secure Client/Server Communications 474
Configuring Certificates 478
Deploying Customized Certificates in User Groups 483

Chapter 29 Trust-Based Security for Java 489
Trust-Based Security for Java Overview 490
Understanding Trust-Based Security 490
Configuring Custom Java Security 496
Customizing Java Permissions for User Groups 502

Chapter 30 Content Advisor 519
Content Advisor Overview 520
Understanding Content Ratings 520
Configuring Content Advisor 522
Deploying Content Advisor Settings to User Groups 526

Chapter 31 User Privacy 531
Understanding User Privacy 532
Configuring Privacy Options 536

Part 8 Internet Explorer 4 Accessibility

Chapter 32 Accessibility Overview 545
Accessibility Benefits 546
Upgrade Considerations 548
Suggested Features for Different Types of Disabilities 549
Accessibility Resources 554

Chapter 33 Accessibility Features and Functionality 557
Using Keyboard Navigation 558
Customizing Display of Fonts, Colors, and Styles 566
Configuring Advanced Internet Accessibility Options 575
Using Accessibility Features in NetMeeting 579

Part 9 Deployment and Maintenance

Overview: The Deployment Process 583

Chapter 34 Planning the Deployment 585
Deployment Planning Overview 586
Assembling Project Teams 586
Identifying Technical Requirements for Deployment 587
Identifying Client/Server Configurations and End-User Needs 590
Identifying Migration Issues 590
Specifying Distribution Strategies 594
Specifying Custom Package Configurations 596
Developing the Deployment Project Plan 601

Chapter 35 Validating the Deployment 603
Deployment Validation Overview 604
Preparing the Test Plan and Checklists 604
Testing the Deployment Process 605
Planning the Pilot Program 606
Conducting the Pilot Program 607
Finalizing the Deployment Plan 609

Chapter 36 Implementing the Deployment 611
Deployment Implementation Overview 612
Announcing the Deployment 612
Providing User Training and Support 612
Building Custom Packages 614
Signing Custom Package Files 666
Producing Distribution Media 667
Distributing Internet Explorer 668
Assisting Users During Installation 670

Chapter 37 Automatic Browser Configuration 671
Automatic Browser Configuration Overview 672
Understanding Automatic Browser Configuration 672
Setting Up Automatic Browser Configuration 673
Using the IEAK Profile Manager 682

Chapter 38 Software Distribution and Updates 689
Software Distribution Overview 690
Understanding Software Distribution Channels 690
Implementing Software Distribution Channels 691
Managing Software Distribution Channels 705

Appendixes

Appendix A Switches, Keys, and Files in Internet Explorer 4 Setup 707
WExtract Command-Line Switches for IE4Setup.exe 707
Other Setup Registry Keys 709
Command-Line Switches for Active Setup AutoUpdate 710
Command-Line Switches for ACME Setup 711
The Sections of the SoftBoot.inf 714
Files Left in the System After Uninstalling Internet Explorer 715

Appendix B Troubleshooting Installation and Uninstallation 719
Troubleshooting Installation 720
Troubleshooting Uninstallation 726

Appendix C Troubleshooting the Browser 733
Font Size Change is Not Saved 734
Determining Your Previous Version of Internet Explorer 735
Slow Connections and Inability to View Non-Microsoft Sites 736
Restoring Functionality to the Downloaded Program Files Folder 737
Changing the Download Location for ActiveX Files 738

Appendix D Troubleshooting the Active Desktop 739
Pointing to Desktop Icons Causes Text to Disappear 740
Changing to Single-Click Mode Using the Registry 740
Images Copied to the Desktop Are Saved as Active Desktop Items 741
"Find on the Internet" Command Appears Twice on the Start Menu 742
Active Desktop Items Stop Working When Offline 742
System Properties Do Not Show That Microsoft Plus! Is Installed 743
Internet Explorer Cannot Find the Active Desktop 744
Do Not Use Web Pages with Frames as Your Desktop Wallpaper 744

Appendix E Troubleshooting Webcasting 745
Unattended Dial-Up Does Not Work Without User Intervention 745
E-Mail Notification Is Not Sent for Updated Web Sites 746
Red Gleams Do Not Appear on Channel Bar When Site Content Updates 747
Subscription Images Are Downloaded Even If Images Are Disabled 748

Appendix F Troubleshooting Outlook Express 749
Problems Changing the Default Mail Client for Internet Explorer 749
Cannot Import Address Book in Outlook Express 751
Preview Pane Does Not Display News Messages When Enabled 752
Error Message: The Command Failed to Execute 753
Internet Mail and News Files Renamed by Outlook Express 753

Appendix G Troubleshooting HTML Authoring 755
Text Doesn't Wrap in Text Boxes 755
Background Images Are Not Displayed 756
Internet Explorer Browser Is Not Automatically Redirected 756

Appendix H Frequently Asked Questions About Cabinet Files in Java 759

Appendix I Internet Client SDK FAQs 763
Setup 763
Active Channels 763
HTML 766
Dynamic HTML 767
WebBrowser Control 768
Component Packaging 769
Control Development 769
Scripting 770
Miscellaneous 772

Appendix J Custom-Build Checklist 773

Appendix K Internet Explorer Administration Kit (IEAK) FAQs 793

Appendix L IExpress Configuration Wizard 803
Using the IExpress Configuration Wizard 803
Signing Package Files 814

Index 815

Welcome to the Microsoft Internet Explorer 4 Resource Kit

Microsoft® Internet Explorer 4 integrates the power of the computer and the exciting content on the Internet to give users a compelling Web experience. Internet Explorer 4 integrates Web tasks into a proven and popular user interface that makes use of current investments in training and solves key customer issues. Internet Explorer 4 is the next step in Microsoft's vision of "information at your fingertips"; finding and using information should be easy, regardless of how, or where, it is stored.

For corporations, Internet Explorer 4 is designed to make users more productive and to ease migration to intranets, while allowing administrators complete control of Web browser management and design. For users, it provides a much richer Web experience, delivers full-featured communication capabilities, and makes finding information on the Internet or an intranet easier than ever. For developers, Internet Explorer 4 offers support for full interactivity and cutting-edge content, while reducing bandwidth constraints.

The Microsoft Internet Explorer 4 Resource Kit is written for system administrators and information systems (IS) professionals. It provides the information required for understanding, customizing, deploying, and supporting Internet Explorer 4 in their organizations. This resource kit is a technical and planning resource that supplements the information included with the Internet Explorer 4 product and the Microsoft Internet Explorer Administration Kit (IEAK). For information about how to use specific Internet Explorer 4 features, see the Help system supplied with the software. For more information about Internet Explorer 4, see the Internet Explorer Web site at **http://www.microsoft.com/ie/**.

What's New in Microsoft Internet Explorer 4

This section gives a brief description of some of the most interesting and exciting new features and functionality in Microsoft Internet Explorer 4. Internet Explorer 4 delivers innovation in nine key areas that correspond to nine parts in this book:

1. **Easy installation.** Internet Explorer 4 Active Setup offers a powerful solution to the challenges associated with current Internet-based installation programs. The new, modular approach used by Internet Explorer 4 conserves bandwidth and offers users a faster, more versatile installation experience.
2. **Best browser.** Having a great experience on the Web starts with the browser. Internet Explorer 4 contains a variety of improvements that make browsing the Web easier, faster, and more productive so that users can find the information they need.
3. **True Web integration.** Internet Explorer 4 integrates the Internet into every aspect of personal computing—the desktop, file folders, the local network, and even the **Start** menu. The Windows® Desktop Update provides users with a single metaphor for exploring and searching for information. Information from the Internet or an intranet becomes a seamless part of the desktop, and browser-like navigation becomes available in every view. Whether looking at a local hard disk, a local area network (LAN) network drive, or a Web site on an HTTP server, the same navigation toolbar provides functionality for going backward and forward, marking favorite locations, and searching for desired information. Because the same menus and navigation toolbar accomplish similar tasks in different scenarios, users need to learn only one navigational model; thus, the cost of training and supporting users is reduced.
4. **Webcasting.** The built-in Webcasting features in Internet Explorer 4 provide the broadest range of intelligent "push" and "pull" solutions for optimized information delivery. Users can subscribe to any Web site, schedule Internet Explorer 4 to automatically check for updated content on that site, be notified when content changes, and even set Internet Explorer 4 to automatically download updated content for offline viewing. In addition, users can target Web page downloads to the Active Desktop so that they can view their content as desktop wallpaper, as a screen saver, in a desktop ticker, or in a desktop Web window. Using the new Channel Definition Format (CDF), it's easy for content providers or corporate administrators to create custom Active Channels that deliver personalized information to users' desktops in an efficient, effective format.

5. **Complete communication and collaboration.** Internet Explorer 4 offers a complete, scalable suite of integrated tools for communication and collaboration. These include electronic mail, news, conferencing, broadcasting, authoring, and publishing tools. This suite of tools is tightly integrated to ensure consistency across all components in the suite and to provide an easy way to switch between them or other applications you may want to use. Built-in extensibility also makes it easy to integrate third-party clients.
6. **A great publishing platform.** Internet Explorer 4 supports a wide range of new features that publishers can use to create interactive, exciting Web pages based on the newest Internet technologies and standards. Using this set of technologies, Web authors can produce more attractive content while taking up less bandwidth. They can develop business applications that deliver more value and create a more rewarding Web experience. A new object model exposes every element of an HTML page to the content developer. This enables the creation of pages where each page attribute—whether a single word, text style, or a graphic object—can respond to user interaction at run time so that Web pages don't require an entire refresh. Thus, Web page effects finally rival the great performance of multimedia CD-ROM titles. And, Microsoft has again designed the fastest, most reliable, and fully compatible Java support on the market; and delivers a new Java security model that allows users to run powerful, interactive Java applications without having to worry about harm to their computers.
7. **Enhanced security and privacy.** Internet security is a primary concern of every user and corporation on the Internet today, and Internet Explorer 4 is the safest, most secure browser on the Internet today. It supports more of today's emerging and existing security standards, such as Authenticode™, Java permission-based security, Secure Sockets Layer (SSL), and Private Communication Technology (PCT). In addition, Internet Explorer 4 introduces new security enhancements, including Authenticode 2.0 technology and the innovative use of security zones. Security zones provide an easy interface for dividing the Internet or an intranet into safe and unsafe regions, each with its own security settings. Administrators can pre-configure security zone settings, providing better safeguards for corporate users and intranets.
8. **Better accessibility.** Internet Explorer 4 introduces a number of advances that make the Web more accessible to computer users with disabilities. In particular, several features allow users to customize the appearance of Web pages to meet their own needs and preferences. Internet Explorer 4 also includes the best features of Internet Explorer 3, including the ability to navigate the Web using the keyboard. Users without disabilities may also be interested in many of these options as ways to customize colors and fonts to their own tastes or to save time by using keyboard shortcuts.

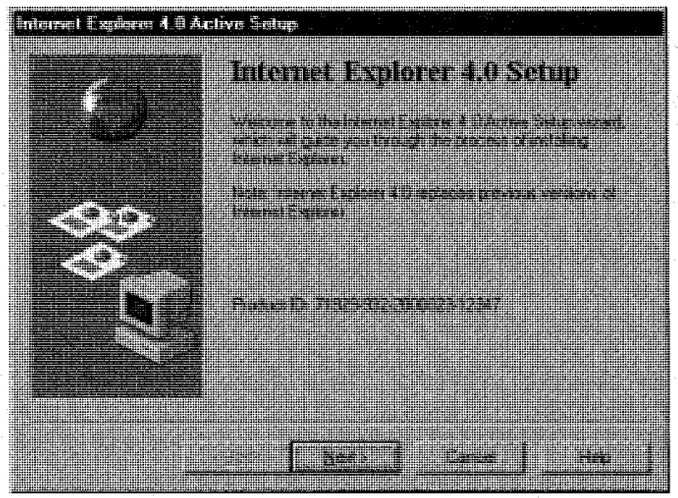
9. **Expanded deployment and accessibility.** A corporation can begin to realize intranet cost reduction by deploying Internet Explorer 4 using the setup functionality provided in the IEAK. The IEAK allows a system administrator to build a custom package that includes all the specific configuration settings mandated within a corporation. Such an Internet Explorer 4 installation package can also include additional components that need to be installed along with Internet Explorer 4, making it easier for corporations to deploy multiple applications to all desktops at the same time as deploying Internet Explorer 4.

   The specific advantage of the IEAK is that it builds Internet Explorer installation packages for all intranet platforms and completely eliminates all end-user setup choices and options, ensuring that all users within an organization experience hands-free setup with the exact same application configuration for Internet Explorer 4 and additional components. Part 9, "Deployment and Maintenance," provides extensive deployment planning, implementation, and maintenance information for Internet Explorer 4.

Easy Installation with Active Setup

Internet Explorer 4 Active Setup is an innovative solution to the challenges of current Internet-based installation programs. Before Internet Explorer 4, all browsers were installed by downloading a single, large, monolithic package. If the download failed, it had to be restarted from the beginning of that large file. If a connection to a server was dropped during download, Setup could not resume and none of the data downloaded could be used.

An executable file cannot run until it has been completely downloaded, so large Setup packages are unable to check system requirements in advance. Users may wait a long time only to find out that they have to free up disk space and try again. A single, monolithic file does not give you control over which components will be installed. If you don't want all the components included in a monolithic package, time is wasted downloading unwanted data, and disk space is wasted storing that data.



Internet Explorer 4 uses a modular approach to solve the problems with previous browser Setups. Internet Explorer Setup is based on an ActiveX® engine called Active Setup, which runs on the client. The process begins with a small Setup package. This self-extracting file can be downloaded to the computer using an existing browser, or it can be copied directly to a computer with no existing browser. Because the file is only 400 kilobytes in size, it downloads very quickly and can fit on a single floppy disk. This file is all that is needed to initiate the installation of Internet Explorer 4. It contains the bare minimum code necessary to give the client Internet access and provide a basic level of file download capabilities.

This small Setup package also allows Setup to collect information about the host computer before the download begins. Active Setup uses this information to intelligently manage the download of Internet Explorer cabinet (.cab) files and make installation as efficient and problem free as possible.

After all the needed user and system information has been gathered and you have selected from the available Microsoft distribution servers on the Internet, the necessary .cab files are downloaded. Because Setup has already inventoried any Internet Explorer components already on the system and has already asked you which components are desired, it knows to download only the .cab files needed. After all the .cab files have been successfully downloaded, Active Setup is complete.

Best Browser

A powerful browser can help your organization reap the important benefits the Web has to offer. As the Internet increases in utility and complexity, more and more computer users have come to rely on Internet Explorer to help them make the most of Internet advancements. Ease-of-use innovations and enhanced personalization features in Internet Explorer 4 truly give users the Web on their terms, or "the Web the way you want it." Internet Explorer 4 also offers key underlying technologies for Web developers.

Internet Explorer 4 contains a variety of improvements that make browsing the Web easier, faster, and more fun than ever before. Part 2, "Configuring the Browser," provides detailed information about the features and functionality of the Internet Explorer 4 browser; a few key features are highlighted here.

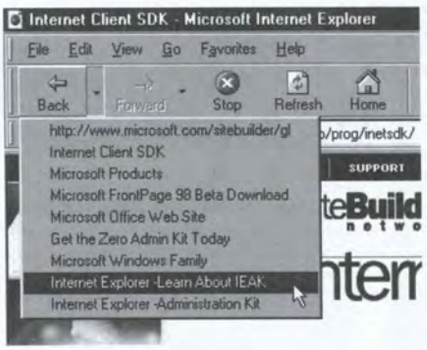
AutoComplete

The Address bar in Internet Explorer 4 automatically completes addresses for you, based on sites you've already visited. By filling in long, complicated URLs automatically, AutoComplete makes it easier for you to type in URLs and reduces the opportunity for typographical mistakes. AutoComplete also automatically adds prefixes and suffixes to Internet addresses.

As an extension of the AutoComplete feature, you can right-click the Address bar to have Internet Explorer 4 provide a list of available completions for a partial URL.

Navigation History on Back and Forward Buttons

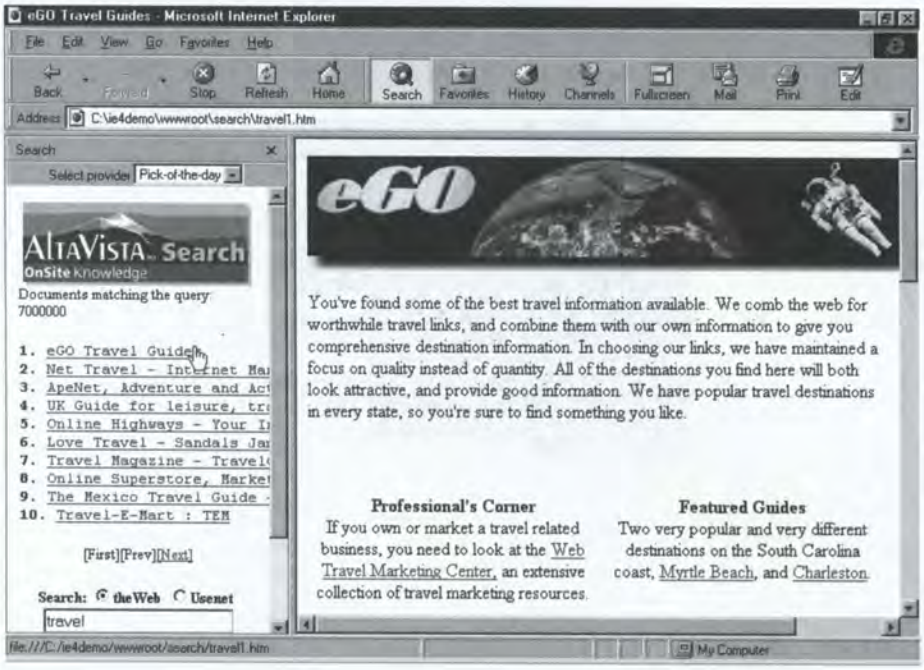
Instead of pressing the **Back** button over and over again to go to a site you visited earlier, you now click the drop-down arrow next to the **Back** or **Forward** button to see a list of your current navigation history. From that list, you can quickly go to the pages you need.



Explorer Bars

The Explorer bar is an innovation exclusive to Internet Explorer 4 that makes it dramatically easier to navigate the Web. Explorer bars for Search, Favorites, History, and Active Channel™ view make it easier to find the sites you need, visit your favorite pages, return to the places you've been, and access the information you use most often. When you select an Explorer bar by clicking its button on the Internet Explorer toolbar, a special pane slides in on the left side of Internet Explorer content window. When you click a link on an Explorer bar, the corresponding page opens in the main content window, so your links are still visible while the content is displayed. To remove the Explorer bar from your desktop, click its **Close** button.

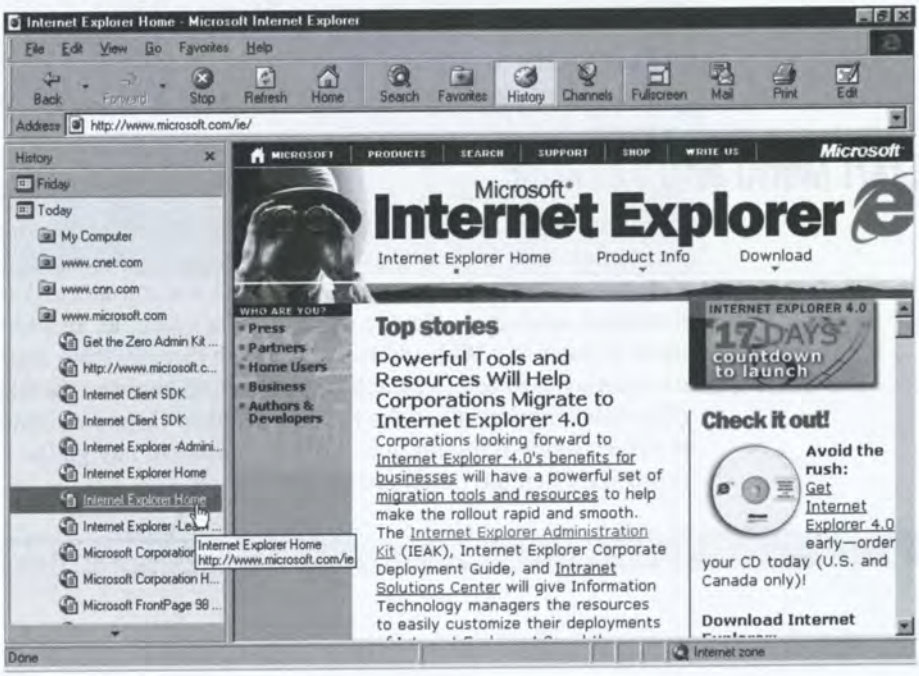
For example, the Search bar displays search results independent of the main browser area. When you rest the pointer over a result, Internet Explorer also displays a summary of the site in a ToolTip. When you select a site from the results list, the site appears in the main browser area, keeping the Search bar available for future searches. The results list remains in the Search bar, so if you need to visit a different site that the search engine revealed, you can easily move from result to result without repeatedly using the **Back** button to return to the search results page. Internet Explorer 4 also preserves the state of the search, so if you click the **Search** button again during the same session, the results of the previous search appear. This way, you easily can move the Search bar out of the way and then bring it back when you need it.



Offline Browsing

When users want to go to a favorite site on the company intranet or on the Web, it's not always convenient to connect to the network or to their Internet service provider (ISP). With offline browsing, users can view sites without these connections. For example, users can browse offline while using their laptop computers at locations where Internet access is not available. Or users can find it helpful to browse offline at home when they don't want to tie up their only phone line. Some users may even prefer to browse offline at the office when server traffic is heavy and load time is slow. (Offline browsing is faster than browsing online.)

Internet Explorer 4 makes offline browsing possible by automatically caching every site you visit when online. All Web pages you access, as well as files related to those pages (such as graphic files), are stored on your hard disk so that they can be viewed later without an active Internet connection. To begin browsing offline, select **Work Offline** from the **File** menu in Internet Explorer and select any page from the History bar. The History bar automatically sorts your visited Web pages by date and then by site, so it's easy to see where you've been in the past, whether you're online or offline.



In addition, you can use the Webcasting technology in Internet Explorer 4 to schedule your browser to automatically check a Web site for changes since your last visit and download any new or updated content so that you're always browsing the most up-to-date information, even when offline. (For more information about Webcasting, see Part 4, "Information Delivery: Webcasting.")

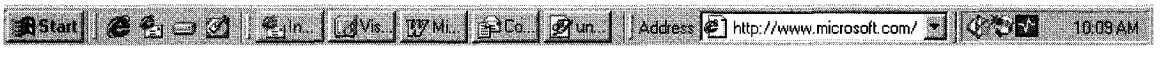
True Web Integration

Until now, business computer users got their information from two distinct worlds: the online world of the Internet and the "local" world of company networks and systems. But, as the Internet advances to permeate users' everyday computing experience, the integration of Web features and functionality with the desktop operating system is an idea that has become increasingly important and valuable.

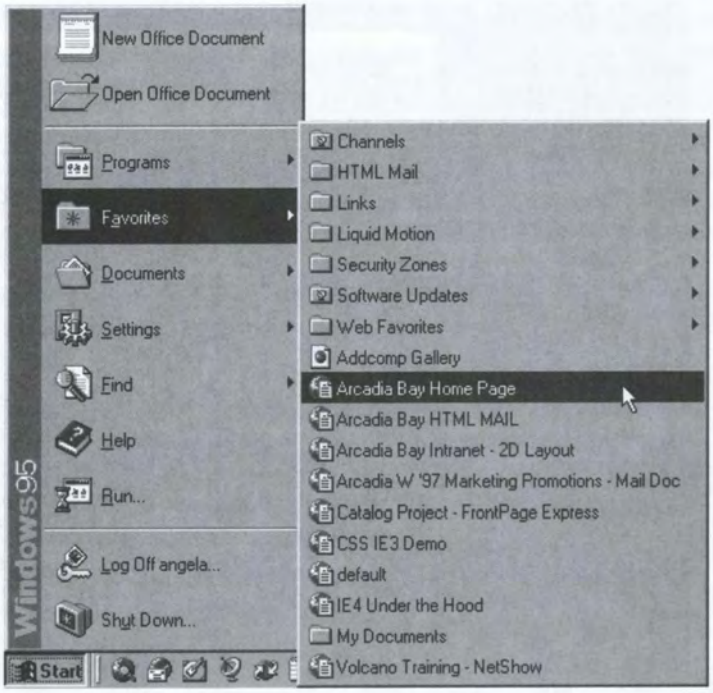
For this reason, Internet Explorer 4 provides true Web integration with the Windows Desktop Update, a set of features that integrate the Internet into every aspect of personal computing—the desktop, file folders, the network, and even the **Start** menu. Part 3, "True Web Integration: Windows Desktop Update," describes how true Web integration works in Internet Explorer 4, details the features included in the Windows Desktop Update, and explains how you can use these features to create a richer, more streamlined, and more productive working environment in your organization. A few key features are outlined here.

Start Menu and Taskbar

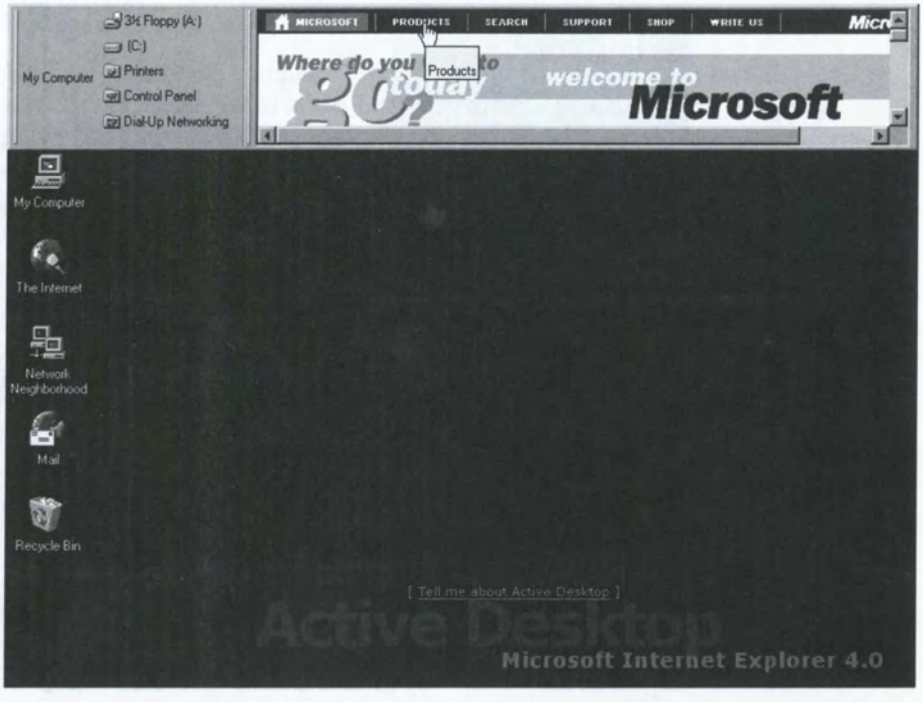
The **Start** menu and taskbar provide an anchor on the Windows desktop, giving users one center for starting and switching tasks. In Internet Explorer 4, these elements are extended to integrate Internet and intranet tasks into a user interface that many users have already mastered. For example, an address bar allows users to enter any Web address or file path directly from their Windows taskbar, eliminating the need to switch back and forth between the file folder window (Windows Explorer) and the browser window (Internet Explorer). This concept is called Single Explorer, and it offers consistent navigation at the same time as it provides flexible functionality.



When using the Windows 95 and Windows NT® operating systems, users know they can use the **Start** button to access almost everything. Inside the Web browser, they know they can rely on their **Favorites** menu to get to their commonly used sites. In Internet Explorer 4, these ideas are combined and the **Favorites** menu attached to the **Start** button, so users can now get to their favorite sites quickly—without having to launch their Web browser first.



Following the same logic, the Internet Explorer 4 taskbar provides access to a wide array of Internet-focused functions, permitting users to get to the information they need quickly and conveniently. With Internet Explorer 4, the taskbar has become extensible to support expandable, dockable, and layered browser toolbars. In addition, custom toolbars can contain any sort of content, including application shortcuts or even Web pages.

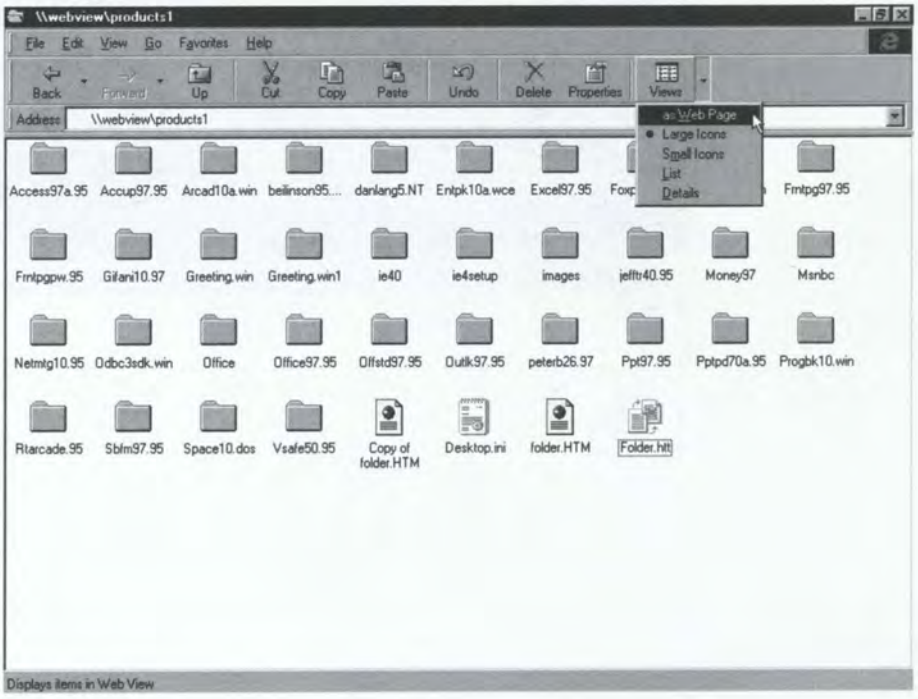


Web View

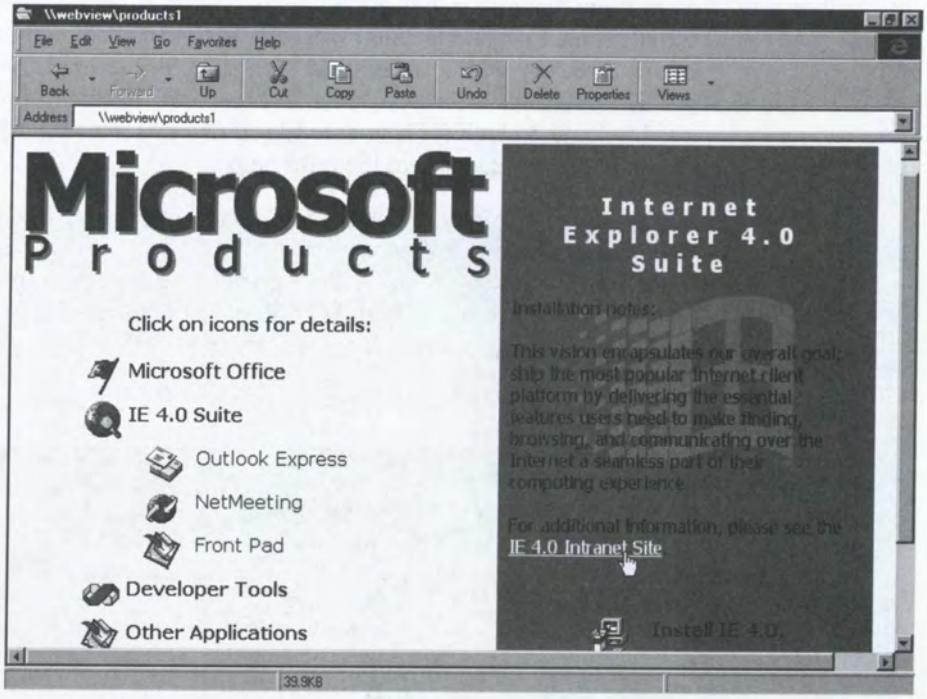
Internet Explorer 4 opens up the user interface so that you can customize any folder (on a floppy disk, on a local hard disk, or on a server) with HTML. The Web View feature extends the original views in Windows 95 (Large Icons, Small Icons, List, and Details) with a fifth view that can represent the contents of any directory with a customized Web page. For example, when viewing a public folder on a network, users today just see a list of files and folders. They need to know to look for a readme file (if one exists) to see what is present in those directories and where to go for help.

With Internet Explorer 4, administrators can set up these folders with Web views so that users can see a Web page with HTML, Java, or ActiveX controls that richly describe the contents of that folder, or whatever information the administrator decides to provide. With Web View, administrators can extend the user interface of Windows with the simplicity of HTML.

For example, a typical directory on a public server may appear this way to a user:

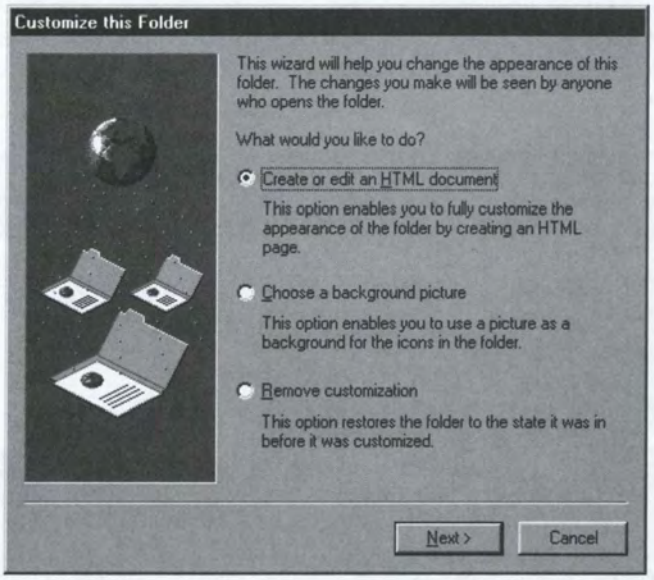


However, administrators can dramatically enhance the appearance of that directory by implementing an HTML Web view. The window in the following illustration is displaying the exact same directory, but Web View has been activated.



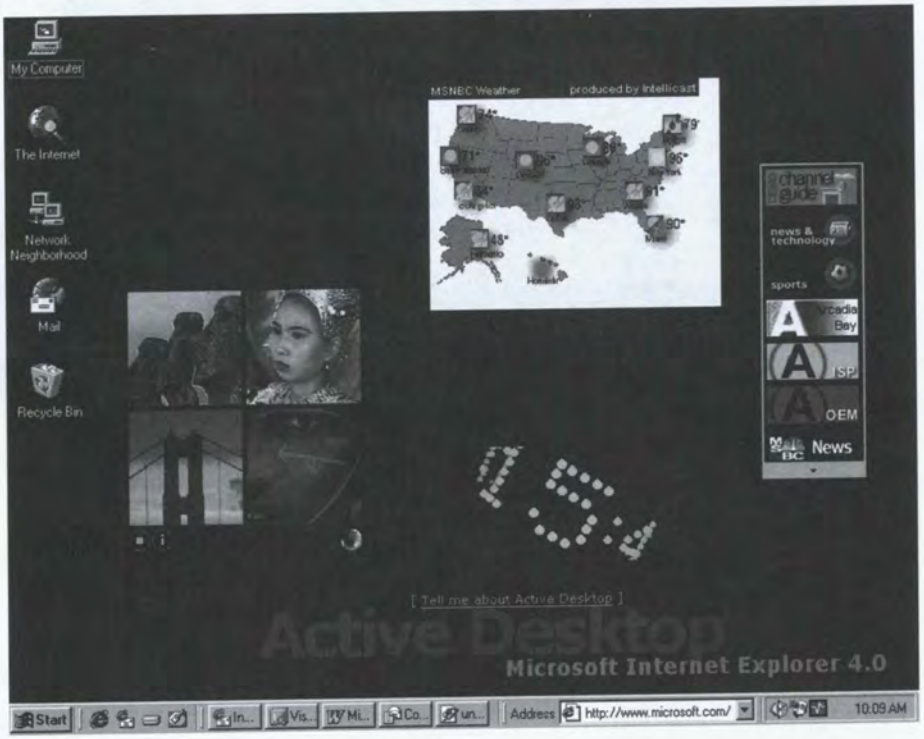
Web View Wizard

The Web View Wizard makes it easy for users and network administrators to create a custom Web view (based on HTML) for any folder, whether it is local or on a network share, without requiring a Web server. Any Internet Explorer 4 user who views that folder will see the customized Web view instead of the usual, nondescript list of files and folders. By default, the wizard will use FrontPage® Express to help create the Web view, but Internet Explorer 4 can be customized to use any HTML editor.

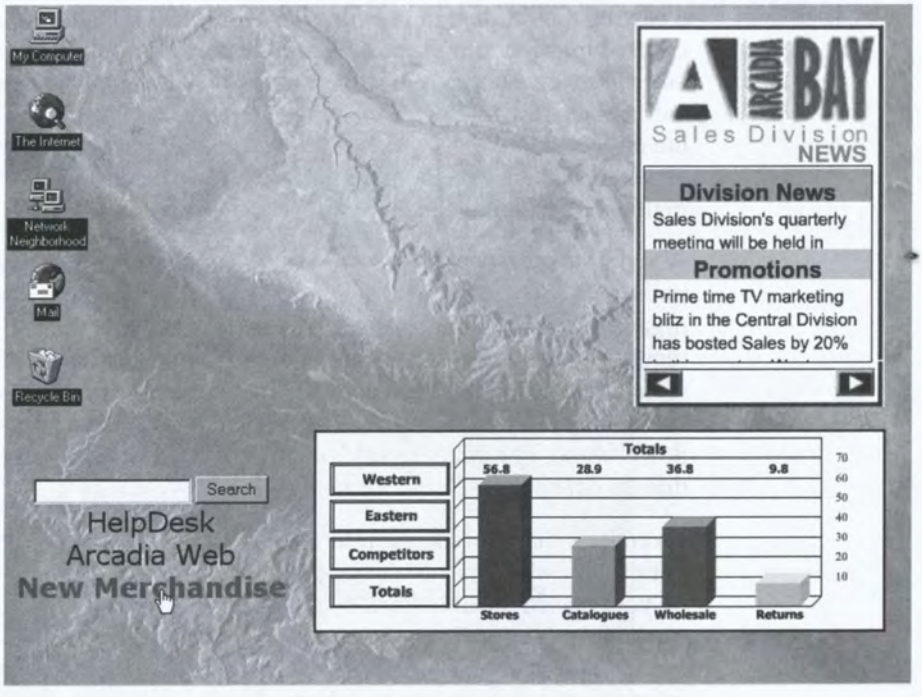


Active Desktop

The Active Desktop interface brings the Web right to users' desktops with HTML wallpaper, screen savers, and Active Desktop items. Now you can use a Web page as desktop wallpaper or as a screen saver, complete with graphics and links to your favorite files and URLs. ActiveX objects, JavaScript, and Visual Basic® Scripting Edition (VBScript) can be used to add even more functionality. Active Desktop items, which are essentially miniature Web pages that reside on the desktop, complement this functionality. They can be used to provide information such as news headlines, weather, traffic information, and stock quotes. For users, adding an Active Desktop item can be as simple as pointing and clicking. For system administrators, creating an Active Desktop item is as easy as creating a Web page.



The flexibility of the Active Desktop interface allows intranet administrators to use "push" and "smart pull" technologies to make information and resources directly accessible from users' desktops, and to customize that information for specific user groups. The Active Desktop turns users' desktops into dynamic, personalized, live Web "bulletin boards" that regularly update with the latest content, for quick access online or offline. Such custom configurations can be easily managed using the IEAK.



Webcasting

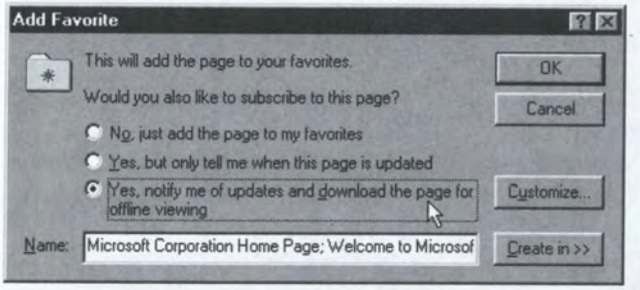
The information delivery architecture in Internet Explorer 4 enables new ways for Web content or applications to reach end users, using the new "push" and "pull" technologies, which are collectively referred to as *Webcasting*. Webcasting is the automated delivery of personalized, up-to-date information to users' desktops. Internet Explorer 4 provides a mechanism for end users to schedule recurring monitoring, or crawling, of Web sites and automatic downloading of updated information. These advanced features for pulling content allow online users to receive notification when content changes, and they allow mobile workers to view favorite Web content even when they work offline. In addition, the information delivery architecture allows intranet administrators to set up servers that push content to corporate employees via bandwidth-efficient multicast protocols.

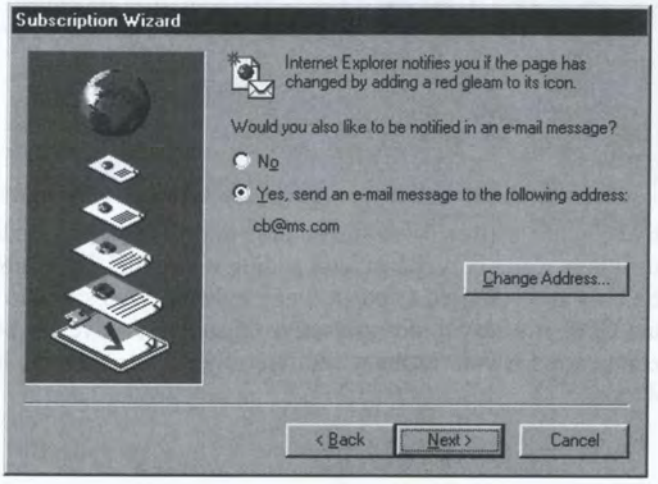
Why introduce a new model of information delivery on the Web? Because traditional Web browsing poses several problems for end users and the organizations that support them. Currently, most Internet and intranet sites require users to manually browse for useful information and then frequently revisit the sites for updated content. This process is time-consuming for users and bandwidth-consuming for the network, as the same pages are repeatedly downloaded. For mobile workers, the problem is exacerbated by painfully slow dial-up connections.

To address these problems, Internet Explorer 4 provides three Webcasting options that allow you to take the best advantage of this new technology, based on your current business needs and technology investments. By automatically tracking updates to Web content, Webcasting can help minimize the time users spend searching the Web for up-to-date information, while ensuring that the latest information is automatically delivered to the people who need it. In addition, Webcasting can reduce the bandwidth necessary to deliver updated content on the Web by reducing the amount of redundant information that is downloaded and by operating at off-peak network load periods, rather than the peak load periods often used with manual browsing. Part 4, "Information Delivery: Webcasting," describes the three types of Webcasting and explains how to incorporate these information delivery technologies into your organization. A brief overview is outlined here.

Basic Webcasting

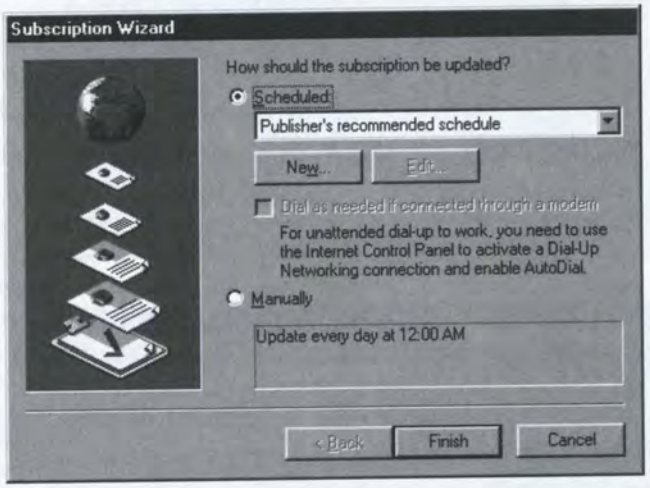
Using basic Webcasting, users can subscribe to any existing Web site and have Internet Explorer 4 automatically visit and crawl the site on a scheduled basis to check for updated content. Internet Explorer 4 can check the site up to three levels deep; when content changes, Internet Explorer 4 notifies users—either by a gleam on their Favorites lists or by e-mail. Internet Explorer can also automatically download updated content for offline use. Site subscription can be thought of as "smart pull" technology.





Managed Webcasting

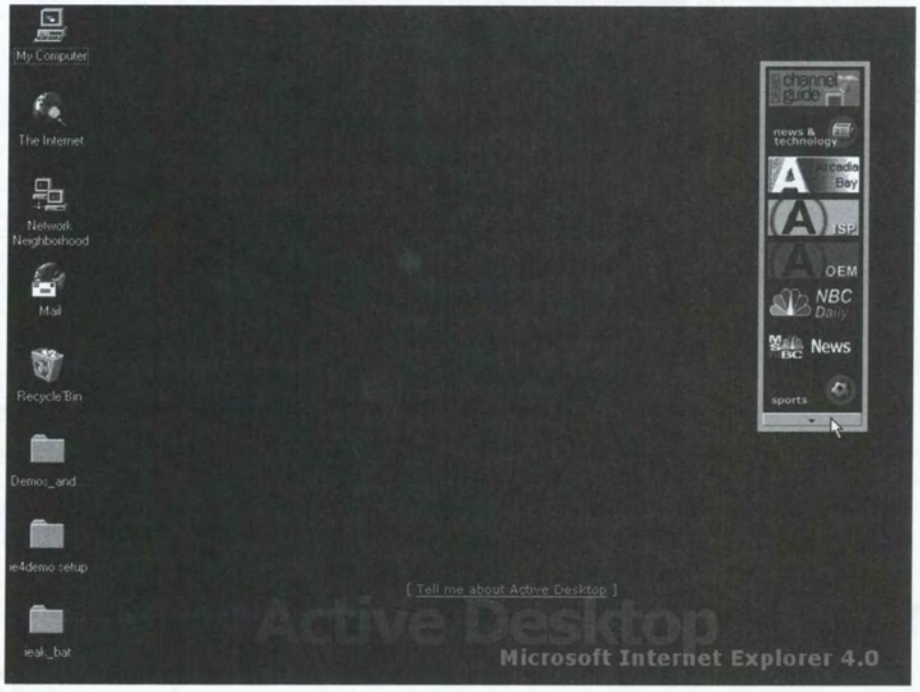
Similar to basic Webcasting, managed Webcasting also works through a subscription process, except that users subscribe to publisher-specified Active Channels; an Active Channel™ delivers a defined range of Web content. This can be thought of as "programmed pull." Users can check for updates on specific types of content and can accept a publisher's predefined update schedule or specify their own custom schedule. When channel content changes, Internet Explorer notifies the user of changes and can be set to download only the requested content for viewing offline.



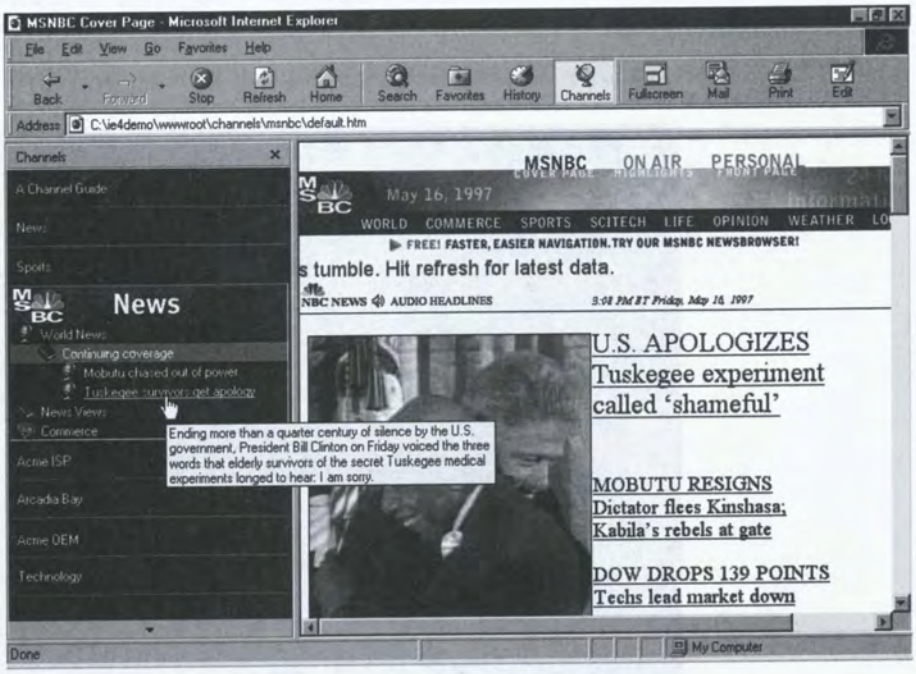
Active Channels allow Web site authors to optimize, personalize, and more fully control how a site is Webcast, and adding a Channel Definition Format (CDF) file is the only step required to convert any existing Web site into an Active Channel. Using CDF, it's easy to design Active Channels, Active Desktop items, or channel screen savers for your intranet to efficiently deliver business information to users' desktops.

After a user subscribes to an Active Channel, Internet Explorer will automatically add the subscribed channel logo to the Channel bar on the Active Desktop and in the browser, making the Web site more prominent and more easily accessible.

Here is the Active Channel bar directly on the desktop.



Here is the Active Channel bar inside Internet Explorer 4.



True Webcasting

True Webcasting, or "true push" technology, uses multicast add-ons such as Microsoft NetShow™ 2.0 to deliver bandwidth-efficient multimedia information to users' desktops. Organizations can use true Webcasting to automatically push updated information to subscribers at scheduled periods. (For more information, see Chapter 19, "NetShow.")

Complete Communication and Collaboration

Collaboration across the Internet is one of the most exciting, avidly discussed topics today. Microsoft is committed to a complete communication and collaboration suite of tools, and the Internet Explorer 4 suite provides a solution for whatever Internet-based communication needs users have. Using its modular installation program or the IEAK, users and administrators can set up only the pieces they need or take advantage of the flexibility of the Internet Explorer suite and integrate it with their existing software solutions.

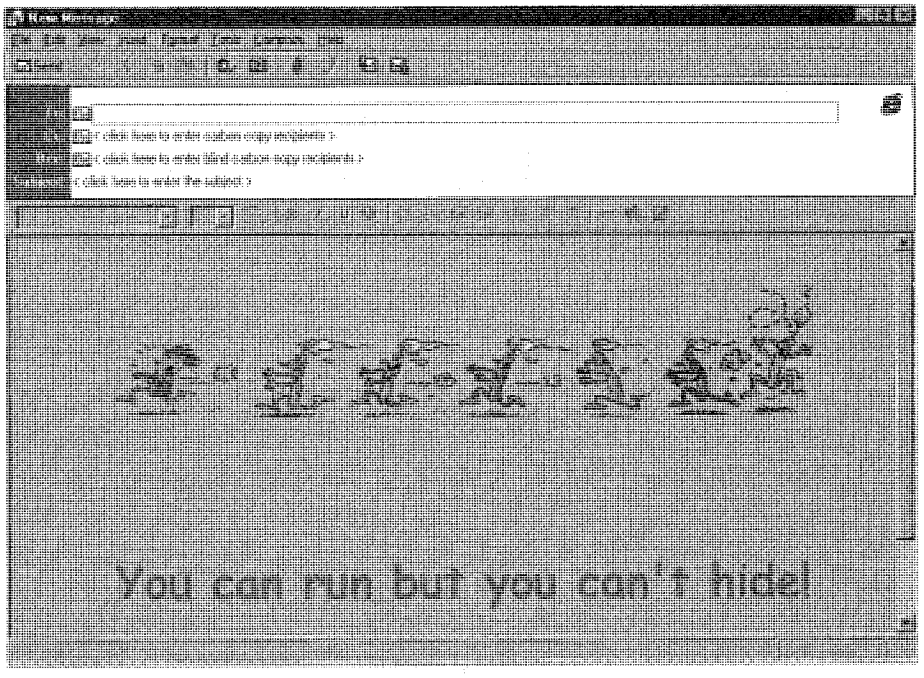
The Internet Explorer suite allows seamless integration from one application to the next because the applications are all tightly integrated and have been developed with a common menu and toolbar. This simplifies training, because a user who learns one application in the suite already has a head start in learning the next one.

The Internet Explorer 4 suite contains these components for communication and collaboration across the Internet or local intranet:

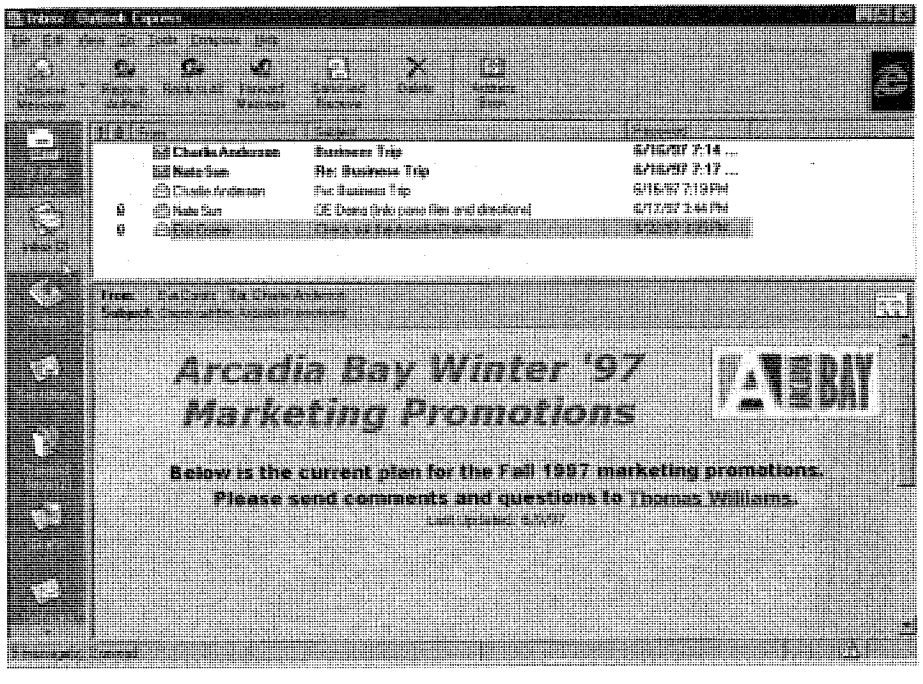
- Outlook™ Express for e-mail and newsgroups
- NetShow for broadcasting
- NetMeeting™ for conferencing and application sharing
- FrontPage Express for Web authoring
- Personal Web Server for Web publishing

Outlook Express

E-mail has become one of the most popular and effective ways for people to communicate, both in business and in their personal lives. Unfortunately, most e-mail has been limited to text-only messages, perhaps with attachments. Internet Explorer 4 allows an entirely new type of standards-based messaging, opening the door to greater richness and detail. Outlook Express supports full HTML, so you can create e-mail messages with the color and functionality of Web pages without knowing HTML.



You can even send full Web pages from the Internet as part of your e-mail message.



In addition, Outlook Express provides users with powerful mail management features, more efficient e-mail and newsgroup use, enhanced security, and full support for Internet standards and technologies. Outlook Express is flexible enough to meet the diverse e-mail needs of users with dial-up Internet access through an ISP and users who work on a LAN based on Internet standards such as Simple Mail Transfer Protocol (SMTP), Post Office Protocol 3 (POP3), Internet Mail Access Protocol 4 (IMAP4), or Lightweight Directory Access Protocol (LDAP).

NetShow

NetShow brings the power of broadcasting to the desktop. With NetShow, Web content comes alive with interactive content, including audio, illustrated audio (images and sound), and video. It includes both client and server components to add the power of traditional broadcasting systems – audio and video – to HTTP. NetShow harnesses Internet technologies and the power of Windows NT server to transform Web communications into a richer and more effective medium: the network show.

NetMeeting

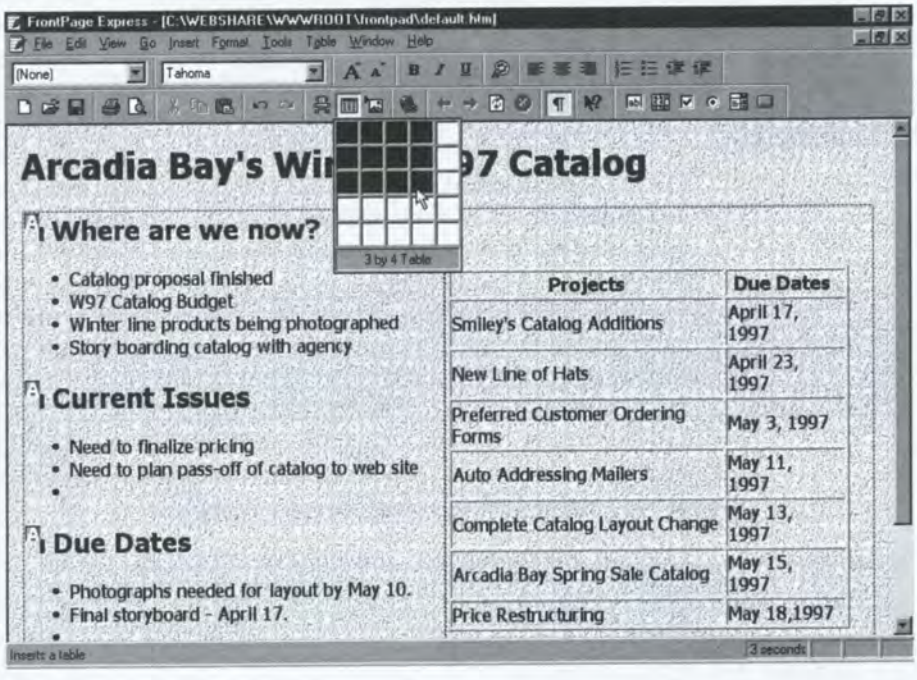
Microsoft NetMeeting is a powerful tool that enables real-time communications and collaboration over the Internet or intranet, providing standards-based audio, video, and multipoint data conferencing support. From a Windows® 95-based or Windows NT 4-based desktop, users can communicate over the network with real-time voice and video technology. They can share data and information with many people through true application sharing, electronic whiteboard, text-based chat, and file transfer features. NetMeeting helps organizations communicate more effectively, thus increasing productivity for their users. Also, NetMeeting helps organizations reduce their support burden by providing real-time application sharing features so that help-desk staff can remotely resolve user problems more quickly and efficiently.

Designed for corporate communication, NetMeeting supports international communication standards for audio, video, and data conferencing. With NetMeeting, people can connect by modem, ISDN, or LAN using the TCP/IP protocol, and communicate and collaborate with users of NetMeeting and other standards-based, compatible products. In addition, support for system policies in NetMeeting makes it easy for administrators to centrally control and manage the NetMeeting work environment.

FrontPage Express

While HTML has made it possible for many people to become Web publishers, it is still not a particularly intuitive language. For this reason, Microsoft FrontPage Express has been included as a component of the Internet Explorer suite. Based on the full-featured Microsoft FrontPage 97 Web authoring and management tool, FrontPage Express features a graphical interface that makes creating HTML pages as easy as creating a document in a word processor.

FrontPage Express allows users to create their own Web pages in a what-you-see-is-what-you-get (WYSIWYG) environment, without knowing HTML. FrontPage Express includes all the features of the FrontPage 97 editor, except for premium features such as Active Server Pages and some of the WebBot® components that rely on specific server extensions. WebBot components from FrontPage 97, such as Include, Search, and Time-Stamp, are still included, however.



Personal Web Server

As millions of users flock to the Internet for information and entertainment, many decide that they also want to share information with the world. Personal Web Server turns any Windows 95 or Windows NT 4 computer into a Web server, enabling easy publication of Web pages. The simplicity of this tool makes the Personal Web Server perfect for home users, schools, and corporate workgroups.

A Great Publishing Platform

The implementation of Internet standards is crucial to enhancing the Web experience. Now users can view and listen to real-time broadcasts, watch videos, run ActiveX controls and Java applets, and play interactive, multiplayer games on the Web. All of this can be experienced with Internet Explorer 4 because of the browser's extensive support for Internet standards and because it includes support for innovative underlying technologies such as Dynamic HTML, ActiveX, and Java. For more information, see Part 6, "Authoring for Internet Explorer 4."

Dynamic HTML

Limitations of current browser technology often require authors using HTML to choose between interactivity and speed. After a page is loaded, changing the display or content of the page typically requires that the entire page be reloaded. Or, depending on how extensive the changes are, additional pages may have to be retrieved from the server. To address that limitation, Internet Explorer 4 includes several new features collectively called Dynamic HTML. These features provide the client-side intelligence and flexibility to enable HTML authors to create interactive pages that dynamically change the display and content of a page entirely on the client machine, without requiring additional server resources. Dynamic HTML enables HTML authors to create innovative Web sites with interactive pages without having to pay a performance penalty. In addition, by reducing requests from the server, Dynamic HTML also reduces server load, thus improving performance.

ActiveX

For those who need more power than Dynamic HTML offers, Internet Explorer 4 provides enhanced support for ActiveX controls. ActiveX controls are software components that provide interactive and user-controllable functions. Small and versatile, they open a wide range of opportunities for creating attractive and effective Web content. Internet Explorer 4 presents a set of new opportunities for control and script engine developers. Controls can now be faster, smaller, and more integrated than ever before.

Java

You can also use Java to extend the capabilities of Dynamic HTML and create even more interactive, exciting Web content. While Internet Explorer 3 brought the fastest, most reliable, and fully compatible Java support to the market, Internet Explorer 4 goes even further, taking Java support to the next level. Java applets not only run faster than in Internet Explorer 3, but also you can optimize Java applets to take full advantage of all the features and speed of the Windows 95 and Windows NT operating systems. In addition, Internet Explorer 4 introduces a Java security model that allows you to run powerful, interactive Java applications without harming your organization's computers or threatening its privacy.

Active Scripting

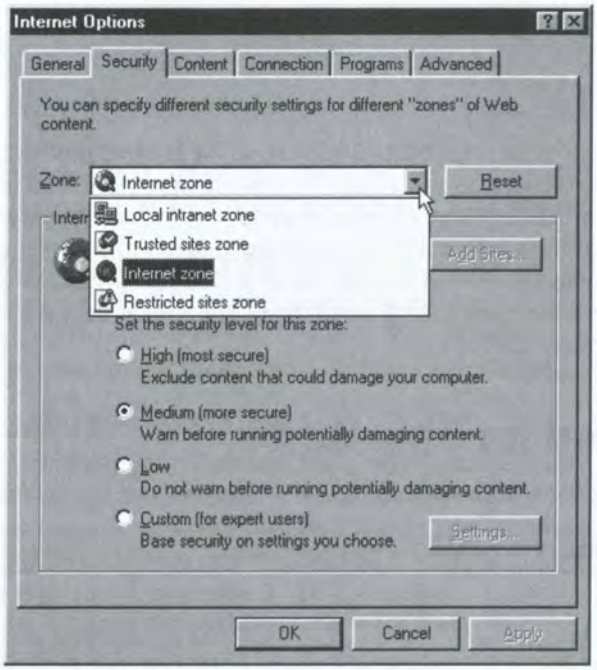
With its support for active scripting, Internet Explorer 4 provides fast, comprehensive, language-independent script-handling capability. Users can view Web pages that use any popular scripting language, including Visual Basic Scripting Edition (VBScript) and JScript™ development software. Internet Explorer scripting support allows authors to develop pages that can ask questions, respond to queries, check user data, calculate expressions, link to other programs, and connect to ActiveX controls, Java applets, or anything else you specify. In addition, Internet Explorer 4 complies with the new Internet scripting standard specified by the European Computer Manufacturers Association (ECMA). Compliance to Internet standards helps ensure that the Web content and active scripting you develop for Internet Explorer 4 will run seamlessly across a wide range of platforms and products.

Enhanced Security and Privacy

Microsoft Internet Explorer 4 provides a full range of security features to help system administrators manage Internet and network security in their organizations. Internet Explorer 4 also provides features to help ensure user privacy. Part 7, "Security and Privacy," describes the security and privacy features of Internet Explorer 4 and explains how these features can be used to help protect the individuals and information in your organization.

Security Zones

Internet Explorer 4 introduces security zones, an innovative approach to secure browsing that offers greater management control and flexibility. Security zones allow users to divide the Internet and intranet into four groups of trusted (safe) and untrusted (unsafe) areas and to designate that specific Web content belongs to these safe and unsafe areas. This Web content can be anything from an HTML file, a graphic, an Active X control, a Java applet, or an executable program. After zones of trust have been established, browser security levels can be set for each zone.



Certificate Management

To verify the identity of individuals and organizations on the Web and to ensure content integrity, Internet Explorer 4 uses industry-standard digital certificates and Microsoft Authenticode 2.0 technology. Together with security zones, these can be used to control user access to online content based on the type, source, and location of the content. For example, system administrators can use security zones in conjunction with certificates to allow users to have full access to Web content on their intranet but limit users' access to Web content from restricted sites located on the Internet.

Trust-Based Security for Java

Internet Explorer 4 provides trust-based security for Java, which is a cross-platform security model for Java that provides fine-grained management of the permissions granted to Java applets and libraries.

Content Ratings

System administrators can control the types of content that users can access on the Internet by using the Internet Explorer 4 Content Advisor. You can adjust the content rating settings to reflect what you think is appropriate content in four areas: language, nudity, sex, and violence.

User Privacy

Internet Explorer 4 supports a wide range of Internet protocols for secure information transfers and financial transactions over the Internet or an intranet. Internet Explorer 4 also provides a variety of features to help users ensure the privacy of their information. For example, Microsoft Wallet is an "electronic wallet" where users can securely store and access private information such as credit card numbers, digital certificates, and digital keys for easy and private online transactions.

Conventions Used for This Guide

The following terms and text formats are used throughout this guide.

Convention	Meaning
Bold	Indicates the actual commands, words, or characters that you type or that you click in the user interface.
Italic	Indicates a placeholder for information or parameters that you must provide. For example, if the procedure asks you to type a *file name*, you type the actual name of a file.
Path\File name	Indicates a Microsoft Windows file system path or registry key—for example, Templates\Normal.dot. Unless otherwise indicated, you can use a mixture of uppercase and lowercase characters when you type paths and file names.
`Monospace`	Represents examples of screen text or code text.

PART 1

Installation of Internet Explorer 4

Part 1 describes the components, processes, and options for installing Internet Explorer 4 on Microsoft® Windows 95 and Windows NT® 4 operating systems, and offers troubleshooting tips. Internet Explorer 4 is also available for Windows 3.1, Windows for Workgroups, and the Macintosh®. For more information about Internet Explorer 4 on these platforms, please refer to the documentation for the specific version you are using.

Chapter 1: Installation Requirements and Issues

This chapter outlines the system requirements for installing Internet Explorer 4 on the Windows 95 and Windows NT operating systems. Issues that may impact the successful installation of Internet Explorer are also discussed.

Chapter 2: Phase 1 - Active Setup

Internet Explorer 4 Active Setup offers an innovative solution to the challenges associated with current Internet-based installation programs. This chapter explains the modular approach used by Internet Explorer 4, which conserves bandwidth and offers users a faster, more versatile installation experience. The chapter guides you through the Active Setup process, step by step.

Chapter 3: Phase 2 - ACME Setup

ACME Setup is the process of extracting, installing, and registering program files on the user's computer, and making the necessary configurations to the system. The processes of the ACME Setup are described in detail here.

Chapter 4: Automated Installation (Hands-Free Mode)

This chapter describes how to install Internet Explorer 4 with a minimum of user prompts. Administrators can use the default settings specified in the Setup program or use the command prompt to choose their own options for installation. Automatic configuration is also explained.

Chapter 5: User Profiles

User profile settings allow multiple users to use a single installation of Internet Explorer while retaining unique, individual settings for each user. This is useful for situations where multiple users share a workstation and for storing roaming settings for mobile users.

Chapter 6: Adding and Removing Components After Setup

After installing Internet Explorer 4, it is possible to further customize the installation by adding new components or removing existing components. This chapter explains how to update your version of Internet Explorer to be sure you're using the latest software.

Chapter 7: Uninstalling Internet Explorer

This chapter describes the processes involved in uninstalling Internet Explorer 4. The files and registry entries that are used for uninstallation are described in detail.

Chapter 8: Configuring Connection Settings

The Internet Connection Wizard (ICW) lets users configure an Internet connection. Users can choose an existing Internet service provider (ISP) account or create a new one. They can establish a connection using either a modem or a local area network (LAN).

CHAPTER 1

Installation Requirements and Issues

In this chapter

Understanding System Requirements 4
Understanding Installation Issues 5

Understanding System Requirements

System Requirements for Internet Explorer 4:

Component	Requirements
Processor	486/66 or better
Operating System	Windows 95, OSR2, or Windows NT 4.0 with SP3
Memory	Windows 95: 8 MB minimum, 16 MB with Active Desktop
	Windows NT: 16 MB minimum, 24 MB with Active Desktop
Disk Space	40 MB for a minimal installation (52 MB with .cab files)
	51 MB for a standard installation (67 MB with .cab files)
	64 MB for a full installation (89 MB with .cab files)
Peripherals	A mouse and a modem or network interface card

At least 70 MB of free disk space is required on the Windows partition if installing to another partition.

For troubleshooting purposes, it is suggested that you add 5 MB to each of the above numbers.

For Setup to run on a computer running Windows NT, Service Pack 3 (SP3) for Windows NT must be installed. If SP3 is not on the system, Setup will fail, with a message that informs the user that SP3 must be installed before Internet Explorer can be installed.

In addition, a user must have administrative privileges to install or uninstall on a Windows NT operating system. This includes having administrative privileges the first time you start your computer after installing or uninstalling. If the Temporary Internet Files cache is on an NTFS volume, all users will need full rights to the folder.

Understanding Installation Issues

Please read through the following issues before installing Internet Explorer 4.

Diskless workstations

A user cannot set up Internet Explorer to run from a network disk. The **Download only** option supports downloading to a network disk, but not running from a network disk.

Coexisting with previous versions of Internet Explorer

It is *not* possible to have Internet Explorer 3 and Internet Explorer 4 coexist on the same computer. Internet Explorer 3 will be restored if Internet Explorer 4 is uninstalled.

Because some clients of MSN™, the Microsoft Network, are based on Internet Explorer 3, MSN users will need to upgrade their MSN client software to MSN version 4 when they install Internet Explorer.

Coexisting with Netscape browsers

Internet Explorer may coexist with any Netscape browser. Although Setup detects the presence of Netscape, there is nothing in the Setup user interface to reflect this. You do not get the choice to upgrade over Netscape and replace it with Internet Explorer. Internet Explorer adopts Netscape user settings at the time of installation. For more information, see Chapter 34, "Planning the Deployment," in Part 9, "Deployment and Maintenance."

Installing on a dual-boot computer

Multiple installations of Internet Explorer 4 on a single computer are not supported. If you plan multiple installations, you should install them to separate operating systems on separate disks or partitions. Internet Explorer provides full support for user profiles so that multiple users of a computer may use a single installation. (See Chapter 5, "User Profiles," for more information.)

Although it is not supported, the Internet Explorer core can be shared on the same disk in separate folders. However, the add-ons always install to the Program Files directory of the Windows disk. If you have Windows NT and Windows 95 installed on the same disk, the add-ons have to be shared. This can cause problems if you uninstall from one operating system, because then you can't uninstall the add-on from the other operating system.

Installing on a system with no browser

For Active Setup to work on a system with no browser installed, a working Dial-Up Networking connection to an ISP must be created before Setup is run. If the distribution server can be accessed on the local intranet, no pre-configuration is required.

Can't reinstall Windows 95 (or OSR2) over Internet Explorer 4

Once Internet Explorer has been installed on a system, the operating system may not be reinstalled without first uninstalling Internet Explorer. Users will get a warning message if they try to do so, but they may ignore the warning. Doing so will result in a mismatch of files and registry settings that will render the system unbootable. The resolution to this state can be found in Appendix B or in the Knowledge Base, in the "Reinstalling Internet Explorer 4" section. You can access the Microsoft Knowledge Base at **http://www.microsoft.com/support/**.

Can't remove Windows Desktop Update during a reinstall

When reinstalling Internet Explorer over itself, you cannot remove the Windows Desktop Update during the reinstall. You can, however, *add* the Windows Desktop Update during a reinstall. For instructions on removing Windows Desktop Update after installation, see Chapter 6, "Adding and Removing Components After Setup."

Can't change the Setup folder during a reinstall

When reinstalling, Active Setup will place the cabinet (.cab) files in the existing Setup folder and will not give you an opportunity to specify a different folder.

Failed trust check

When installing Internet Explorer, all the components in the installation checklist may quickly fail. This may be caused by a corrupt .cab file in the Setup folder (to identify, check file sizes) or a bad version of Wintrust.dll. The .cab file can be replaced by another download.

The log file can be used to see if trust check failed. If it has, there are a variety of possible resolutions:

- Install the Authenticode™ 2 update.
- Rename Wintrust.dll before launching IE4Setup.exe again.
- Add the following key to the registry to disable the trust check:

  HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup
  "DisableCheckTrust"="y"
- Try renaming Digsig.dll and Schannel.dll.

For information about additional issues related to installation and uninstallation of Internet Explorer 4, see Appendix B.

CHAPTER 2

Phase 1: Active Setup

In this chapter

Understanding Active Setup 8
Downloading and Extracting IE4Setup.exe 9
- Understanding Language Considerations 9
- Understanding Cabinet Files in Internet Explorer 4 10
- Downloading IE4Setup.exe 12
- Understanding System Checks 13

Understanding the Active Setup Log File 15
Using the Setup Wizard 16
- End-User License Agreement 17
- Download Options 18
- Installation Options 19
- Windows Desktop Update 20
- Active Channel Language 22
- Destination Folder 23
  - Errors 24
- Distribution Server List Download 24
  - Errors 25
- Download Sites 26
- The .Cab Information File 28
  - Final Disk-Space Check 28
  - Errors 28
- Preparing Setup 29

Downloading to Your Computer 30
Errors 31
.Cab File List 31
Understanding Smart Recovery 34

Note Active Setup is needed only to install Internet Explorer from the Internet, not from a compact disc. If you are installing from a compact disc, skip this chapter, and begin reading at Chapter 3, "Phase 2: ACME Setup."

Understanding Active Setup

Internet Explorer 4 Setup is an innovative solution to the challenges of current Internet-based installation programs. Before Internet Explorer, all browsers were installed by downloading a single, large, monolithic package. If the download failed, it had to be restarted from the beginning of that large file. If a connection to a server was dropped during download, Setup could not resume, and none of the data downloaded could be used.

An executable file cannot run until it has been completely downloaded, so large setup packages are unable to check system requirements in advance. Users may wait a long time, only to find out that they have to free up disk space and try again. A single, monolithic file does not give you control over what components will be installed. If you don't want all the components included in a monolithic package, time is wasted downloading unwanted data, and disk space is wasted storing that data.

Internet Explorer uses a modular approach to solve the problems with previous browser setups. Internet Explorer Setup is based on an ActiveX™ engine called Active Setup, which runs on the client. The process begins with a small setup package. This self-extracting file can be downloaded to the computer, using an existing browser, or it can be copied directly on to a computer with no existing browser. Only 400 kilobytes in size, it downloads very quickly or fits on a single floppy disk. This file is all that is needed to initiate the installation of Internet Explorer. It contains the bare minimum code necessary to give the client Internet access and provide a basic level of file download capabilities.

This small setup package also allows Setup to collect information about the host computer before the download begins. Active Setup uses this information to intelligently manage the download of Internet Explorer .cab files and make installation as efficient and problem-free as possible.

After all the needed user and system information has been gathered, and you have selected from the available Microsoft distribution servers on the Internet, the necessary .cab files are downloaded. Because Setup has already inventoried any Internet Explorer components already on the system and has already asked you which components are desired, it knows to download only the .cab files needed. Once the .cab files have all been successfully downloaded, Active Setup is complete.

The next phase of Internet Explorer 4 Setup uses the traditional ACME Setup engine used by most Microsoft products. This process follows the typical methodology of an ACME-driven installation using a collection of .cab files. The only difference in this case is that the .cab files are in a folder on the local hard disk, rather than on a removable disk or network disk.

Downloading and Extracting IE4Setup.exe

To install Internet Explorer from the Internet, you must first establish a connection to a download site (such as **http://www.microsoft.com/ie/**) with whatever browser is currently installed. From this site, you will download IE4Setup.exe. IE4Setup.exe may also be copied to the computer from a network connection or a removable disk (for unsupported custom installations). Note that if you choose to save IE4Setup.exe to your hard disk when downloading IE4Setup.exe, you will need to manually launch it from your local disk to begin Active Setup.

Understanding Language Considerations

From the download site, you will be asked to select the language version of Internet Explorer you want. Internet Explorer is language-specific. For example, the English version of Internet Explorer will only run properly on English-language-based operating systems. If you have accidentally installed the English version over a non-English-language-based operating system, you can use only limited Internet Explorer functionality. You will not be able to use the Active Desktop™ interface or access Active Channels™, for instance.

Users are not informed if they pick a language that is different from their current shell. (It is assumed that users will choose the correct language.) Setup checks to see if the chosen language matches the currently installed operating system. If it doesn't match, you will get an error saying that the Windows Desktop Update shell will not be installed, and the option to choose the Windows Desktop Update will not be displayed during setup. This error is also displayed if you try to install the wrong language version of the shell from the add-on page. The Windows Desktop Update and the add-on page are discussed later in this chapter.

For troubleshooting information on languages, see "Shell Integration Not Enabled After Installing Internet Explorer" in Appendix B.

Understanding Cabinet Files in Internet Explorer 4

For a number of years, Microsoft has been using cabinet (.cab) files to compress software that was distributed on diskettes. Originally, these files were used to minimize the number of diskettes shipped with packaged product. Today, .cab files are used to reduce the file size and the associated download time for Web content that is found on the Internet or corporate intranet servers. The .cab file format is a nonproprietary compression format, also known as MSZIP, that is based on the Lempel-Ziv data-compression algorithm.

Cabinet files and ActiveX controls can be digitally signed when they are created. A digital signature provides accountability for software developers by associating a given file with a software vendor. Authenticode™ is the name of the client-side technology that reads the digital signatures of .cab files and ActiveX controls and prompts the user with the name of the software vendor before downloading them. (For more information about digital signatures and Authenticode, see Part 7, "Security and Privacy.") There is Visual Basic Scripting Edition (VBScript) that checks the computer for the presence of Authenticode 2.0 and presents the prompt if Authenticode 2.0 files are not found. It is best if you download Authenticode 2.0 and then reboot before installing Internet Explorer 4.

WExtract is a new technology that expands what we can do with .cab files. WExtract gives a .cab files the ability to create a temporary directory, self extract, and then automatically launch a process using an .inf file (in which case a GenInstall is executed) or an executable file. The launched .inf or .exe file can be contained in the .cab file. In the case of IE4Setup.exe, WExtract expands into the files needed to run Active Setup and automatically launches the Setup Wizard. When the process is complete, the temporary directory containing the extracted contents of the .cab file is automatically deleted.

During the course of a successful installation, users will never realize that IE4Setup.exe is actually many files. This is because the extracted files are placed in a special temporary folder that is automatically removed from the user's system after installation has finished or been canceled.

The Setup Wizard creates the folder C:\Temp\Ixp000.tmp and extracts into it. The following files are extracted into the C:\Temp\Ixp000.tmp folder:

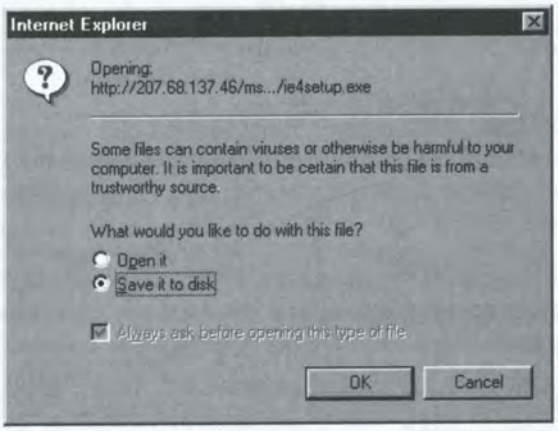
File Name	Byte Size	Description
IE4Setup.inf	40,088	This is the master .inf file for Active Setup. It identifies the other settings files, contains all the registry keys updated during installation, and tells Setup how and when to install components.
Homepage.inf	1,594	Contains information for setting the default home page in the registry.
Content.inf	23,337	Contains language version information for Channel Guide selection.
Wininet.dll	291,600	Core Setup file, provides ActiveX support on non-IE systems.
Urlmon.dll	166,160	Provides HTTP services on non-IE systems. (OSI Application layer).
IE4Wzd.exe	124,672	The Active Setup Wizard.
Advpack.dll	79,824	The Setup engine, it extracts .cab files and changes registry keys.
Inseng.dll	58,528	The Active Setup engine, it controls installing components.
Jobexec.dll	42,688	Parses the .cif file and gives instructions to Inseng.dll
Globe.ani	6,300	Animation for the Setup screens.
IE4Setup.hlp	12,734	Setup Help file.
W95inf16.dll	2,272	Companions to Advpack.dll used for thunking.
W95inf32.dll	4,608	Companions to Advpack.dll used for thunking.
IE4.txt	26,643	The Internet Explorer readme file containing the latest product issues.
This.txt	331	A text document describing the purpose of the setup folder.
License.txt	11,026	Microsoft Internet Products License Agreement.

If you cancel the Setup Wizard at any time, the C:\Temp\Ixp000.tmp folder is deleted before Active Setup quits. You will only see this folder if Active Setup quits unexpectedly and the folder is not deleted.

Downloading IE4Setup.exe

▶ **To download IE4Setup.exe using Internet Explorer 3**

1. If Authenticode 2.0 is not on the computer, install it, and then restart the computer before continuing. See **http://www.microsoft.com/ie/security/** to download Authenticode 2.0.
2. Download IE4Setup.exe from **http://www.microsoft.com/ie/ie40/download/**.
3. While following the instructions from the download site, choose a download server that is close to you.
4. In the **Internet Explorer** dialog box that appears when you begin the download, select the **Save it to disk** option, and then click **OK**.



5. In the **Save As** dialog box, specify the root directory of C:\ as the target path, and then click **Save**.
6. Using a file management tool, such as Windows Explorer, create the folder C:\Savepack, and copy (do not move) IE4Setup.exe from C:\ to the C:\Savepack folder.
7. Delete any files in the C:\Windows\Temp folder.
8. From the command prompt, type the following to extract the .cab file but suppress its execution:

   C:\IE4Setup /c

9. When you get the following prompt, type **C:\Windows\Temp**.

From the C:\Windows\Temp folder, support engineers can check for corruption and attempt an installation by running IE4Wzd.exe, which was suppressed by using the **/c** switch.

Tip To be sure the Setup package has not been customized, support engineers troubleshooting Setup should verify the size of IE4Setup.exe.

Note For a full list of WExtract command line switches that can be used on IE4Setup.exe, refer to Appendix A.

Understanding System Checks

When IE4Setup.exe is executed, it extracts and prepares the system to run Active Setup by performing a series of system checks and installing the files needed to run Phase 2, ACME Setup. The progress of the installation is recorded in a log file that is automatically created in the Windows directory. This file can be useful for troubleshooting installation problems.

IE4Setup.exe performs the following system checks:

- **Operating system** Checks the version of the operating system (Version.dll). If Windows NT 3 is detected, a Setup failure message will be displayed, stating that Setup will only run on Windows NT 4.
- **Version** Checks to see if a previous version of Internet Explorer is installed on Iexplore.exe and Shadocvw.dll.
- **OLE** OLE is initialized.
- **The Shell** Shell32.dll is checked for its version and its language version (409 is the code for English).

- **Smart Recovery** Checks for a previous failed installation of Internet Explorer.
- **Encryption Check** Internet Explorer 4 contains two versions of Schannel.dll: one is an update for domestic 128-bit encryption, and the other supports exportable 40-bit encryption.

  The 128-bit domestic version of Schannel.dll is encrypted using 128-bit encryption, so it is impossible to hack it in without already having 128-bit encryption. This allows Internet Explorer to be shipped out of the country, without worrying about encryption export laws, policies, or procedures.

  To upgrade Internet Explorer from 40-bit encryption to domestic 128-bit encryption, the update may be downloaded from **http://www.microsoft.com/**.

  Both versions of Schannel.dll support Server Gated Cryptography (SGC). SGC is different from straight 128-bit encryption. SGC 128-bit encryption is implemented as an extension to the standard SSL 3.0 protocol. A 128-bit SSL session is set up only if the server presents a special SGC certificate; otherwise, a 40-bit session is used. (For more information about SGC, see Chapter 28, "Digital Certificates.")

If the version check fails, you will get a message stating that the version of the browser cannot be determined, and setup will not continue. If Internet Explorer is the installed browser, just rename Shdocvw.dll and IExplore.exe.

The following files are loaded directly into memory from the IXP000.tmp folder:

- Advpack.dll
- W95inf16.dll
- W95inf32.dll

The following files are copied from C:\Temp\Ixp000.tmp to the C:\Windows\System folder and registered:

- Wininet.dll
- Urlmon.dll
- Jobexec.dll
- Inseng.dll

Wininet.dll and Urlmon.dll are not copied if a released version of Internet Explorer 3 was the previous browser and the existing files pass the check.

Understanding the Active Setup Log File

The Active Setup Log.txt file is a log of the entire Active Setup process from the moment IE4Setup.exe is executed until the download of the last .cab file is complete. When IE4Setup.exe is executed, Active Setup Log.txt is created in the C:\Windows folder. If an Active Setup Log from a previous Internet Explorer Setup session exists, it is renamed to Active Setup Log.bak.

The log begins with the date and time Active Setup was launched and ends with the date and time it successfully downloads the last .cab file. As the user goes through the Setup Wizard, logging entries are continually written to this file. It is the most informative log file for determining what caused a download failure and when the failure occurred. Most entries logged in this file are also written to the registry.

The following HResult error codes indicate the download phases in which an error occurred. This can help to determine exactly what Setup was doing when it failed and also give an idea of the cause:

HResult Error Code	Download Phase
0	Initializing (Making a temp folder, checking disk space)
1	Dependency - (Checking for all dependencies)
2	Downloading (Server to download folder)
3	Copying (Download folder to temp installation folder)
4	Retrying (Restarting download due to timeout or some other download error)
5	Checking Trust
6	Extracting
7	Running (.inf or .exe)
8	Finished (Installation complete)
9	Download finished (Downloading complete)

The following are other common error codes:

- **80100003.** During install, one or more files are missing from download folder.
- **800b*xxxx*.** Anything starting with 800b is a trust failure.
- **800C*xxxx*.** Anything starting with 800C is a Urlmon failure (for example, 800C005 - file or server not found, 800C00B - connection timeout).
- **8004004.** User canceled.

Using the Setup Wizard

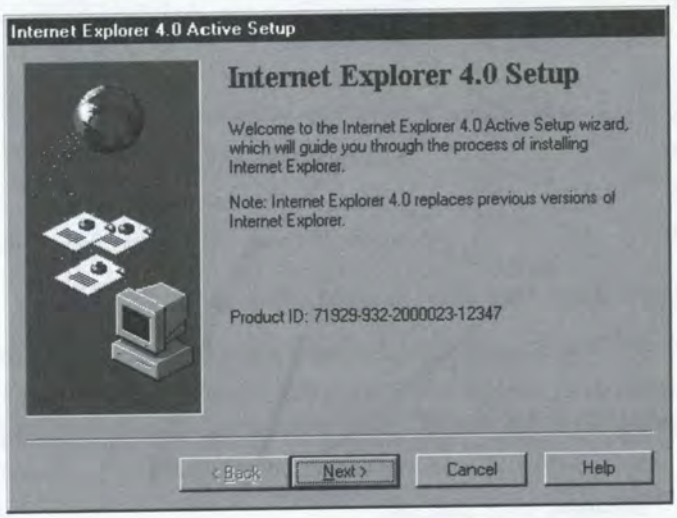
After extraction, IE4Setup.exe automatically launches the Setup Wizard, IE4Wzd.exe, unless you suppressed it earlier. If you did, you can run it from the command prompt by typing **C:\Windows\Temp\IE4Wzd** and pressing ENTER. To control the wizard, you can use the following command-line switches:

Command-line Switch	Description
/m:[0|1|2]	Specifies the installation option (default is **1**)
/I:[y|n]	Specifies whether to install the integrated desktop or not (default is **y**)
/r:n	Disables SoftBoot
/v	Skips version check

For a full list of WExtract command-line switches that can be used on IE4Setup.exe, refer to Appendix A.

Note All the dialog boxes in this chapter show the default settings.

After performing the system checks, the Setup Wizard walks the user through the Active Setup. The rest of this section details the Setup Wizard dialog boxes, explains what actions are being performed "behind the screens," and offers relevant, administrative information related to each screen, such as corresponding registry keys.



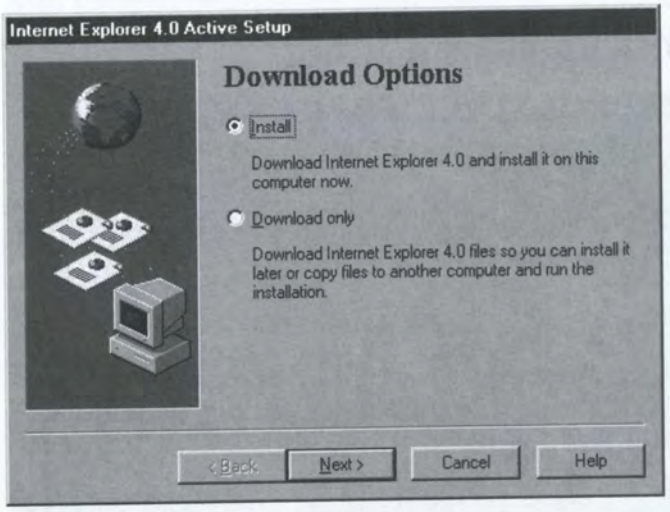
The Setup Wizard uses the Times New Roman font, Timesbd.ttf, installed on the system, for the titles. If this font does not exist, the System font will be used.

End-User License Agreement

The License agreement text comes from License.txt. At this time, the version of any installed ActiveX support files is checked, including Wininet.dll, Urlmon.dll, Shdocvw.dll, and Shlwapi.dll. Trust files are checked: Wintrust.dll, Digsig.dll, Inseng.dll, and Jobexec.dll. If these files need updating, Active Setup does so using the AXControl.Register section of the IE4Setup.inf to install and register them:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IE4\SETUP
"ActiveXFiles"="C:\ WINDOWS.95\SYSTEM\..."

Download Options



This option specifies whether Setup should stop after the download phase or continue on into the installation phase. The download option is recorded in the following log, where Install=1, Download=2:

```
Setup Options :[1|2]:
```

A preliminary space check is done on all partitions, and free space is logged. The text string, "Microsoft Corporation," corresponding to the digital signature of each .cab file, is added to the Trust Database in the registry so that users do not get security warnings during download:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing\Trust Database\...

Installation Options

This dialog box offers the following installation options:

Option	Disk Space	.cab File Size	Components
Browser Only	40 MB	12 MB	Internet Explorer core, Java VM, DirectShow™ (DirectX® mini, DirectDraw™, DirectAnimation™), and ActiveMovie™.
Standard	51 MB	16 MB	Internet Explorer, Outlook™ Express, Connection Manager, Internet Connection Wizard, Java VM, Microsoft Interactive Music Control, DirectShow, ActiveMovie, VDO Player, Microsoft Wallet, and Web fonts.
Full	64 MB	24.3 MB	Internet Explorer, Outlook Express, NetMeeting™, NetShow™, FrontPage™ Express, Web Publishing Wizard, Chat 2.0, Connection Manager, Internet Connection Wizard, Java VM, Microsoft Interactive Music Control, DirectShow, ActiveMovie, VDO Player, Real Player, Indeo™ 5, Microsoft Visual Basic® 5.01 Runtime, Microsoft Wallet, and Web fonts.

The installation option is recorded in the log, where Standard= 0, Enhanced=1, and Full=2:

```
Setup mode selected :[0|1|2]:
```

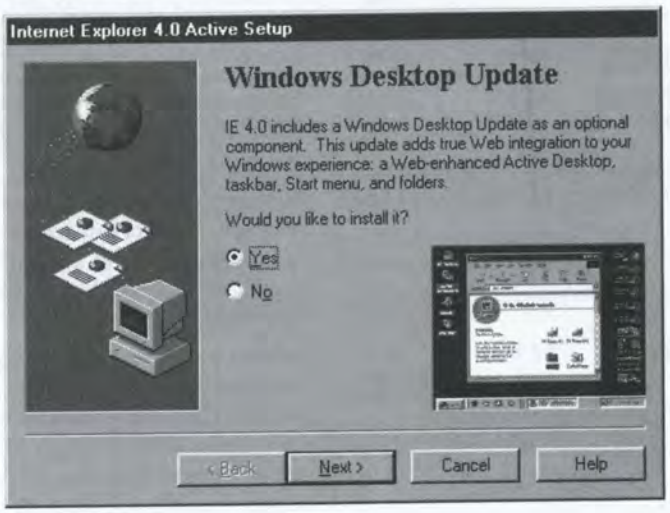
The installation option is also recorded in the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\ InstallInfo\"Setup mode"="2"

The IE4Setup.inf contains a list of all possible components and the mode numbers that correspond to each. Active Setup will install only those components whose Setup mode flag corresponds to the installation option selected by the user. For example, a component that would be installed in all three options would have the flag: Modes="012", whereas a component that would only install in the Full Installation option would have the flag, "Setup mode"="2".

Minimal installation is not a valid option when installing from a compact disc. Only the Standard Installation and Full Installation options are available. This is to ensure that the new Internet Connection Wizard is installed.

Windows Desktop Update



This dialog box allows users to choose if they want to install the Windows Desktop Update (integrated browser). For more information about the Windows Desktop Update, see Part 3, "True Web Integration." You will not see this dialog box if the Windows Desktop Update is already installed. If you do see it and click **No**, the example image will change to the following:

The option is recorded in the log, where the default is **Yes**:

Integrated Browser : [0|1]:

The option is also recorded in the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\ InstallInfo\"Integrated Shell"="1"

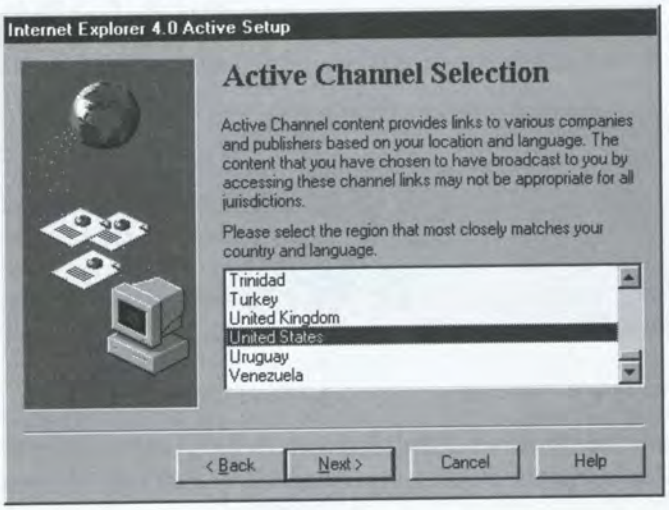
If you choose **Yes**, one of the following additional .cab files will be downloaded, which contains all the files required for the Windows Desktop Update:

- IE4Shl95.cab for Windows 95 systems
- IE4Shlnt.cab for Windows NT systems

If you choose not to install Windows Desktop Update at this time, you can add it later. This process requires reconnection to a download site to get IE4Shl*xx*.cab, even if the file exists in the Setup folder.

Jobexec.dll will parse the AddInstallShell.win section of the IE4Setup.inf when installing the Integrated Shell, otherwise the No_Integrated_Shell section is used.

Active Channel Language



At this point, the user has already chosen the language version of the product when downloading IE4Setup.exe. This dialog box only relates to Active Channels™. It gets its information from Content.inf.

Setup performs a language check on Shell32.dll (409=English).

The choice is registered in the log:

```
Content selected:United States:
```

This choice determines the version of Chl*xx-xx*.cab (English example: Chlen-us.cab) that will be downloaded. This file is a collection of the default channels also known as the Channel Guide.

Destination Folder

This dialog box specifies the location of the folder to which the .cab files will be downloaded. The default is to create this folder on the partition with the most free disk space. This dialog box is displayed only if a Setup folder does not exist. When reinstalling Internet Explorer, you will not see this dialog box.

The path is recorded in the registry as:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\ InstallInfo\"Download Dir"="C:\Internet Explorer 4.0 Setup"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\Download Directories\"C:\Internet Explorer 4.0 Setup"="1"

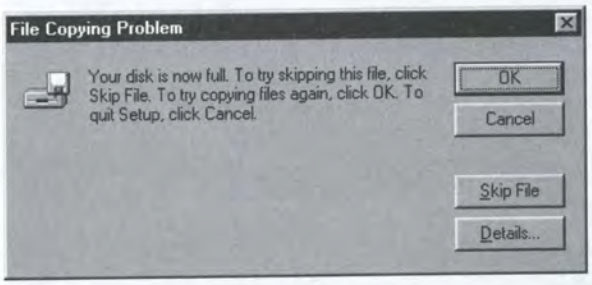
The path is also recorded in the log:

```
Download Dir: C:\Internet Explorer 4.0 Setup
```

Note that no connectivity is required up to this point.

Errors

If there is not enough disk space for the chosen installation option on the partition specified, the following error message will be displayed:



This is a rough-estimate disk-space check. It is not as accurate and is more forgiving than the final disk-space check. The C:\Temp folder and Temporary Internet Files folder are not automatically cleared to free up disk space, but it's a good idea to do this manually before installing Internet Explorer.

Distribution Server List Download

An HTTP GET request for IE4sites.dat is sent to the address specified in the IE4Setup.inf:

```
[Strings]
url="http://www.microsoft.com/…"
```

The IE4sites.dat file is a table of download locations. It contains four fields: region, language, friendly name, and URL. The language-specific Setup package determines which language version of IE4Sites.dat is pointed to in the IE4Setup.inf.

The C:\Windows\Msdownld.tmp folder is created. The IE4Sites.dat is downloaded to the C:\Windows\Temporary Internet Files folder and is used to display the list of download servers. The location of the site list is copied to the registry from IE4Setup.inf:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\Jobs\Job.IE4\"DownloadSiteURL"="http://…/ie4sites.dat"

Errors

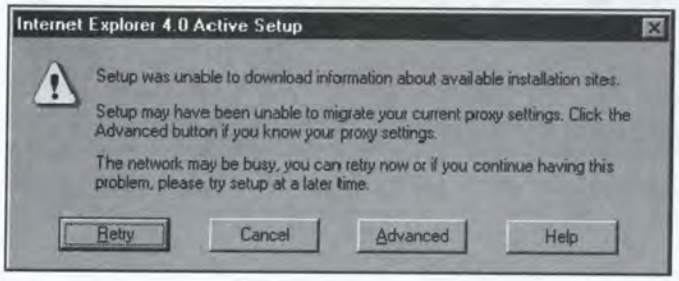
There is a potential for Proxy errors at this point.

Urlmon.dll uses the existing proxy settings in the following registry key to connect to the site specified:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\"ProxyServer"="*data*"

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\"ProxyEnable"="01 00 00 00"

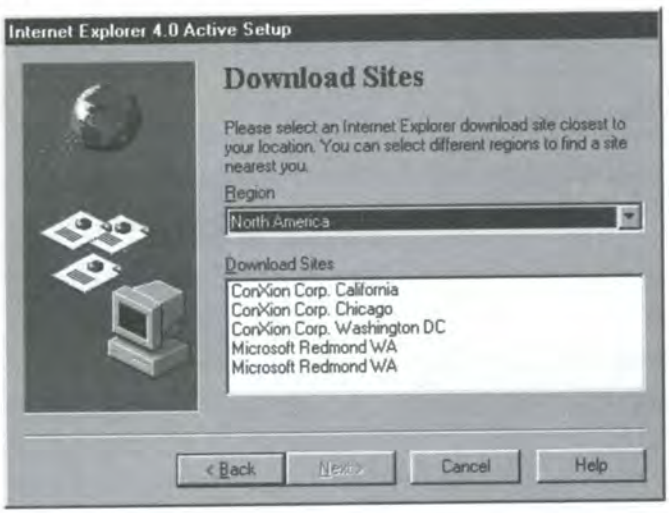
If the system was configured to use an ISP that uses proprietary proxy settings, Setup may be unable to find and adopt those proxy settings.



The **Advanced** button gives the user the following convenient interface to edit the registry keys that Setup uses to find the proxy server:

Even though proxy settings are adopted correctly, server-side problems with the proxy server, download server, or network traffic could prevent the successful download of IE4sites.dat. It is suggested that the connection be retried a few times before changing the proxy settings.

Download Sites



The subfolder C:\Windows\Msdownld.tmp\Ase000.tmp is created. The selected download region is recorded in the log:

```
Selected region: North America:
```

The IE4sites.dat is sorted by region, and only sites for the selected region are displayed in the user interface. The selected download region and all corresponding URLs are recorded in the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\Jobs\Job.IE4\"DownloadSiteRegion"="North America"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ActiveSetup\Jobs\Job.IE4\URLList

http://mschus.www.conxion.com/msdownload/ie4/rtw/x86/en/cab = "ConXion Corp. Chicago"

http://mssjus.www.conxion.com/msdownload/ie4/rtw/x86/en/cab = "ConXion Corp. California"

http://msvaus.www.conxion.com/msdownload/ie4/rtw/x86/en/cab = "ConXion Corp. Washington DC"

http://207.68.137.46/msdownload/ie4/rtw/x86/en/cab = "Microsoft Redmond WA"

Note This registry key is deleted at the end of Active Setup. The download sites are listed with the protocol that Active Setup will use to connect to the distribution server. FTP is slightly faster than HTTP.

The selected download site and its corresponding URL are also recorded in the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\Jobs\Job.IE4\DownloadSiteURL

Note The keys in **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup**, previously mentioned in this chapter, are not actually written until the **Next** button on the **Download Sites** dialog box is clicked.

The .Cab Information File

As soon as the user selects the download site, a GET request for IE40cif.cab is sent to the URL corresponding the chosen distribution server. IE40cif.cab is downloaded to the Internet Explorer 4 Setup folder. It is immediately copied to C:\Windows\Msdownld.tmp\Ase000.tmp, where it self-extracts to a single file, IE40.cif.

The IE40.cif file contains instructions for downloading the .cab files and installing the components they contain. It includes a separate section for each component. Each section contains the name and size of the .cab file, the amount of disk space needed to install that component, a dependency (all components are dependent on the successful installation of the Internet Explorer core files), and more.

Final Disk-Space Check

At this point, Active Setup has enough information to perform a final disk-space check, including:

- The amount of free space on each partition.
- The list of components required for the selected installation option (from the IE4Setup.inf).
- The target location to which the .cab files will be downloaded.

And now that the IE40.cif has been downloaded, Active Setup can use the size information it contains for each component to be installed. The size of each component included in the selected option is added up. This number is compared the amount of free disk space on the target partition.

Errors

If the connection to the distribution server fails to be established or is dropped at any point during the download phase of Setup, the following connection error will be displayed:

A simple traffic issue may cause this. Just clicking **OK**, going back to the previous dialog box, and clicking **Next** until a connection is made may solve the problem.

A corrupt IE40.cif will prevent any component from installing. The best way to recover from this situation is to choose a different distribution server to get a new IE40cif.cab.

Preparing Setup

A connection is established to the download site. Active Setup logs the time that JobExec.dll begins parsing IE4Setup.inf and IE4.cif for jobs (instructions about which .cab files to download based on the chosen installation option). Each job with the appropriate mode flag is executed, and its corresponding .cab file is downloaded.

The Filelist.dat, IE4Setup.dir, and IE4Setup.ini are created in the C:\Internet Explorer 4 Setup folder. They log the .cab files as they are successfully downloaded and track which files are contained in the Setup folder. Because the Setup folder could contain Internet Explorer files for both Windows NT and Windows 95 operating systems, .cab file version information is also kept here.

At this point, C:\Internet Explorer 4 Setup should contain:

- Filelist.dat
- IE40cif.cab
- IE4Setup.dir
- IE4Setup.ini

The C:\Windows\Temporary Internet Files folder is created.

Downloading to Your Computer

Active Setup provides the user with valuable download information, including a progress bar, bytes received, approximate download time in minutes, and the name of the selected download site. Download time is calculated using .cab file size information in the .cif file and an evaluation of the connection speed. For each .cab file, the following occurs:

1. A dependency check is done (by Jobexec.dll).
2. The download takes place.
3. A CheckSum is performed, and the size of the file is then recorded in Filelist.dat.

   A trust check is performed against its digital signature.

Active Setup has the ability to shift servers during an installation to maintain maximum throughput or recover from a down distribution site. If a connection times out, Setup will try to connect to the next download site in the list and pick up where it left off. If a connection to that site cannot be established, Setup will ask you if you wish to abort the installation or try again.

Errors

Most problems that may be encountered at this point will be related to a trust issue. Perhaps a newer version of Wintrust.dll is needed. Problems may also be encountered in a reinstallation scenario if the wrong version of Inseng.dll or Jobexec.dll is running.

.Cab File List

After the download phase of a full installation, the contents of the C:\Internet Explorer 4 Setup folder should look like the following list, plus these four files that existed prior to download phase:

- IE4Setup.dir (created)
- Filelist.dat (created)
- IE40cif.cab
- IE4Setup.ini (created)

Component	File Name	Byte Size	Description
Core Components			
Browser Only Install			
Standard Install			
Full Install			
ActiveMovie	Amov4ie.cab	512,263	
DirectAnimation	Axa.cab	802,557	
	Branding.cab	10,608	Setup information from the IEAK
Channel Guide	Chlen_us.cab	154,779	
DirectDraw® Extensions	Dxddex.cab	33,166	
DirectX® Layering	Dxmini.cab	339,279	
Web Fonts	Fontcore.cab	627,056	
Front Pad	Fpesetup.cab	1,386,589	

Component	File Name	Byte Size	Description
Internet Connection Wizard	Icw95.cab	315,605	
	IE4.txt	26,643	The Internet Explorer readme file
	IE4_s1.cab	1,427,294	
	IE4_s2.cab	1,458,534	
	IE4_s3.cab	1,458,534	
	IE4_s4.cab	1,458,534	
	IE4_s5.cab	1,313,010	
	IE40cif.cab	14,614	The .cab information file
	IE4data.cab	329,575	
Integrated Desktop	IE4shlxx.cab	800,014	Where x = 95 or NT (optional)
Java Support	IEjava.cab	26,057	
Indeo™ 5	Ir50_32.cab	483,521	
Java Support	Javi386.cab	2,892,161	
Outlook Express	Mailnews.cab	1,472,540	
Microsoft NetMeeting	Mnm2095.cab	1,885,958	
Microsoft Chat	Mschat2.cab	875,227	
Internet Music Control	Msmus_1.cab	176,434	
Visual Basic® Support	Msvbvm50.cab	704,015	
Microsoft Wallet	Mswallet.cab	314,493	
Microsoft NetShow	Nsie4.cab	1,545,651	
Real Player	R32msie4.exe	1,226,680	
	Setupw95.cab	526,607	
	Thisfo~1.txt	331	Text explaining the setup folder
VDO Live	Vdolive.cab	141,175	
Microsoft Publishing Wizard	Wpie4x86.cab	364,605	

You may notice that .cab files IE4_s1.cab through IE4_s5.cab actually have multiple .cab files inside them. The .cab files need to be signed for download security checking, but ACME Setup doesn't work with signed .cab files, so the signed .cab files are downloaded and extracted into unsigned .cab files that ACME Setup can use. This also gives the bonus of separating the process of

building a working component from the signing process. The compact disc contains the same signed .cab files that are downloaded to the Setup folder during an installation from the Web.

When the last .cab file has successfully downloaded, the following things occur:

- The Ase000.tmp folder is deleted, but the C:\Windows\Msdownld.tmp folder remains.
- The success string is written to the Active Setup Log.txt:

  ```
  "Install Engine - Download complete"
  ```

- IE4Setup.exe is moved into the C:\Internet Explorer 4 Setup folder, and the original is deleted.

If the **Download only** option was selected in the **Download Options** dialog box, the user will see this message:

The Setup Wizard will stop here. To begin installing Internet Explorer in this case, you must manually execute IE4Setup.exe.

If the **Install** option was selected in the **Download Options** dialog box, the user will see this message:

In this case, Setup gives you 30 seconds to disconnect from the Internet before proceeding to the installation phase.

For information on Active Setup registry keys not mentioned yet, refer to Appendix A.

Understanding Smart Recovery

Smart Recovery is the feature that allows Setup to recover from interrupted Internet downloads. Smart Recovery allows Setup to pick up where it failed previously, rather than starting over from the beginning.

When rebooting or relaunching, Setup will check the log files, registry, and IE4Setup.ini to determine if a previous installation attempt was incomplete. If so, the following dialog box is displayed:

The status of the two phases of setup are logged in the registry, where 0 = not started, 1 = started, 2 = finished:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\InstallInfo\"Download"={0|1|2}

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\InstallInfo\"Install"={0|1|2}

Smart Recovery and IE4Setup.exe use this information to identify the state of the system.

CHAPTER 3

Phase 2: ACME Setup

In this chapter

Installing Internet Explorer 4 on Your Computer 36
Destination Folder 36
Items Are Being Installed 37
Errors 38
Understanding System.1st Backup 39
Configuring the System 40
Completing Installation 40
Emergency Repair Disk on Windows NT 41
The Welcome Screen 41
Understanding the IE4 Setup Log File 42
Understanding SoftBoot 43
Errors 44

Installing Internet Explorer 4 on Your Computer

The next phase of Internet Explorer 4 Setup uses the traditional ACME Setup engine used by most Microsoft products. Users who are installing Internet Explorer from a compact disc begin the installation process here.

Note If, in the Active Setup part of the installation process, the **Download Only** option was chosen in the **Download Options** dialog box, you must now manually launch ACME Setup by running IE4Setup.exe again. Because the state of the system may have changed since Active Setup completed the .cab file download, or a new user may be launching ACME Setup, some portions of Active Setup are repeated. The system checks are performed again, and the first several user options are gathered again, as described in "Using the Setup Wizard" in Chapter 2. If the **Install** option was chosen in the **Download Options** dialog box, then earlier dialog boxes are not displayed, because it is unlikely the system or user has changed since Active Setup.

Destination Folder

The first prompt for user input is the target folder for the installation. (If the **Install** option was chosen prior to downloading the .cab files, this dialog box is not displayed.)

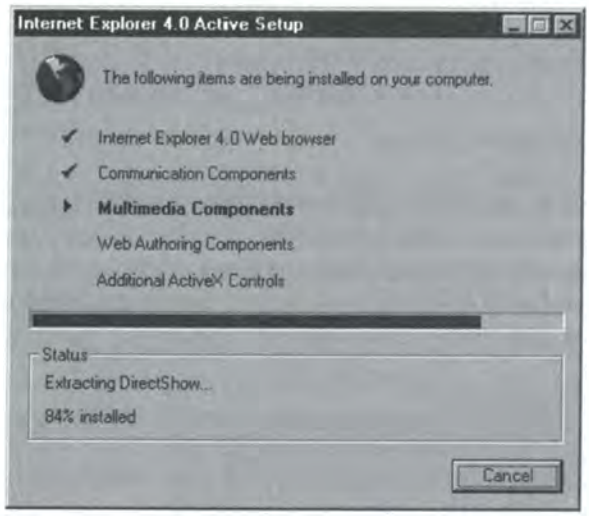
When upgrading a previous version of Internet Explorer, the installed location is used.

The installation folder is recorded in the registry. The default is C:\Program Files\Internet Explorer.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\ InstallInfo\"Install Dir"="C:\Program Files\Internet Explorer"

For a list of ACME 3.0 Setup command-line switches, see Appendix A.

Items Are Being Installed



ACME Setup first copies files, makes registry changes, and then registers .ocx files and .dll files. A check is performed on IExplore.exe to see if there is an existing installation of Microsoft Internet Explorer. Before the Internet Explorer core files are installed, Setup backs up existing files and configuration information needed to restore the system to its pre-Setup state during an uninstall. This data is moved into IE4bak.dat, and the IE4bak.ini is created.

Then the .cab files are expanded and installed. The following steps are performed for each component:

1. **Trust check.** The .cab is found in the Setup folder, and the signature is checked for Authenticode trust verification.
2. **CheckSum.** If trust check is not available, a file size CheckSum is done on the .cab file, based on information in the .cif file. If it fails, the user is prompted to download the .cab file again.
3. **Extraction.** The .cab file is extracted to C:\Windows\Msdownld.tmp\Ase000.tmp.
4. **Install.** Files are moved from Ase001.tmp to C:\Program Files\Internet Explorer\Setup and C:\Windows\System.

The configuration is updated in the registry.

Errors

If any components fail to install, a red X appears next to that component, rather than a check mark. All components are dependent on the core Internet Explorer files. If the core components are not installed successfully, all remaining components will immediately fail, and a red X will appear next to all the items in the install list. If any component fails to install, the following error message is displayed after the last component has installed or failed:

Low disk space or an inaccurate determination of free disk space, a file still in use, an invalid system configuration, or insufficient system requirements can cause this error. It can also be caused by a corrupt .cab file (to identify, check file sizes) or a bad .cif file. This can be resolved by moving IE4Setup.exe to the root, deleting the Setup folder, and re-running the download portion of Setup. This can also be caused by an Authenticode failure.

If Setup is interrupted while ACME setup is updating the OLE files, Smart Recovery will be unable to recover, because the OLE files will not initialize at startup, and the system becomes unbootable. The operating system will need to be reinstalled. The point at which the OLE files are being updated changes, based on the installation option. For example, Full Installation is between approximately 18% and 25%.

Understanding System.1st Backup

If Windows Desktop Update is installed, the registry is backed up as follows:

- The old desktop files are backed up to the file, Integrated Browser.dat, and the corresponding Integrated Browser.ini is created.
- Additionally, for Windows 95 and OSR2, the original System.1st is also backed up and stored in the Integrated Browser.dat.
- A command that copies System.dat to C:\System.1st is added to the **RunOnce** key of the registry. System.dat will be copied to C:\System.1st at the next reboot or when SoftBoot processes the **RunOnce** key.

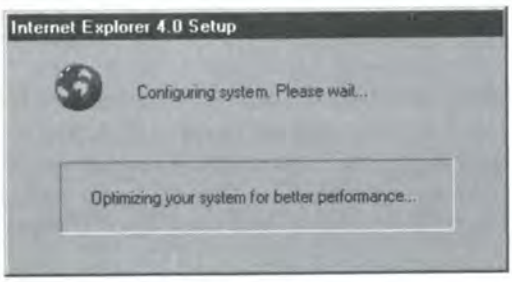
If the original System.1st is not present during the install of the new shell, the new System.1st will not be created. Note that it does not check the validity of the System.1st file. A dummy System.1st will be enough to trigger the backup.

In the case of low disk space, the creation of the new System.1st will be skipped in order to ensure the successful installation of the new shell.

Because a new System.1st may not be created in some circumstances, you will want to verify its creation before using it to replace the registry. Do this by viewing the Integrated Browser.ini file. If System.1st was updated, it will be listed in this file usually at the second entry.

Configuring the System

As the Setup Wizard configures your system, you'll see this screen:



In the subwindow of this screen, you will see three phases:

- Closing all programs
- Optimizing your system for better performance
- Preparing to run Internet Explorer 4

Shortcuts are also updated while this dialog box is displayed.

There are three log files associated with the processes behind the Configuring System screen:

- **SoftBoot Log.txt.** Logs the SoftBoot process.
- **RunOnceEx Log.txt.** Logs the registration of .dll files.
- **Bind List Log.txt.** Logs problems during binding.

Completing Installation

A successful installation will end with this dialog box:

Emergency Repair Disk on Windows NT

After Internet Explorer is successfully installed on Windows NT 4, the backup of the registry that stored on the Emergency Repair Disk (ERD) becomes obsolete. To remedy this, setup will prompt the user to update or create a new ERD.

After rebooting a computer running Windows NT, a final Setup dialog box is displayed as the final changes are made to the system.

The Welcome Screen

The first time the browser is run, it will try to connect to the default Welcome screen. The URL for the Welcome screen is written from the Homepage.inf:

HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main\"Start Page"="http://home.microsoft.com/"

Understanding the IE4 Setup Log File

IE4 Setup Log.txt is the log file for the ACME Setup phase of Internet Explorer 4 installation. It is found in the C:\Windows folder along with the other Internet Explorer log files. If an installation is interrupted during this second phase, this log is part of what tells Smart Recovery where to pick up.

IE4 Setup Log.txt logs the following:

- The time installation begins.
- The successful completion of each step, with one of the following flags:
  - Succeeded
  - Failed
  - Completed successfully
  - Incomplete
- The time the current configuration backup information is saved.
- The backup files placed in IE4bak.dat.
- The files moved from C:\Windows.95\Msdownld.tmp\Ase001.tmp to \Program Files\Internet Explorer\Setup and Windows95\System.
- The successful creation of the following new directories:
  - Program Files\Internet Explorer: Succeeded
  - Program Files\Internet Explorer\Setup: Succeeded
  - Program Files\Common Files: Succeeded
  - Program Files\Common Files\Microsoft Shared: Succeeded
  - Program Files\Common Files\Microsoft Shared\TextConv: Succeeded
  - Windows\Help\Frames: Succeeded
  - Windows\Help\Topics: Succeeded
  - Windows\System\ShellExt: Succeeded
- Registry key change instruction JobExec finds.
- The country code information, which is added to the Telephon.ini.
- The time the log is completed.

"Safe = 1" or "Safe = 0" in the last section of the log describes when it's safe to quit ACME Setup. "Safe = 1" means it's safe. This tells JobExec if it needs to complete a process before responding to a click of the **Cancel** button.

The Telephon.ini file stores the settings for all TAPI-enabled programs, such as Fax and Dial-Up Networking. If Telephon.ini is deleted and regenerated by running Tapiini.exe, it will no longer contain the country code information added to it during the Internet Explorer installation.

Understanding SoftBoot

Normally, when a Setup program runs, it modifies registry keys and replaces files. As long as those keys and files are not in use, this can be done on the fly. A Setup program that needs to modify registry keys and files that are in use, however, writes the instructions for key modifications and file replacements to the Wininit.ini, then prompts the user to reboot the system. During reboot, the Wininit.exe processes the instructions in the Wininit.ini. Then Runonce.exe processes instructions in the **RunOnce** branch of the registry. Finally, the system is loaded, and control is given to the user.

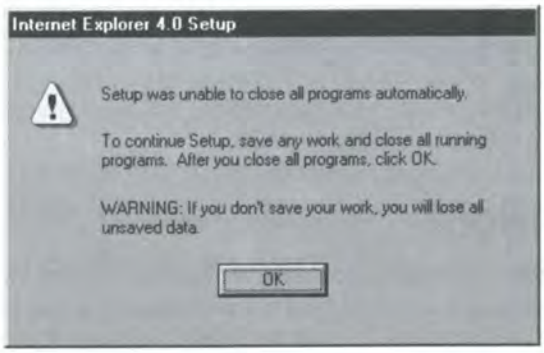
SoftBoot is a way to update the registry and new files without actually rebooting. SoftBoot only works on Windows 95 and OSR2 systems. When installing on Windows NT, the system must always be restarted.

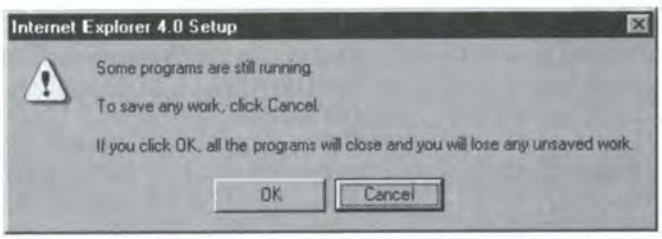
Since SoftBoot is trying to simulate a reboot, it must first take full control of the system. Using information in the Softboot.inf, it closes down running programs. The Softboot.inf contains a listing of programs and processes and instruction for closing them. Processes are listed in the Softboot.inf by section. Each section defines how the process should be terminated. The sections are defined in Appendix A.

Once all running processes have been successfully closed, SoftBoot takes control of the system. At this point, the user cannot use ALT+TAB to focus the system away from Setup. It then executes the instructions in the Wininit.ini, updating registry keys and replacing files. If no errors occur during this process, the Wininit.ini is cleared of entries.

The Wininit.ini is only cleared if all instructions are executed successfully. If any error is encountered, the contents in the Wininit.ini are left so that Wininit.exe will have a chance to do its job upon reboot.

If SoftBoot is unable to close any running programs, the user will see one of the following two messages:





SoftBoot also runs any time Windows Desktop Update is installed or uninstalled separately.

Errors

The way that Setup closes programs may prevent them from saving recent changes or cause them to hang.

Troubleshooting SoftBoot is unnecessary. Because the Wininit.ini entries are not cleared until after SoftBoot is successful, simply rebooting after a SoftBoot failure allows Windows 95 to handle the registry updates as it normally would.

To solve .dll registration problems, use the command below to unregister the .dll files. The entries will re-register automatically upon the next reboot:

regsvr32 /u *filename*.dll

CHAPTER 4

Automated Installation (Hands-Free Mode)

In this chapter

Using Automated Installation 46
Using Debug Install 47
Using Automatic Configuration 47

Using Automated Installation

To run the Setup executable file hands-free, using Setup's defaults, save IE4Setup.exe to your disk, and then type **ie4setup /q**.

This will install Internet Explorer 4 in quiet mode. Setup will bypass the license agreement, will use the Full Installation option, and will install the Windows Desktop Update. You will only be prompted to pick the region and download location. Setup defaults will automatically upgrade to the same folder of any previous Internet Explorer version.

System administrators may want users to run in quiet mode with a different set of defaults. For example, you may want a different install folder and to install some programs other than Internet Explorer. To do this, use the Internet Explorer Administration Kit (IEAK) to customize any of the Setup configurations, or use additional switches with the standard Setup package in conjunction with the **/q** switch, as in the following example:

```
ie4setup /q /c:"ie4wzd.exe /x /q /m:0 /i:n /r:n /s:#e"
```

Switch	Description
/q	Quiet mode (applied to the Setup package)
/c:"ie4wzd.exe	Specifies the executable file to run after extraction
	(all following switches are for the wizard)
/q	Quiet mode (applied to the wizard)
/m:0	Installation option [0|1|2] (0 = browser only)
/i: [y|n]	Install the Integrated Desktop Update
/r: [y|n]	Reboot after installing
/s:#e"	Destination folder (default is #)

When IE4Setup.exe is run from a network share (such as: *server**share* \IE4Setup.exe) containing an IE4Setup.ini file, Setup will not look for a site list but will download all the .cab files from that share.

No Setup folder is created on the client computer when this installation method is used. Installing over the LAN is faster than using HTTP.

Using Debug Install

In Stepping mode (debug install), Setup will stop and present a status dialog box after each step. Stepping mode can be initiated by typing this:

```
ie4setup.exe /c:"ie4wzd.exe /s:#e /g:debug
```

or by creating this registry key prior to running Setup:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\"SteppingMode"="Y"

Using Automatic Configuration

Automatic Configuration is an extension of the Automatic Proxy feature in Internet Explorer 3 that allows administrators to centrally control the default settings for all browser installations on a network. These settings are controlled using the IEAK Configuration Wizard and the IEAK Profile Manager. Settings, such as the browser title, Start page, Search page, Quick Links, and proxy settings, can be configured using these tools.

Administrators can use the IEAK to pre-configure an Automatic Configuration URL that Internet Explorer uses to update its own configuration settings. This file is checked each time Internet Explorer is launched.

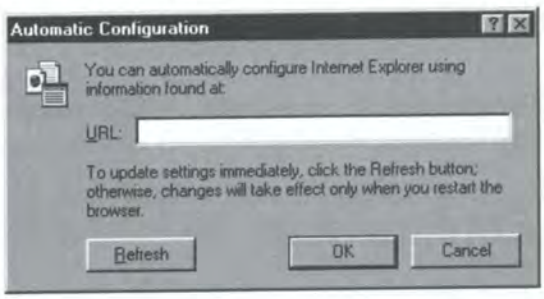
The Automatic Configuration URL is stored in:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\"AutoConfigURL"=

If an Automatic Configuration URL has not been pre-configured, users can manually select one by using the following procedure.

▶ **To manually select an automatic configuration file in Internet Explorer**

1. On the **View** menu, select **Internet Options**.
2. In the **Internet Options** dialog box, click the **Connection** tab.
3. On the **Connection** tab, click the **Configure** button.



4. Type the URL to a Web page with a configuration script or .ins file, and click **OK**.

For more information, see Chapter 37, "Automatic Browser Configuration."

CHAPTER 5

User Profiles

In this chapter

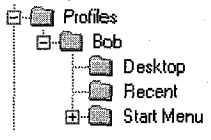
Understanding User Profiles 50
Understanding Profile Structure 52
Understanding Roaming Profiles 52
Platform Dependence 53
Home Directories 53
Creating User Profiles 54
Specifying the Profile Supervisor 58
Understanding Advanced User Profile Functionality 58

Understanding User Profiles

Internet Explorer 4 allows multiple users to use a single installation of Internet Explorer while retaining unique individual settings for each user.

In general, user settings are put in the **HKEY_CURRENT_USER** hive of the registry, and computer settings go under **HKEY_LOCAL_MACHINE**. Per-computer settings that might change when a laptop is in its docking station versus when it's undocked go under the **HKEY_CURRENT_CONFIG** hive.

When multiple users log on at a workstation, they each get their own profile. A profile is a folder for each user that contains user-specific files and configuration information, including the user's copy of the **HKEY_CURRENT_USER** registry hive in the form of a User.dat file. The name of the profile folder is the user name. It is a subfolder of C:\Windows\Profiles.



If a network is used, profiles can roam. This means the user's copy of User.dat is stored on a central server and downloaded to any workstation the user logs on to. This way, users can see the same environment no matter what workstation they use. It also allows administrators central control over individual user settings.

When a new user profile is added to the system, it is not simply a copy of the existing User.dat. A new User.dat is created for the new user with the same system defaults the original user saw after installing the operating system. This process of creating a new User.dat is handled by the operating system. The problem is that this process is not Internet Explorer-aware. Internet Explorer did not exist when these operating systems were written. After installing Internet Explorer, new user profiles created by the operating system have none of the user settings needed to run Internet Explorer.

There are two possible ways that a new User.dat can be updated with Internet Explorer settings: if the Windows Desktop Update is installed, Explorer.exe and Shdocvw.dll perform this task; if not, Loadwc.exe will do it. Loadwc.exe and Explorer.exe are always in memory.

When a new user profile is created, one of these components will compare the data in the following registry keys:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ActiveSetup\ Installed Components

HKEY_CURRENT_USER\Software\Microsoft\ActiveSetup\ Installed Components

If the data matches, then no action is taken. If it does not match, then the two hives are synchronized.

A variable named StubPath is specified for most component keys in the aforementioned section of **HKEY_LOCAL_MACHINE**. The data of that variable provides instructions on what that setting in the new user profile should be. It points to an executable file or .inf files. For executable files, it commonly contains instructions, such as values and APIs. For .inf files, it will point to a specific section containing the data to be added to the User hive.

For each key that is not a match in both the User and Machine hives, the key is created in User (if it doesn't exist), and the StubPath is executed to set its variable. This results in the User keys being updated with the correct data.

Hive synchronization occurs in only one of the following scenarios:

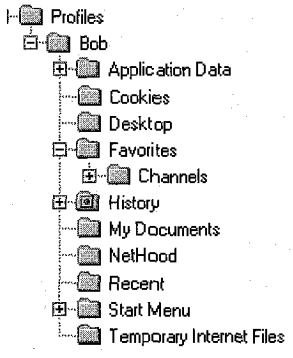
- When a new user profile is created after Internet Explorer has been installed.
- The first time a user logs on after Internet Explorer has been installed on a computer that already has multiple profiles.
- When any user installs a newer version of one of the components.
- When a user on a multiuser system uninstalls Internet Explorer.

The method of configuring new user profiles works the same way in Windows 95 as in Windows NT.

Understanding Profile Structure

The settings in the **Personalized Items Settings** dialog box (for more information, see "Creating User Profiles" later in this chapter) corresponds to the types of files and settings that will be stored in a Users Profile folder.

In addition to containing a copy of the **HKEY_CURRENT_USER** hive (User.dat), an Internet Explorer profile contains subfolders for Cookies, Recent, History, Favorites, Desktop, and Application Data.



Understanding Roaming Profiles

Using the **Personalized Item Settings** dialog box to control the content and settings saved on a per-user basis is especially important when considering disk space or implementing roaming profiles. Allowing all the content in a profile to pass back and forth between the server and a client that the profile is using is not a realistic option. Though the server may have unlimited disk space to store the content, the impact on network traffic would be too great. Thus, when using roaming profiles, it is prudent to minimize the amount of data stored in the profiles:

- **Desktop folder and Documents menu.** If the Active Desktop is part of the corporate installation, this can be left on.
- **Start Menu and Program folders.** Since these are dependent on the programs installed in a standard corporate installation, this could be turned off.

- **Favorites folder.** This would be left on so that users could maintain a personal favorites list.
- **Downloaded Web pages.** This is the Temporary Internet Files folder, which contains Cookies and History. If strict size limitations are imposed, this could be left on, but in most cases this would be left off.
- **My Documents folder.** Network traffic would be minimized if this were turned off and a network share were used for document storage.

If roving profiles are already enabled on a computer, installing Internet Explorer will not automatically move the new folders to the profile. You will need to select the items from the User Profiles Manager in Control Panel.

Platform Dependence

Because roaming profiles are handled by the operating system, they are still bound by the differences between Windows 95 and Windows NT user profiles.

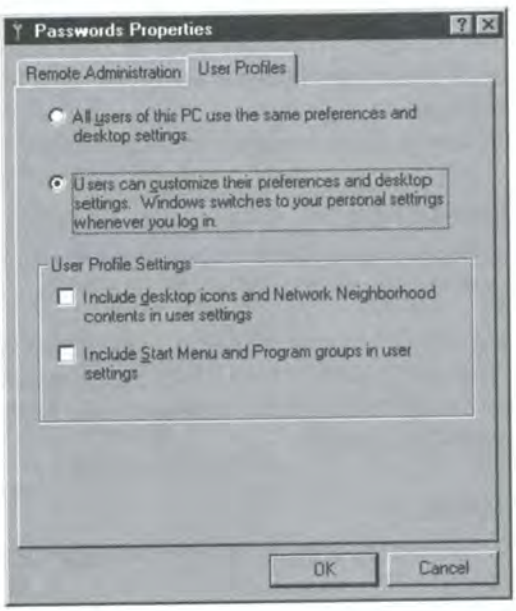
It is possible to log on to an Internet Explorer 4 roving profile using an Internet Explorer 3 system. Settings such as MRU, Start page, Search page, and Favorites are available from the Internet Explorer 3 browser. The main limitation will be orphaned links to Active Channels.

Home Directories

If profiles are enabled and the primary logon is a network logon such as NT or NetWare, Windows 95 will automatically try to roam your user profile to your home folder on a logon server. If accounts are not configured with valid home directories, logon/logoff may take a long time, or unexpected system policies may be applied. To prevent this default roaming behavior, create a DWORD value named UseHomeDirectory in **HKEY_LOCAL_MACHINE\Network\Logon**, and set it to 0.

Creating User Profiles

First, user profiles need to be enabled on the **User Profiles** tab of the **Passwords Properties** dialog box. (Click the Passwords icon in the Windows Control Panel.)



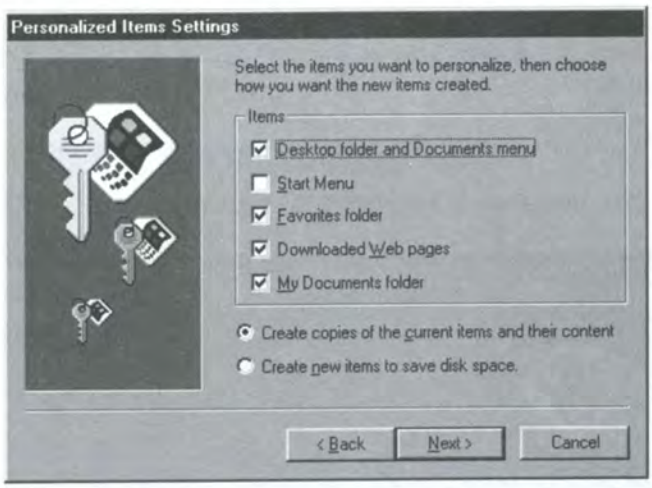
Simply log on using a user name that is new to the system (and therefore has no corresponding .pwl file). The system prompts you to confirm the new user password. Then you are asked if you would like to use global settings or per-user settings.

In addition to simply logging on with a new user name, there is a new way to create a user profile. You will notice a Users icon in Control Panel. This is a new interface and wizard for adding and configuring user profiles. This is simply an alternative user interface for creating a new profile.

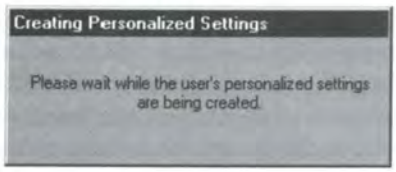
Users

The first time you click this icon, the wizard will run.

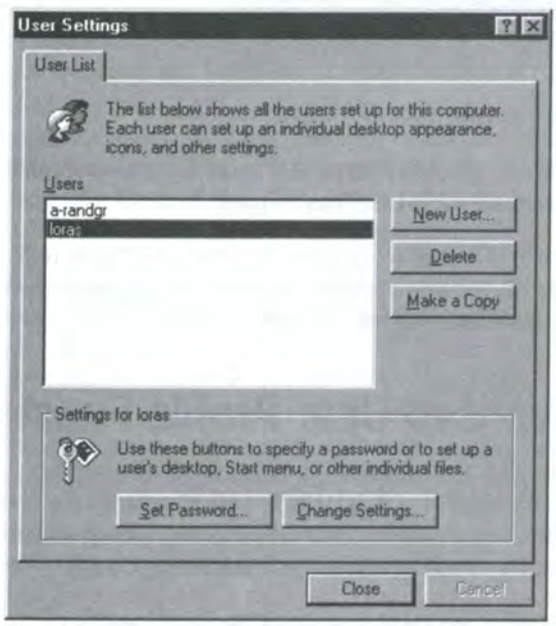
After clicking **Next**, you'll be prompted for a new user name and a password for that user. Here you specify the items to be included in the new profile:



Profiles are now enabled. The user is prompted to reboot. The **Logon** dialog box will appear with the name of the new user. You will see this message after the reboot:



The User Profile Wizard only runs once. After that, the User List is displayed when the Users icon in Control Panel is clicked and the correct supervisor password is entered.



The **Change Settings** button brings up the **Personalized Items Settings** dialog box.

Specifying the Profile Supervisor

The first user who launches the User Profile Wizard is automatically the administrator of all consecutive user profiles on that computer. On systems that have not enabled profiles, the profile supervisor will be the original user who installed the operating system. Non-administrators need the administrator's password to change the settings in their profiles. The prompt for the supervisor's password is displayed as soon as the Users icon in Control Panel is double-clicked.

The supervisor is specified in:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current Version\ProfileList\Supervisor

If the supervisor's Windows password in the .pwl file is null, then no supervisor key is created. By editing the above registry key, more than one supervisor can be specified or supervisor warnings can be disabled.

Understanding Advanced User Profile Functionality

Some functionality was added to the Internet Explorer Shlwapi.dll to make it easier to specify **HKEY_CURRENT_USER** settings for components. The following functions in Shlwapi.dll will retrieve user-specific registry settings that are normally stored in **HKEY_CURRENT_USER**:

- **SHRegOpenUSKey()** is equivalent to **RegOpenKeyEx()**
- **SHRegQueryUSValue()** is equivalent to **RegQueryValueEx()**
- **SHRegDeleteUSValue()** is equivalent to **RegDeleteValue()**
- **SHRegCloseUSKey()** is equivalent to **RegCloseKey()**
- **SHRegGetUSValue()** is the same as calling, **RegOpenKeyEx()**, **RegQueryValueEx()**, and **RegCloseKey()**.

These functions will look in **HKEY_LOCAL_MACHINE** if **HKEY_CURRENT_USER** is empty. This will fix problems for users who didn't install the product. Also, the functions have an optional default value parameter, which will be used if nothing is found in either **HKEY_CURRENT_USER** or **HKEY_LOCAL_MACHINE**. The functions will expand Windows NT environment variables. And a parameter will allow the caller to ignore the **HKEY_CURRENT_USER** value and use the **HKEY_LOCAL_MACHINE** value. This can be used for values that an administrator may not want a user to be able to customize (**SHRestricted**()).

If a local path is written under **HKEY_CURRENT_USER**, it may name a folder that does not exist on other computers (or even cannot exist, because it references a nonexistent drive letter). There are two things that can be done here. First of all, keys can be added to **HKEY_CURRENT_USER** to direct the operating system's user profile code to generate a per-user folder with a specific name and set a registry value under **HKEY_CURRENT_USER** to point to that path. This is how the **Desktop** and **Start** menu folders are made per-user. Secondly, it can store plain filenames and relative paths, wherever possible, and combine them with a per-user base path at runtime. For example, if the HTML template for the Control Panel is Control.htm, don't store C:\Windows\Web\Control.htm in the registry; store just Control.htm, call an API to get the per-user version of C:\Windows\Web, and combine the two.

If a path is stored in the per-user registry, it probably points to a folder with contents that are also per-user. You should consider whether these contents also should roam from one computer to another. Keep in mind that while any files can be made to roam, not all files should. Aside from the performance implications of copying large numbers of files to and from the user's logon server, there are security issues: a user may not want or expect sensitive documents to be copied to any workstation on which they log on and to remain there even after they log off.

Be particularly careful with hyperlinks to programs. They usually have hard-coded paths in them (such as a shortcut to C:\Program Files\Internet Explorer\IExplore.exe), which may not roam well. If you want the contents of a folder full of shortcuts to roam, try to make sure the shortcuts don't have absolute paths in them.

There are actually several choices for how a folder setting behaves:

- **Not per-user.** The setting is the same for all users and does not follow the user to other computers.
- **Optionally per-user.** The user or network administrator can choose whether or not users get their own version of the setting. (To get even more complicated, it's possible for this optional configuration, to itself, be per-user. That is, either all users get their own version of the setting or none of them do, or some users get their own setting and others use the default.
- **Optionally both per-user and roaming.** Either all users get the same setting and the contents do not follow them from computer to computer, or users get their own settings and the contents do follow them. This is how the **Desktop** and **Start** menus work in Windows 95. Again, the determination of whether a particular user gets his own copy or just uses the default is done on a user-by-user basis.
- **Always per-user, optionally roaming.** All users on a computer get their own version of the setting, but the contents of the folder do not follow the user to other computers on the network. This is the way the WININET cache is expected to behave (several megabytes of cache would be far too much to copy all over the network).
- **Always per-user, always roaming.** Users each get their own content; their individual contents always follow them. This is the way ordinary per-user registry items work and is also how the user's Favorites, Cookies, and History folders is expected to work.

To keep the user interface simpler, the optional choices should only be used if there are good reasons why you would want a setting to be per-user or roaming and good reasons why you may want it not to be. If there are compelling arguments on both sides, make it optional. If the arguments on one side are weak or far-fetched, do not make it optional.

CHAPTER 6

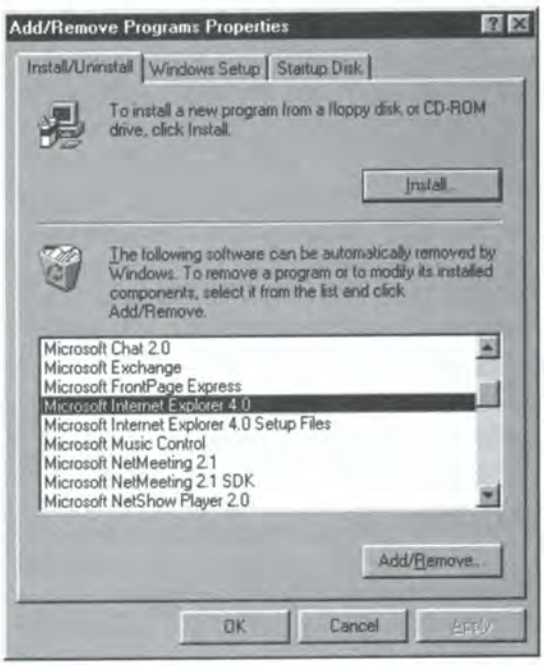
Adding and Removing Components After Setup

In this chapter

Understanding the Add/Remove Dialog Boxes 62
Removing Components 63
Deleting the Setup Folder 63
Adding Components 64
Adding Components from the Internet 64
Adding Components from a Compact Disc 66
Reinstalling Components 67
Using IExtract.exe 69
Adding and Removing the Windows Desktop Update 69
Updating Internet Explorer 4 70
Using Product Update 72

Understanding the Add/Remove Dialog Boxes

There are two **Add/Remove** dialog boxes used by Internet Explorer, each with distinctly different functionality. The first dialog box is the traditional **Add/Remove Programs** applet in the Windows Control Panel. This is used to uninstall only Internet Explorer add-on components with corresponding .inf files containing uninstall information. This is not used to uninstall the Internet Explorer core files:



The second dialog box appears when the user selects **Microsoft Internet Explorer 4** from the software list in the **Add/Remove Programs** dialog box and then clicks the **Add/Remove** button.

This dialog box is displayed by Setupwbv.dll, which knows how to invoke maintenance mode ACME, call an .inf file uninstall, or launch Internet Explorer to download new components from the Web.

Removing Components

▶ **To remove a component**

1. In the Control Panel, double-click the Add/Remove Programs icon.
2. In the software list, click the component you want to remove, and then click the **Add/Remove** button.

When removing a component, its .cab file is always left in the Setup folder so that the user may reinstall it without downloading the .cab file again.

Deleting the Setup Folder

▶ **To remove the Setup Folder**

1. In the Control Panel, double-click the Add/Remove Programs icon.
2. In the software list, click **Microsoft Internet Explorer 4.0 Setup Files**, and then click the **Add/Remove** button.

This will remove the Setup folder and all the .cab files it contains. This procedure is no different than manually deleting the folder.

Warning Unless you have a compact disc containing Internet Explorer, deleting this folder is not recommended, because you will not be able to perform a reinstall if Internet connectivity is lost for any reason.

For information on how to remove Internet Explorer core files or use the Uninstall Log, see Chapter 7, "Uninstalling Internet Explorer."

Adding Components

You can add optional components to Internet Explorer 4 from the Internet or from the Internet Explorer compact disc.

Adding Components from the Internet

▶ **To add or remove components from the Internet**

1. In the Control Panel, double-click the Add/Remove Programs icon.
2. In the software list, click **Microsoft Internet Explorer 4.0**, and then click the **Add/Remove** button. The resulting dialog box will present you with three options.

3. Select **Add a component to Internet Explorer**, and click **OK**.

Setup will automatically launch the browser (if it's not already running) and connect to one of the two sites below, depending upon the platform being used:

- **http://www.microsoft.com/ie/ie40/download/rtw/x86/en/download/addon95.htm**
- **http://www.microsoft.com/ie/ie40/download/rtw/x86/en/download/addonnt.htm**

Note The URLs to the add-on pages were moved from the IE4Setup.inf [AddonPagesW95.reg] and [AddonPagesWNT.reg] to the registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\Addon-URL

Tip You can also access this Web page by selecting **Product Update** from the **Help** menu in the Internet Explorer window. From the Download Area Web site, click the **Component Downloads** hyperlink.

These pages contain VBScript that will perform some system checking and determine what components are already installed on the system. The results of this check are displayed on the right side of the web page as **Not Installed**, **Already Installed**, or **Newer Version Available**. You simply check the boxes on the left side of the page that correspond to the components you want, then select the site from which the .cab files will download.

The **Upgrade All** button on the **Components Download** page can be used to reinstall all components that have a "Newer Version Available" result.

Some components available on the **Components Download** page are not included in any of the three installation options when installing Internet Explorer 4. Those components are:

- VRML
- Microsoft Agent
- Macromedia Shockwave Director

- Macromedia Shockwave Flash
- Supplemental Web Fonts
- Internet Explorer Sound Pack
- Task Scheduler
- Multi Language Support (Japanese, Korean, Pan-European, Chinese Traditional, Chinese Simplified)

Also, the Microsoft Script Debugger is available on the Site Builders Network (SBN) rather than as a component on the Internet Explorer 4 Component Download page.

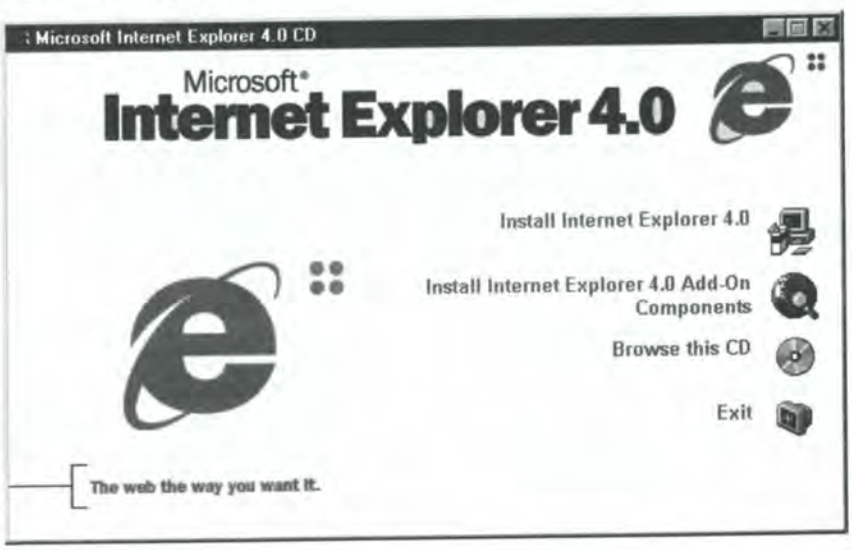
Setup will go through trust, dependency, and CheckSum checks before downloading the .cab files and installing their contents. The .cab files are not downloaded to the Setup folder; instead they are stored temporarily in C:\Windows\Msdownld.tmp\Ase000.tmp.

For troubleshooting information about installing Internet Explorer components, see "Invalid Page Fault When Installing Additional Components" in Appendix B.

Adding Components from a Compact Disc

▶ **To add components from a compact disc**

1. Insert the Internet Explorer 4 compact disc into the drive, and the Autorun feature will display this screen:



2. Click the **Install Internet Explorer 4.0 Add-On Components** hyperlink, which will open the add-on page from the compact disc:

   [CD:]\EN\DownLoad\AddOnxx.htm

All add-on components are available on the compact disc, including the script debugger.

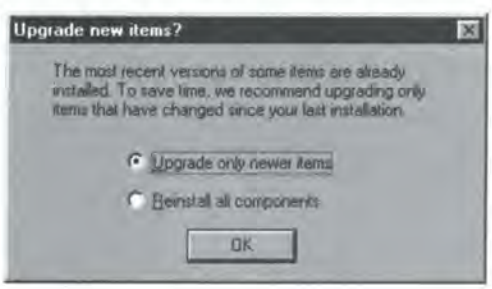
Reinstalling Components

If Internet Explorer 4 has been uninstalled or any of the components from its original installation have been removed, they can be reinstalled from the Setup folder—as long as the Setup folder has not been removed. When Setup is run in maintenance mode, it compares the installed components to the .cab files in the Setup folder. (If the Setup folder has been removed, you'll need to reinstall Internet Explorer 4 by repeating the steps of downloading from the Internet, as explained in Chapter 2, "Phase 1: Active Setup." Or you can follow the instructions for adding new components, as explained in the previous section.)

To reinstall Internet Explorer 4 or any of its components using your local Setup folder, run IE4Setup.exe from the Setup folder. Each time IE4Setup.exe is run in maintenance mode (after initial installation), you'll see the Welcome Screen, agree to the License Agreement, select an installation option, and confirm the Destination folder, as detailed in Chapter 2, "Phase 1: Active Setup."

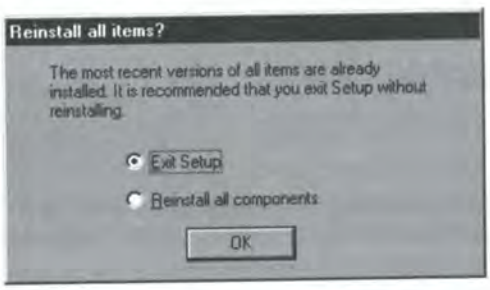
When re-running IE4Setup.exe from the Setup folder, you are limited to the same installation option (or a subset thereof) that was selected during the original download. Cabinet (.cab) files for components added from either the Internet or compact disc add-on page will not exist in the Setup folder; they are not included in a local reinstallation.

If the Setup folder contains a .cab file for a component that is not currently installed, then the **Upgrade new items?** dialog box is displayed.



To reinstall components that have previously been removed, click the **Upgrade only newer items** option. This will install any .cab files in the Setup folder not already installed.

If the Setup folder contains no uninstalled .cab files, the **Reinstall all items?** message is displayed.



If, during a local reinstallation, all the components for the selected installation option are not installed and one of the .cab files needed to install a required component is missing from the Setup folder, this message is displayed:

Following those instructions, when IE4Setup.exe is run a second time, you will see the Welcome screen, the End-User License Agreement, the Smart Recovery screen, and, if you choose **Recover**, you will be prompted to choose a download site.

After the Preparing Setup screen is displayed, the Downloading Internet Explorer screen will show status information as the missing .cab files for the selected option are downloaded.

Using IExtract.exe

IExtract.exe is an MS-DOS-mode utility, found in the C:\Windows\Command folder, used to extract files from the IE4bak.dat file. It has the following syntax:

iextract [/w] [/l *folder*] *datafile* [*filename*]

- **/w.** Warn before overwriting a file. The default is to overwrite existing files.
- **/l.** Save extracted files to the following location. The default is current folder.
- ***datafile.*** Full qualified path to the backup information .dat file (required).
- ***filename.*** Name of the file to extract from the backup information file. You can use multiple file names (separated by blanks). The default is all files.

Here is an example:

```
iextract /w /l c:\windows\temp c:\windows\ie4bak.dat schannel.dll
```

This example will extract Schannel.dll from the IE4bak.dat file and place it in the Temp folder.

The Iextract.exe utility is particularly useful in troubleshooting problems resulting from a mismatch of shell or OLE files.

Adding and Removing the Windows Desktop Update

▶ **To add or remove the Windows Desktop Update**

1. In the Control Panel, double-click the Add/Remove Programs icon.
2. In the software list, click **Microsoft Internet Explorer 4.0**, and then click the **Add/Remove** button. The resulting dialog box will present you with three options. The third option allows you to add or remove the Windows Desktop Update (integrated browser). This dialog box changes depending on whether the Windows Desktop Update is installed or not:

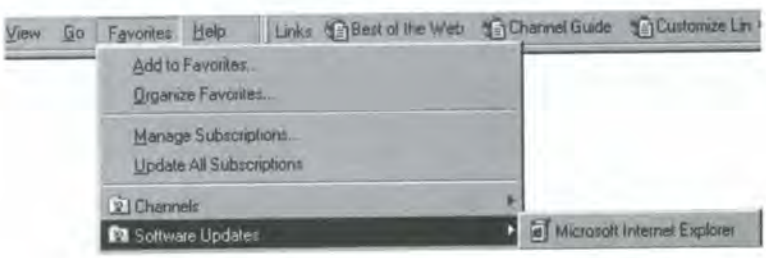
For more information about the Windows Desktop Update, see Part 3, "True Web Integration."

The Integrated Browser.dat and Integrated Browser.ini are required for removing the Windows Desktop Update, and they are created when the Windows Desktop Update is added. During the uninstallation of the shell, the original System.1st is extracted from Integrated Browser.dat and overwrites C:\System.1st. This step will not take place if the creation of the new System.1st was skipped because of a low amount of disk space.

SoftBoot will attempt to simulate a reboot when the shell is being changed. For more information about System.1st and SoftBoot, see "Installing Internet Explorer 4 on Your Computer" in Chapter 3.

Updating Internet Explorer 4

Internet Explorer 4 comes with a Software Updates channel already listed in the Favorites menu. A subscription to this channel is already set up, so it's easy to check the Internet to see if a newer version of Internet Explorer has been posted for download. (For more information about Active Channels, see Chapter 16, "Managed Webcasting.")



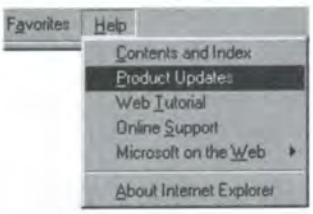
The Internet Explorer Update Channel (linked to IEUpdate.cdf,) can be used to initiate the install of the newer version or to schedule an automatic upgrade.

By default, you must go to the Update Channel to see if there is a newer version of Internet Explorer available, but the subscription can easily be changed from manual to automatic, which will give you automatic notification when a newer version has been made available. Then, when you sign on to Internet Explorer, if a new update is available, you'll be presented with a **Software Update Available** dialog box.

On this dialog box, the **Details** button reveals details about the currently installed version. The **Update Now** button will automatically connect to the Internet Explorer Update Channel, shown above. The **Don't Update** button allows you to postpone the update.

Using Product Update

The Product Update feature is not related to the Internet Explorer Update Channel. It gives users an easy way to navigate to the Microsoft Download Area Web site and the Download Components page to compare installed components to those available. (For more information about the **Download Components** page, see "Adding Components from the Internet" earlier in this chapter.)



This can be helpful if you have deleted your subscription to the Update Channel or are between automated update checks.

CHAPTER 7

Uninstalling Internet Explorer

In this chapter

Removing Internet Explorer Core Files 74
Understanding the Uninstall Process 76
Using Maintenance Mode ACME Setup 78
Errors 79
Files Used by Uninstall 79
Registry Entries Used by Uninstall 80
Deleting Uninstall Files 81
Understanding the Uninstall Log 82
Understanding Emergency Uninstall: IERemove.exe 82

Removing Internet Explorer Core Files

As explained in the previous chapter, there are two **Add/Remove** dialog boxes used by Internet Explorer, each with distinctly different functionality. The first dialog box is the traditional **Add/Remove Programs** applet in the Control Panel. This is used to uninstall only Internet Explorer add-on components with corresponding .inf files containing uninstall information. This is not used to uninstall the Internet Explorer core files. (For information on removing add-on components, see Chapter 6, "Adding and Removing Components After Setup.")

The second dialog box appears when **Microsoft Internet Explorer 4.0** is selected from the software list in the **Add/Remove Programs** dialog box, and the **Add/Remove** button is clicked. This dialog box is displayed by Setupwbv.dll, which knows how to invoke Maintenance-mode ACME, call an .inf file uninstallation, or launch Internet Explorer to download new components from the Web.

▶ **To uninstall Internet Explorer**

1. In the Control Panel, double-click the **Add/Remove Programs** icon.
2. In the software list, click **Microsoft Internet Explorer 4.0**, and then click the **Add/Remove** button. The resulting dialog box will present you with three options. The first of the three options is used to uninstall core files and any add-on components you select.

3. Select **Uninstall Internet Explorer 4.0 and all its components**, and then click the **Advanced** button.

4. In the resulting dialog box, select **Uninstall Internet Explorer 4.0 and the components selected below**, and click the check boxes next to the components you want to uninstall.

For each component, the registry contains a minimum version and an Internet Explorer build number with which the component will work. For example, one component may work if Internet Explorer 3, build 1155, or greater, is installed. Another may only work if build 1300, or greater, is installed. Some only work if Internet Explorer 4 is installed. Setup recorded the previous browser version in the registry when Internet Explorer was installed. That value is compared to the version requirement of each component. If the previous version is less than the required version of the component, the component's box will be checked.

Warning In the **Advanced Uninstall** dialog box, if you select **Delete the backup information for Internet Explorer 4.0**, you will delete the IE4bak.dat, IE4Bak.ini, Integrated Browser.dat, and Integrated Browser.ini. files, which contain important uninstall information that you will need if you decide to reinstall and uninstall Internet Explorer again. It is recommended that you not delete these files. (For more information, see "Deleting Uninstall Files" in the next section.)

For troubleshooting information about removing Internet Explorer 4, see Appendix B.

Understanding the Uninstall Process

ACME Setup is used to uninstall Internet Explorer 4 core files, whereas the Internet Explorer add-on components use the standard .inf method of uninstallation, as per Windows 95 and Windows NT compatibility specifications.

ACME Setup relies on the following files:

- C:\Program Files\Internet Explorer\Setup\Setup.stf
- C:\Windows\IE4bak.dat
- C:\Windows\IE4bak.ini
- C:\Windows\IE4RegUn
- C:\Program Files\Internet Explorer\Uninstall\AINF0000
- C:\Program Files\Internet Explorer\Uninstall\Integrated Browser.dat (if the Integrated Browser is installed)
- C:\Program Files\Internet Explorer\Uninstall\Integrated Browser.ini (if the Integrated Browser is installed)

If, for some reason the .dat and .ini files are damaged or missing, the user may be able to get them from another system and successfully complete the uninstallation. Keep in mind that both systems must be running the same operating system, must have chosen the same Internet Explorer installation option, and must have had the same browser installed prior to installing Internet Explorer; and that the install path for each browser must be the same.

The registry information needed to uninstall is stored in the AINF0000 files. Installation instructions for each component includes the RegSave function, instructing Advpack.dll to move data from the registry into the AINF0000 files. During uninstallation the RegSaveRestore function is called, the AINF0000 files are loaded into the registry as hives, and the registry is updated with the original data. Similarly, an IE4RegUn file is created under the Windows folder, which contains backed-up registry information for the Internet Explorer core.

You should remove Internet Explorer 4 before performing any of the following actions:

- Installing an earlier version of Internet Explorer.
- Restoring a registry (System.dat or User.dat) that was created before you installed Internet Explorer 4. (This applies to Windows 95 operating systems only.)
- Using a Windows NT Emergency Repair Disk (ERD) that you created before you installed Internet Explorer 4. (This applies to Windows NT operating systems only.)
- Upgrading to a later release of Internet Explorer 4.
- Reinstalling your operating system (Windows 95 or Windows NT).
- Installing or uninstalling operating system upgrades (for example, Windows 95 Service Pack 1).

Uninstalling does not support the following versions or situations:

- The beta 2 version of Internet Explorer 4 installed over a previous version of Internet Explorer 4.
- The final version of Internet Explorer 4 installed over a previous version of Internet Explorer 4, other than beta 2.
- Internet Explorer 4 installed under multiple operating-system installations (Windows 95, OSR2, or Windows NT 4.0) on the same logical drive. If you need to run Internet Explorer in multiple operating systems, you must install each operating system to a separate logical drive letter.
- You have removed the Internet Explorer 4 backup files. If you removed these files, you cannot uninstall Internet Explorer 4 without reinstalling Windows to a different folder.

Uninstalling does not completely return your computer to its previous state. It leaves behind some files, registry entries, and shortcuts. Additionally, some user settings are not restored with the previous version of Internet explorer. System files such as Setupapi.dll are left behind to support other .inf file installs, or OLEAUT32 to prevent breaking OLE functionality in Office. For a complete list of files that uninstalling does not delete from the system, see Appendix A.

Because of the larger number of changes made to the system by Setup, a manual uninstall would require many, many steps and a great deal of time. Attempting a manual uninstall is not recommended.

For the latest list of issues you need to be aware of while running Setup and while uninstalling, check the Internet Explorer 4 Readme file.

Using Maintenance Mode ACME Setup

Uninstalling relies on the existence of this registry key:

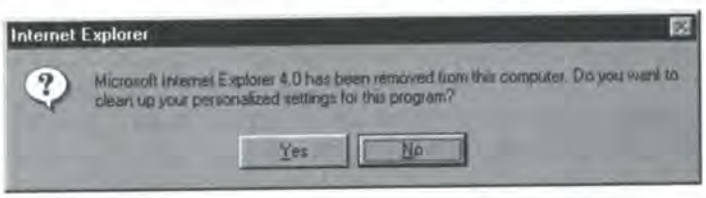
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\IE40 "C:\Program Files\Internet Explorer\Setup\SETUP.EXE" /g C:\WINDOWS\IE4 Uninstall Log.txt

This value launches maintenance-mode ACME Setup with hands-off operation.

Uninstalling also relies on the following back up files: IE4bak.dat, IE4bak.ini, Integrated Browser.dat and Integrated Browser.ini. If these files are missing, uninstall will fail. Note that the Integrated Browser files are only needed if the Windows Desktop Update is installed at the time of uninstallation.

After maintenance mode, ACME Setup uninstalls Internet Explorer. The Setup folder will remain on the system so that Internet Explorer may be installed again without requiring connectivity to the Internet.

If the **User Profiles** option has been enabled, Setup will ask if you want to remove Internet Explorer settings from your user profile (User.dat).



You can also run maintenance-mode Setup manually, by running Setup.exe from the Setup folder in your Internet Explorer installation folder. To be in hands-off mode or to create an uninstall log file (IE4unin.txt) you will need to specify the following command line:

```
C:\program files\internet explorer\setup\setup /g C:\ie4unin.txt
```

Or, if you don't have long file-name support:

```
C:\progra~\intern~1\setup\setup /g C:\ie4unin.txt
```

Errors

If Shell32.dll and Explorer.exe are not successfully extracted from the Integrated Browser.dat, the newer versions of those files will still load, and you'll be presented with this message:

The SHLWAPI.dll made a bad call to Explorer.

To fix this, extract the original versions of Shell32.dll and Explorer.exe from your Windows CD-ROM and copy them to the system, overwriting the newer versions.

Files Used by Uninstall

There are several key files needed for a proper uninstall:

Setup.exe Runs the ACME Setup program.

Setup.stf Contains instructions for the ACME portion of Setup or uninstallation and calls IE4.dll.

IE4.dll Custom Action Setup engine used for logging and for custom file or registry additions, changes, or removals.

IE4.inf Contains instructions for IE4.dll.

Advpack.dll Used for Internet Explorer Express cabpack extraction, and for .inf file parsing or execution while uninstalling programs installed with Internet Explorer Express. Used in combination with W9INF32.dll and W9INF16.dll on Windows 95 systems.

Setupapi.dll Originally a Windows NT file, Internet Explorer uses this file in both Windows 95 and Windows NT for locating .cab file and .inf file installations.

Integrated Browser.dat Contains Integrated Browser modes files, including the previous versions of Explorer.exe and Shell32.dll, and is used for installing or uninstalling the integrated browser mode. This file is called Integrated Browser NT.dat on the Windows NT operating system.

Integrated Browser.ini Contains information about Integrated Browser.dat.

IE4bak.dat Contains many of the files needed to uninstall Internet Explorer.

IE4bak.ini Contains information about IE4bak.dat.

Setupwbv.dll Used to control mode switching by parsing IE4Shell.inf and calling Advpack.dll for IE4shell.cab extraction.

IE4shell.cab Contains new shell components and is used in mode switching and uninstall.

IE4shell.inf Contains instructions for mode switching. Also contains the contents of IE4shell.cab.

IE4data.inf Contains instructions for uninstalling data components.

IE4data.cab Backup file for data components used during installation and uninstallation, and that rely on Internet Explorer (Task Scheduler, ActiveMovie, Multimedia Controls, etc.) This file is called by Advpack.dll.

There are several other uninstall .inf files; one exists for each external component. All are called using Advpack.dll.

Registry Entries Used by Uninstall

There are several registry keys that need to be present and populated for proper uninstalling:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
contains entries for everything that shows up under Control Panel **Add/Remove Programs** and some that do not. Each item listed has its own method (command line) for launching an uninstall. Note that entries containing a **QuietUninstall** key refer to external components (.inf files) called during an Internet Explorer uninstallation.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\IE4\Options
contains location and other info for IE4bak.dat and 4bak.ini.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IE4\SETUP
contains some miscellaneous file location information for various logs, etc.

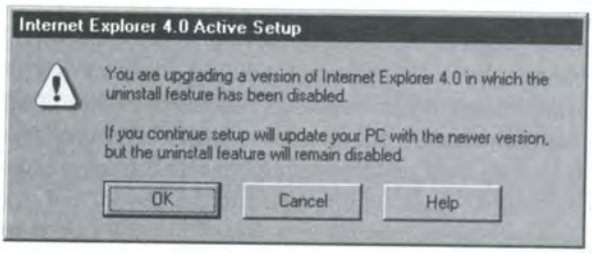
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IE4\SETUP\History
contains information concerning previous browser information.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IE4\SETUP\IE4RegBackup
contains backup registry information saved by Setup. The 0 section contains the actual data, and the 0.map contains table-of-contents type tracking of the information in 0. In cases of reinstallation and upgrade, the *.map keys (starting with 3.map and incrementing by 1 per reinstallation) are created to track differences.

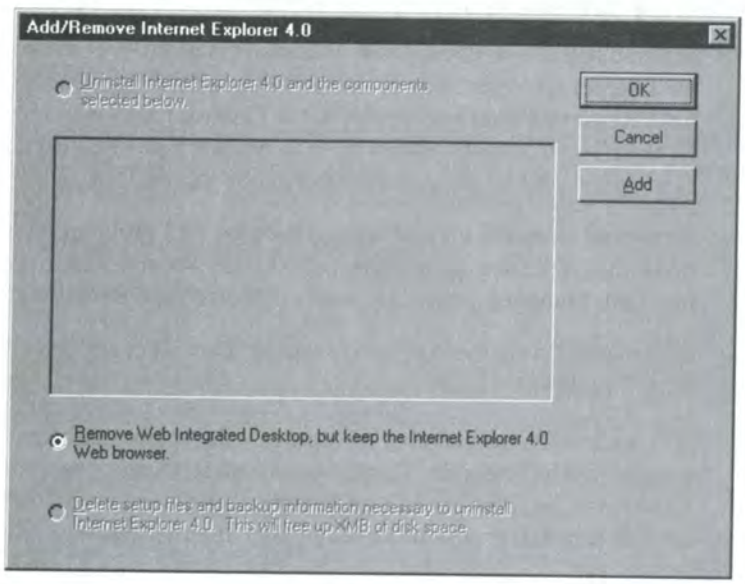
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Advanced INF Setup\Integrated Browser\RegBackup
contains 0 and 0.map sections used to back up and track registry information used to uninstall the Integrated Browser. Map files are not created in this section, and the 0 and 0.map sections should be empty if Integrated Browser mode is not enabled.

Deleting Uninstall Files

If you delete the primary uninstall information (IE4bak.dat, IE4bak.ini, Integrated Browser.dat, Integrated Browser.ini), you will see this message if you reinstall:



After the reinstallation is complete, you may have options disabled in the **Advanced Add/Remove** dialog box.



Understanding the Uninstall Log

The most important tool for troubleshooting an uninstallation is the uninstall log, called IE4 Uninstall Log.txt, located in the Windows folder. This log is very comprehensive. It covers the entire uninstallation process sequentially from beginning to end, including every file addition or removal; every registry addition, change, or removal; and any dialog boxes shown to the user. The log is divided into Passes which denote the different phases of an uninstall. Entries in the log also have an Object number—which corresponds to the line entry in Setup.stf. Lines without an Object number result from custom actions specific to Internet Explorer 4—and are contained in the IE4.inf file or in an .inf file from an external component uninstallation. The Uninstall Log is not created when IERemove.exe is run.

Understanding Emergency Uninstall: IERemove.exe

IERemove.exe is not a separate uninstall program. It is a forced removal of the Internet Explorer core. It should be used only in circumstances where the **Add/Remove Programs** applet in the Control Panel is unable to uninstall Internet Explorer, or if you absolutely have to replace a dead registry and have already tried System.1st or an updated Emergency Repair Disk (ERD).

A no-boot situation may be caused because of a problem with the new shell. To make the shell files compatible with the shell registry settings, boot to safe mode, run Task Manager, press CTRL+ESC, and then type the following:

```
c:\progra~1\intern~1\setup\ieremove /i:c:\progra~1\intern~1\
uninst~1\integr~1.dat
```

(Do not use the **/n:** switch.) IERemove will replace only the shell files, and the system will be bootable. Then you can run IE4Setup.exe out of your Internet Explorer 4 Setup folder (if you have one) to solve any problems. You will still need to reinstall programs installed after Internet Explorer was installed.

IERemove attempts to restore the system to the state it was in prior to installing Internet Explorer. This includes restoring the previously used browser. IERemove only replaces core files. It will not uninstall any additional components, even if they rely on the Internet Explorer core to function. Internet Explorer files with no original counterpart stored in the .dat files are not removed. Also, none of the folder structure associated with Internet Explorer will be removed, and **Start** menu items will remain. If the user had Internet Explorer 3 installed previous to Internet Explorer 4, IERemove will restore Internet Explorer 3 so that it works, although some settings may be lost.

IERemove.exe is found in the C:\Program Files\Internet Explorer\Setup folder. It is a Windows program. When it is run, an uninstall log.txt is created. SetupWbV.dll is not removed by IERemove, so the **Advanced Add/Remove** dialog box should still be available to users after they are able to boot.

IERemove relies on the backup files and will fail if the backup files are not present. The command-line switches listed below are not required. IERemove looks to the following registry keys for the locations of IE4bak.dat and its corresponding .ini file, and the Integrated Browser.dat and its corresponding .ini file, respectively:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\IE4\options

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Advanced INF Setup\Integrated Browser

If you get an error such as the following when running IERemove.exe, it is likely that the registry keys mentioned above are absent:

Setup could only remove the shell integration option. Internet Explorer 4.0 was not completely removed.

To resolve this problem, use the command-line switches to specify the locations of the .dat and .ini files. Also, you can force IERemove to remove only the shell or only the core by using only one of the following switches:

- **/i:"*path*\integrated browser.dat"**
- **/n:"*path*\ie4bak.dat"**

Example:

```
ieremove /i:"C:\program files\uninstall information\
integrated browser nt\integrated browser nt.dat"
/n:"C:\windows\ie4bak.dat"
```

Or, if you don't have long file name support:

```
C:\progra~1\intern~1\setup\ieremove /i:C:\progra~1\intern~1\
uninst~1\integr~1.dat /n:C:\windows\ie4bak.dat
```

If IERemove runs on a computer that did not have Internet Explorer 3 previously installed, the user may not be able to successfully install Internet Explorer 3 because of version conflicts with Internet Explorer files left on the system. These files need to be renamed.

To find procedures for using IERemove.exe to uninstall Internet Explorer 4 for Windows 95 and Internet Explorer 4 for Windows NT, see Appendix B.

CHAPTER 8

Configuring Connection Settings

In this chapter

Using the Internet Connection Wizard 86
Choosing Manual Setup: Using an Existing Internet Account 88
Choosing Automatic Setup: Creating a New Internet Account 89
Using Manual Configuration 93
Using the Proxy Bypass List 95
Errors 96

Using the Internet Connection Wizard

The first time Internet Explorer is launched after Setup, the Internet Connection Wizard (ICW) runs. The ICW is a fully automated process that takes a user through the steps of setting up a connection to the Internet.

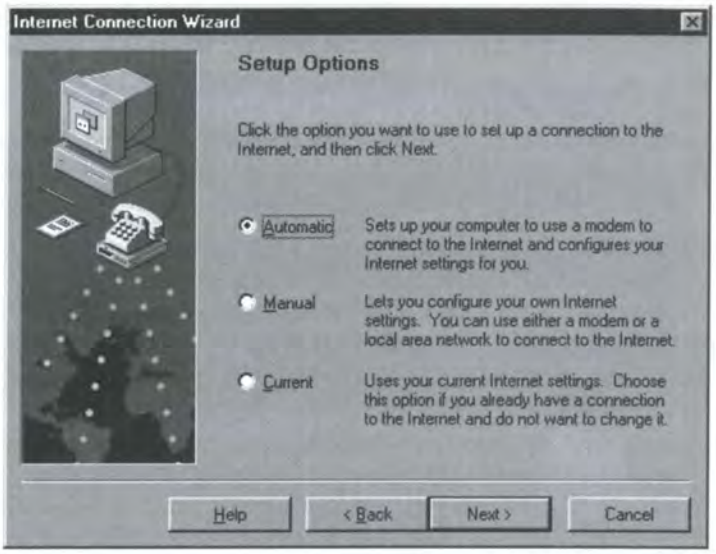
The following registry key is set to 0 when Internet Explorer is installed and indicates that the ICW has not yet run:

HKEY_CURRENT_USER\Software\Microsoft\Ineternet Connection Wizard\"Completed"="00 00 00 00"

After the ICW has completed, it sets this registry key to 1:

HKEY_CURRENT_USER\Software\Microsoft\Ineternet Connection Wizard\"Completed"="01 00 00 00"

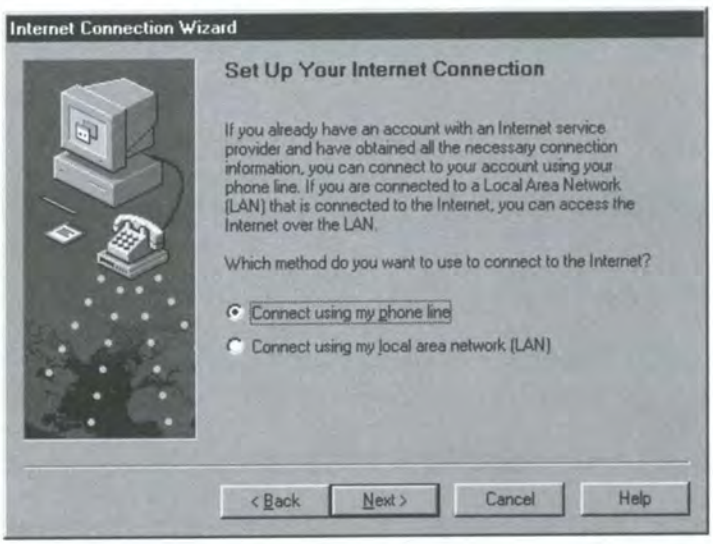
The ICW can also be launched from the **Start** menu or by selecting **Internet Options** from the **View** menu and clicking the **Connection** tab. The Internet Connection Wizard steps are described below.



The first (automatic) option and second (manual) option lead the user through a similar process. The third option does nothing but take you to the end of the wizard; it accepts your current connection settings and does not check to see if the existing configuration is valid.

Choosing Manual Setup: Using an Existing Internet Account

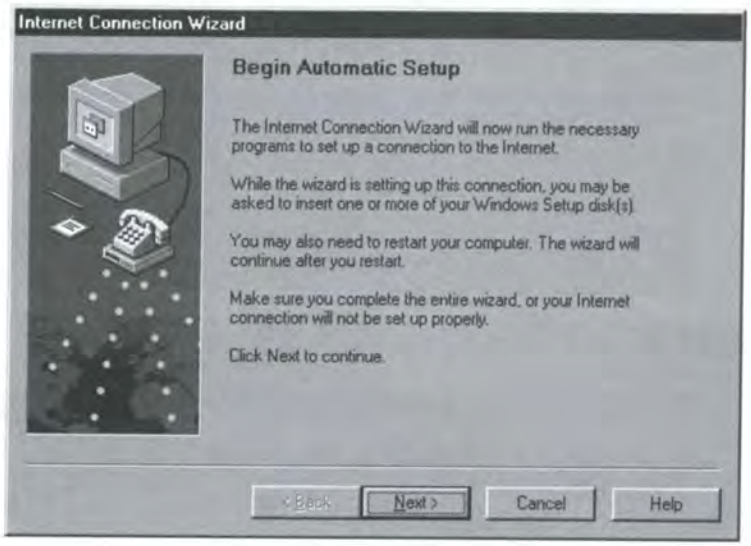
The second option leads the user through the following screens. Users can choose a phone line or a Local Area Network, and they can choose whether to use Windows Messaging, an Internet mail client:



Both the manual and automatic options will install (or reinstall) Windows 95 networking drivers from the Windows 95 compact disc. No detection occurs to verify that these drives are already installed, but version conflict warnings will be displayed. TCP/IP and Dial-Up Networking (DUN) will be installed (or reinstalled), and the Install New Modem Wizard will run. The ICW checks that your modem is set up properly. If it's not, the Install New Modem Wizard is launched automatically so that you can configure a modem and dial-up settings.

Choosing Automatic Setup: Creating a New Internet Account

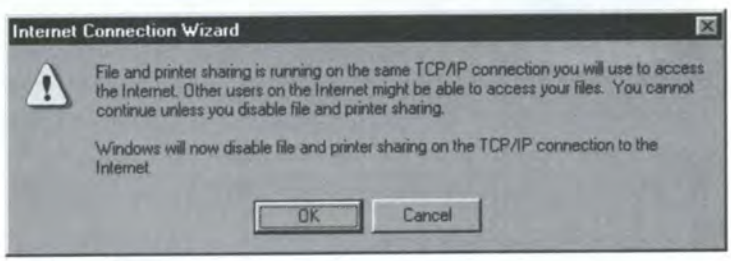
In addition to configuring the modem and a connection to an ISP, as described above, this option lets users create a new account with an ISP. However, the user must be using a modem; a LAN connection is not an option.



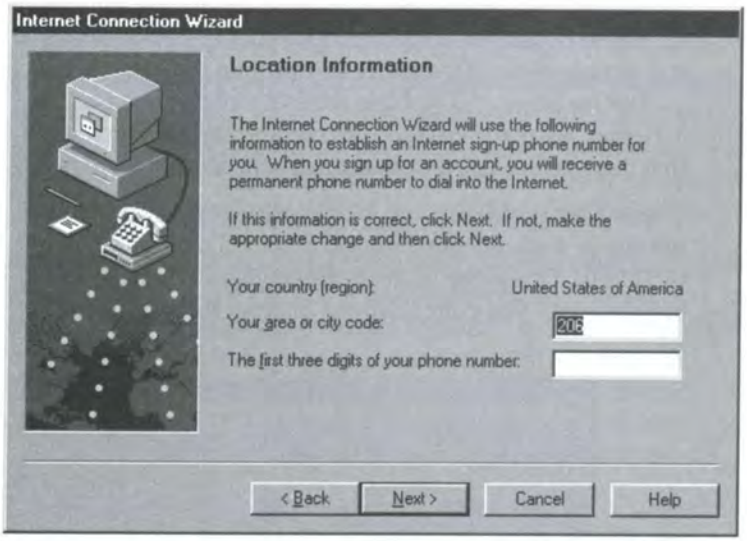
The ICW checks for an installed modem. If no modem is installed, it will launch the Modem Control Panel to begin modem detection. If not installed, TCP/IP and DUN will be installed. The user will be prompted for the Windows 95 compact disc.

If new drivers need to be added, the system is automatically rebooted after they are installed.

Users may receive this standard security warning:



The Begin Automatic Setup screen is displayed, and then location information is gathered. (At this point, the user cannot use ALT+TAB to turn away from the wizard.)



The browser then dials out to a Microsoft Referral Server.

The browser is opened in kiosk mode to display a list of ISPs in your area that offer immediate accounts. The **Sign Me Up** button configures a DUN connectoid to the corresponding ISP.

Once you select an ISP, the ICW will attempt to connect to one of the ISP's servers to obtain further information for the sign-up process. If you have problems connecting to the ISP's server or finishing the sign-up after connecting, contact the ISP for support. To obtain the ISP's phone number, click the **More Info** hyperlink for the ISP on the Internet Referral Servers list of ISPs. (This is the first page you see after connecting to the Referral Server.)

If you have problems with the automatic option, Internet Explorer checks that the modem is working correctly and that the Dial-Up Adapter is installed and bound to TCP/IP. Otherwise, it treats the problem as a typical Dial-Up Networking problem until you connect to the Internet Referral Server.

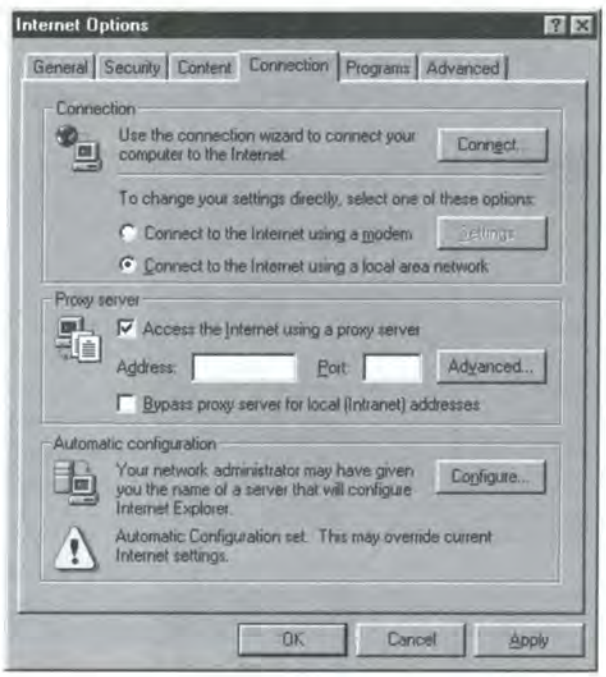
If you have trouble connecting or maintaining a connection to the Referral Server, Internet Explorer tries to reduce the modem speed and configure it for a standard modem. If this fails, there may be a problem with the Referral Server, and you should try using the ICW at a later date.

After a user has signed up for an Internet account, the ICW will start a Setup program to install and configure the Internet software required by your service provider. For product support phone numbers in your area, refer to the file C:\Program Files\Internet Explorer\Support.txt.

Using Manual Configuration

A proxy server acts as an intermediary between your computer and the Internet. It is usually only used in a corporate intranet situation when users are connected to a LAN. It also acts as a security barrier between your internal network and the Internet. This helps secure important data, because it keeps other people on the Internet from gaining access to confidential information on your internal network. In addition, administrators can use this for proxy load balancing and site blocking. Proxies are becoming more advanced in their ability to reduce network traffic by caching content that is frequently requested by the browsers they serve.

Internet Explorer 4 includes a new dialog box with the most commonly used proxy settings on one tab and more detailed proxy settings on a separate tab. On the **View** menu, select **Internet Options**, and click the **Connection** tab.



In the **Address** box, enter the name or IP address of the proxy server you want to handle requests from your browser. In the **Port** box, enter a port specified for passing Internet protocols. Typically, port 80, the reserved port for HTTP, is used.

A check box is available to bypass the proxy when connecting to intranet Web servers. This prevents requests for addresses with no period in the hostname string from being sent to the proxy server.

The corresponding registry settings can be found in:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings

"ProxyEnable"="01 00 00 00"

"ProxyServer"="*data*"

"ProxyOverride"="*local*"

Clicking the **Advanced** button on the **Connection** tab of the **Internet Options** dialog box brings up the **Proxy Settings** dialog box.

By default, the **Use the same proxy server for all protocols** check box is selected. All Internet protocols will use the same port. As soon as the user clicks the **Advanced** button on the previous dialog box, this setting is disabled, allowing the user to specify a separate proxy server and port number for each individual Internet protocol. Selecting the check box makes all the other entries unavailable and copies the proxy information in the HTTP setting into the other protocol settings. Selecting the check box also hides the information in the **Socks** setting.

The **Secure** settings are for HTTPS requests. Enter the proxy location and port number for the appropriate protocols.

Proxy locations that do not begin with a protocol (such as **http://** or **ftp://**) are assumed to be a CERN-type HTTP proxy. For example, the entry **proxy** would be treated the same as the entry **http://proxy**. For FTP gateways, such as the TIS FTP gateway and the WinGate Win95 Modem FTP proxy, the proxy should be listed with the protocol **ftp://** in front of the proxy name. For example, an FTP gateway FTP proxy should be entered as **ftp://ftpproxy**.

The proxy bypass list in the **Exceptions** section allows users to specify addresses that will bypass the proxy server and be accessed directly.

Using the Proxy Bypass List

Some network requests need to bypass the proxy. The most common reason to bypass the proxy is for local (intranet) addresses. Generally, these addresses do not contain periods in them. By selecting the **Bypass proxy server for local (intranet) addresses** check box on the **Connection** tab of the **Internet Options** dialog box, all addresses without a period will bypass the proxy and be resolved directly. To bypass more complex addresses, click the **Advanced** button on the **Connection** tab of the **Internet Options** dialog box, and use the **Proxy Settings** dialog box to enter the addresses in the **Do not use proxy server for addresses beginning with:** box.

A proxy bypass entry may begin with a protocol type. These may be **http://**, **https://**, **ftp://**, or **gopher://**. If a protocol type is used, the exception entry only applies to requests for that protocol. Note that the protocol value is case insensitive. Multiple entries should be separated by a semicolon (**;**).

If no protocol is specified, any request using the address will be bypassed. If a protocol is specified, requests with the address will only be bypassed if they are of the indicated protocol type. As with the protocol type, address entries are case insensitive. If a port number was added, the request is only processed if all previous requirements are met and the request uses the specified port number.

The **Exceptions** section of the **Proxy Settings** dialog box allows a wildcard (*) to be used in the place of zero or more characters. The following list contains examples showing how to use wildcards:

- To bypass servers, enter a wildcard at the beginning of an Internet address, IP address, or domain name with a common ending. For example, use ***.example.com** to bypass any entries ending in.example.com (such as some.example.com and www.example.com).
- To bypass servers, enter a wildcard in the middle of an Internet address, IP address, or domain name with a common beginning and ending. For example, the entry **www.*.com** matches any entry that starts with www and ends with com (such as www.someplace.com, www.almost.anywhere.com, and so on).
- To bypass servers, enter a wildcard at the ending of an Internet address, IP address, or domain name with a common beginning. For example, use **www.someplace.*** to bypass any entries that began with www.someplace. (such as www.someplace.com, www.someplace.org, and www.someplace.else.com).
- To bypass addresses with similar patterns, use multiple wildcards. For example, use **123.1*.66.*** to bypass addresses such as 123.144.66.12, 123.133.66.15, and 123.187.66.13.

Although wildcards are powerful, they must be used carefully. For example, the entry **www.*.com** will cause Internet Explorer to bypass the proxy for most Web sites.

If you need to bypass the proxy for a local domain, try using ***.domain.com**. This will not use the proxy for any computer name ending in .domain.com. You can use the wildcard for any part of the name.

Errors

Some features such as RealAudio and Secure Sockets Layer (SSL) may not work over a proxy server. Many of these features require special configurations on the proxy server.

PART 2

The Internet Explorer Browser

A powerful browser will help your organization reap the important benefits the Web has to offer. As the Internet increases in utility and complexity, more and more computer users have come to rely on Microsoft® Internet Explorer to help them make the most of it. Internet Explorer 4 contains a variety of improvements that make browsing the Web easier, faster, and more fun than ever before.

Ease-of-use innovations and enhanced personalization features in Internet Explorer 4 truly give users the Web on their terms, or "the Web the way you want it." Internet Explorer 4 also offers key underlying technologies for Web developers. Part 2 describes the new features, functionality, and technologies of the Internet Explorer 4 Web browser and explains how to use this browser to create the best Internet and intranet experience.

Chapter 9: Browser Features and Functionality

Having a great experience on the Web starts with the browser. Internet Explorer 4 contains a variety of improvements that make browsing the Internet and local intranets easier, faster, and more productive. For example, the Address bar in Internet Explorer automatically completes addresses for you, based on sites you've already visited. It also adds prefixes and suffixes to Internet addresses. Explorer bars for Search, Favorites, History, and Active Channels™, make it easier to find the sites you need, visit your favorite pages, return to the places you've been, and access the information you use most often. Administrators can customize Explorer bars and even create their own. Internet Explorer 4 also introduces the ability to browse offline. In addition, Internet Explorer 4 supports an open architecture that allows easy integration with third-party programs, offering users a flexible environment without loss of integration benefits.

Chapter 10: Browser Performance

As increased traffic on the Web continues to reduce browser responsiveness, Internet Explorer 4 has introduced enhanced features and implemented new technologies that improve the browser's ability to access Web sites more quickly. Internet Explorer 4 supports key technologies, including the latest HTML and HTTP standards, to make your browsing experience the fastest and most efficient on the Internet.

Chapter 11: The WebBrowser Control

The Internet Explorer WebBrowser Control allows developers to use the best of the Internet Explorer feature set in their own programs, offering a rich set of functionality in simple, reusable, ActiveX™ controls.

CHAPTER 9

Browser Features and Functionality

In this chapter

Ease of Use and Personalization Features 100
- AutoComplete 100
- AutoDomain Searching 102
- Navigation History on Back and Forward Buttons 103
- Explorer Bars 104
- Full Screen Mode 105
- Improved Favorites 105
- Movable Menu Bar 106
- Improved Links Bar 106
- Friendly User Feedback 107
- Printing Enhancements 107

Using Explorer Bars 108
- Understanding the Search Bar 108
- Configuring and Customizing the Search Bar 109
  - Creating Your Own Search Bar 112
- Understanding the Favorites Bar 113
- Configuring Favorites 114
- Understanding the History Bar 116
- Configuring the History Bar 117
- Understanding the Channel Bar 118
- Configuring the Channels Bar 118
- Creating a New Explorer Bar 119

Browsing Offline 119
- Understanding Offline Browsing 120
  - Understanding CD Caching 120
- Using Offline Browsing 120

Setting Advanced Browser Options 121

Integrating Third-Party Programs 126

Ease of Use and Personalization Features

The features that distinguish Internet Explorer 4 as the best browser include the following:

- AutoComplete
- AutoDomain Searching
- Navigation history on **Back** and **Forward** buttons
- Explorer bars
- Full Screen mode
- Improved Favorites
- Movable Menu bar
- Improved Links bar
- Friendly user feedback
- Printing enhancements

AutoComplete

The Address bar in Internet Explorer 4 automatically completes addresses for you, based on sites you've already visited. By filling in long, complicated URLs automatically, AutoComplete makes it easier for you to type in URLs and reduces the opportunity for typographical mistakes. You can easily override the suggestions by typing over them. The functionality of AutoComplete will be familiar to many users, because it is based on the IntelliSense® technology implemented in Microsoft Office and is similar, for example, to the AutoFill feature in Microsoft Excel.

AutoComplete works by reading the information stored in the C:\Windows\History folder. This means that the AutoComplete feature becomes more effective over time, as you browse the Web and your History folder fills up with shortcuts. AutoComplete will work only if there are shortcuts in the History folder.

As an extension of the AutoComplete feature, Internet Explorer will offer you suggestions for the completion of partial URLs that you enter in the Address bar. Simply type the first few characters of a URL, then right-click in the Address bar, click **Completions** in the shortcut menu, and Internet Explorer will search your History file and list all the available completions of the URL you started to type.

Internet Explorer 4 also makes it easier to get to Web sites by adding prefixes and suffixes to Internet addresses. Users can forego the common typing of "http://www." and ".com" before and after most URLs; just enter the site name and press CTRL+ENTER to add the prefix and suffix information. Internet Explorer will then navigate to that page.

To get the most of AutoComplete, you'll want to use the following features and shortcuts:

- Add to the string that has been automatically completed by pressing the RIGHT ARROW key and then typing the additional characters.
- Skip over separation characters in URLs (such as // and /) by pressing and holding CTRL, and then pressing the LEFT ARROW or RIGHT ARROW key.
- Search your History file by typing the beginning of an Internet address and right-clicking the Address bar to complete it. A shortcut menu appears; select **Completion** for a list of available URLs that match the address you started to type; press the DOWN ARROW on your keyboard to scroll though the list.
- Type a partial URL, then press CTRL+ENTER to add "http://www." before the entry and ".com" after it. For example, if you type "microsoft" in the Address bar and press CTRL+ENTER, the URL instantly becomes "http://www.microsoft.com/".

AutoDomain Searching

Internet Explorer 4 introduces a new feature called AutoDomain Searching. If you aren't sure of the exact address of a Web site, enter any word in the Address bar and Internet Explorer will search for the site. If it doesn't get a response from the DNS server for the name, it will use different prefixes and suffixes to try to find the site. For example, if you type the word Microsoft into the Address bar, Internet Explorer will search:

- http://www.microsoft.com
- http://www.microsoft.edu
- http://www.microsoft.org
- http://microsoft.com
- http://microsoft.edu
- http://microsoft.org

Using the Registry Editor, you can customize the way Internet Explorer searches for sites.

▶ **To add a new suffix to the default search routine**

1. Open the Registry Editor (type **regedit** in the **Run** dialog box), and go to the following registry key:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\URLTemplate

2. Right-click on any blank area in the right pane, and select **Add**, then **String Value**.
3. The Name of each entry is a numerical value representing the order in which each item will be searched. Type the next number in the sequence, and press **Enter** twice.
4. Type ***%s.your suffix*** as the Value Data (for example, %s.gov).
5. Repeat steps 2 to 4 to add additional suffixes. Notice that each suffix has two entries, one with www. preceding the wildcard, and one without.
6. Click **OK**, quit the Registry Editor and restart Internet Explorer to make the changes take effect.

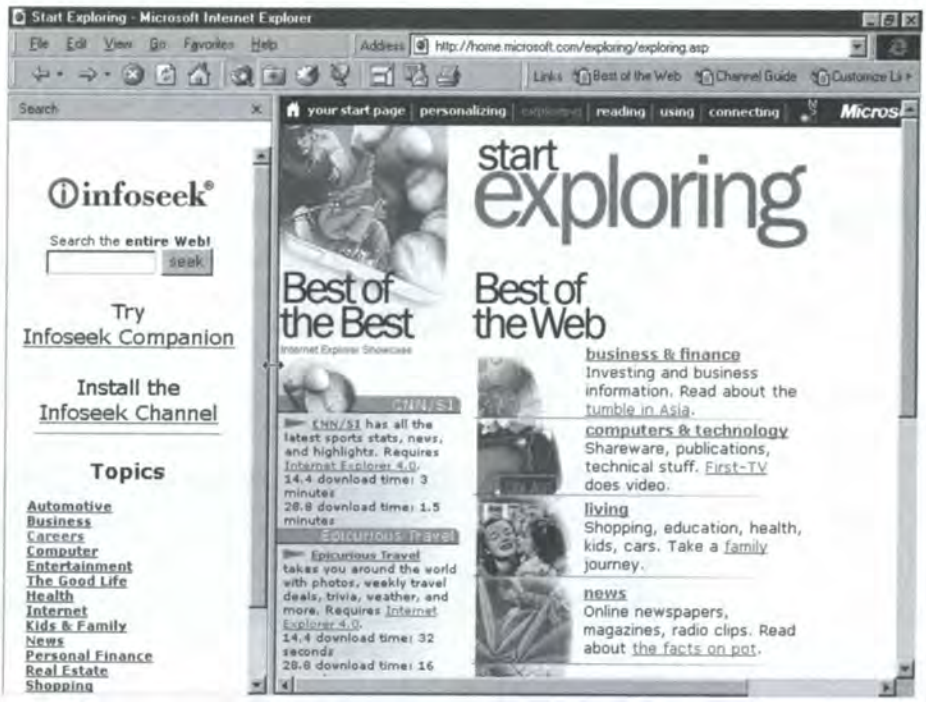
Navigation History on Back and Forward Buttons

In Internet Explorer 4, the **Back** and **Forward** buttons have drop-down menus that display the navigation history of your current session. From that list, you can quickly go to the page you want. Simply click the drop-down arrow next to the **Back** or **Forward** button, or right-click either button, to view the menu. These navigation history menus save you from having to click the **Back** button multiple times to return to a page you have visited.

The **Back** and **Forward** buttons display only the sites visited in the current session (reading from the C:\Windows\History\Today folder). If you need to go further back in time, you can use the History bar to display all Web pages visited in the past 20 days (the default setting). See the following section for more information about the History bar.

Explorer Bars

The Explorer bar is an innovation exclusive to Internet Explorer 4 that makes it dramatically easier to navigate the Web. Explorer bars for Search, Favorites, History, and Active Channels make it easier to find the sites you need, visit your favorite pages, return to the places you've been, and access the information you use most often. When you select an Explorer bar by clicking its button on the Internet Explorer toolbar, a special pane slides in on the left side of Internet Explorer's content window. When you click a link in an Explorer bar, the corresponding page opens in the main content window, so that your links are still visible while the content is displayed. To dismiss the Explorer bar, simply click its close button. For more information, see "Using Explorer Bars," later in this chapter.

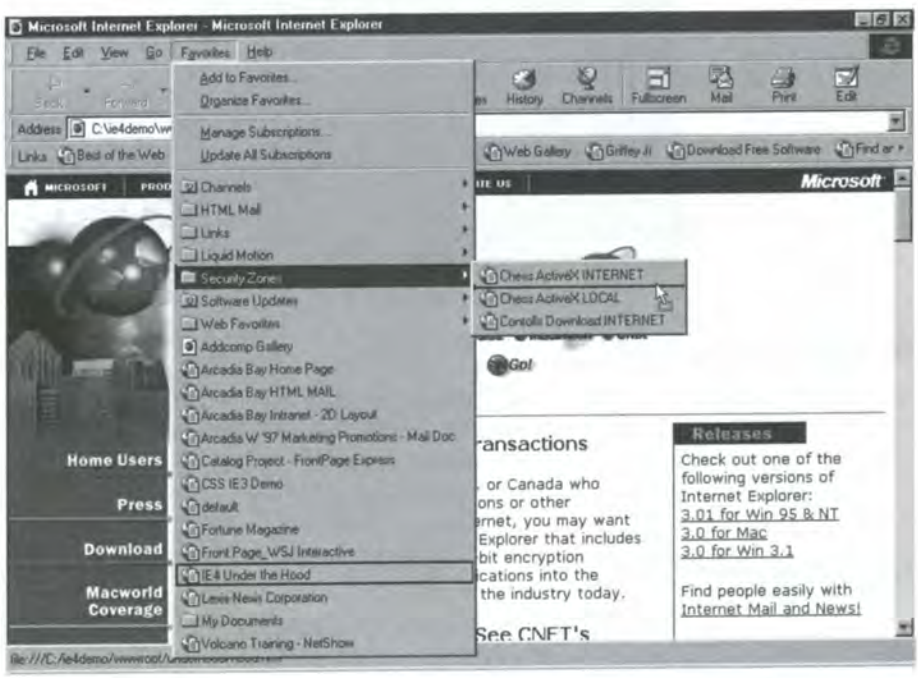


Full Screen Mode

Full Screen mode removes all toolbars, title bars, and scrollbars from the screen so that you can view Web pages in a larger area. You can toggle between Full Screen and normal modes by clicking the **Full Screen** button on the Internet Explorer toolbar.

Improved Favorites

Internet Explorer 3 took a big step forward with the way it handled Favorites. They weren't a list in an .ini file or an HTML file; they were actual shortcuts—just like any other shortcut in the system. This way, you could easily copy and paste them, send them to someone, or arrange them in your file system as you wanted. The **Favorites** menu has been updated in Internet Explorer 4 so that you can rearrange your favorites within the menu.



In addition, the **Favorites** menu has been opened up so that you can interact with the Favorites just as you would with any other file, with right-click functionality such as **Cut**, **Copy**, and **Paste**.

Movable Menu Bar

In Internet Explorer 4, the Menu bar has been implemented as a toolbar, so it can be moved and sized according to user preferences. This new feature helps you keep all your menu commands and toolbars available and still have a large area for browsing.

Improved Links Bar

Quick Links were introduced in Internet Explorer 3, but they had some limitations. For example, you were limited to five Internet shortcuts on the Links bar, and it was too difficult for most users to figure out how to customize them. In Internet Explorer 4, the Links bar allows you to drag URLs from the Address bar to create new Quick Links, and you can also drag buttons within the toolbar to rearrange your Quick Links. In addition, you can drag buttons off the Links bar to the Windows® desktop or other applications. Users can now have as many Quick Links as they want, and the Links bar provides scrolling arrows if your links run off the screen. You can also right-click on the Links bar to use shortcut menus to set properties or delete Quick Links buttons.

In Internet Explorer 3, the Links bar was limited to point to five sites on the Internet as specified in the registry. With Internet Explorer 4, any number of links can now be stored in the Links subfolder, within the Favorites folder. The default path depends on whether the User Profiles feature is turned on:

With User Profiles

C:\Windows\Profiles*username*\Favorites\Links

Without User Profiles

C:\Windows\Favorites\Links

Friendly User Feedback

With most Web browsers, you spend a long time waiting for pages to download, without being certain of the status of the process. Internet Explorer 4 provides improved user feedback, so you always know what the browser is doing. An improved Status bar provides visual feedback about whether you are online or offline, the download status of a specific page, and whether you are connected to the Internet, an intranet, or a local file system. When you have the Internet Explorer Sound Pack installed (available for free download), you can set two new Control Panel sound events, Start Navigation and Complete Navigation, to provide audible signals when the browser starts looking for a site and when loading has been completed. If Internet Explorer can't find a particular site that was entered into the Address bar, instead of displaying technical error messages, it will first search common domain names, such as .com, .edu, and .org. Then it will ask if you would like to search for the site.

Printing Enhancements

Internet Explorer 4 offers several printing enhancements that allow you to control how Web pages are printed. Frame-set improvements enable three options for printing frame sets: print frames as laid out on the screen; print all frames separately; or print only the selected frame. You can even configure Internet Explorer to print all linked documents for a print selection. Plus, support for style-sheet extensions allow Web-page designers to control placement of page breaks within a Web page, so users benefit from "smart page breaking."

Using Explorer Bars

Explorer bars make it easier for users to navigate the Web: they improve users' ability to find relevant information on the Internet during searches; they make working with Favorites simpler; they make it easier for users to identify what sites they have visited in their Web-browsing sessions so they can return to them again; and they improve the discoverability of Active Channels.

Explorer bars are implemented in a split-screen browser pane that is accessible at the same time as the user browses the Web. When you select an Explorer bar by clicking its button on the Internet Explorer toolbar, the Internet Explorer window divides into two panes—a navigation pane (the Explorer bar) on the left and a content pane on the right. When you click a link in the left pane, its content is displayed in the right pane. The Explorer bar takes advantage of the component architecture of Internet Explorer 4 and is not simply a frame, but a separate browser control. The Explorer bar has a title bar, a close button, and can be resized by the user.

The Explorer bar buttons on the toolbar are toggle buttons; when you click the button once, it will open the corresponding Explorer bar, and when you click the button a second time, it will close the Explorer bar. When the Explorer bar is closed, the content pane will fill the entire browser window.

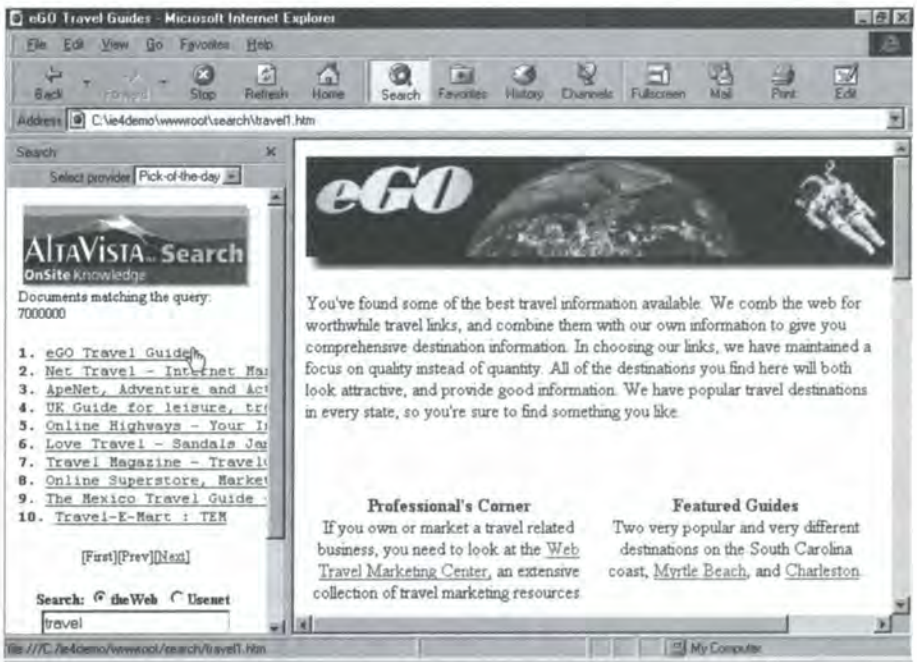
There are four different Explorer bars:

- Search bar
- Favorites bar
- History bar
- Channel bar

Understanding the Search Bar

Finding information on the Web can be a frustrating experience. Not only do users commonly receive thousands of successful matches for a search query, but they then confront the difficulty of navigating from site to site. Internet Explorer 4 attempts to solve the navigation issue by displaying a Search bar when users click the **Search** button on the toolbar. The Search bar displays search results independent of the main browser area and remains visible until the user presses the **Search** button again.

Each time you open the Search bar, it displays a list of search engines from which you can choose. When you start a search, the results appear in the Search bar only. When you rest the pointer over a result, Internet Explorer also displays a summary of the site in a ToolTip.



When you select a site from the results list, the site appears in the main browser area, keeping the Search bar available for future searches. The results remain in the Search bar, so if you need to visit a different site that the search engine revealed, you can easily move from result to result without repeatedly using the **Back** button to return to the search-results page. Internet Explorer 4 also preserves the state of the search, so if you click the **Search** button again during the same session, the results of the previous search are displayed. This way, you can move the Search bar out of the way and bring it back when you need it.

Configuring and Customizing the Search Bar

When you bring up the Search bar, Internet Explorer links to a default page at **http://home.microsoft.com/search/search.asp** that offers a Provider-of-the day. (This is done so that the Search bar page code can be constantly updated and improved by Microsoft.) This Active Server Page also provides a drop-down menu that allows a different search site to be chosen. You can customize the Search bar to point to a search engine that is not on the default menu or to point to your own search site. There are two ways to do this:

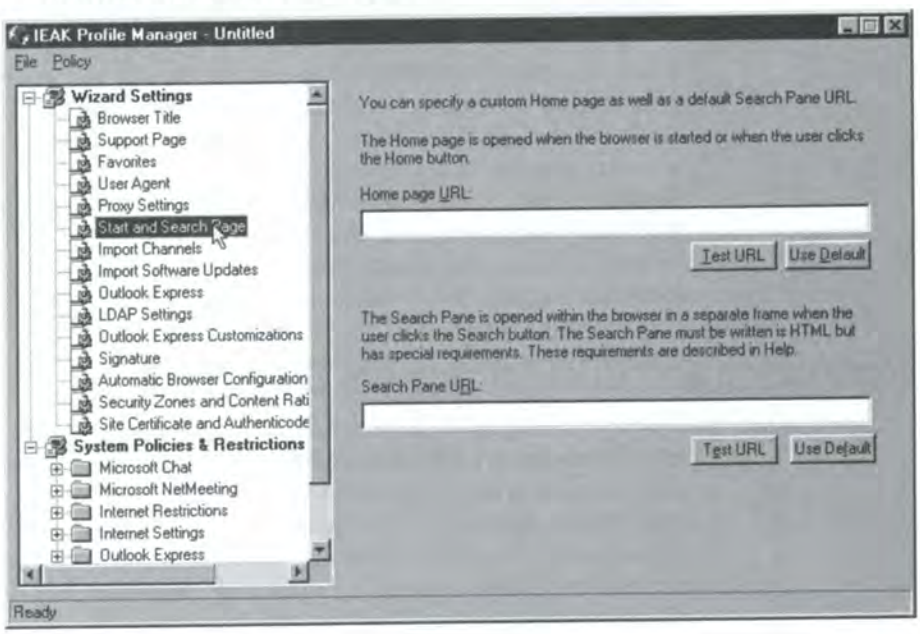
If	Then
You have set up the Internet Explorer Administration Kit (IEAK) automatic browser configuration feature.	Use the IEAK Profile Manager. (For more information, see Chapter 37, "Automatic Browser Configuration."
You have not set up automatic browser configuration.	Use the Registry Editor.

Warning Using the Registry Editor incorrectly can cause serious problems that may require you to reinstall Windows 95. Microsoft cannot guarantee that problems resulting from the incorrect use of the Registry Editor can be solved. Use the Registry Editor at your own risk.

Note It is recommended that you make a backup copy of the registry files (System.dat and User.dat) prior to editing the registry. For information about how to edit the registry, see the Help menu in the Registry Editor (Regedit.exe).

▶ **To point to a different Search bar page using the IEAK Profile Manager**

1. From the IEAK Profile Manager, on the **File** menu, click **New**.
2. In the left pane, click **Start and Search Page**.
3. In the **Search pane URL** box, type the URL that you want, and then click **Test URL** to be sure it is correct.



▶ **To point to a different Search bar page using the Registry Editor**

1. From the **Start** menu, click **Run**, type **regedit**, and then click **OK**.
2. In the Registry Editor, navigate to the following registry key:

   HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
3. Click the **Main** registry key.
4. In the right pane of the Registry Editor, double-click **Search Bar** in the Name column. Then, under Value Data, type the complete URL or entire file path for the Search page you want to use in the Search bar.
5. If the value does not exist, complete the remaining steps.
6. Right-click a blank part of the right pane, click **New**, and then click **String**. The new value appears with a temporary name.
7. Type **Search bar**, and then press ENTER twice.
8. Under Value data, type the complete URL or entire file path you want for the Search page. This can be any address or a location on the local computer or network. For example:

   ```
   http://my server/my folder/my search page.htm
   ```
9. Close the Registry Editor, and then restart Internet Explorer 4.

Tip If you make a mistake that results in your computer not starting properly, it is possible to restore the registry. For instructions, see the Registry Editor Help topic, "Restoring the registry."

You can also make the **Find** menu (on the Windows **Start** menu) point to a different Search page when **On The Internet** is selected.

▶ **To make the Windows Find menu point to a different Search page**

1. In the Registry Editor, navigate to the following registry key:

   HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main
2. In the right pane, double-click **Search Page** in the Name column.
3. In the **Edit String** dialog box, replace the path under Value Data with the complete URL or entire file path you want for your Search page. This can be any address or a location on the local computer or network, for example:

   ```
   file://inetsdk/rtw-image/installed/help/default.htm
   ```
4. Close the Registry Editor, and then restart Internet Explorer.

Creating Your Own Search Bar

You can create your own custom search page for the Search bar in much the same way that you would create a Search page on an ordinary Web site. The following guidelines and tips will help you to get the most from your custom Search page:

- Result pages should include buttons for modifying the current search or running a new search. These buttons should be in the top 20 percent of the content area.
- The default size for the Search bar is 200 pixels wide by the current height of the browser window. Although the user can widen it, your page (including tables and elements) should fit within this area to avoid horizontal scroll bars.
- You can use Dynamic HTML, ActiveX technology, Java, and Cascading Style Sheets (CSS) to achieve dynamic effects and implement multimedia. Remember, however, that the space for the Search bar is limited.
- For direct links to the content area in the main browser window, you can:
  - Add the tag <Base Target="_main"> to the header of the Search page to direct all the links

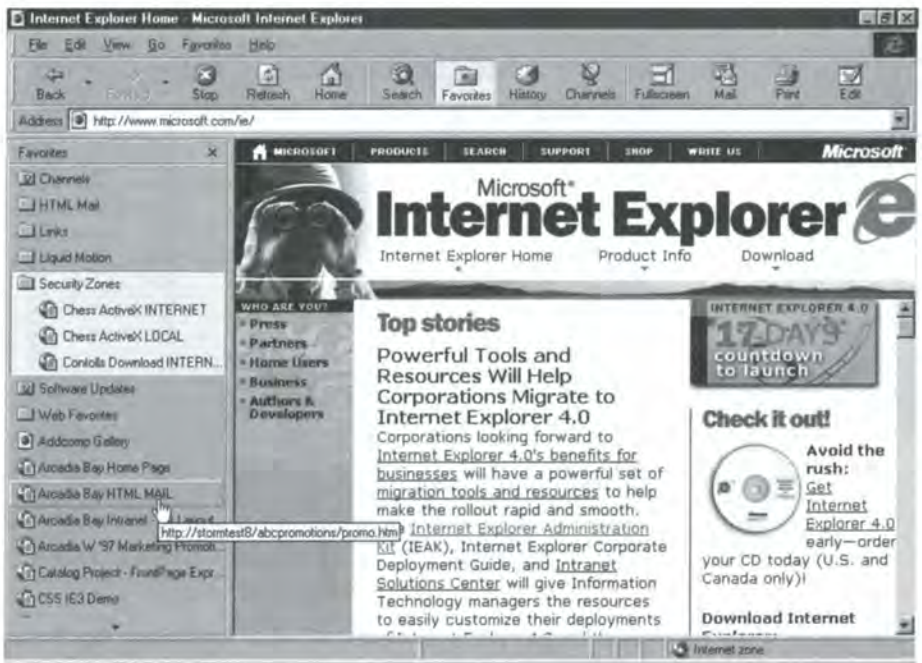
    or
  - Add the attribute Target="_main" to the <A HREF> tag of each link, as you would reference a frame in a frameset.
- To make an exception to <Base Target="_main"> and direct all links to the contents area, add the attribute TARGET="_top" to the <A HREF> tag of the link, such as for a link to the next 10 searches. For example:

  ```
  <A HREF="URL" TARGET="_top">
  ```

- Use the TARGET=_self attribute for links that update the Search window.
- Any links on the page that don't target the main browser area should be used with caution. The Search bar has no visible navigation controls. If you change the content in the Search bar, you must provide your own navigation controls in your content, such as backward and forward links in the HTML of your new page.

Understanding the Favorites Bar

The best way to view the sites and files that you use most frequently is by using the Favorites bar. Click the **Favorites** button on the Internet Explorer toolbar to display a list of your favorite Web sites and files in the navigation pane in the browser window. From here, favorites can be easily accessed and organized. You can add sites to the Favorites bar by simply dragging a link into the bar. You can drag links or folders within the Favorites bar to rearrange the order of your list according to your preferences.



The path names used by the Favorites bar are stored in the C:\Windows\Favorites folder unless the User Profile feature is turned on; in that case, the Favorites folder is stored in your Profile folder. By default, Internet Explorer comes with the following Favorites folders:

- **Channels.** Includes Active Channels that are installed.
- **Links.** Includes a collection of Microsoft links.
- **Software Updates.** Contains a link to the Microsoft Channel Web site that checks for a new version of Internet Explorer or other Microsoft programs. Other programs can write .inf files to add themselves to this folder.
- **My Documents.** Opens the local My Documents folder.

Configuring Favorites

The list of favorites in the Favorites bar is the same list that is used in the **Favorites** menu on the browser and the **Favorites** menu on the **Start** button. Any changes to this list will be reflected in each place in which the favorites are displayed. You can add, rearrange, rename, or delete individual Favorites and favorites folders using the **Organize Favorites** dialog box.

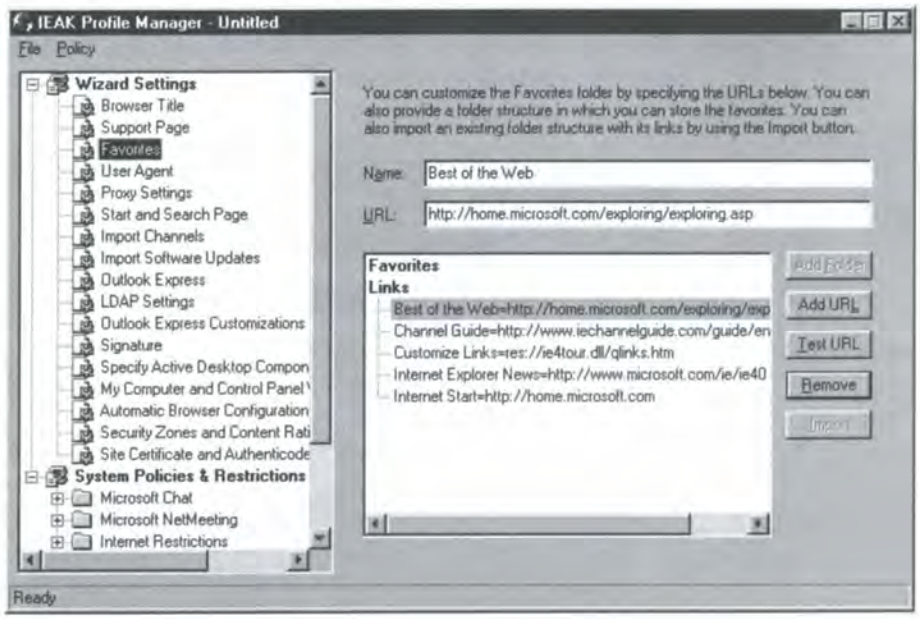
▶ **To access the Organize Favorites dialog box**

- From the Favorites menu, select Organize Favorites.

If you have set up the automatic browser configuration feature in the Internet Explorer Administration Kit (IEAK), you can change the default URLs that are included in the Links folder, or you can create an organization-wide standard set of favorites and distribute them to all users by using the IEAK Profile Manager. You can also use the Profile Manager to import a folder of existing Favorites. (For more information, see Chapter 37, "Automatic Browser Configuration.")

▶ **To edit favorites using the IEAK**

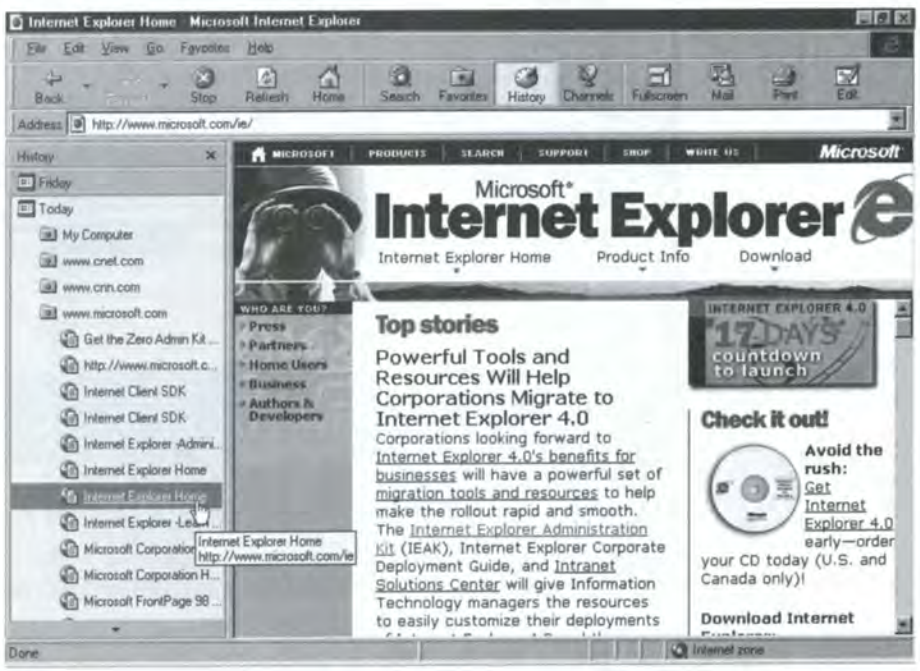
1. From the IEAK Profile Manager, on the **File** menu, click **New**.
2. In the left pane, click **Favorites**.



3. The URLs appear in the box at the bottom of the right pane. Here, you can:
   - Add a new URL or subfolder to the Favorites folder by selecting a subfolder and clicking **Add URL** or **Add Folder**. The new item will be added under the subfolder you select.
   - Edit an existing URL by clicking it and changing its path or friendly-name information in the text boxes above.
   - Remove an unwanted URL from the Favorites folder by selecting the URL and clicking **Remove**.
4. When you are finished editing, save the file in the subfolder within the IE4site folder that contains the installation package for your browser. In English versions, the folder is named En.

Understanding the History Bar

In previous browsers, it was difficult to make sense of the History directory, because it likely was filled with a dizzying array of .HTML, JPEG, .gif files, and other random information from the Web. In Internet Explorer 4, it's easy to see where you've been, because the browser automatically sorts your navigation history by date and then by site, displaying only useful shortcuts inside the History bar and ignoring other file types (such as graphics and cookies).



The History bar contains shortcuts for URLs and files that you have visited recently (within the last 20 days, by default). These are broken down according to when you last visited each page (the 3 previous weeks, the current week, and the current day). This information is stored in C:\Windows\History unless the User Profile feature is turned on; in that case, the History folder is stored in your Profile folder.

Note Internet Explorer still caches its complete file history in C:\Windows\Temporary Internet Files. This is useful if you need to see not only shortcuts but also every file that was downloaded in a past session.

To allow faster access, only the path names of recently visited sites are stored in the History folder. The actual contents of the sites or files you view are also downloaded as you view them, but these are saved in the Internet Explorer 4 cache, located at C:\Windows\Temporary Internet Files. In addition, you can find registry information about the History folder at:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Cache\Extensible Cache

Configuring the History Bar

The only configurable feature of the History bar is the number of days the History list will retain its information.

▶ **To change the number of days pages will be saved in History**

1. From the **View** menu, select **Internet Options**. The **Internet Options** dialog box will appear.
2. On the **General** tab, in the **History** box, change the number next to **Days to keep pages in History**.

3. Click **OK**.

You can clear the entire history folder by clicking the **Clear History** button in the **Internet Options** dialog box.

Administrators can prevent users from modifying the History settings by checking the **Disable changing history settings** option in the IEAK Configuration Wizard or the IEAK Profile Manager. This option is located under **Internet Policies and Restrictions, Internet Restrictions, General**. (For more information, see Chapter 36, "Implementing the Deployment.")

Understanding the Channel Bar

The Channel bar displays the Active Channels currently available. When Internet Explorer is first launched, the cache is preloaded with a collection of channels that is stored in C:\Windows\Favorites\Channels. This is intended to introduce users to the concept of Active Channels by giving them the opportunity to sample from a variety of existing channels. For more information about Active Channels, see Chapter 16, "Managed Webcasting."

Configuring the Channels Bar

Administrators can deliver a custom collection of channels to their users by using the IEAK Configuration Wizard or the IEAK Profile Manager. This is done under **Internet Settings, Import Channels**. In addition, many subscription parameters can also be set in the IEAK tools under **System Policies and Restrictions, Subscriptions**. For more information, see Chapter 36, "Implementing the Deployment" and Chapter 37, "Automatic Browser Configuration."

▶ **To customize channels using the IEAK Profile Manager**

1. In the left pane of the IEAK Profile Manager, select **Import Channels**.
2. Customize the channels on the configuration machine. (Add new channels and remove unwanted ones.) These settings will be imported to the users in your workgroup.
3. Select **Import current channel configuration**.
4. When you are finished editing, save the file in the subfolder within the IE4site folder that contains the installation package for your browser. For example, in English versions, the folder is named En.

▶ **To restrict subscription settings using the IEAK Profile Manager**

1. In the left pane of the IEAK Profile Manager, select **Subscriptions** from the System Policies and Restrictions section.
2. Edit the subscription settings you want to change.
3. When you are finished editing, save the file in the subfolder within the IE4site folder that contains the installation package for your browser. In English versions, the folder is named En.

Creating a New Explorer Bar

You can create a custom Explorer bar to conveniently display a corporate folder, library, site map, or any other list or menu that members of your organization might frequently use to navigate the Internet or intranet. You might want to create several new Explorer bars containing sites that are specific to the task related to each department or workgroup in your organization.

You can create new Explorer bars in two ways: You can create a new name on the **View, Explorer bar** menu that points to an existing page, or you can program new components. To program new components for an Explorer bar, you should be familiar with OLE COM, Visual C++® programming, and component categories. For more information about creating an Explorer bar, refer to the "Band Objects" topic in the *Microsoft Internet Client SDK*.

Browsing Offline

When users want to go to a favorite site on the company intranet or on the Web, it's not always convenient to connect to the network or to their Internet service provider (ISP). With offline browsing, users can view sites without these connections. For example, users can browse offline while using their laptop computers at locations where Internet access is not available. Or users may find it helpful to browse offline at home when they don't want to tie up their only phone line. Some users may even prefer to browse offline at the office when server traffic is heavy and load time is slow. (Offline browsing is faster than browsing online.)

Internet Explorer 4 makes offline browsing possible by automatically caching every site you visit when online. All Web pages you access, as well as files related to those pages (such as graphics files), are stored on your hard disk, so they can be viewed later without an active Internet connection. To begin browsing offline, simply select **Work Offline** from the **File** menu in Internet Explorer and select any page from the History bar.

When browsing offline, if you hover over a link to a page that is not cached, the cursor will change, denoting that the content behind the link is not locally available. If you proceed to click on that link, a dialog box will offer the option of connecting to the Internet to access the page.

In addition, you can use the Webcasting technology in Internet Explorer to schedule your browser to automatically check a Web site for changes since your last visit and download any new or updated content, to ensure that you're always browsing the most up-to-date information, even when offline. For more information about Webcasting, see Part 4 "Information Delivery: Webcasting."

Understanding Offline Browsing

Internet Explorer 4 automatically caches every site you visit while browsing online and stores cached files in the C:\Windows\Temporary Internet Files folder. Each file in the cache is assigned an alphanumeric file name. This prevents Web developers from determining the relative path and file name of a file and attempting to execute it. If you view the contents of the Temporary Internet Files folder using Windows Explorer, you will see the original file names rather than the new file names. These names are mapped in the cache index (Index.dat). The Temporary Internet Files folder is invisible from the MS-DOS prompt. (Removing its attributes will not change this.) When you access one of the cached files, Internet Explorer checks the index to map the original file name to its new random file name, and the file is retrieved from the cache to the browser.

Understanding CD Caching

The new CD Caching utility in Internet Explorer 4 allows users to access part of the http namespace directly from a CD-ROM drive to avoid having to repeatedly connect and download frequently used content. The Cache Container Tool (**cdcache.exe**) is the AutoRun executable file that allows this. It associates selected HTTP content with a URL that the user specifies. When the user browses for this content, Internet Explorer first looks for the content on the CD-ROM drive. If it does not find the content there, Internet Explorer allows the user to look in the usual location. This tool can be found in the \bin folder of the *Microsoft Internet Client SDK*. For more information about creating your own cache container, see the "Cache Container Tool (**cdcache.exe**)" topic in the *Microsoft Internet Client SDK*.

Using Offline Browsing

To view Web content offline, it must first be downloaded from its online location, and offline browsing must be enabled.

▶ **To save a site or file to the hard disk**

- While Internet Explorer is online, browse to the desired site so that it gets saved in the cache.

  or

- Use Web site or Active Channel subscriptions to schedule the browser to download the latest content to your computer. (For more information, see Part 4, "Information Delivery: Webcasting.")

▶ **To enable offline browsing**

- On the **File** menu, click **Work Offline**.

  Internet Explorer will always start in Offline mode until you click **Work Offline** again to clear the check mark.

▶ **To open a site offline**

- Click the **History** button on the Internet Explorer toolbar and then select a shortcut from the History bar, or click the **Channels** button on the Internet Explorer toolbar; then, from the Channel bar, select a channel to which you have subscribed. You can also view any sites and files that are stored in the C:\Windows\Temporary Internet Files folder.

▶ **To check whether sites or files are saved on your hard disk**

1. On the **File** menu, click **Work Offline**.
2. Click the **History** button.
3. Move your mouse pointer over the History item you want to check. If the hand pointer appears, the item is saved on your hard disk and available for offline browsing. If a circle with a line through it appears, then you will not be able to view this content offline. Clicking that site brings up a dialog box asking you if you want to connect to the Internet.

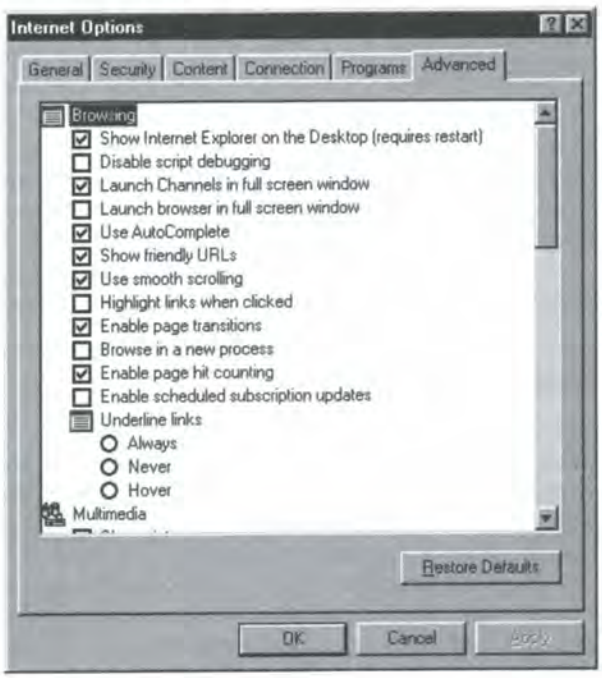
Setting Advanced Browser Options

In Internet Explorer 4, the **Internet Options** dialog box is laid out in a more discoverable manner. Easy-to-access tabs have been updated for **General** options (including History and Cache settings), Security settings (including security zone configurations, Connections settings (for ISP and proxy configurations), and **Advanced** options (for browsing, multimedia, and other settings).

Some of the most important changes to the **Internet Options** dialog box are on the **Advanced** tab.

▶ **To configure Advanced browser settings**

1. On the **View** menu, click **Internet Options**.
2. Click the **Advanced** tab.



3. Click the check boxes to select or clear the options you prefer, as described in the table below.

Option	Description
Browsing	
Show Internet Explorer on the Desktop (requires restart)	Displays the Internet Explorer icon on the Active Desktop.™
Disable script debugging	Turns off the script debugger if installed.
Launch Channels in full screen window	Launches Active Channels in Full Screen mode.
Launch browser in full screen window	Automatically displays all Web pages in Full Screen mode.
Use AutoComplete	Automatically completes URLs as you type them in the Address bar, based on your navigation history.

Option	Description
Browsing *(continued)*	
Show friendly URLs	Shows the full URL for the current Web page on the Status bar at the bottom of the browser window.
Use smooth scrolling	Uses scrolling that will display content at a predefined speed.
Highlight links when clicked	Displays a thin border around selected hyperlinks.
Enable page transitions	Fades out the page you are leaving and fades in the new page as you move from one Web page to another.
Browse in a new process	Starts a new instance of Internet Explorer every time you open it (for example, if you click an HTML file in My Computer).
Enable page hit counting	Allows the tracking of page usage.
Enable scheduled subscription updates	Allows Webcasting technology to automatically perform subscription updates that you have scheduled (including downloading of new and changed content).
Underline links: Always	Underlines all links on pages.
Underline links: Never	Never underlines links.
Underline links: Hover	Underlines a link when your mouse pointer is "hovering" over it.
Multimedia	(For more information about how multimedia options affect browser performance, see the following chapter, "Browser Performance").
Show pictures	Includes graphical images when pages are displayed.
Play animations	Plays animations when displayed in Web pages.
Play videos	Plays video clips when displayed in Web pages.
Play sounds	Plays music and other sounds when included in Web pages. (If you have an audio player such as RealAudio installed, sounds may still play, even if this option is disabled.)
Smart image dithering	Smooths images so that they appear less jagged when displayed.

Option	Description
Security	(For more information about security options, see Part 7, "Security and Privacy.")
Enable Profile Assistant	Accepts sites' requests for Profile Assistant information and prompts you to choose which parts, if any, of your demographic information to share.
PCT 1.0	Sends and receives secured information through PCT (Private Communications Technology).
SSL 2.0	Sends and receives secured information through the SSL 2.0 (Secured Sockets Layer Level 2) protocol.
SSL 3.0	Sends and receives secured information through the SSL 3.0 (Secured Sockets Layer Level 3) protocol.
Delete saved pages when browser closed	Deletes saved (cached) pages that have been downloaded to your hard disk when the browser exits.
Do not save encrypted pages to disk	Prevents secured information from being saved to your hard disk.
Warn if forms submit is being redirected	Displays a warning when you send information using a Web-based form if the information will be sent to a location different than that of the form.
Warn if changing between secure and insecure mode	Displays a warning when you switch from Internet sites that are secure to sites that are not secure.
Check for certificate revocation	Checks certificates before accepting them as valid to see if they have been revoked.
Warn about invalid site certificates	Displays a warning when the URL in the digital certificate of an Internet site is not valid.
Always accept cookies	Accepts cookies without being prompted. A cookie is a file sent by an Internet site and stored on your computer; it stores information about your identity and preferences when you visit that site.
Prompt before accepting cookies	Displays a warning prompt before receiving a cookie from a Web site.
Disable all cookie use	Prevents cookies from being sent to your computer by Internet sites and prevents existing cookies on your computer from being read by Internet sites.

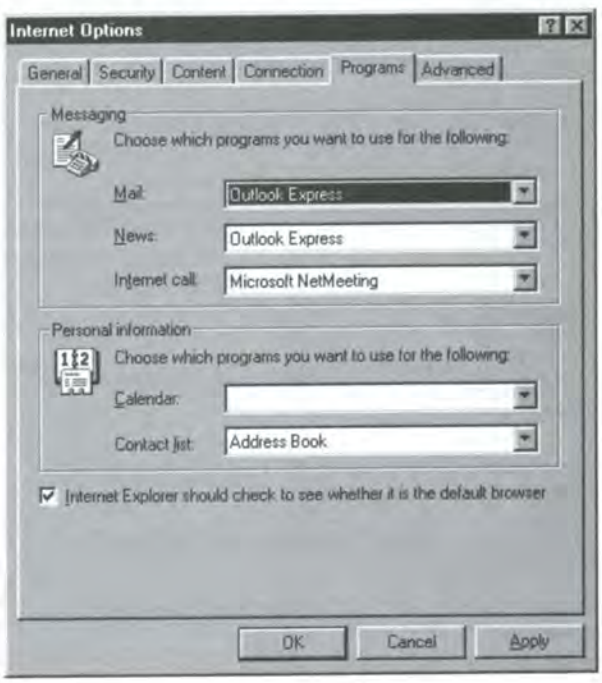
Option	Description
Java Virtual Machine (VM)	(For more information about Java VM and Java JIT, see the following chapter, "Browser Performance.")
Java JIT compiler enabled	Creates all Java applets automatically, using the Internet Explorer Java Just In Time compiler.
Java logging enabled	Creates a log of all Java program activity.
Printing	
Print background colors and images	Prints background colors and images when you print a page.
Searching	
Autoscan common root domains	Uses AutoDomain Searching to search for a URL using common root domains.
Search when URL fails: Never search	Disables searching for an accurate URL when the entered URL fails.
Search when URL fails: Always ask	Displays a prompt before searching for an accurate URL when the entered URL fails.
Search when URL fails: Always search	Always searches for an accurate URL when the entered URL fails.
Toolbar	
Show font button	Displays the **Font** button on the toolbar.
Small Icons	Displays the toolbar using Microsoft Office–style buttons without text descriptions.
HTTP 1.1 settings	(For more information about HTTP 1.1, see the following chapter, "Browser Performance.")
Use HTTP 1.1	Sets intranet connections to HTTP 1.1.
Use HTTP 1.1 through proxy connections	Sets proxy connections to HTTP 1.1.

Integrating Third-Party Programs

Internet Explorer 4 supports an open architecture for all non-HTTP clients. Third-party client software can be registered as Internet Explorer clients and included in the programs listed on the Internet Options **Programs** tab, for easy integration with Internet Explorer. You can use the **Programs** tab to select the default program used for:

- Mail
- News
- Internet call (Conferencing)
- Calendar
- Contact list (Address Book)

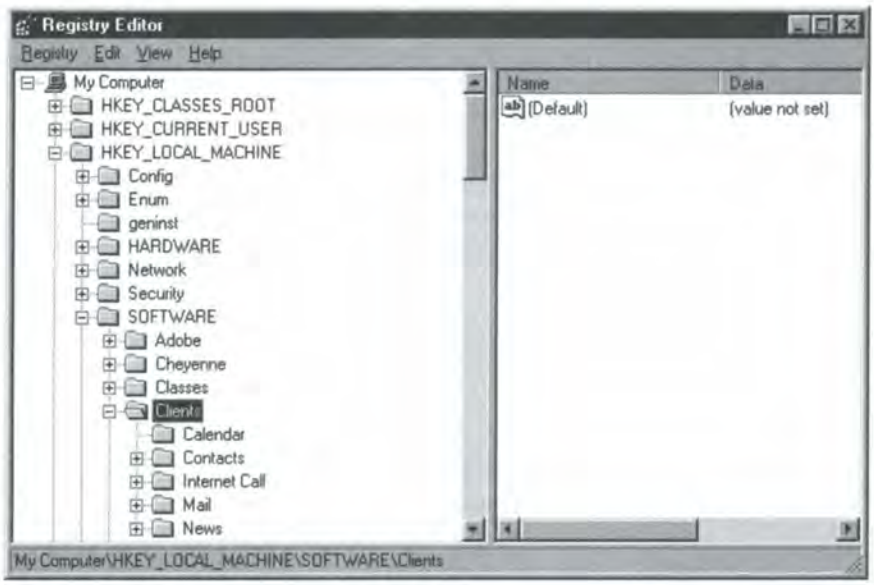
The default programs are launched when you click the corresponding menu option on the Internet Explorer **Go** menu. For example, if you select Outlook™ Express as your default mail and news client, Internet Explorer launches Outlook Express when, from the **Go** menu, you click **Mail** or **News**. Likewise, Internet Explorer launches the default Internet call, calendar, and contact list programs you specify as the defaults on the **Programs** tab.



Client programs can register as Internet Explorer clients by creating subkeys in the registry at:

HKEY_LOCAL_MACHINE\SOFTWARE\Clients

The client subkeys are: **Calendar**, **Contacts**, **Internet Call**, **Mail**, and **News**.



The following tables describe the Mail and News subkeys. (The other subkeys work the same way.)

Key	Description
HKEY_LOCAL_MACHINE\ SOFTWARE\Clients\Mail	Contains subkeys for registered mail clients; for example, Microsoft Outlook and Outlook Express.
HKEY_LOCAL_MACHINE\ SOFTWARE\Clients\News	Contains subkeys for registered news clients; for example, Microsoft Outlook and Outlook Express.

Subkeys for registered clients contain the following types of information:

Mail

Key	Description
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\Mail\Outlook Express	Specifies friendly name for this client ("Outlook Express") that appears in the **Programs** tab, **Mail** selection list.
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\Mail\Outlook Express\Protocols\mailto	Specifies that Outlook Express is used as the mail client for mailto: URLs.
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\Mail\clientname\shell\open\command	Specifies the path, program file, and command-line options that Internet Explorer uses to launch Outlook Express. For example, when the user clicks a mailto: URL, Internet Explorer runs the Outlook Express program with any specified command-line options.

News

Key	Description
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\News\Outlook Express	Specifies friendly name for this client ("Outlook Express") that appears in the **Programs** tab, **News** selection list.
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\News\Outlook Express\Protocols\news	Specifies that Outlook Express is used as the news client for Usenet news: URLs.
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\News\clientname\Protocols\nntp	Specifies that Outlook Express is used for communications requiring the NNTP protocol.
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\News\Outlook Express\Protocols\snews	Specifies that Outlook Express is used for secure news (snews:) URLs.
HKEY_LOCAL_MACHINE\SOFTWARE\Clients\News\Outlook Express\shell\open\command	Specifies the path, program file, and command-line options that Internet Explorer uses to launch Outlook Express to send news messages.

CHAPTER 10

Browser Performance

In this chapter

Optimizing Browser Performance 130
Using HTTP 1.1 130
Using the Java Just In Time Compiler 131
Using Cached Content 132
Configuring Multimedia Settings 134

Optimizing Browser Performance

Internet Explorer 4 includes a number of performance optimization features, as well as support for advanced protocols and technologies that deliver enhanced browser performance. At the most basic level, the rendering engine in Internet Explorer 4 has been greatly improved, allowing faster Web-page download on a local area network (LAN) or dial-up connection. Internet Explorer is fully integrated with the Windows shell to provide maximum performance of ActiveX controls. In addition, Internet Explorer supports the new HTTP 1.1 protocol, which compresses packets of data transferred via HTTP to boost performance as much as 50 to 100 percent over HTTP 1.0, for faster, more efficient downloads of Web content.

Internet Explorer also supports the Portable Network Graphics (PNG) specification, Dynamic HTML, and the Microsoft DirectX® multimedia extensions, which enable developers to design faster, more bandwidth-efficient graphics and interactive and animated Web content. (For more information about Dynamic HTML, see Part 6, "Authoring for Internet Explorer 4.") Plus, Internet Explorer includes the fastest Java Virtual Machine (VM) and Just In Time (JIT) compiler to provide optimum performance for Java applets.

Using HTTP 1.1

HTTP 1.1 provides enhanced Web performance between browsers and servers. Basically, browsers and servers communicate more efficiently using HTTP 1.1 than when using the older HTTP 1.0. Web communications using HTTP 1.1 require fewer delays and consume less bandwidth, and Web content downloads faster to the browser. By default, HTTP 1.1 communications are enabled in Internet Explorer to optimize browser performance when communicating with servers that support HTTP 1.1.

You can use the **Advanced** tab on the **Internet Options** dialog box to enable or disable HTTP 1.1. If HTTP 1.1 is disabled, all Web communications are transacted using the older and slower HTTP 1.0. The default settings for Internet Explorer do not allow HTTP 1.1 to be used through a proxy server. If you want to use HTTP 1.1 to communicate with Web servers through a proxy server, you'll need to select the **Use HTTP 1.1 through proxy connections** option. For more information about setting Advanced browser options, see "Setting Advanced Browser Options" in the previous chapter, "Browser Features and Functionality."

Using the Java Just In Time Compiler

Java applets are compiled into an intermediate form called byte code. The Java Virtual Machine (VM) included with Internet Explorer 4 translates the byte code into the machine code needed for the computing platform when it runs an applet. This translation process typically results in much slower performance for Java applets than for programs that are compiled in native machine language, such as standard Windows programs. The Java VM included with Internet Explorer 4, however, includes a Just In Time (JIT) compiler that significantly improves the performance of Java applets and offers the fastest Java experience available.

By default, the Java JIT is enabled to provide optimum performance for Java applets. You can use the Internet Options **Advanced** tab to disable the Java JIT compiler. For more information about setting Advanced browser options, see "Setting Advanced Browser Options" in the previous chapter, "Browser Features and Functionality."

Using Cached Content

The first time you connect to a Web page, Internet Explorer downloads the page and its supporting content to a cache stored in the Temporary Internet Files folder on your hard disk. By default, Internet Explorer uses the cached content instead of downloading new content when you return to browse a cached page. Using cached content provides faster performance when browsing Web sites, because the same content is not downloaded over and over again.

Each time Internet Explorer is restarted, however, it clears the cache and treats all Web pages you browse as new pages, downloading all the pages and their supporting content to the cache. During that session, Internet Explorer uses cached content whenever it is available until the next restart.

You can use the cache **Settings** dialog box to change the defaults and dictate when Internet Explorer downloads content when revisiting Web pages. For example, you can specify that Internet Explorer never clears its cache and always uses the stored content for a Web page, if available. Or you can require the browser to never use the cache and always download every Web page.

▶ **To configure the cache settings**

1. On the **View** menu, click **Internet Options**.
2. On the **General** tab, in the **Temporary Internet Files** box, click **Settings**.
3. Configure the cache options, as described in the table below.

4. Click **OK**.

Option	Description
Every visit to the page	Select to specify that Internet Explorer downloads Web-page content each time you browse a Web page. Choosing this option typically results in slower browsing, because content is always downloaded.
Every time you start Internet Explorer	Select to specify that Internet Explorer only downloads Web-page content the first time you visit a Web page in a given session. This option can speed up browsing, because cached content is reused rather than downloaded again. Internet Explorer downloads new content for this page only when you restart Internet Explorer. This is the default setting for Internet Explorer.
Never	Select to specify that Internet Explorer always uses cached content when you revisit Web pages. If you choose this option, you won't see updates or changes to cached Web pages.
Amount of disk space to use	Select to specify the percentage of disk space to use for the Temporary Internet Files folder. When you view a new page on the Web, Internet Explorer temporarily stores it (and some of its contents, such as graphics files) on your hard disk. This increases the speed at which previously viewed pages are displayed. The more disk space you allot to the folder, the more pages Internet Explorer can store on your hard disk. If you are low on disk space, however, you may want to set this option to a lower percentage.

Note No matter what cache settings you choose, you can always manually update the cache while browsing Web pages by clicking the **Refresh** button on the browser toolbar to see any changes to the page.

Configuring Multimedia Settings

By default, Internet Explorer shows or plays a wide range of standard multimedia content, including graphics, video, and audio. Internet Explorer also uses image-dithering technology to smooth images so they appear less jagged. However, multimedia content and dithering can cause Web pages to download and display slowly.

You can use the Internet Options **Advanced** tab to configure multimedia settings to allow Web pages with multimedia content to download faster. For example, you can specify that Internet Explorer not show graphics or play animations, videos, or sounds. You can also turn off image dithering to speed up image display time. Be aware, however, that when multimedia features are disabled, the pages will not display as intended, and you may miss some relevant content. For more information about setting Advanced browser options, see "Setting Advanced Browser Options" in the previous chapter, "Browser Features and Functionality."

Note When graphics, animations, or videos are turned off, Internet Explorer shows a small icon for the item in the browser window. You can selectively choose to display images or play animations and videos by right-clicking the icon for the image, animation, or video and clicking **Show Picture**.

CHAPTER 11

The WebBrowser Control

In this chapter

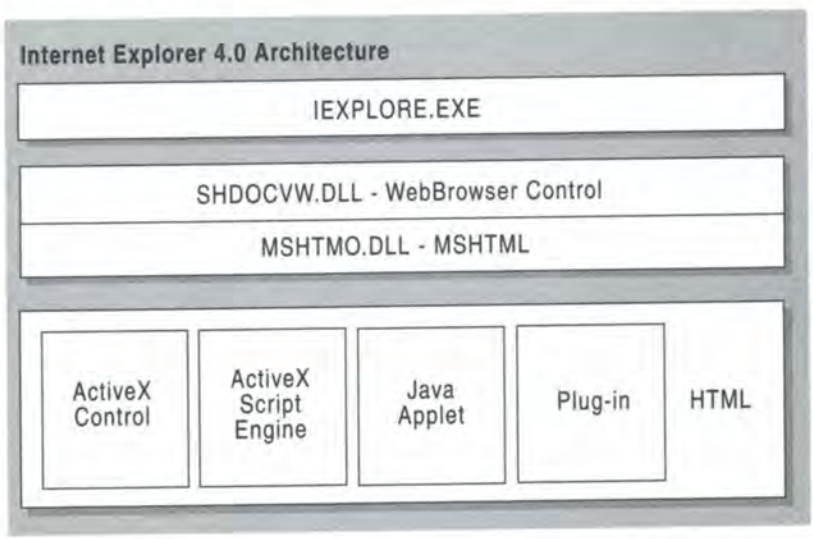
Understanding Browser Architecture 136
Reusing the WebBrowser Control 139
Changing Error Messages 140
Customizing WebBrowser Control Functionality 141
Adding HTML Resources 141
Distributing Browser Controls with Applications 142

Understanding Browser Architecture

The Internet Explorer 4 program, Iexplore.exe, is basically a wrapper program for two browser ActiveX controls, Shdocvw.dll (also called the WebBrowser Control) and Mshtml.dll, which provide all of the functionality of the browser. These controls can be reused with any application that supports ActiveX controls to add HTML browsing functionality to the application. These controls provide the following features:

- **Hyperlink navigation and in-place hyperlinking.** In-place hyperlinking refers to the opening of documents in the same instance of the WebBrowser Control, rather than in a new window.
- **Uniform Resource Locator (URL) navigation.**
- **Favorites and History lists.**
- **Security.** (For more information, see Part 7, "Security and Privacy.")
- **Platform for Internet Content Selection (PICS) protocol.** This is the information exchange standard that allows you to control what content can be viewed in the browser. (For more information, see Chapter 30, "Content Advisor.")

The following diagram shows the basic architecture for Internet Explorer 4.



Internet Explorer uses ActiveX controls and Active Document interfaces to connect components. Iexplore.exe is a small program that hosts Shdocvw.dll, which in turn hosts Mshtml.dll. Shdocvw.dll, the WebBrowser Control, supplies the functionality associated with HTML navigation, in-place hyperlinking, Favorites and History management, and PICS support. Many companies, such as America Online, use Shdocvw.dll within their programs to provide Internet browsing. Shdocvw.dll can be found in the C:\Windows\System folder.

Although Shdocvw.dll provides navigation, Favorites, and History functionality, it cannot open, interpret, or render HTML pages. For that functionality, Shdocvw.dll calls on the following components, each of which is needed to create a rich, Internet experience:

- **Mshtml.dll** renders HTML pages.
- **Urlmon.dll** provides HTTP connectivity by processing URLs.
- **Wininet.dll** provides the Internet connection.
- **Java VM** runs Java applets.

Mshtml.dll is an Active Document (also known as an OLE document object) that exposes the HTML document through the Dynamic HTML Object Model and performs the HTML parsing and rendering in Internet Explorer 4. This component hosts the scripting engines, Java VM, ActiveX controls, and other objects that may be referenced in the HTML document that is loaded. Mshtml.dll implements the Active Document server interfaces, which allows it to be hosted using standard COM interfaces. This architecture is recursive in that Shdocvw.dll is in turn hosted by Mshtml.dll in a frameset situation. In this case, each instance of a frame is another instance of Shdocvw.dll hosting Mshtml.dll.

Earlier versions of Internet Explorer did not allow an ambient property, such as the ambient background color, to affect the OLE document objects. As an OLE-based architecture, Internet Explorer 4 does allow the ambient properties that are commonly used by ActiveX controls to be applied to its components. In this way, a host of the WebBrowser Control can set an ambient property that will filter down to all the frames and controls hosted in the loaded document.

Should I Reuse Shdocvw.dll or Mshtml.dll?

Developers can choose to use only Mshtml.dll in their programs or components. Hosting Mshtml.dll, however, is only recommended for specialized programs such as parsing HTML. Mshtml.dll does not support in-place hyperlinking, so clicking a hyperlink would run a new instance of the browser.

For the majority of programs, developers should use Shdocvw.dll, because it supports navigation and in-place hyperlinking and because it is very likely that the WebBrowser Control will already be loaded in memory. For more information about hosting Mshtml.dll, see the WalkAll sample in the *Microsoft Internet Client SDK* topic, "Samples List."

HTML Resources Within Shdocvw.dll and IE4tour.dll

Embedded within Shdocvw.dll and IE4tour.dll are several informational HTML pages. These pages are stored in this way to reduce the number of files actually included within the package and to prevent the need for Internet Explorer to maintain HTML files. These pages provide the Internet Explorer 4 Tour, seen the first time that Internet Explorer is opened, and provide any error messages that appear while using Internet Explorer.

To access the HTML pages, the **res://** command is used with the .dll file name and the name for the HTML page to be loaded. (The term "res" stands for *resource*.) The **res://** command will load an HTML resource from a specified .dll file. The following is a list of all the available resources. These URLs can be typed in the Run line or Address bar to open the resource.

Shdocvw.dll error messages

These messages are specified in the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\AboutURLs

- **res://shdocvw.dll/navcancl.htm** An information page appears when the user gets a failed connection to a URL. The many types of connection errors include proxy server errors and invalid URLs.
- **res://shdocvw.dll/offcancl.htm** An informational error appears when the user tries to view in offline mode a page that is not available in the cache.

IE4tour.dll Information Tour Resources

These resources are specified in the following registry key:

HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main

- **res://ie4tour.dll/channels.htm** This resource loads the Channels tour demonstration page.
- **res://ie4tour.dll/tour.htm** The Internet Explorer 4 tour of features and usability.
- **res://ie4tour.dll/welcome.htm** The page that appears the first time Internet Explorer 4 is opened. This is the page with the three tour options: Tour, Channels, and Register.

Reusing the WebBrowser Control

The WebBrowser Control provides rich functionality for you to include in any program where you want to see elements of a browser interface or to have just HTML navigation, History and Favorites management, HTML parsing and rendering, and so on. Because Shdocvw.dll is registered as a control in the registry, you include it in your programs by adding **Microsoft Internet Controls** to the toolbox in your development program, such as Visual Basic.

▶ **To create your own Web browser using Internet Explorer 4 and Visual Basic 5**

1. Launch Visual Basic 5.
2. Choose **Standard EXE** from the **New Project** dialog box.
3. Right-click **Toolbox** on the left side of the screen, and click **Components**.
4. Check the box next to **Microsoft Internet Controls**, and click **OK**. Notice the two new icons in the toolbox for the WebBrowser Control and the ShellFolderViewOC.
5. Make your form a bit larger so that you have more room to work.
6. Select the **WebBrowser** control, and draw a box on your form covering approximately three-quarters of the bottom of your form. You have just created the area in your Web browser where the Web pages will be viewed.
7. To make an Address bar, select the **TextBox** control from the toolbox, and draw a rectangular box above the **WebBrowser** control on your form.
8. Add a button to your form by selecting the **CommandButton** control and placing it just next to your Address bar. You can change the caption Command1 to anything you want by typing a new caption.
9. Finally, to make this work, double-click the button you just created, and add the following line of code:

   ```
   WebBrowser1.Navigate Text1.Text
   ```

10. Press the **Play** button or F5 to start your new Web browser. Type any URL in your Address bar, press the navigation button that you created, and the content appears below.

Your new Web browser contains all of the HTML rendering capabilities of Internet Explorer 4, because it is using Shdocvw.dll, which was described earlier in this chapter.

You can also use the Internet Explorer 4 object model to expose only the functionality you want your users to have. For example, you've just created a browser with no Forward, Backward, or Refresh button. If this suits your needs, this would be fine. To add this functionality, however, you only need to create a few more buttons on your form and add the following lines of code:

```
WebBrowser1.GoForward
WebBrowser1.GoBack
WebBrowser1.Refresh
```

You can experiment with the rest of the functionality inside Shdocvw.dll so that you can build the browser that fits your needs.

You can also make the following changes to Shdocvw.dll components in your programs:

- Change Error messages
- Customize WebBrowser Control functionality
- Add HTML resources

Changing Error Messages

To change the error messages that come with Shdocvw.dll, you must edit the registry.

Warning Using the Registry Editor incorrectly can cause serious problems that may require you to reinstall Windows 95. Microsoft cannot guarantee that problems resulting from the incorrect use of the Registry Editor can be solved. Use the Registry Editor at your own risk.

Note For information about how to edit the registry, see "Changing Keys and Values," a Help topic in the Registry Editor (Regedit.exe). You should make a backup copy of the registry files (System.dat and User.dat) before you edit the registry.

▶ **To change the Shdocvw.dll error messages**

1. From the **Start** menu, click **Run**, type **regedit**, and then click **OK**.
2. In the Registry Editor window, click the plus signs in the left pane to go to the following location:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer
3. Click the **AboutURLs** registry key.

4. In the right pane of the Registry Editor, double-click the error message you want to change in the **Name** column, and then, under Value data, type the complete URL. For example, type:

   http://my server/my folder/my error message.htm

5. Close the Registry Editor, and then restart Internet Explorer.

Tip If you make a mistake that results in your computer not starting properly, you can restore the registry. For instructions, see the Registry Editor Help topic, "Restoring the Registry."

Customizing WebBrowser Control Functionality

You can customize WebBrowser Control functionality in your own programs by:

- **Overriding context menus, setting the 3-D border, or extending the Dynamic HTML Object Model.** You do these things by implementing the IDocHostUIHandler and IDocHostShowUI interfaces. These interfaces are obtained from the hosting program by either the WebBrowser Control or Mshtml.dll calling QueryInterface on the IOleClientSite interface.
- **Extending existing context menus.** You do this by changing registry setting
- **Controlling what content is downloaded.** You do this by implementing the download control ambient property, which can restrict frames, images, Java, and so on.
- **Overriding security settings.** You do this by implementing the IInternetSecurityManager interface.

For more information about reusing the WebBrowser Control or Mshtml.dll, or controlling their functionality, see the *Microsoft Internet Client SDK* topic, "Reusing the WebBrowser Control."

Adding HTML Resources

Adding HTML to your program requires you to create your own .dll file that includes customized HTML pages as resources. With customized .dll files, you can control functionality, such as viewing the HTML source. You might also develop your own toolbar with your own set of features.

To create .dll file resources, you use the **ShowHTMLDialog** function to pass parameters to and receive parameters from an HTML dialog box. The HTML code also accesses the parameters that are passed into the dialog box and sets the return parameters from the dialog box using script.

To add the HTML as a resource, you include a file that contains nothing but the actual HTML in your application's resource script. This file is included as an HTML type resource. The following example shows how to include a file called MyHTML.htm as an HTML resource:

```
HTML_RESOURCE HTML "MyHTML.htm"
```

In this example, HTML_RESOURCE is the identifier of the resource. This can be either a string or a numerical identifier. HTML is the resource type. Visual C++ 5 will interpret this as the numeric value 23 and will substitute 23 for HTML when the resource file is opened for editing. MyHTML.htm is the file that contains the HTML source that will be added. The resource compiler adds this file as is and will not attempt to interpret the contents of the file.

You can specify the HTML resource in any function that will accept a res: protocol. **ShowHTMLDialog** is one example of how the res: protocol can use HTML from a resource. You can also use **FindResource** and **LoadResource** to obtain the HTML resource. In **FindResource**, specify the RT HTML value for **lpType** to find an HTML resource.

For more information on creating resources within a .dll file, see the HTMLDlg sample in the *Microsoft Internet Client SDK* topic, "Samples List," and the *Microsoft Windows SDK.*

Distributing Browser Controls with Applications

You can distribute Internet Explorer browser controls with your applications royalty free. However, you must first obtain a redistribution license for Internet Explorer 4 from Microsoft. You can obtain the necessary redistribution license as part of the license agreement for the IEAK. When you register for the IEAK and accept the licensing agreement, you are free to distribute the components of Internet Explorer to your customers or users within the terms of your licensing agreement. For more information, visit **http://www.microsoft.com/ie/ieak/**. For more information about how to obtain a redistribution license, see the Microsoft Customer Support Web page at **http://microsoft.com/Support/**.

PART 3

True Web Integration: Windows Desktop Update

Until now, business computer users got their information from two distinct worlds: the online world of the Internet and the "local" world of company networks and systems. But, as the Internet advances to permeate users' everyday computing experience, the integration of Web features and functionality with the desktop operating system is an idea that has become increasingly important and valuable.

For this reason, Internet Explorer 4 provides True Web Integration with the Windows® Desktop Update, a set of features that integrate the Internet into every aspect of personal computing—the desktop, file folders, the network, even the **Start** menu. The Windows Desktop Update provides users with a single metaphor for exploring and searching for information. Whether looking at a local hard disk, a LAN network drive, or a Web site on an HTTP server, the same navigation toolbar provides functionality for going backward and forward, marking favorite locations, and searching for desired information. Because the same menus and toolbar accomplish similar tasks in different scenarios, users need to learn only one navigational model, and the cost of training and supporting users is reduced. Plus, features such as auto-URL-completion and the convenient search-results pane make daily Web browsing easier and more efficient.

To add to the navigational benefits provided by the Web integration in Internet Explorer 4, the flexibility of the Active Desktop™ interface allows intranet administrators to use push and smart-pull technologies to make information and resources directly accessible from users' desktops and to customize that information for specific user groups. The Active Desktop turns users' desktops into dynamic, personalized, live Web "bulletin boards" that regularly update with the latest content, for quick access online or off.

Part 3 describes how True Web Integration works in Internet Explorer 4, details the features included in the Windows Desktop Update, and explains how you can use these features to create a richer, more streamlined, and more productive working environment in your organization.

Chapter 12: Start Menu and Taskbar

The **Start** menu and taskbar provide an anchor on the Windows desktop, giving users one center for starting and switching tasks.When using the Windows 95 and Windows NT® operating systems, users know they can use the **Start** button to access almost everything. Inside the Web browser, they know they can rely on their **Favorites** menu to get to their commonly used sites. In Internet Explorer 4, these ideas were combined and the **Favorites** menu attached to the **Start** button, so users can now get to their favorite sites quickly—without even launching the Web browser first. Following the same logic, the Internet Explorer 4 taskbar provides access to a wide array of Internet-focused functions, permitting users to get to the information they need, quickly and conveniently.

Chapter 13: Web View

In Internet Explorer 4, elements are extended to integrate Internet and intranet tasks into a user interface that today's users have already mastered. For example, an address bar allows users to enter any Web address or file path right from their Windows taskbar, eliminating the need to switch back and forth between the file folder window (Windows Explorer) and the browser window (Internet Explorer). This concept is called Single Explorer, and it offers consistent navigation at the same time as it provides flexible functionality.

Web View expands Web browser functionality to many parts of the standard Windows interface. Now it is possible to view files and Web content in the same window, regardless of file location (Internet, intranet, network, or local hard disk). Plus, it's easy to switch between Web View and the classic Windows interface, or to combine features of both. In Web View, folders can display Hypertext Markup Language (HTML) content, making the folders easier to use and enhancing the information they offer. Custom Web views can be designed for each folder on a system, making it possible to design complete solutions for distributing files on an intranet, local network, compact disc, or even a floppy disk.

Chapter 14: Active Desktop

The Active Desktop interface brings the Web right to users' desktops with HTML wallpaper, screen savers, and Active Desktop items. Now you can use a Web page as desktop wallpaper or as a screen saver, complete with graphics and links to your favorite files and URLs. ActiveX™ objects, JavaScript, and Visual Basic® Scripting Edition (VBScript) can be used to add even more functionality. Active Desktop items, which are essentially miniature Web pages that reside on the desktop, complement this functionality. They can be used to provide information such as news headlines, weather reports, traffic information, and stock quotes. For users, adding an Active Desktop item can be as simple as pointing and clicking. For system administrators, creating an Active Desktop item is as easy as creating a Web page. The Active Desktop can be designed to meet the unique needs of multiple workgroups, improving access to information for users of all levels. Such custom configurations can be easily managed using the Internet Explorer Administration Kit (IEAK).

True Web Integration Architecture

The key file that makes True Web Integration possible is Shdocvw.dll. As with Internet Explorer 3, this file is the Web browser control for Internet Explorer. It is an OLE-COM (ActiveX) application that, when hosted within a parent application, provides Internet browser functionality. Companies such as America Online use Shdocvw.dll within their proprietary applications to provide Internet browsing.

When Shdocvw.dll is loaded, it provides Internet functionality such as Web navigation, viewing of HTML pages, and linking to Favorites. All of the ability to open, interpret, and render HTML pages, however, is not within Shdocvw.dll. It calls upon other services such as Mshtml.dll to render HTML content, the Java run time to process Java applets, and so on. Each of these components is needed to get a rich Internet experience:

- Shdocvw.dll provides Internet functionality, such as Web navigation control. The Internet Explorer program Iexplore.exe is a host for the Shdocvw.dll file. Iexplore.exe is responsible for loading the .dll file and trapping errors for the application, but does not really perform any other tasks.
- Mshtml.dll is the HTML rendering engine.
- Urlmon.dll is used for HTTP connectivity.
- Wininet.dll processes ActiveX controls.
- VBScript.dll processes VBScript.
- Java VM runs Java applets.

Note Shdocvw.dll can be found in the Windows\System folder.

When Windows Desktop Update is selected to be installed, Internet Explorer 4 replaces Explorer.exe and Shell32.dll during the installation process. These two files, along with Shdocvw.dll, make the Active Desktop and Web View functional.

Shell32.dll is needed to implement the updated layering model of the desktop and to fix some issues with the bounding outline in Active Desktop mode. Explorer.exe has been updated to be an OLE host for Shdocvw.dll. With this update, Internet Explorer can display HTML components on the desktop or provide full HTML support while a user is viewing local and network resources with either Internet Explorer or Windows Explorer.

CHAPTER 12

Start Menu and Taskbar

In this chapter

Start Menu and Taskbar Overview 148
Configuring the Start, Favorites, and Programs Menus 155
Configuring the Taskbar and Toolbars 157
Importing Custom Toolbar Configurations for User Groups 159

Start Menu and Taskbar Overview

With Internet Explorer 4, the Windows **Start** menu and taskbar become Web savvy. They now contain many of the commands and functions that make sense in this more Web-centric world.

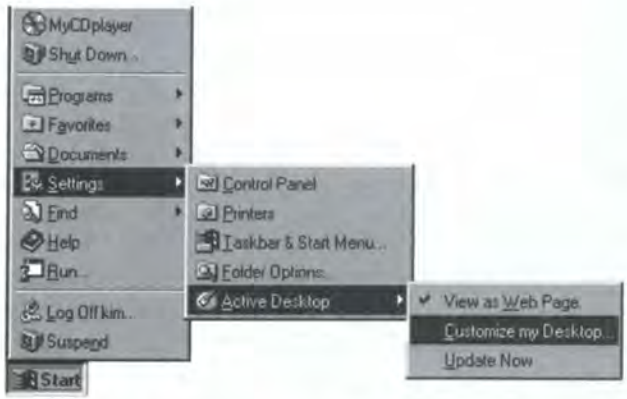
The Internet Explorer **Favorites** menu is now part of the **Start** menu, giving users easy access to their favorite URLs and channels.

Just as users can rearrange the order of their favorites in the Favorites bar and **Favorites** menu inside the browser, they can do the same in the **Favorites** menu on the **Start** button. Internet Explorer 4 also adds this functionality to the **Programs** menu, allowing users to rearrange the order of their applications by dragging them where they prefer.

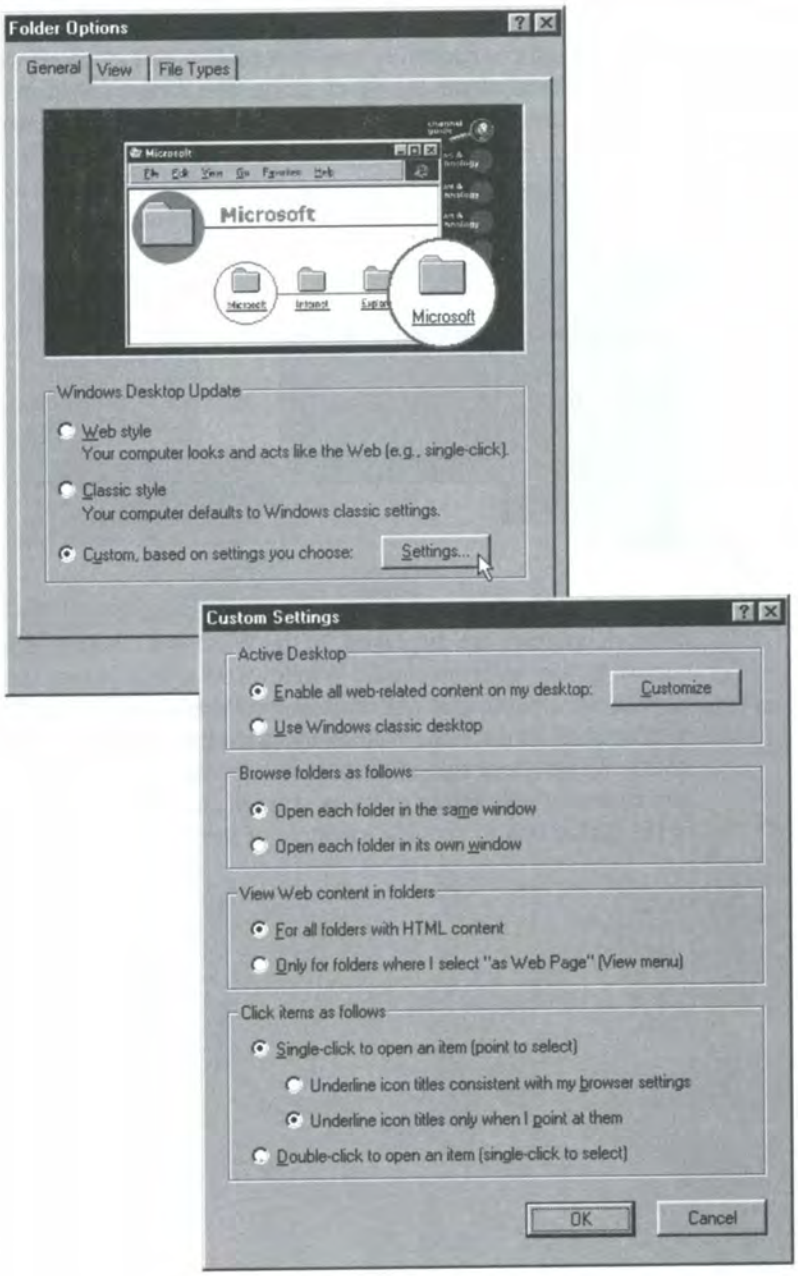
Chapter 18, "Outlook Express," describes support for Lightweight Directory Access Protocol (LDAP) directory services. Essentially, these services provide access to the Internet virtual white pages, which makes it easy to find anyone on corporate LDAP servers or on the Internet. In order to make finding e-mail addresses for individuals even easier, the **Find** command on the **Start** menu has a new **People** option, which enables users to perform LDAP searches directly from the **Start** button.

On the Internet is another new addition to the **Find** command. The **On the Internet** option will launch the Web browser and connect to your default search engine page, which is normally Allinone.htm.

The **Settings** command has also been integrated with Internet Explorer 4, so it is now possible to control settings for folders and icons, as well as the Active Desktop, directly from the **Start** menu. The Active Desktop can be easily activated, turned off, or updated directly from the **Start** button.



The **Folder Options** dialog box gives users a way to control how their local and network contents are displayed. In the Windows Desktop Update pane, users can choose to stay with the classic Windows 95 style of using their file system or have the computer look and act like the Web. For example, folders can now be displayed all in the same window or in multiple windows. Files can be launched with a Web-style single-click or remain with existing double-click action. Also, any custom combination of these settings can be applied. Changes are illustrated in the dialog box to provide a sample of the new user interface.


Folder Options
General
View
File Types
Microsoft
Windows Desktop Update
Web style
Your computer looks and acts like the Web (e.g., single-click).
Classic style
Your computer defaults to Windows classic settings.
Custom, based on settings you choose:
Settings...
Custom Settings
Active Desktop
Enable all web-related content on my desktop:
Customize
Use Windows classic desktop
Browse folders as follows
Open each folder in the same window
Open each folder in its own window
View Web content in folders
For all folders with HTML content
Only for folders where I select "as Web Page" (View menu)
Click items as follows
Single-click to open an item (point to select)
Underline icon titles consistent with my browser settings
Underline icon titles only when I point at them
Double-click to open an item (single-click to select)
OK
Cancel

In Windows 95, the taskbar was simply the place where users could find the **Start** button, taskbar buttons for their open applications, the clock, and other utilities. With Internet Explorer 4, the taskbar has become extensible to support expandable, dockable, layered browser toolbars. Plus, custom toolbars can contain any sort of content, including application shortcuts or Web pages and support for full dragging functionality.

The new toolbars include:

- **Quick Launch toolbar** A quick and easy place to launch commonly used applications and documents. Internet Explorer will provide a few default applications, but the Quick Launch toolbar can be customized simply by dragging any icon onto the toolbar. Also, any icon can be deleted by right-clicking the icon and choosing **Delete**. Notice the new **Show Desktop** button, which quickly hides everything so users can get instant access to the desktop. Pressing it again restores the applications to their previous state.

- **Address bar** The address bar makes the Internet, intranet, and local network servers easily accessible at all times. Users can type in any URL before launching their Web browser, and Internet Explorer 4 will automatically take them to their desired location. It can even be used as a modified **run** command line as users enter UNC names or local content file names (C:\Windows) or URLs.
- **Quick Links bar** The same Quick Links bar that is viewed inside Internet Explorer 4 can also be displayed on the taskbar. This makes it easy to get to the places that users visit most often, without launching the Web browser first.
- **Desktop bar** This custom toolbar dynamically displays all icons that are present on the desktop. This way, if users have multiple applications open that are covering their desktop, they can still get access to those files or applications simply by clicking on the appropriate icon on the Desktop bar.

- **New Toolbar** By using the **New Toolbar** command, users can generate their own custom toolbars simply by choosing a folder or entering a URL. If a folder is chosen, the contents of the folder will be displayed in the toolbar. If a URL is chosen, the toolbar will display the corresponding Web page.

Any of these toolbars can be set up with **Always On Top** or **AutoHide** turned on. This prevents toolbars from taking up available space on the desktop, while still providing quick access to them.

Administrators can import custom toolbar settings to users' machines using the Internet Explorer Administration Kit (IEAK). Users can configure each of these options by right-clicking the taskbar and selecting the appropriate menu option, unless access has been restricted by the administrator.

For information about troubleshooting the **Start** menu and taskbar, see Appendix D.

Configuring the Start, Favorites, and Programs Menus

The Internet Explorer 4 Windows Desktop Update simplifies the process of customizing the contents of **Start**, **Favorites**, and **Programs** menus. System administrators can pre-configure custom settings for these menus and import them to user groups. To create custom menus for user groups, first configure the menus on the computer with the IEAK, and the settings will be carried over to users when the IEAK Configuration Wizard is run. Take note, however, that these settings cannot be adjusted in the IEAK Profile Manager.

When customizing the **Start, Favorites,** and **Programs** menus, the following points should be considered:

- Add the folders, documents, and applications that the workgroup uses most frequently to the **Start** menu. You also may want to add some of these items to the Windows taskbar, the Quick Launch toolbar, or to new toolbars that you create.
- Add all of the workgroup's custom program groups and applications to the **Programs** menu. Whenever a Windows 95–compatible program is installed on a computer, Windows creates a program group for that application on the **Programs** menu. However, you might want to create a program group called Inventory that contains only the inventory applications specific to the workgroup.
- If the workgroup accesses distributed applications that use the Distributed Component Object Model (DCOM), you might want to customize application shortcuts to point to the appropriate local component or to a component on a network server.

The **Start** menu reads the folder location out of the following registry key and displays the contents on the **Start** menu:

HKEY_USERS\.Default\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

If the folder specified in this registry entry is missing or corrupt, then the **Favorites** menu will be empty on the **Start** menu. The registry entry is tied to the Long File Name of the Favorites folder, so if this value does not match the registry, you lose the items on the **Start** menu.

▶ **To edit the contents of the Start and Programs menus**

1. Click the **Start** button, point to **Settings,** and then click **Taskbar & Start Menu**.
2. In the **Taskbar Properties** dialog box, click the **Start Menu Programs** tab.
   - Click **Add** or **Remove** to edit the contents of the **Start** menu.
   - Click **Advanced** to bring up the **Start** and **Programs** menus in Windows Explorer.

▶ **To edit the contents of the Favorites menu**

The contents of the Favorites folder can be edited from within Internet Explorer or from within the folder itself.

- **To edit Favorites from within Internet Explorer**
  1. Open Internet Explorer.
  2. Click the **Favorites** menu.
  3. Select **Add to Favorites** to add the current URL to the folder, or select **Organize Favorites** to edit the contents of the folder.
- **To edit the Favorites folder from within the folder itself**
  - Open **C:\Windows\Favorites**, and add or remove files, as desired.

    You will notice that Channels and Links are subfolders within the Favorites folder.

▶ **To reorder items on the Favorites and Programs menus**

1. Click the **Start** button, and then point to the folder you want to change.
2. On the folder's submenu, click the item you want to move, and drag it to the new location.

The location information is only saved at exit time, so you have to close Windows and log on again for changes to be written to the registry. If a crash occurs during a session, you may lose the location data.

The arrangement data is stored in the registry at:

HKEY_CURRENT_USERS\Software\Microsoft\Windows\CurrentVersion\Explorer\Start Menu\Menu

Configuring the Taskbar and Toolbars

You can configure the taskbar to simplify a user's or workgroup's access to files, applications, and URLs. There are four ways to do this:

- **Customize the Quick Launch toolbar.** You can customize the default Quick Launch toolbar by adding files, folders, applications, and URLs.
- **Customize the default Windows taskbar.** You can customize the Windows taskbar by adding files, folders, applications, and URLs.
- **Creating new toolbars.** You can create new toolbars containing files, folders, applications, and URLs, and then add them to the Windows taskbar or the Active Desktop.
- **Add toolbars as windows on the Active Desktop.** You can add toolbars directly to the Active Desktop as toolbar windows that list the contents of the toolbar in a menu.

When customizing the Quick Launch toolbar, Windows taskbar, or creating new custom toolbars, an administrator should consider the following:

- Decide which files, folders, applications, and URLs the workgroup uses most frequently.
- Determine the best method of presentation for the workgroup. For example, if the workgroup consists of writers working on a specific Microsoft® Word document, you might want to create a shortcut to Microsoft Word or to that document on the Quick Launch toolbar. If the workgroup regularly accesses a folder of related files, applications, or URLs, you might want to create a toolbar containing the contents of that folder and place it on the Windows taskbar or as a floating toolbar on the Active Desktop.
- If you want to create new toolbars, group the files, applications, and URLs that the workgroup uses most frequently into one or more appropriately named folders. A new toolbar consists of the contents of a folder on a local or network drive, and the name of the toolbar will be the same as the name of the folder.

▶ **To customize the Quick Launch toolbar**

- Drag a file, folder, application, or URL from My Computer or Windows Explorer to the Quick Launch toolbar, or add files directly to the Quick Launch folder in: <windir>\Application Data\Microsoft\Internet Explorer\Quick Launch.

▶ **To add the Address toolbar**

1. Right-click the taskbar, and select **Toolbars,** then **Address**.

   The Address toolbar will appear in the taskbar.
2. You can drag the Address toolbar to another location on the desktop and resize it, as necessary.

▶ **To add and customize the Quick Links toolbar**

1. Right-click the taskbar, and select **Toolbars**, then **Links**.

   The Links toolbar will appear in the taskbar.
2. You can drag the Links toolbar to another location on the desktop and resize it, as necessary. Items can be added and removed by dragging them in and out of the toolbar.

▶ **To add and customize the Desktop toolbar**

1. Right-click the taskbar, and select **Toolbars**, then **Desktop**.

   The Desktop toolbar will appear in the taskbar.
2. You can drag the Desktop toolbar to another location on the desktop and resize it, as necessary. Items can be added and removed by dragging them in and out of the Desktop toolbar or the desktop itself.

▶ **To create a custom toolbar**

- **To create a toolbar from the contents of a file folder**
  1. Right-click the taskbar, and select **Toolbars**, then **New Toolbar**.

     The **New Toolbar** dialog box will appear, showing a hierarchical display of the files on your machine.
  2. Create a file folder, and add the files, folders, and shortcuts you want its toolbar to contain.
  3. Select the folder, and click **OK**.

- **To create a toolbar from the contents of a Web page**
  1. Right-click the taskbar, and select **Toolbars**, then **New Toolbar**.

     The **New Toolbar** dialog box will appear, showing a hierarchical display of the files on your machine.
  2. Enter the URL in the text box, and click **OK**.

     The custom toolbar will appear in the taskbar. You can drag the new toolbar to another location on the desktop and resize it, as necessary. Items can be added and removed by dragging them in and out of the toolbar, or by changing the contents of the folder itself.

Importing Custom Toolbar Configurations for User Groups

System administrators can use the IEAK Configuration Wizard to import custom packages with pre-configured toolbar configurations for user groups. (For more information, see "Building Custom Packages" in Chapter 36. Administrators can also restrict users from changing custom toolbars.

▶ **To add custom toolbar settings to user group configuration**

- At the **Customize Desktop Toolbars** page of the IEAK Configuration Wizard, select **Import the current desktop Toolbar settings**.

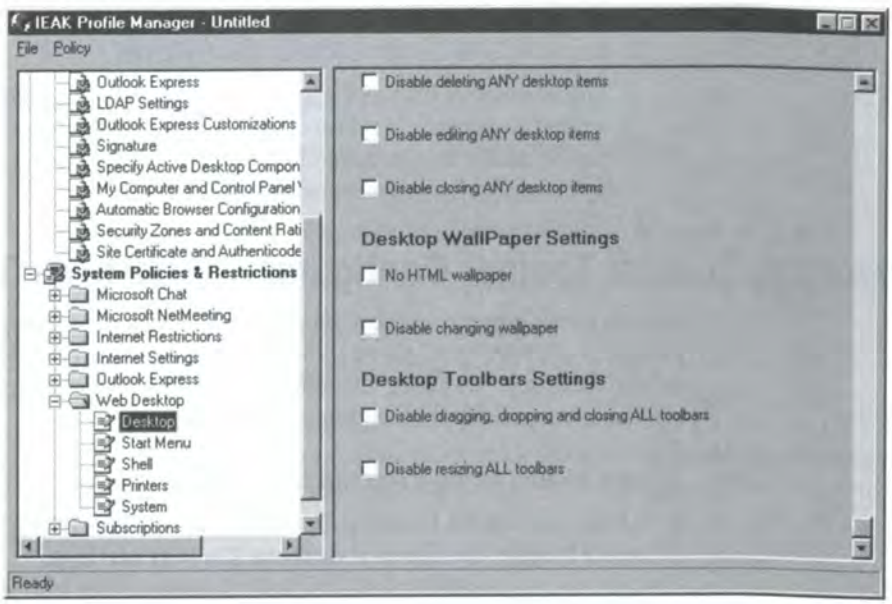
  Your current desktop settings will be applied to the installation.

▶ **To lock down custom toolbars**

- **In the IEAK Configuration Wizard**
  1. At the **System Policies & Restrictions** page, expand the Active Desktop item.
  2. Scroll down to Desktop Toolbar Settings, and select the desired options. Check boxes will offer the option to **Disable dragging, dropping, and closing of ALL toolbars** and **Disable resizing of ALL toolbars**.

- **In the IEAK Profile Manager**

  1. In the left pane, in the System Policies and Restrictions hierarchy, select **Web Desktop**, and then click **Desktop**.



Note The IEAK Profile Manager is shown here, but the IEAK Configuration Wizard works similarly.

  2. In the right pane, under Desktop Toolbars Settings, select the desired options. Check boxes will offer the option to **Disable dragging, dropping and closing of ALL toolbars** and **Disable resizing of ALL toolbars**.

CHAPTER 13

Web View

In this chapter

Understanding Web View 162
Custom Web View Samples 163
Web View Architecture 164
Customizing Web View 165
Customizing a Folder with the Web View Wizard 166
Customizing the Default Folder.htt File 167
Customizing the Default Left Pane 167
Adding Links to the Default Left Pane 167
Customizing the Default Header Graphic 168
Replacing the Default Folder.htt with a Custom HTML Page 168
Customizing Printers.htt and Safemode.htt 170
Customizing Web View for My Computer and Control Panel 170
Selecting Thumbnail View in a Folder 171

Understanding Web View

Web View folders use Hypertext Markup Language (HTML) to display information in Windows Explorer. (Remember, Windows Desktop Update must be installed in order for Web View to work.) You can accept the Internet Explorer 4 default Web View, or system administrators can customize their own Web Views for the folders they choose. Standard HTML, Dynamic HTML, graphics, ActiveX objects, JavaScript, and VBScript can all be used to customize Web View folders, allowing you to leverage existing development resources.

A customized Web View can be used to give users a richer, more intuitive experience. Navigation can be vastly improved, and greater graphic consistency can be achieved. For example:

- A software download site on a network server can be simplified using Web View. Instead of being confronted with a confusing assortment of different files, users will see only links to things they can use.
- Crucial information can be prominently displayed in a folder, so it can't be missed by users.
- The appearance and functionality of Web View folders can be redesigned to match online resources, so users will have a consistent experience whether using file folders, Web pages on an intranet, or Web pages on the Internet.

For information about troubleshooting the Web View, see Appendix D.

Custom Web View Samples

The following samples illustrate some useful applications of Web View. The samples include solutions for three hypothetical scenarios and one stand-alone example. All samples can be run directly from the Internet Explorer 4 Resource Kit Compact Disc. To view the samples, simply click **Start**, then **Run**, and enter **D:\Samples** (substituting the letter of your CD-ROM drive if it is different than D). On the **View** menu, select **View as Web Page**.

- **Scenario #1: Arcadia Bay Travel Resources**

  Location: D:\Samples\travelpublic

  Many useful features of Web View have been incorporated into a special business-travel folder for this fictitious company. This example illustrates the potential for presenting information and resources in an intuitive manner while still retaining traditional Windows functionality. The FileList control has been used to make a file folder available to users in a familiar way.

- **Scenario #2: Acme Software Driver Floppy Disk**

  Location: D:\Samples\driverfloppy

  This example shows how Web View can transform an ordinary (and potentially confusing) floppy disk into an attractive one-click solution for a hardware driver installation. Try switching between Web View and the normal file views. Which view would you rather use?

- **Scenario #3: Software Installation Folder**

  Location: D:\Samples\products1

  This company has a potentially bewildering array of software available for its employees to install. In the days before Internet Explorer 4, users would have to hunt through a maze of cryptically named folders and then guess which installation option was the right one. Here, Web View and Dynamic HTML have been applied to help users find the software they want and get the information they need to know before installing it.

- **Media Preview Folder**

  Location: D:\Samples\preview

  This folder uses the ActiveMovie™ control to give a preview of sound and video files stored inside it.

Note The names of companies, products, people, characters, and/or data mentioned herein are fictitious and are in no way intended to represent any real individual, company, product, or event, unless otherwise noted.

Web View Architecture

To create the new, Internet Explorer 4 Web View, an extended view was added to the Windows Explorer default implementation of the **IShellView** interface (the shell code responsible for displaying files and folders in the shell's name space—the Large Icons, Small Icons, List, and Details views that you are already familiar with in Windows 95). To view a folder in the default Web View, all you have to do is select the **View as Web Page** option in the **View** menu. The default Web View for all folders is defined by the HTML in folder.htt. (.htt is the file name extension for HTML Template files.)

All Web content for Web View and Web/shell integration is stored in the C:\Windows\Web folder. These files are used to create the appearance and functionality of the Web/shell integration and maintain the user's desktop. The files that provide the main functionality are:

Mycomp.htt	HTML template file used to create the Web View for the My Computer folder.
Controlp.htt	HTML template file used to create the Web View for the Control Panel folder.
Deskmovr.htt	HTML template used to load the Deskmover.ocx file that controls the Active Desktop.
Folder.htt	The default HTML template for Web View in all folders. This file is duplicated when a folder is customized.
Printers.htt	HTML template file used to create the Web View for the Printers folder.
Safemode.htt	HTML template file that displays the Active Desktop Safe Mode (seen when Explorer.exe is shut down, then auto-restarted by the system).

Customizing Web View

Folders can be individually customized by opening the folder and selecting the **Customize This Folder** option in the **View** menu. When this option is selected, the Web View Wizard is launched. After the wizard is completed, two new files are created in the target folder: Desktop.ini and Folder.htt.

Desktop.ini

The Desktop.ini file is a standard .ini file, used to tell Internet Explorer that a folder has a customized Web View. Each customized folder has one Desktop.ini file. The shell reads the Desktop.ini, which directs the shell to host the MSHTML.dll component of Internet Explorer 4 and points it to the appropriate .htt file.

This is a sample Desktop.ini file:

```
[ExtShellFolderViews]
Default={5984FFE0-28D4-11CF-AE66-08002B2E1262}
{5984FFE0-28D4-11CF-AE66-08002B2E1262}={5984FFE0-28D4-11CF-
    AE66-08002B2E1262}
[{5984FFE0-28D4-11CF-AE66-08002B2E1262}]
PersistMoniker=file://folder.htt
[.ShellClassInfo]
ConfirmFileOp=0
```

Folder.htt

The Folder.htt file is a template that contains customized HTML content and an ActiveX control to display the normal directory listing. Each customized folder has one Folder.htt file. It is the same file referenced in the **PersistMoniker=file://folder.htt** desktop.ini setting earlier on this page. To create Folder.htt, the Web View Wizard launches your HTML editor with the <windir>\Web\Folder.htt template. If you save the default HTML provided by the wizard, the folder will appear the same as it did before you ran the wizard. The Folder.htt template file can be edited at any time, and it is liberally commented so you can find your way around in the file. A sample of an uncustomized Folder.htt file can be found in the <windir>\Web\directory.

The default Folder.htt page is a robust example of what can be done with Web View. Most of the functionality is provided by JavaScript and ActiveX controls. We recommend that you make changes to this file carefully and incrementally. Be sure to make a backup of the file before you begin editing.

Customizing a Folder with the Web View Wizard

When initially customizing Web View, we recommend that you use the Web View Wizard, since it automatically creates the Folder.htt and Desktop.ini files that will reside in the customized folder. After that, you can open and edit Folder.htt with the editor of your choice.

▶ **To customize a folder using the Web View Wizard**

1. Open the folder you want to customize.
2. On the **View** menu, select **Customize this Folder**.
3. Select **Create or Edit an HTML Document**, and then click **Next**.

   Your default HTML editor will start with the Folder.htt file open.
4. Insert the code for your custom HTML page.
5. Save and close the file.
6. Click **Finish**.

   The Folder.htt and Desktop.ini files will appear in the folder.
7. On the **View** menu, choose **View as Web Page** to test the new page.

Now, each time you run the wizard for that folder, it will automatically open that folder's Folder.htt file.

▶ **To remove customization from a folder**

After editing Folder.htt, you can remove customization to return the folder to its unedited state.

1. Open the folder you want to edit.
2. On the **View** menu, select **Customize this Folder**.
3. Select **Remove customization,** and then click **Next**.
4. Click **Next** again to proceed.

   The Folder.htt and Desktop.ini files will be removed from the folder.

Note To change the default Web View for *all* file folders, open the \Windows\Web folder and customize the master Folder.htt file. Afterwards, if you want to undo your customization of this master file, open the Windows\Web folder, and go through the standard process for removing customization from a folder; Folder.htt will be replaced by a backup file tucked away inside Webvw.dll.

Customizing the Default Folder.htt File

Folder.htt files may be edited in order to accomplish a variety of customization goals. For example, you may want to:

- Customize the file information that will be displayed in the folder's left pane when a file is selected.
- Add links to other files or URLs.
- Customize the header graphic.

To undo any edits you make to a Folder.htt file, follow the steps for removing customization from a folder, detailed in the previous section.

Customizing the Default Left Pane

The file information that is displayed in the left pane is controlled by a JavaScript function at the top of the Folder.htt file. The first few lines look like this:

```
// name
        text = "<b>" + fldr.GetDetailsOf(items, 0) + "</b>";
// type
        data = fldr.GetDetailsOf(items, 2);
        if (data != "")
            text += "<br>" + data;
```

This part of the JavaScript function is responsible for displaying the name and type of the file in the left pane. Notice the <b> tag that makes the name appear in bold, and the "if" statement that inserts a line break if the file property is null. If your files have additional properties, you can display these by adding new statements to this function, substituting the appropriate number for the property after "items" (items, *x*). File properties start at "0" (name) and increase by one for each column to the right. If you wish to prevent a property from being displayed, simply remove its section from the function.

Adding Links to the Default Left Pane

To add hyperlinks to the left pane, open Folder.htt, and search for the word "here." The following lines of text will be displayed:

```
<!-- HERE'S A GOOD PLACE TO ADD A FEW LINKS OF YOUR OWN -->
    <!-- (examples commented out)
    <a href="http://www.mylink1.com/">Custom Link 1</a>
            <br>
    <a href="http://www.mylink2.com/">Custom Link 2</a>
            <br><br>
    -->
```

Remove the comment, tags and replace the dummy link information with your own links.

Customizing the Default Header Graphic

The default header graphic is read from the Webvw.dll file. You can substitute it with a graphic of your choice by replacing this line in Folder.htt:

```
background: URL(res://Webvw.dll/folder.gif)
```

with something like this:

```
background: URL(%THISDIRPATH%\mylogo.gif)
```

Replace "mylogo.gif" with the name of your graphic. In this example, the graphic is in the same path as Folder.htt. You may need to modify some of the HTML code to adjust the positioning of the graphic.

You can point to a graphic stored in the template directory (C:\Windows\Web) by using %TEMPLATEDIR% in place of %THISDIRPATH%.

Replacing the Default Folder.htt with a Custom HTML Page

Rather than editing the default Folder.htt file, as explained above, you may prefer to replace it entirely with a new, custom HTML page. When you replace Folder.htt with a custom HTML page, you have the choice of reusing some of the code from the Folder.htt file or completely throwing it away and starting from scratch.

Note We do not recommend that you replace the My Documents folder with a custom Web page, due to the way this folder has been designed to interact with Windows. It is still possible, however, to customize the default Folder.htt file to add links and background graphics and to edit the information displayed in the folder's left pane.

The contents of Folder.htt can be replaced by any HTML code, scripts, or ActiveX objects. Although you will use the same tools and techniques as those used to develop traditional Web pages, keep in mind that Web View can incorporate functionality that is not used in the Internet environment. The following guidelines can help you make the most of Web View:

- Always save your custom file with the .htt file extension. This way, the .htt MIME filter will be activated to let you use things such as variable substitution strings. If you change the name of the file, be sure to update the PersistMoniker line in Desktop.ini to point to the new file. Some variable substitution strings that can be directly incorporated into the HTML code are:
  - **%THISDIRPATH%** (inserts the directory path of the folder).
  - **%THISDIRNAME%** (inserts the name of the folder).
  - **%TEMPLATEDIR%** (inserts the template directory path, usually C:\Windows\Web).

- When creating hyperlinks to files and subfolders, use the HREF tag, but omit the file-name extension for subfolders:

  For files: `<A HREF="filename.ext">`

  For subfolders: `<A HREF="foldername">`

- To display the contents of a folder, insert the FileList object:

```
<OBJECT ID="filelist" classid="clsid:EAB22AC3-30C1-11CF-A7EB-
0000C05BAE0B" width=150 height=470>
    <PARAM NAME="Location" value="%THISDIRPATH%\folder">
    <PARAM NAME="AlignLeft" value=1>
    <PARAM NAME="AutoSize" value=7>
    <PARAM NAME="AutoSizePercentage" value=100>
    <PARAM NAME="AutoArrange" value=1>
    <PARAM NAME="NoClientEdge" value=true>
    <PARAM NAME="ViewMode" value=3>
</OBJECT>
```

The parameter values of the FileList control affect the following settings:

- **Location** defines the path to the folder to be displayed.
- **AlignLeft/AlignRight** defines the alignment of the items when displayed.
- **AutoSize.** Leave this value set to 7.
- **AutoSizePercentage.** Leave this value set to 100.
- **AutoArrange.** A value of 0 allows items displayed in Large Icon or Small Icon views to be moved, while a value of 1 enforces autoarrange at all times.
- **NoClientEdge** defines whether or not the control will have a visible border.
- **ViewMode** defines the view in which the contents of the folder will be displayed (1=Large Icons, 2=Small Icons, 3=List, 4=Details).

Caution When using this control, NEVER set the Location parameter value to the same folder that is hosting the control, as this could cause a looping VBScript error. Another important thing to keep in mind when using the FileList control is that whatever folder is referenced in the Location parameter will appear in Web View. Pointing the FileList control to uncustomized folders is the safest approach.

Customizing Printers.htt and Safemode.htt

You can customize the Web View of the Printers folder by modifying the Printers.htt file in the C:\Windows\Web directory. A nice addition to this folder might be contact information for technical assistance with printers and a link to an alias to report problems. The Printers.htt folder is set up similarly to Folder.htt, with a few minor differences.

The Safemode.htt HTML template file is displayed only when Explorer.exe is shut down and restarted by the system for any reason. It has a special control that is used to reset Active Desktop settings, along with instructions for what to do in this situation. This page could be modified to include additional support information for users. It is recommended that the reset object not be tampered with, to retain the functionality of this page. The reset object is identified by the following code:

```
<OBJECT ID="suih" width=0 height=0 classid="clsid:64AB4BB7-111E-
    11d1-8F79-00C04FC2FBE1">
```

Use caution when editing either of these folders, as they provide vital functionality to users.

Customizing Web View for My Computer and Control Panel

The My Computer and Control Panel folders must be edited differently than other folders. Since they are virtual folders, Desktop.ini and Folder.htt files cannot be created for them. The Web versions of these folders are controlled by registry entries that point to HTML files located in the Web folder (<windir>\Web\). They are called Controlp.htt and Mycomp.htt, respectively.

The registry settings for Control Panel are:

HKEY_LOCAL_MACHINE\Software\Classes\Clsid
\{21EC2020-3AEA-1069-A2DD-08002B30309D}
\shellex
\ExtShellFolderViews
\{5984FFE0-28D4-11CF-AE66-08002B2E1262}

**Default value (Win95): "\Web\controlp.htt"
**
Default value (WinNT): "%systemRoot%\Web\controlp.htt"</lt>

The registry settings for My Computer are:

HKEY_LOCAL_MACHINE\Software\Classes\Clsid
\{20D04FE0-3AEA-1069-A2D8-08002B30309D}
\shellex
\ExtShellFolderViews
\{5984FFE0-28D4-11CF-AE66-08002B2E1262}

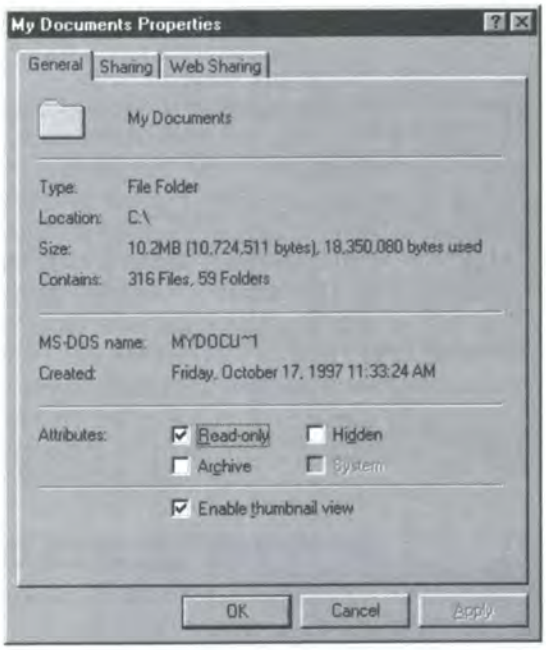
**Default value (Win95): "\Web\mycomp.htt"
**
Default value (WinNT): "%systemRoot%\Web\mycomp.htt"</lt>

Selecting Thumbnail View in a Folder

With Internet Explorer 4, it is possible to create a thumbnail view of the content within any given directory. Thumbnails are a good way to get a quick snapshot of multiple Web sites or graphics files at the same time. When Thumbnail view is selected, the contents of graphics files, HTML files, and Office Documents are displayed in a 2 by 2-inch window. An icon is displayed for files that cannot display their contents in a thumbnail. A file called Thumbvw.dll controls the Thumbnail view. This feature is turned on in the **Properties** dialog box of each folder.

▶ **To enable Thumbnail view**

1. Right-click the folder for which you would like to enable Thumbnail view, and then select **Properties** on the shortcut menu.
2. On the **General** tab, select **Enable thumbnail view**.



3. Click **OK** to add Thumbnails to the folder's view options.
4. Open the folder and, on the **View** menu, select **Thumbnails**.

   The contents of the folder will be displayed in Thumbnail view.

Note You cannot combine Thumbnail view and Web View for the same folder. Due to the complexity of each of these display options, they are mutually exclusive. If Thumbnail view is selected for a folder, Web View will automatically be turned off for that folder (although subfolders may still use Web View, as long as they don't also use Thumbnail view).

CHAPTER 14

Active Desktop

In this chapter

Understanding the Active Desktop 174
- Historical Background of the Active Desktop 174
- Layers of the Active Desktop 175
- The Active Desktop and Webcasting 176

Customizing the Active Desktop 177
- Customizing Active Desktop Wallpaper 178
  - HTML Wallpaper Samples 178
- Adding Channel Screen Savers 179
- Configuring Active Desktop Items 180

Developing Active Desktop Items 182
- Designing Active Desktop Items 182
- Creating CDF Files for Active Desktop Items 185
  - CDF Example for an Active Desktop Item 186
- Offering the Active Desktop Item to Users 186

Understanding the Active Desktop

The Internet Explorer 4 Active Desktop interface is capable of hosting any Hypertext Markup Language (HTML) item, such as Web pages, Java applets, ActiveX controls, floating frames, and images. With this technology, users can create a highly customized environment that automatically brings their favorite Web content to their desktop. Web publishers can leverage their existing tools and production processes to enhance their Web content with Dynamic HTML support. The resulting desktop and Web integration benefits users through a richer and easier Web experience. Web publishers also benefit from the increased presence in the users' working environment.

Active Desktop technology for Internet Explorer 4 marries the best of the Internet to the Windows shell. It creates a unique opportunity to push and pull content and applications from internal and external Web sites, providing a powerful, customizable, information delivery architecture. Essentially, the Active Desktop allows users to place Web sites directly on their desktops, in the form of either Active Desktop items, wallpaper, or screen savers. You can also add an item from the Microsoft Active Desktop gallery, such as a Java clock or a stock ticker.

Historical Background of the Active Desktop

The seeds for integrating the Windows shell and browser were planted with the release of Internet Explorer 3, which componentized the Web browser as an ActiveX control so it could be embedded in a Visual Basic form or an HTML page. But to completely integrate the browser component with the container's frame, the browser piece would have to be extended further; it would need to become an Active Document server. (For more information, see the *Internet Client Software Development Kit.*) Active Documents make it possible to view non-HTML content without leaving the browser, and an Active Document server is simply an OLE Document server with additional interfaces that enable embeddable in-place objects to take over the entire document window. Active Documents thus unified the browsing model for applications, documents, and objects.

The brainstorm that drove design of the Active Desktop was to turn the Windows desktop into an ActiveX Document container. The Active Desktop doesn't include all of the browser's user interface elements, but as you will see, you can do quite a bit with an HTML view of the world.

Layers of the Active Desktop

The Active Desktop in Internet Explorer 4 is built from two separate layers. The desktop icon layer exposes all the user's existing desktop shortcuts, and the background HTML layer hosts the HTML wallpaper or screen saver and all Active Desktop items. The HTML layer is described by a single, local HTML file called Desktop.htm, which is created and edited automatically by Internet Explorer 4. This file contains the following:

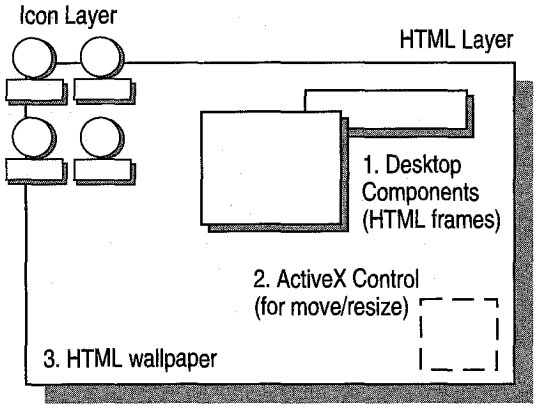
- HTML tags that represent each Active Desktop item. Each Active Desktop item consists of a single HTML tag with arbitrary x- and y-positions. The HTML tag used for an Active Desktop item can be either an image (IMG tag) or a floating frame (IFRAME tag) and is generated automatically by Internet Explorer 4. The floating frame is the most commonly used approach, because it neatly encapsulates an entire arbitrary HTML document that can contain anything the publisher desires. In either case, a single URL points to the actual content.
- An ActiveX control that enables moving and resizing of the Active Desktop items and also helps manage the list of items.
- Any other static HTML that the user wants to have in the background. By default, this is just a reference to the user's chosen wallpaper, which is exposed as the background watermark for the HTML page.

Active Desktop items provide dynamic links from the desktop to Web content. An Active Desktop item can be an ActiveX control, a Java applet, an image, or an HTML document. Because these are all essentially things that you can put in an HTML document, they are no different from the HTML wallpaper, except that they live in a layer above it. You can see this by dragging a desktop item around your Active Desktop; it will always be drawn underneath the icons and on top of the wallpaper. Active Desktop items are hosted in an instance of the WebBrowser control. An Active Desktop item can act as a desktop shortcut or a quickly referenced source of regularly updated information. When combined with Web site subscriptions (explained in detail in Chapter 15, "Basic Webcasting.") you can present timely information to your users without requiring any effort on their part.

The Active Desktop is the part of the Windows shell that keeps track of Active Desktop objects. The persistent information about installed desktop items is located in the registry at

HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Desktop

The desktop icon layer extends all the features of standard Web-page hyperlinks, such as single-click navigation and hot tracking, to your desktop icons. Plus, it extends this Web-like feel to the familiar features in the Windows 95 shell, such as click-through transparency, dragging, file-type associations, double-clicking, and more.



The Active Desktop and Webcasting

You can easily place any ActiveX control, Java applet, graphic, or HTML document on your desktop and leave it at that. To get the most from the Active Desktop, however, you'll want to utilize the Webcasting technologies that are available in Internet Explorer 4. Part 4, "Information Delivery: Webcasting," provides a complete explanation of these features, but a few ideas are important to understand now.

Active Desktop items and Active Desktop wallpaper are, essentially, Web sites that appear directly, in borderless frames, on users' desktops. Once a Web site is placed on the Active Desktop, the site's content is cached on the computer's hard disk, so it's always available for offline viewing. The user can then choose to "subscribe" to the Web site, and the Active Desktop item or wallpaper will be automatically updated at regularly scheduled times, so the content will always be up-to-date.

For optimum control of the update schedule and the type of content that is downloaded during these updates, Channel Definition Format (CDF) files can be used to create Active Channels. Again, Part 4 offers a complete explanation of Active Channels and CDF files. For our purposes now, it is only necessary to understand that CDF files can be used to create custom Active Desktop items and wallpaper and also to bring channel screen savers to the Active Desktop.

It's important to note that Webcasting is typically used to deliver Web content that appears in a subscriber's browser window, while Active Desktop items are used to deliver miniature Web pages that appear directly on subscribers' desktops. Desktop items live on the Windows desktop, alongside existing desktop shortcuts. A desktop item lets you create dynamic links to a workgroup's favorite Web content. Desktop items are typically designed to provide summary information in a small amount of screen space. Therefore, information in desktop items often includes hyperlinks or hot spots, so users can click a designated area to open a browser window and obtain the details they want.

You can configure individual update schedules for each Active Desktop item. For example, a user's Active Desktop can display three different desktop items: an inventory ticker that is updated every 15 minutes, a workgroup home page that is updated once a day, and a newsletter that is updated once a week. Internet Explorer 4 automatically refreshes the desktop item whenever the update occurs.

Customizing the Active Desktop

One of the best benefits of the Active Desktop for administrators is the ability to add custom HTML wallpaper, screen savers, and Active Desktop items for workgroups. This provides a way to display relevant, up-to-date information directly on users' desktops. HTML wallpaper and screen savers can be targeted to specific user groups, and Active Desktop items can be used to complement the information presented there.

For example, a company with many different departments can create a custom HTML page specific to each department. This page might include links to files, folders, and URLs of interest to the specific department, as well as company-wide resources such as organizational charts, a human resources intranet site, and the company's official Internet site. Active Desktop items could be created to alert employees of company news and events and provide a top-level summary of news feeds and trade-related information. A special Active Desktop item could be targeted to new employees that would make it easier to set up their computers and access company resources. Using Webcasting, the new user information could be automatically replaced after a specified period of time. A smaller organization might simplify this approach, by having one page for all employees.

To import a custom Active Desktop to user groups, first configure the Active Desktop on the computer with the Internet Explorer Administration Kit (IEAK), and the wallpaper, screen savers, and desktop items will be carried over to users when the IEAK Configuration Wizard is run. User access to Active Desktop settings can be restricted to prevent users from changing the configurations on their machines.

For information about troubleshooting the Active Desktop, see Appendix D.

Customizing Active Desktop Wallpaper

You can display any HTML page as the wallpaper on the Active Desktop. Remember that the page you select on your configuration machine will be the page seen by all users in the workgroup.

▶ **To use an HTML page as the default Active Desktop wallpaper**

1. Create the HTML page you want to use as wallpaper.
2. Right-click the Active Desktop to display the shortcut menu, and then click **Properties**.
3. Click the **Web** tab, and then select the **View My Active Desktop as a Web Page** check box.
4. Click the **Background** tab, and navigate to the location of the HTML file you created in step 1.
5. Select the file, and click **OK** to display the file.

Tip Try incorporating Dynamic HTML into your Web page to provide a more engaging experience for your users. For more information about Dynamic HTML, see: **http://www.microsoft.com/workshop/author/dhtml/**.

HTML Wallpaper Samples

These samples show how a company can create specialized HTML wallpaper to help meet the information needs of different workgroups. All samples can be run directly from the Internet Explorer 4 Resource Kit Compact Disc. Click **Start**, then click **Run**, and enter **D:\SAMPLES** (substituting the letter of your CD-ROM drive if it is different than D).

The HTML wallpaper samples are located in D:\Samples\desktops. You can view these samples in the browser window by double-clicking the HTML file in the Desktops directory; or, to view the samples directly on the desktop, copy D:\SAMPLES\DESKTOPS to your hard disk, and select the various HTML files as your wallpaper.

Here are two sample desktops that illustrate how custom wallpaper can be used effectively:

- **Corpdesk.htm** contains a floating frame that displays the current date and a short bulletin page. It also features an intranet search window (disabled for this demo).
- **Corpdesk1.htm** features a floating frame that displays current inventory data and a bar graph that displays various sales statistics. Although the information in this demo is read from a delimited text file, it can be easily adapted to interact with a database.

Adding Channel Screen Savers

You can configure the Active Desktop to use a screen saver provided by a subscribed Active Channel Web site, even if that usage is not specified in the Active Channel's CDF file. With a channel screen saver, a user can move the mouse and click objects on the screen without immediately dismissing the screen saver. Clicking a link on the screen saver opens a browser window and dismisses the screen saver.

In a corporate intranet, you might want to configure a channel screen saver that displays a stock ticker, broadcast bulletins, or an inventory counter. For more information about channel screen savers, see Chapter 16. Remember that the page you select on your configuration machine will be the page seen by all users in the workgroup.

▶ **To configure a channel screen saver**

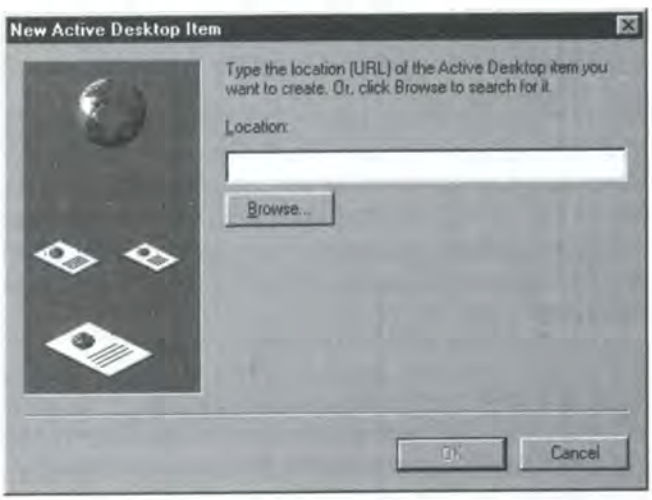
1. Right-click the Active Desktop to display the shortcut menu, and then click **Properties**.
2. Click the **Screen Savers** tab, and then, in the **Screen Savers** dialog box, click **Channel Screen Saver**.

Configuring Active Desktop Items

You can add any graphic, URL, or HTML file to your desktop as an Active Desktop item. Adding a URL will automatically subscribe you to the site you selected. When you subscribe to a URL, site information is automatically cached on your computer for offline viewing, and Internet Explorer 4 will automatically update the desktop item on a schedule you choose. (For more information about Web subscriptions, see Part 4.)

▶ **To add an Active Desktop item**

1. Right-click the Active Desktop to display the shortcut menu, and then click **Properties**.
2. Click the **Web** tab; then click **New**.



3. Enter the URL or path to the item, or click **Browse** to navigate to the file's location on your local or network drive.
4. Click **OK** to add the item to your desktop.

   Alternatively, you can add a desktop item by subscribing to an Active Channel that specifies a desktop component. For more information, see Chapter 16, "Managed Webcasting."

Note To download examples of Active Desktop items, visit the Active Desktop gallery at: **http://www.microsoft.com/ie/ie40/gallery/**.

▶ **To disable or delete an Active Desktop item**

1. Right-click the Active Desktop to display the shortcut menu, and then click **Properties**.
2. Click the **Web** tab.
   - To disable an item, clear the check box next to the item. It will still be there if you want to use it again.
   - To delete an item, select the item, and click **Delete**. This will permanently remove the item from your system.
3. Click **OK**.

System administrators can place Active Desktop items on users' desktops during configuration by setting them up on the configuration machine prior to running the IEAK Configuration Wizard. Users can be restricted from modifying, adding, and removing Active Desktop items by using the IEAK Configuration Wizard or the IEAK Profile Manager.

▶ **To lock down Active Desktop items**

In the IEAK Configuration Wizard or the IEAK Profile Manager, on the **System Policies and Restrictions** page, under **Web Desktop**, select the following options:

- **Disable all desktop items**
- **Disable adding any new desktop items**
- **Disable deleting any desktop items**
- **Disable editing any desktop items**
- **Disable closing any desktop items**

Developing Active Desktop Items

The process for developing a custom Active Desktop item is essentially the same as the process for developing an Active Channel. First, you design the desktop item (the Web page), and then you create its CDF file. In Chapter 16, the "Creating CDF Files" section offers a detailed explanation, with examples, of how to build CDF files, plus some special instructions for creating CDF files for Active Desktop items. Some additional information, specific to desktop items, is detailed below.

Designing Active Desktop Items

When first designing the Active Desktop item, there are some special guidelines to keep in mind. Again, Chapter 16 offers general guidelines for designing Web sites to be used as Active Channels, but if you want the channel to function with optimum effectiveness as a desktop item, also consider these guidelines and tips:

- **Try implementing Dynamic HTML in your Active Desktop item.** It is common for an Active Desktop item to cycle through a set of headlines or summary information. Dynamic HTML makes it possible to achieve this effect without using Java or ActiveX code.
- **The item must work offline.** Since most dial-up users don't have a constant connection to the Internet, an Active Desktop item should display some reasonable content in the absence of a connection. This is important, because the desktop is immediately visible when the user launches the operating system, even before the user has a chance to initiate a dial-up session. An Active Desktop item that displays a blank, gray square or an error message while the user is offline is neither useful or attractive.

  To avoid problems, any necessary data files the item requires should be configured to be cached on the user's computer. The Internet Explorer 4 subscription feature can be configured to cache any object referenced by an Active Desktop item by including it in the CDF file with the ITEM element. Subscriptions can be set to automatically download any images, sounds, objects, or applets, and also any number of HTML links *n* levels deep. It is important to note that Java applets cannot be pre-cached if the Java classes are not wrapped together in a .cab, .zip, or .jar file. If the classes are in separate files, Internet Explorer cannot determine the Java class dependencies and cache appropriately, because Java binds classes dynamically.

Java- and ActiveX-based desktop items should also be authored to give an error or informational message when not connected to the network, to prevent an empty, gray box from being displayed. Also, to read from the cache when not connected to the network, Java applets should call setUseCaches(TRUE) in their java.net objects. Active Desktop items can also specify individual subscriptions for any data files or objects they may reference. Please note that Java applets and ActiveX controls must be signed so they can access content from the offline cache. For more information about signing Java applets or ActiveX controls, see **http://www.microsoft.com/java/** or **http://www.microsoft.com/windowsdna/**.

- **Avoid using in-place navigation controls.** Active Desktop items are meant to be lightweight and therefore have no navigation controls available for the user to move backward and forward. By default, clicking a link inside an Active Desktop item creates a new browser window. The intended purpose of an Active Desktop item is to provide quick summary information and an easy browser-launching mechanism if the user wants more information. Therefore, authors shouldn't create in-place navigation controls, and they shouldn't depend on the Back and Forward commands being available to the item. Active Desktop item authors should not make any assumptions about the context in which the item is running.
- **Smaller is better.** An Active Desktop item needs to share the desktop space with any other icons or Active Desktop items the user has selected. Because this space is completely customizable by the user, Active Desktop items should be authored to be as small as possible to display their desired content. This gives Active Desktop users the maximum flexibility to arrange their desktop as they wish. A good benchmark is to make sure that the Active Desktop item takes up no more than one-sixth the area of the screen. By default, Internet Explorer lays out new Active Desktop items on a 3 by 2 grid. As more items are added to the desktop, they will start to overlap each other. For a desktop at a resolution of 640 x 480, allowing for reasonable spacing between items, this would translate into a maximum size of no more than 200 × 200 pixels. This size specification is merely a guideline. Internet Explorer 4 does not itself enforce any restrictions on the size of an Active Desktop item.

- **Avoid pop-up messages.** Today it is common for ActiveX controls, Java applets, and scripting content to notify the user of error messages or warnings by using modal dialog boxes that overlap the browser window. This is fine when the browser window is in the foreground. However, launching a modal dialog box from an Active Desktop item may confuse the user. If an Active Desktop item delivers a pop-up message to the user while the desktop is not in the foreground, the user may not be sure where the message came from or what to do about it. The typical situation to avoid is showing error messages when an Active Desktop item can't connect to the server. In these cases, the item should display some type of status message in the Active Desktop item itself, preferably with a hot spot or link that lets the user get more information and figure out how to proceed.
- **Use the default color palettes.** A large percentage of Internet Explorer 4 users have displays that are limited to 256 colors. For these users, Internet Explorer chooses a default palette that is intended to give a broad range of hues across the entire color spectrum. Therefore, when authoring an Active Desktop item, it is important to keep in mind exactly what the image colors will look like on a 256-color display.
- **Include a rating tag in your item.** Content Advisor is a feature in Internet Explorer that allows parents and supervisors to filter out Web content based on rating labels. This feature is based on the World Wide Web Consortium (W3C) standard for content ratings, known as PICS (http://www.w3.org/PICS/), or Platform for Internet Content Selection. (For more information about PICS, see Part 7.) Web publishers should include a rating tag in their HTML files because someone browsing with the Content Advisor turned on will not be able to see content that is not rated. For more information, see the *Microsoft Internet Client SDK*.
- **Handle expiration dates.** Web publishers can define an expiration date for time-sensitive desktop items. This information is specified by the EndDate attribute of the SCHEDULE element in the CDF file. For example, a typical desktop item schedule might look like this:

```
<SCHEDULE ENDDATE="1998.11.05T08:15-0500">
    <INTERVALTIME DAY="1" />
    <EARLIESTTIME HOUR="12" />
    <LATESTTIME HOUR="18" />
</SCHEDULE>
```

This schedule shows that on November 5, 1998, Internet Explorer will stop refreshing this desktop item's data from the source URL. The desktop item will remain on the user's desktop, but its content will become stale. A few days before the desktop item expires, Web publishers should use a notification page to replace the content to which the desktop item points. This page should explain to users that the desktop item is no longer valid and that they should either close it or get a new one. The actual desktop item should not be deleted from the HTTP server until after the expiration date, to avoid having broken desktop items on users' desktops.

Creating CDF Files for Active Desktop Items

The primary rule in creating Active Channels is that you are required to create a separate CDF file for each Active Desktop item or Active Channel you author. The basic syntax includes the ITEM and USAGE elements. (Remember, for a full explanation of how to create CDF files, see "Creating CDF Files" in Chapter 16.)

The CDF for an Active Desktop item must include at least one ITEM element under the top-level CHANNEL element, with a USAGE child element marked as "DesktopComponent" (see the example below). It is this USAGE element that tells Internet Explorer that the CDF file refers to an Active Desktop item, as opposed to another type of Active Channel. The CDF file should also include a SCHEDULE element to indicate when the item will be updated.

Since an Active Desktop item comprises a single HTML page, the only ITEM you need to include in the CDF file is the small Web page you want to embed in the user's desktop. This should be marked with the USAGE element as indicated in the example below. If you like, you can include additional ITEM elements for each page to which the Active Desktop item links. Including these as ITEMs in the CDF file has the additional benefit of keeping those pages "subscribed to" and available for offline viewing. The ITEMs, other than the actual Active Desktop item URL, are not exposed to the user. Note that this differs from Active Channels, which present a hierarchical list of ITEM tags to the user in the form of the Internet Explorer Channel Bar.

The following restrictions are placed on CDF files for Active Desktop items:

- If a CDF file has more than one ITEM element marked `<Usage VALUE="DesktopComponent">`, Internet Explorer uses the first one it encounters.
- Only the top-level CHANNEL element is valid. Nested CHANNEL elements have no effect, because these additional items don't correspond to any list or hierarchy that the user can access.
- USAGE elements set to anything other than "DesktopComponent" are ignored.

CDF Example for an Active Desktop Item

The following example shows a CDF file for an Active Desktop item.

```
<?XML VERSION="1.0"?>
<CHANNEL>

    <SCHEDULE ENDDATE="1998.11.05T08:15-0500">
        <INTERVALTIME DAY="1"  />
        <EARLIESTTIME HOUR="12"/>
        <LATESTTIME   HOUR="18"/>
    </SCHEDULE>

    <ITEM HREF="http://www.microsoft.com" LASTMOD="1996.11.05T08:15-
        0500">
        <Title>My News Ticker</Title>    <!-- Friendly name -->
        <USAGE VALUE="DesktopComponent"> <!-- Required -->
            <OPENAS VALUE="HTML"/>  <!-- "HTML" (default) or "Image" -->
            <WIDTH  VALUE="200" />   <!-- Width in pixels            -->
            <HEIGHT VALUE="80"  />   <!-- Height in pixels           -->
            <CANRESIZE VALUE="Yes"/> <!-- Yes (default) or No        -->
        </USAGE>
    </ITEM>

    <!-- Other Item tags may go here.  IE4 won't display them anywhere,
        but -->

    <!-- will use them for site crawling and place them in the user's
        cache. -->

    <ITEM HREF="http://www.samplesports.com/topstories.html"
        LASTMOD="1996.11.05T08:15-0500" >
    </ITEM>

</CHANNEL>
```

Offering the Active Desktop Item to Users

Once you've designed your Active Desktop item and created its CDF file, you will need to make the CDF file available on your Web server. As with Active Channels, a link or button should be placed on any Web page that references the CDF file to allow users to subscribe to the new Active Desktop item. When the user clicks the link, Internet Explorer 4 automatically detects the USAGE element, which tells it that the CDF refers to an Active Desktop item.

Internet Explorer 4 then activates the Active Desktop item Subscription Wizard on the user's computer and allows the user to receive updated information about this Active Desktop item. As the content contained in the Active Desktop item is added, removed, or modified, Web publishers should update the CDF file to reflect the changes.

PART 4

Information Delivery: Webcasting

The information delivery architecture in Internet Explorer 4 enables new ways for Web content or applications to reach end users, using new push and pull technologies that are collectively referred to as *Webcasting*. Webcasting is the automated delivery of personalized, up-to-date information to users' desktops. Internet Explorer 4 provides a mechanism for end users to schedule recurring monitoring or crawling of Web sites and automatic downloading of updated information. These advanced features for pulling content allow online users to receive notification when content changes, and they allow mobile workers to view favorite Web content even when offline. The information delivery architecture further allows intranet administrators to set up servers that push content to corporate employees via bandwidth-efficient multicast protocols.

Why introduce a new model of information delivery on the Web? Because traditional Web browsing poses several problems for end users and the organizations that support them. Currently, most Internet and intranet sites require users to manually browse for useful information and then frequently revisit the sites for updates. This process is time-consuming for users and bandwidth-consuming for the network, as the same pages are repeatedly downloaded. For mobile workers, the problem is exacerbated by painfully slow dial-up connections.

To address these problems, Internet Explorer 4 provides three Webcasting options that allow you to take the best advantage of this new technology, based on your current business needs and technology investments. By automatically tracking updates to Web content, Webcasting can help minimize the time users spend searching the Web for up-to-date information, while ensuring that the latest information is automatically delivered to the people who need it. Plus, Webcasting can reduce the bandwidth necessary to deliver updated content on the Web, by reducing the amount of redundant information that is downloaded, and operating at off-peak network load periods, rather than the peak load periods often used with manual browsing.

Part 4 describes the three types of Webcasting and explains how to incorporate these information delivery technologies into your organization.

Chapter 15: Basic Webcasting

Using basic Webcasting, users can "subscribe" to any existing Web site and have Internet Explorer automatically visit and "crawl" the site on a scheduled basis to check for updated content. Internet Explorer will check the site up to three levels deep, and when content changes, notify users—either through a gleam on their Favorites lists or via e-mail. Internet Explorer can also automatically download updated content for offline use. Site subscription might be thought of as "smart pull" technology.

Chapter 16: Managed Webcasting

Similar to basic Webcasting, managed Webcasting also works through a subscription process, except users subscribe to publisher-specified Active Channels™, which deliver a defined range of Web content. This might be thought of as "programmed pull." Users can check for updates on specific types of content and can accept a publisher's predefined update schedule or specify a custom schedule. When channel content changes, Internet Explorer notifies the user of changes and can be set to download just the requested content for viewing offline.

Active Channels allow Web-site authors to optimize, personalize, and more fully control how a site is Webcast, and adding a Channel Definition Format (CDF) file is the only step required to convert any existing Web site into an Active Channel. Using CDF, it's easy to design Active Channels, Active Desktop™ items, or channel screen savers for your intranet to efficiently deliver business information to users' desktops.

Chapter 17: True Webcasting

True Webcasting, or "true push" technology, uses multicast add-ons, such as Microsoft NetShow™ 2.0, to deliver bandwidth-efficient multimedia information to users' desktops. Organizations can use true Webcasting to automatically push updated information to subscribers at scheduled periods.

CHAPTER 15

Basic Webcasting

In this chapter

Understanding Basic Webcasting 190
Understanding Web Site Subscriptions 190
Link-Crawl Architecture 191
Web Crawling and the If_Modified_Since Header 192
Subscribing to Sites 193
Customizing Subscriptions 196
Locking Down Subscriptions 203
Beyond Basic Webcasting: Addressing Site-Crawling Issues 206

Understanding Basic Webcasting

Basic Webcasting with site subscriptions offers the following features and benefits:

- Automated visiting and crawling of sites on a scheduled basis to check for updated content reduces the time users spend on online browsing.
- Notification to users when updated Web content is available—with a red, asterisk-like gleam on updated favorites or via an e-mail message—helps users stay current easily.
- Automated download of subscribed content for dial-up users means remote workers and laptop users can browse their favorite sites even when offline, saving time and reducing online costs.
- Updates can be scheduled to take place during off-peak times, and only new or changed content is downloaded, so the burden on network bandwidth is reduced.
- Subscriptions work with any existing site that allows Web crawls, so Web site authors do not have to modify existing sites or their content to allow for subscriptions.
- Administrators can centrally lock down and manage a variety of subscription options for users.

Understanding Web Site Subscriptions

Internet Explorer 4 automatically enables Webcasting of any existing Web site without requiring modifications to the site. In order to Webcast content from a conventional Web site, Internet Explorer 4 performs a scheduled site-crawl of the site content, checking for updated content and optionally downloading information for offline use. Users initiate this process by subscribing to a Web site, using the **Favorites** menu in Internet Explorer. Users can also subscribe to Web sites while browsing the site. Once a user subscribes to a site, Internet Explorer periodically checks the site for updated content and notifies the user of changes. Users can configure a variety of subscription options using the Subscription Wizard and can later change configurations, using the tabs in the subscribed site's **Properties** dialog box.

Note Some sites disable Web crawls for selected pages or for the entire site. Site subscriptions won't work for Web pages that disable Web crawls.

Users can specify Web crawls up to three levels deep from a subscribed Web page. Internet Explorer crawls all Web links to the specified level and can optionally download and cache all linked Web pages to users' computers, including all linked images, audio and video, and ActiveX™ controls and Java applets.

When a user chooses to download updated content for offline access, Internet Explorer stores the updated content in a cache on the user's hard disk. Users can browse cached content offline when disconnected from the Web. When offline, if a user clicks on a link for content that is not cached, Internet Explorer queries the user whether to connect to the Web. Users can also complete and submit Hypertext Markup Language (HTML) forms when offline. Internet Explorer will cache the HTML forms and send the form data the next time the computer is connected to the Web.

System administrators can use the Internet Explorer Administration Kit (IEAK) Configuration Wizard and the IEAK Profile Manager to lock down subscription options for users. For example, you can restrict the number of subscriptions allowed at users' desktops or limit the depth level of site crawls.

For information about troubleshooting Webcasting, see Appendix E.

Link-Crawl Architecture

Basic Webcasting uses a general link-crawl process. Literally, Internet Explorer 4 will connect to a Web site, and given a starting Web page, will crawl each link on that page. If configured to do so, it will crawl multiple levels deep, or just look at the first page. During the link crawls, Internet Explorer can check to see if the page has changed, based on the current file size and date, and optionally download that page to the user's machine. Once the page has been checked, a notification of changes or content download is sent to the user.

- Webcheck.dll creates the subscription and launches the crawl at the scheduled time.
- Icmp.dll gets the HTTP header of the Web page to check if the page has changed.
- Shdocvw.dll is used to look for the META tags on the page.
- Mshtml.dll is used to download the page to the local cache and parse the content.

All subscriptions are stored in the registry at:

HKEY_CURRENTUSER\Software\Microsoft\Windows\CurrentVersion\NotificationManager\ScheduledItems

This key contains a CLSID (or GUID) for each subscription. The data is primarily hexidecimal and binary, so is not readable. Deleting any entry from this key will completely remove the subscription. A subscription has no file components.

Link crawl uses the Scheduling Agent to perform its scheduling of downloads.

Web Crawling and the If_Modified_Since Header

The sitecrawler compares the date of a file in the Temporary Internet Files folder to the date of the same file on the Web server on which it resides. If the dates are different, it knows the page has changed. The file date from the server is sent in a special header field in an HTTP response.

The **If_Modified_Since** header field is used with the GET method to make it conditional: if the requested page has not been modified since the time specified in this field, an item will not be returned from the server; instead, a 304 (not modified) response will be returned without any message-body.

If_Modified_Since = "If_Modified_Since" ":" HTTP-date

An example of the field is:

If_Modified_Since: Sat, 29 Oct 1994 19:43:31 GMT

The purpose of this feature is to allow efficient updates of cached information with a minimum amount of transaction overhead.

It's also helpful to be aware of the following notes:

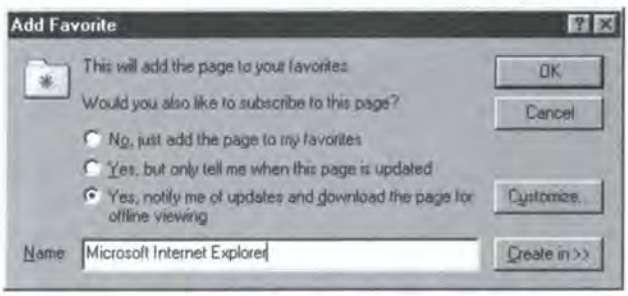
- The range-request header modifies the meaning of **If_Modified_Since**.
- **If_Modified_Since** times are interpreted by the server, whose clock may not be synchronized with the sitecrawler.

Subscribing to Sites

You can subscribe to a Web site while browsing the site, by using the **Add Favorite** dialog box.

▶ **To subscribe to a site while browsing the site**

1. On the **Favorites** menu, click **Add to Favorites**.
2. In the **Add Favorite** dialog box, select one of the following subscription options:
   - **Yes, but only tell me when this page is updated** to subscribe but not download updated content to the cache.
   - **Yes, notify me of updates and download the page for offline viewing** to subscribe and download updated content to the cache.

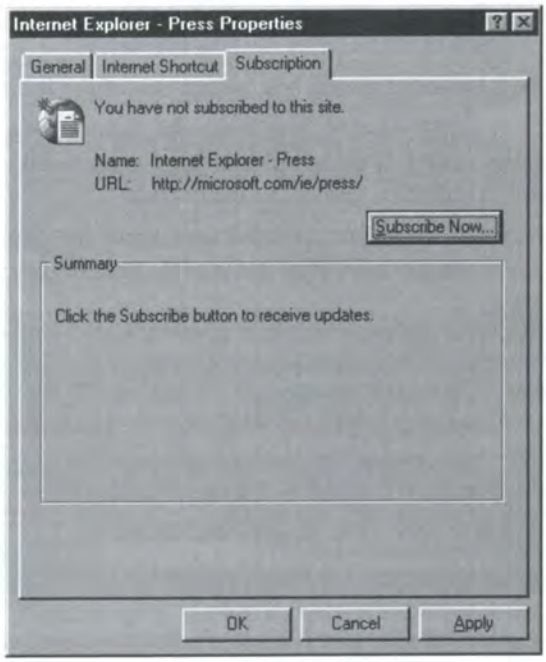


3. To customize subscription options, click **Customize**, and use the Subscription Wizard to configure subscription options. For more information, see "Customizing Subscriptions" next in this chapter.
4. Click **OK**.

You can subscribe to existing Favorites or unsubscribe from subscribed sites with the **Favorites** menu, either directly from the menu or using the **Subscription** tab in the site's **Properties** dialog box.

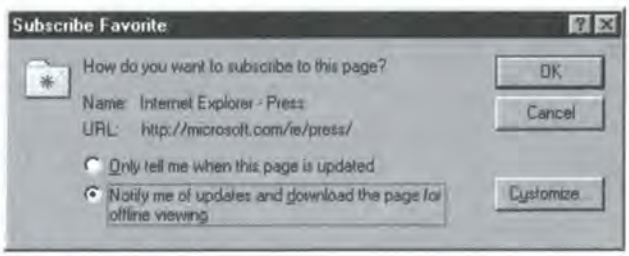
▶ **To open the Subscription tab**

1. On the **Favorites** menu (or on the **Organize Favorites** dialog box, available from the **Favorites** menu), right-click the site to which you want to subscribe, and click **Properties** on the shortcut menu.
2. Click the **Subscription** tab.



▶ **To subscribe to an existing site in the Favorites list**

1. Perform one of the following:
   - On the site's **Subscription** tab, click **Subscribe Now**.
   - On the **Favorites** menu, right-click the site to which you want to subscribe, and click **Subscribe** on the shortcut menu.
2. In the **Subscribe Favorite** dialog box, select one of the following subscription options:
   - **Only tell me when this page is updated** to subscribe but not download updated content to the cache.
   - **Notify me of updates and download the page for offline viewing** to subscribe and download updated content to the cache.



3. To customize subscription options, click **Customize**, and use the Subscription Wizard to configure subscription options. For more information, see "Customizing Subscriptions" next in this chapter.
4. Click **OK**.

▶ **To unsubscribe from a site**

- Perform one of the following:
  - From the subscribed site's **Subscription** tab, click **Unsubscribe**.
  - On the **Favorites** menu, right-click the site from which you want to unsubscribe and click **Unsubscribe** on the shortcut menu.

Customizing Subscriptions

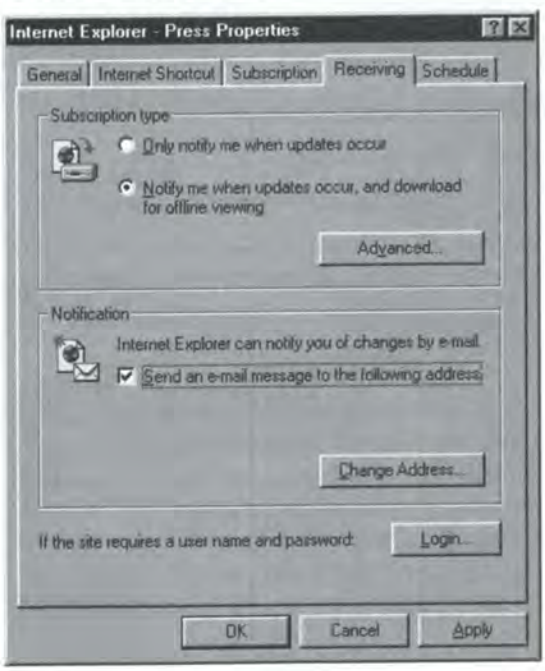
You can customize most subscription options when you subscribe to sites by using the Subscription Wizard. After subscribing to a site, you can use the **Receiving** and **Schedule** tabs in the site's **Properties** dialog box to configure all subscription options.

Use the **Receiving** tab to specify the following options:

- Disable automatic download of updated content for offline viewing.
- Specify up to three levels of links—in addition to the subscription page—to download for offline viewing.
- Prevent images, sound and video, or ActiveX controls and Java applets from downloading during updates.
- Limit downloads to a specified number of kilobytes (KB) per update.
- Activate notification of updates by e-mail.
- Specify the user's e-mail address information.
- Specify a user name and password, if needed, to authenticate site logins.

▶ **To access the Receiving tab**

1. On the **Favorites** menu (or from the **Organize Favorites** dialog box, available from the **Favorites** menu), right-click the site to which you want to subscribe, and click **Properties** on the shortcut menu.
2. Click the **Receiving** tab.



▶ **To configure advanced download options**

1. On the **Receiving** tab, click **Advanced**.
2. Configure the options.

Option	Description
Follow links outside of this page's Web site	Select to specify whether you want Internet Explorer to download more than just the top-level page of the selected Web site.
Download linked pages within a depth of __	Select the depth level for Web crawls and downloads. Select **0** to crawl and download just the top-level subscribed page. You can specify that you want to download the specified Web site up to three levels deep. For example, if you specify two levels deep, the first page and all the pages to which the first page links will be downloaded. Specifying three levels will download all the pages linked to the second-level pages. Consider these levels carefully; automatic download of multiple site levels can fill your hard disk with large amounts of Web content.
Images	Clear to prevent downloading graphical images when Internet Explorer updates the selected Web site. If you choose not to download images, the Web site may appear incomplete or not display properly when browsed offline.

Option	Description
Sound and video	Clear to prevent downloading sound and video files when Internet Explorer updates the selected Web site. If you choose not to download these files, the Web site may not function properly when browsed offline.
ActiveX Controls and Java applets	Clear to prevent downloading ActiveX controls and Java applets when Internet Explorer updates the selected Web site. If you choose not to download these programs, the Web site may not function properly when browsed offline.
Never download more than __ Kb per update	Select this option, and enter a number of kilobytes to limit the maximum amount of information that Internet Explorer downloads to your hard disk from this Web site for each update.

3. Click **OK**.

▶ **To configure mail address options**

1. On the **Receiving** tab, click **Change Address**.
2. Configure the options.

Option	Description
E-mail address	Enter the Internet e-mail address where you want to receive update notifications.
E-mail server name (SMTP)	Enter the name of your outgoing (SMTP) Internet e-mail server.

3. Click **OK**.

▶ **To configure login options**

1. On the **Receiving** tab, click **Login**.
2. Configure the options.

Option	Description
User name	Enter the user name necessary to authenticate logins to the site.
Password	Enter the password necessary to authenticate logins to the site.

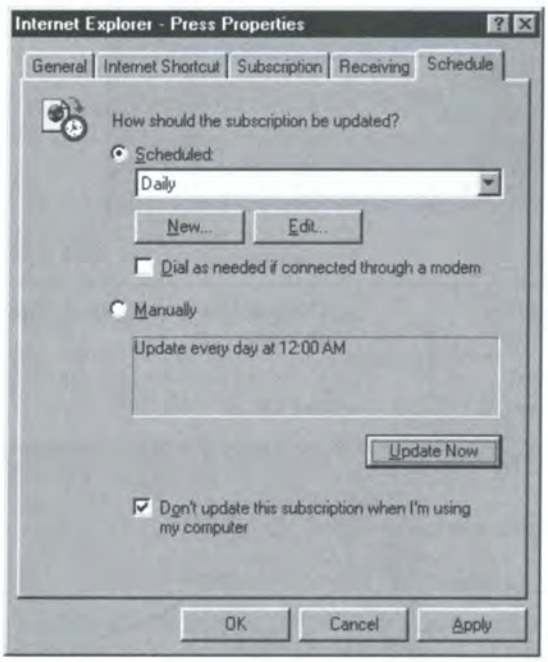
3. Click **OK**.

Use the **Schedule** tab to specify the following options:

- Create a customized, automated update schedule.
- Disable automated updates. (User performs updates manually.)
- Permit updates only when the computer is idle.

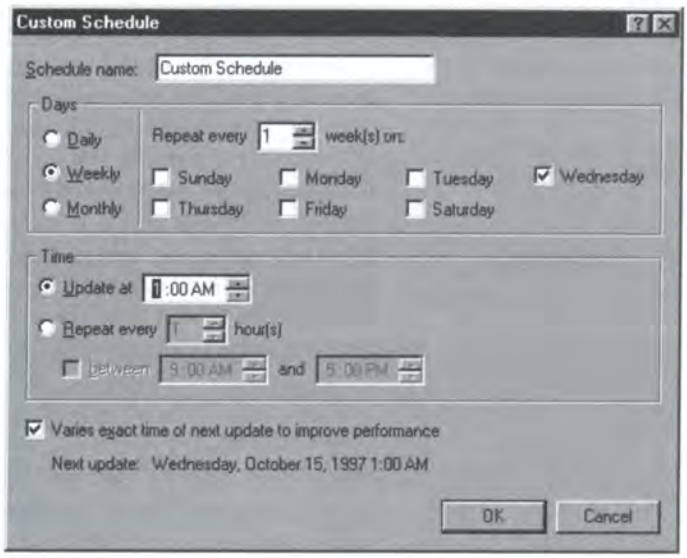
▶ **To access the Schedule tab**

1. On the **Favorites** menu (or from the **Organize Favorites** dialog box, available from the **Favorites** menu), right-click the site to which you want to subscribe and click **Properties** on the shortcut menu.
2. Click the **Schedule** tab.



▶ **To create or edit a custom schedule**

1. On the **Schedule** tab, click **New** or **Edit**.
2. Configure the options.



Option	Description
Schedule name	Enter a unique name for this schedule.
Daily	Select to specify an update interval based on days.
Every __ day(s)	Select this option, and enter an interval in days from 1 to 999 when automated updates are to occur. For example, enter **3** to specify updates every three days.
Every weekday	Select to update the site every weekday.
Weekly	Select to specify an update interval based on weeks.
Repeat every __ week(s) on	Enter an interval in weeks from 1 to 99, and select a day of the week (**Sunday** through **Saturday**) when automated updates are to occur. For example, you can enter **2** for **week(s)** and select **Wednesday** to specify biweekly updates on Wednesdays.
Monthly	Select to specify an update interval based on months.

Option	Description
Day __ of every __ month(s)	Enter the day of the month and an interval in months from 1 to 6 when automated updates are to occur. For example, you can enter **15** for **Day** and **2** for **month(s)** to specify updates to occur on the fifteenth day, every other month.
The __ __ of every __ month(s)	Select **first**, **second**, **third**, **fourth**, or **last** from the first selection list. Select **Sunday**, **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, or **Saturday** from the second selection list. Enter an interval in months from 1 to 6 when automated updates are to occur. For example, you can select **first** and **Saturday** and enter **1** for **month(s)** to specify updates to occur on the last Saturday of each month.
Update at __	Select this option, and enter the time of day when automated updates are to occur. For example, enter **1** A.M. to update the site at 1 AM on the days of scheduled updates.
Repeat every __ hour(s)	Select this option and enter an interval in hours to repeat automated updates. Use this option only if you want updates on an hourly basis. This option overrides other update schedules. You can also use the **between __ and __** option to specify a range of time for repeating updates.
Between __ and __	Select this option and enter a range of time when repeated updates are to occur. Use this option if you want repeated updates to occur during a range of time each day. For example, enter **8** A.M. and **5** P.M. to specify repeated updates only during normal business hours. You must also select **Repeat every __ hour(s)** and enter the repeating interval in hours.
Varies exact time of update to improve performance	Select to vary automated updates by several minutes in actual start time for the next update. Selecting this option increases the likelihood of offsetting the demands other people may make at the same time for the same site and thus improve the efficiency of downloading from the Web.

3. Click **OK**.

▶ To configure the Schedule tab options

1. Configure the options.

Option	Description
Scheduled	Select this option, and then select a schedule from the selection list to specify the automated update schedule.
Manually	Select to disable automated updates and perform all updates manually.
Don't update this subscription when I'm using my computer	Select to permit updates to occur only when the computer is idle. Clear to enable updates to occur when you are using the computer.

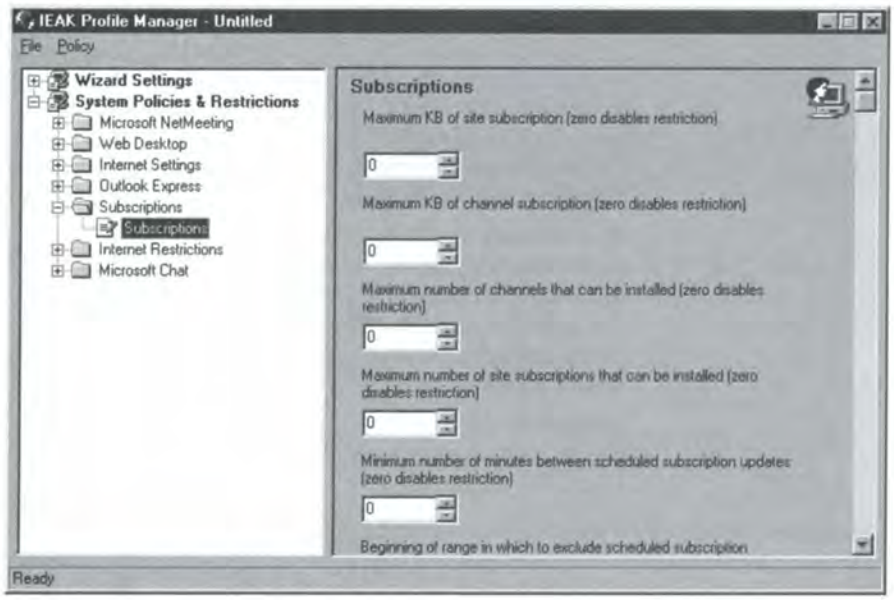
2. Click **OK**.

Locking Down Subscriptions

System administrators can use the IEAK Configuration Wizard to lock down subscription options. After browsers are deployed, you can use the IEAK Profile Manager to lock down subscription options for user groups that have automatic browser configuration implemented. For more information, see "Building Custom Packages" in Chapter 36 and "Setting up Automatic Browser Configuration" in Chapter 37.

▶ **To lock down subscription options**

1. On the **System Policies and Restrictions** page of the IEAK Wizard or the IEAK Profile Manager, in the System Policies & Restrictions hierarchy, select **Subscriptions**, and then select the Subscriptions object.

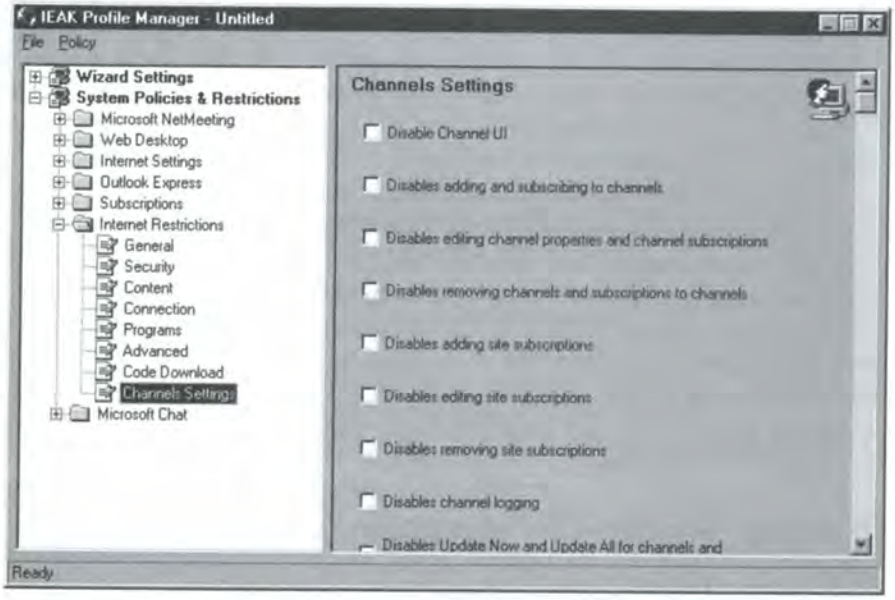


Note The IEAK Profile Manager is shown here, but the **System Policies and Restrictions** page of the IEAK Configuration Wizard works similarly.

2. In the right pane, configure the following options:

Option	Description
Maximum KB of site subscription (zero disables restriction)	Enter the maximum kilobytes that can be downloaded per site update for each user. Users are prevented from downloading more than the specified number of kilobytes per site update.
Maximum number of site subscriptions that can be installed (zero disables restriction)	Enter the maximum number of site subscriptions that can be installed on users' computers. Users are prevented from subscribing to more than the specified number of sites locally.
Minimum number of minutes between scheduled subscription updates (zero disables restriction)	Enter the minimum number of minutes allowed between scheduled updates. Users are prevented from scheduling automated updates closer together than this number of minutes.
Beginning of range in which to exclude scheduled subscription updates. Measured in minutes from midnight (zero disables restriction)	Enter the starting time measured in minutes from midnight of a range of time when you want to exclude scheduled updates. Users are prevented from scheduling automated updates during this range of time. For example, you could enter **420** to start the range of time at 8 A.M. to exclude normal business hours.
End of range in which to exclude scheduled subscription updates. Measured in minutes from midnight (zero disables restriction)	Enter the ending time measured in minutes from midnight of a range of time when you want to exclude scheduled updates. Users are prevented from scheduling automated updates during this range of time. For example, you could enter **960** to end the range of time at 5 P.M. to exclude normal business hours.
Maximum site subscription crawl depth	Enter the maximum depth level allowed for site Web crawls and downloads to users' computers. Select **0** to restrict user crawls and downloads to just the top-level subscribed page. Users are prevented from specifying Web crawls and downloads deeper than the restriction you specify here.

3. In the System Policies & Restrictions hierarchy, select **Internet Restrictions**, and then select the **Channel Settings** object.



Note The IEAK Profile Manager is shown here, but the **System Policies and Restrictions** page of the IEAK Configuration Wizard works similarly.

4. In the right pane, configure the channel settings options.

Option	Description
Disables adding site subscriptions	Select to prevent users from adding site subscriptions.
Disables editing site subscriptions	Select to prevent users from using the **Properties** tab to change site subscription settings.
Disables removing site subscriptions	Select to prevent users from removing or unsubscribing to site subscriptions.
Disables Update Now and Update All for channels and subscriptions	Select to prevent users from manually updating channels or site subscriptions.
Disables all scheduled channel and site subscriptions	Select to prevent automated Web crawls and updates from users' computers.
Disables unattended dialing by subscriptions	Select to prevent automated dial-up connections to update channel and site subscriptions for users who have dial-up connections.

Option	Description
Disables password caching for channel and site subscriptions	Select to prevent users' authentication passwords from being stored and submitted automatically to channels and sites that require user authentication. When this option is selected, users will be prompted to enter the authentication password each time Internet Explorer connects to secure channels or sites.
Disable downloading of site subscription content—change notification will still work	Select to prevent the download of site content to the cache on users' computers. When this option is selected, users cannot browse the site offline.
Disables editing and creating of schedule groups	Select to prevent users from creating custom automated schedules for updates to subscribed channels and sites.

5. Save your changes to the auto-configuration (.ins) file or finish building the custom package, as appropriate.

Beyond Basic Webcasting: Addressing Site-Crawling Issues

While basic Webcasting offers several significant benefits, the technology that it employs, the sitecrawler, has some fundamental limitations and disadvantages that sometimes make it necessary to go beyond basic site subscriptions, namely:

- **Site crawling may be difficult for end users.** While it appears fairly easy to set up a subscription to any Web site, the user is left to make numerous choices, including schedules, process for content delivery, levels and amount of content, and more. This can be confusing for the general end user.
- **Too much information may be downloaded to the hard disk.** If the user wants to monitor a specific piece of information that is a couple levels down on a Web site, the sitecrawler will automatically download all the updated information down to the level the user requested, since the sitecrawler always starts from the top level of the Web tree.

- **Updates offer hit-or-miss usefulness.** With the sitecrawler, users cannot specify types of information; they only can grab all the new and updated information and then sift through it to determine what they need. Additionally, the Web site author has no way to offer organized groups of information to users. The HTML in conventional Web sites provides no cues to help a sitecrawler decide which links point to useful content and which point to useless content. Because no such cues exist, the crawler must rely on a set maximum number of levels and a disk-space limit, and hope that the content downloaded proves useful.
- **A need exists for smart scheduling and site-based control.** With the sitecrawler, the user sets the schedule for when new information should be retrieved, which may not coincide with the actual content updates of the Web site. Conventional Web sites do not advertise content-update schedules, and therefore, a scheduled site crawl may check for updates too often or not often enough. For example, if a Web site publishes updates every day at 3 P.M., but the user has the sitecrawler set to get updates at 2 P.M., the user will clearly miss key updates to the Web site. It would be better if the Web site could instruct the user's PC when to schedule updates.
- **Some Web sites disable site crawling.** Site crawling can cause a heavy load for the Web server, since the site will be entirely crawled by any user who has subscribed to it. Therefore, many Web sites simply do not allow crawling, which prevents basic Webcasting of their information.
- **Organizations have limited control over site subscriptions.** Organizations cannot use basic subscriptions to implement controlled information delivery solutions over the Web, and administrators cannot centrally schedule site updates to optimize intranet performance (although they can specify a block of time when automated site updates are *not* allowed).

To address these site-crawling limitations, Microsoft has introduced the next level of Webcasting—managed Webcasting, discussed in detail in the next chapter.

CHAPTER 16

Managed Webcasting

In this chapter

Active Channels Overview 210
- Analogies 210
- Scenarios 211

Understanding Active Channels 212
- Using Active Channels 212
- Using Active Desktop Items 213
- Using Channel Screen Savers 214

Understanding CDF Files 215
- CDF Link-Crawl Architecture 215
- Creating CDF Files 216
  - Example of a CDF File 220
  - Another Example of a CDF File 222
  - Item Usage 223
  - Publisher's Schedule 223
  - Active Desktop Items 224
  - Channel Screen Savers 225
  - Personalized Webcasts 225
  - Design Guidelines for Active Channels 228

Subscribing to Active Channels and Desktop Items 229

Customizing Active Channels and Desktop Items 230

Deploying Custom Packages with Predefined Channels for User Groups 234
- Locking Down Active Channels and Desktop Items 235
- Managing Channels for Deployed Browsers 239

Active Channels Overview

Managed Webcasting uses the Channel Definition Format (CDF) to create Active Channels, Active Desktop items, and channel screen savers to deliver information to subscribers' desktops. Managed Webcasting overcomes many of the limitations of basic Webcasting and allows Web publishers to optimize, personalize, and control the way sites are seen by subscribers. Managed Webcasting also provides organizations with a streamlined approach to information delivery that can be centrally controlled and fully managed using the Internet Explorer Administration Kit (IEAK).

Managed Webcasting with CDF provides many advantages over simple site subscriptions. By using managed Webcasting, you can:

- Define the subset of a Web site's pages that are actively monitored for changes (site-crawled), which can reduce the amount of network traffic required to monitor for new content.
- Provide a structured view of the content of a site to make user navigation easier and faster.
- Selectively control what subset of pages is downloaded and cached on subscribers' computers for offline viewing to deliver personalized content to individual users' desktops and use the minimal required amount of hard disk space.
- Smart-schedule the channel updates to coincide with scheduled Web site content updates or to take place during optimum network load times.
- Convert an existing Web site into an Active Channel without having to re-author the site or alter it in any way; all that is required is the addition of a CDF file.

Analogies

To understand the benefits that managed Webcasting offers, consider the following analogies. First, CDF offers the ability for users to select the specific content they wish to be downloaded, rather than simply pull large amounts of data from a Web site in the hope that some subset of that information is what they need. In this case, CDF is like a restaurant menu: users select the food (information) they want, and only the food they ordered is delivered to them. Browsing without CDF is like ordering everything on the menu and having to search through the many items on the table for the one dish you want to eat.

CDF also offers the ability to manage the amount of information delivered to users. In this case, CDF could be compared to an automatic sprinkler system that manages the scheduled flow of water to different parts of a lawn. Just as you wouldn't flood your yard to water a small area, users and corporate administrators want to control the "flood" of information to desktops. In this example, CDF is the brains of the sprinkler system (the Web site), which allows users and administrators to control the type and amount of water (information) delivered to the desktop on a scheduled basis.

Scenarios

System administrators can use managed Webcasting to provide a wide variety of information delivery solutions. Possible scenarios include:

- The channel screen saver from the Human Resources department alerts a user that the 401(k) open enrollment period ends in one week. Clicking on the screen-saver hyperlink takes employees directly to the Human Resources Web site.
- A corporate attorney subscribes to a fee-based Active Channel for a legal service. A list of tax-law news summaries displays in an Active Desktop item on the attorney's desktop. A click on a specific news item launches a browser window that provides detailed information about a change to the tax laws.
- A sales representative receives updated price lists on a laptop via one Active Channel, while another Active Channel provides an entire online catalog. In the evenings, a scheduled automatic dial-up connection updates the channel information and highlights changes on the sales representative's laptop. Because the browser downloads only new content, the update takes only a few moments. During the workday, the sales representative accesses the up-to-date information on the laptop while at customer sites.
- A purchasing agent gets notification of a supplier's special pricing programs through an Active Desktop item, which updates daily from the supplier's secured Internet site.
- A credit manager subscribes to an external (fee-based) Active Channel that provides a personalized desktop ticker that delivers news about significant changes in customer credit ratings. A click on the ticker launches a browser window that provides detailed information.
- An executive subscribes to Active Channels providing industry news. Some channels are supplied directly by news providers, while other channels are fed by several electronic news feeds, brought in-house, filtered and categorized, and published internally. When notified that something new has arrived, the executive can scan headlines and decide what to read first.

Understanding Active Channels

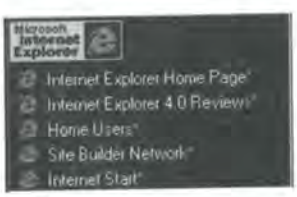
Active Channels are typically used to deliver Web content that appears in a subscriber's browser window, but they can also deliver information in two other, innovative ways—as Active Desktop items and as channel screen savers. Active Desktop items deliver miniature Web pages that appear directly on subscribers' desktops and usually contain summary information. Channel screen savers supply Web content that appears as screen savers on users' desktops when their computers are idle.

For information about troubleshooting Webcasting and for Frequently Asked Questions (FAQs) about Active Channels, see Appendix E and I, respectively.

Using Active Channels

Active Channels are Web sites that are defined by Channel Definition Format (CDF) files. Publishers can easily create a CDF file and turn existing Web sites into Active Channels. Active Channels can include any Web content supported by Internet Explorer 4, including Dynamic HTML, ActiveX controls, Visual Basic® Scripting Edition), Java applets, ECMA Script, and Active Server Pages.

The CDF file for an Active Channel typically specifies a channel hierarchy or table of contents for the Web pages that make up the Active Channel. When users subscribe to a channel, the channel appears in the Desktop Channel bar and the Internet Explorer Channel bar. The channel hierarchy defined in the CDF file appears in the Internet Explorer Channel bar. Channels can contain items that appear under the top-level channel in the Channel bar. When a user clicks the top-level channel, the lower-level items appear. Each item typically represents a Web page, but can represent any URL. When a user clicks an item in the Channel bar hierarchy, the corresponding Web page displays in the browser.



Channels can also include subchannels. Subchannels appear under the top-level channel, similar to items. Each subchannel, however, can also contain items. When a user clicks a subchannel, the items appear under the subchannel. Subchannels provide a good way to organize channels that contain a large number of Web pages.

When a user subscribes to an Active Channel, Internet Explorer reads and downloads the CDF file to the user's computer. All customization options available for site subscriptions are available for channel subscriptions, except users do not specify the depth levels for site crawls. Users can choose not to download any updated content, download all content specified by the publisher (in the CDF file), or download just the Web pages for the channel and subchannels (in which case Web content referenced by items is not downloaded). Downloaded Web content is cached, and users can browse the channel offline.

Channel updates use scheduled Web crawls. Users can accept the publisher's schedule (specified in the CDF file) for updates or specify a custom update schedule. When updated content is available for a channel, the red gleam appears in the left corner of the icon for the top-level channel. A red gleam does not appear for individual channel items or subchannels.

Users can also choose to be notified of updates by e-mail, and publishers can specify that the actual updated Web pages can be e-mailed to users. The updated Web pages display directly in mail programs that support HTML Mail and appear as attachments for mail programs that do not support HTML.

Using Active Desktop Items

Active Channels may be used as Active Desktop items, which appear directly in borderless frames on users' desktops. Desktop items can contain any standard Web content, including Dynamic HTML, ActiveX controls, VBScript, Java applets, ECMA Script, and Active Server Pages. Desktop items are typically designed to provide summary information in a small amount of screen space. Therefore, information items in desktop components often include hyperlinks or hot spots, so users can click a designated area to open a new browser and obtain the details they need. Possible desktop item applications include:

- **A ticker showing current stock quotes for a corporation.** An employee can click a hotspot to open a browser window that displays the home page for the corporation's employee stock option program.
- **A scrolling headline list of announcements or news stories.** When a user clicks a headline, a new browser window opens, displaying an executive summary of the news item or announcement with links to more detailed information, such as the full text of news articles.

- **A list of key employee services provided by an organization, such as help desk, payroll, and employee benefit services.** When a user clicks a list item, a new browser window opens displaying a page with relevant information and links for the appropriate service.
- **A scrolling list of available new software updates.** When a user clicks a list item, a new browser window opens, displaying a page with instructions for downloading the software update to the user's computer.

Desktop items should be designed to take up the minimum amount of desktop space necessary. For example, you could use scrolling lists to show a long list of items in a small amount of space.

Subscriptions to Active Desktop items work like regular Active Channel subscriptions. When updates for Active Desktop items are available, however, users are not notified by a gleam, because updated content automatically appears in the desktop frame.

For more information about the Internet Explorer 4 Active Desktop interface and desktop items, see Part 3, "True Web Integration: Windows Desktop Update."

Using Channel Screen Savers

The Internet Explorer 4 screen saver supports HTML pages. Screen-saver pages can use Dynamic HTML, ActiveX controls, VBScript, Java applets, or ECMA Script to provide a page that dynamically changes the content, style, and position of HTML elements on the screen. These pages are displayed on the full screen when the computer is idle. The Internet Explorer screen saver cycles through all enabled channel screen savers, allocating the same amount of display time to each one (by default, 30 seconds per channel).

Unlike many common screen savers, in a channel screen saver, the user can move the mouse and click objects without immediately dismissing the Internet Explorer screen saver. Clicking a link opens a new browser window and dismisses the screen saver. Clicking a point that is not an image, link, or object closes the screen saver. The user can also click the **Pause**, **Close**, and **Properties** buttons in the screen saver toolbar to change the current behavior.

Understanding CDF Files

Channel Definition Format files are standard text files. A bare-bones CDF file contains nothing but a list of URLs pointing to content. This type of file is easy to create and requires no changes to existing HTML pages. A more advanced CDF file can also include a schedule for content updates, a hierarchical organization of the URLs describing the Web site structure, and title/abstract information describing individual content items. Structured similarly to HTML files, CDF files are based on the open Extensible Markup Language (XML) standard.

CDF provides structured content indexing, independent of content format. It provides an index or map of a Web site that describes the type of information contained on the site. Specifically, CDF describes logical groupings of information (such as sports news versus financial information), providing hierarchical structure and category information about a site. Because this information is completely independent of content format, a CDF-based channel can include any kind of Web content or applications built on HTML, JavaScript, Java, or ActiveX technology. Because the CDF file defines what site content is included in the channel, but contains no Web content itself, a channel's CDF file can remain unchanged, even if the Web content changes regularly.

The CDF mechanism makes channel creation a simple, two-step process that does not require authoring of new content, does not require re-authoring of existing content, and does not require any programming—on either client or server. Creating a channel with CDF can be as simple as (1) writing a CDF file with a list of URLs pointing to existing content and (2) linking to this CDF file to make it discoverable.

CDF Link-Crawl Architecture

Channels add an extra dimension to the link-crawl process. The channel, using a CDF file, lets the content provider customize the content download; it's more like a programmed link crawl. Using CDF files, content providers can segment their data into specific types of content—without having to alter the Web site. A CDF file contains a list of HTML tags that reference specific Web pages on the site. A content provider would create one CDF file for each type of content he or she wants to designate on the site. For example, a news channel might have separate CDF files for sports, entertainment, politics, and so forth. Subscribers can choose to monitor specific CDF files and only download the pages pointed to by that file.

- Webcheck.dll creates the subscription and maintains the schedule of updates.
- Icmp.dll uses the page header to check whether the CDF file has changed.
- Msml.dll parses the CDF file. This file is also involved during the MIME download of the CDF.

- Cdfview.dll processes and applies the CDF file and updates the Channel bar for viewing purposes. It also makes registry entries for the channel.
- Shdocvw.dll and Mshtml.dll actually display the Web content in the browser and download the content to the local cache.

Other files are involved in the process, but these are the major items involved and their primary focus.

Creating CDF Files

The common tags used in a CDF file to create a channel are listed below, followed by two examples of code showing the tags in use. For a detailed CDF reference and guidelines for creating Active Channels, Active Desktop items, and channel screen savers, see the *Microsoft Internet Client SDK*.

For information about troubleshooting Webcasting and for Frequently Asked Questions (FAQs) about Active Channels, see Appendix E and I, respectively.

Define an XML document

<?XML VERSION="*n.n*"?>

This tag specifies that this file is an XML document and indicates the version of XML that this file supports. This header is required in every CDF file.

Create a channel

<CHANNEL> and **</CHANNEL>**

These tags note the beginning and ending of a channel or subchannel. The <CHANNEL> tag is used to specify the channel structure and can reference a URL, normally a Web page, that appears when the channel is clicked in the Channel bar. A channel can also include an unlimited number of <ITEM> tags. Items are used to reference URLs, such as HTML pages and other supporting Web content for the channel. Items can appear under the channel in the Internet Explorer Channel bar. When a <CHANNEL> tag is found within another <CHANNEL> tag, this is called nesting, and it creates a subchannel. A subchannel typically focuses on a specific area of content.

The CHANNEL element can include the following attributes:

- **HREF="*url*"** where *url* specifies the Web page for the channel or subchannel.
- **BASE="*url*"** where *url* specifies the base URL for the channel. A base URL can be used to resolve relative URLs specified in ITEM elements and subchannels contained within this channel.
- **LEVEL="*n*"** where *n* specifies the number of levels or links deep that the sitecrawler crawls from this channel.
- **SELF="*url*"** where *url* is used to specify the URL of this CDF file. This attribute is unnecessary and is only supported for backward compatibility.

```
<CHANNEL HREF="http://www.samplesports.com/samplesports.htm">
</CHANNEL>
```

Title of channel

```
<TITLE> and </TITLE>
```

These tags provide a text title for the channel.

```
<TITLE >SampleSports</TITLE>
```

Mouse-over ToolTip information

```
<Abstract> and </Abstract>
```

These tags mark text to be displayed as a ToolTip when the pointer is held over a particular item within the channel.

```
<ABSTRACT>The latest in sports and athletics from
SampleSports</ABSTRACT>
```

Automatic download scheduling of channel

<SCHEDULE>, <EARLIESTTIME>, <LATESTTIME>, <INTERVALTIME>, and **</SCHEDULE>**

Use these tags to let the content provider schedule the best time to check for changes to the channel. These can be overridden by the user.

```
<SCHEDULE ENDDATE="1994.11.05T08:15-0500">
<INTERVALTIME DAY="1" />
<EARLIESTTIME HOUR="12" />
<LATESTTIME HOUR="18" />
    </SCHEDULE>
```

Visual picture or logo

<LOGO>

This tag points to a graphic used as the logo of the Active Channel. The <LOGO> tag can have a style of IMAGE or ICON. The Image type defines the large image displayed on the Channel bar on the desktop and in Internet Explorer. The Icon type defines the image displayed in the **Favorites** menu.

```
<LOGO HREF="http://www.samplesports.com/images/logo.gif" STYLE="ICON" />
```

Channel content

<ITEM> and **</ITEM>**

These tags typically represent individual pages of content on the Web site to be displayed in the channel. ITEM elements can contain the following attributes:

- **HREF=**"*url*" where *url* specifies the Web page.
- **LASTMOD=**"*date*" where *date* indicate the last time the Web page was modified. You can use this attribute to report Web content updates to the sitecrawler, so the sitecrawler doesn't need to crawl the site, but just downloads content that has changed.
- **LEVEL=**"*n*" where *n* specifies the number of levels or links deep that the sitecrawler crawls from this item.
- **PRECACHE=**"Yes" | "No" where Yes or No specifies whether to download the Web page to the cache on users' computers. Specify No to prevent sitecrawlers from downloading the Web page to the cache.

An <ITEM> tag can point to any Web content, HTML pages, ActiveX controls, Java applets, ActiveMovie™ files, AVIs, animations, and so on.

```
<ITEM HREF="http://www.samplesports.com/articles/
a1.html" LastMod="1994.11.05T08:15-0500"></ITEM>
```

Usage of the channel

<USAGE VALUE="*UsageType*" />

A <USAGE> tag can follow an <ITEM> tag or the top-level <CHANNEL> tag to specify how the item or channel should be used. You can specify one or more USAGE elements per ITEM element. The *UsageType* can be one of the following:

- **"Channel"** to specify that an item appears in the Channel bar hierarchy. This is the default usage if now USAGE element is included.
- **"Email"** to specify that the Web page can be sent to users via e-mail to notify them of updated content. Users must enable e-mail notification for the channel before the Web page is actually sent via e-mail. Browsers that support HTML will view the Web page online; otherwise, the Web page is included as an attachment.
- **"None"** to specify that an item does not appear in the Channel bar hierarchy. You can use this value to include support content, such as ActiveX controls or Java Applets, you don't want to appear on the Channel bar.
- **"DesktopComponent"** to specify that a channel is a desktop item that appears directly in an invisible frame on the desktop instead of on the Channel bar. (Desktops items also require a closing tag: **</USAGE>**, as shown in the second example CDF file below.)
- **"ScreenSaver"** to specify that an item is used as a channel screen saver.
- **"SoftwareUpdate"** to specify that the channel is a software distribution channel and appears in the Favorites **Software Updates** folder instead of on the Channel bar.

```
<USAGE VALUE="ScreenSaver" />
```

Logging user activities back to the Internet

<LOG> and **<LOGTARGET>**

These tags are used to specify that user actions get submitted back to the Web server. The <LOG> tag is used to specify which items need to be tracked, and the <LOGTARGET> tag specifies where to send the data and how.

```
<LOGTARGET HREF="http://www.samplesports.com/tracking"
METHOD="POST" SCOPE="OFFLINE" >
```

You can use the LOG and LOGTARGET elements to gather statistics on which Web pages users access on your Web site, even when you are offline. Statistics are logged when users browse cached Web pages offline, and the statistics are reported back to the Web server the next time users connect to the Internet.

Example of a CDF File

The following example shows a CDF file for a fictional news service named Action News. When expanded, this channel appears in the Internet Explorer Channel bar with the following structure:

Action News Channel
  The Market
    Business nirvana: low inflation, high growth
    Tech stocks soar on XYZ Corp. news

```
<?XML VERSION="1.0" ENCODING="UTF-8"?>
<CHANNEL HREF="home.htm"
    BASE="http://www.action.com/channel/" >
    <TITLE>Action News Channel</TITLE>
    <ABSTRACT>Description of the Action News Channel</ABSTRACT>
    <LOGO HREF="ActionLogo16x16.gif" STYLE="ICON"/>
    <LOGO HREF="ActionLogo32x80.gif" STYLE="IMAGE"/>
    <SCHEDULE STARTDATE="1997-10-17" ENDDATE="1997-10-17">
        <INTERVALTIME DAY="1"/>
    </SCHEDULE>

    <CHANNEL HREF="bus001.htm">
        <TITLE>The Market</TITLE>
        <ABSTRACT>"Summary of today's market news"</ABSTRACT>
        <LOGO HREF="marketLogo.gif" STYLE="ICON"/>

        <ITEM HREF="bus002.htm" LASTMOD="1997-10-17T11:12"
            PRECACHE="Yes" LEVEL="1">
            <TITLE>Business nirvana: low inflation, high growth</TITLE>
            <ABSTRACT>Best environment for business in 25 years due to
                convergence of factors.
            </ABSTRACT>
            <LOGO HREF="TopStoryLogo.gif" STYLE="ICON"/>
        </ITEM>

        <ITEM HREF="bus003.htm" LASTMOD="1997-10-17T11:15">
            PRECACHE="Yes" LEVEL="1">
            <TITLE>Tech stocks soar on XYZ Corp. news </TITLE>
            <ABSTRACT>XYZ Corp. showed unexpected growth this quarter,
                dragging up prices through the technology segment.
            </ABSTRACT>
            <LOGO HREF="TopStoryLogo.gif" STYLE="ICON"/>
        </ITEM>

        <ITEM HREF="control1.ocx">
            PRECACHE="Yes">
            <USAGE VALUE="None" />
        </ITEM>
    </CHANNEL>

</CHANNEL>
```

In this example CDF file:

- **CHANNEL HREF ="home.htm"** references the home page for Action News. When users click the top-level channel, the home page displays in the right pane of the browser.
- **CHANNEL HREF="bus001.htm"** is the top-level page for the news category "The Market." When users click the subchannel, this page displays in the right pane of the browser.
- **ITEM HREF="bus002.htm"** is a news article. When users click the item, the news article displays in the right pane of the browser.
- **LASTMOD="1997-10-17T11:12"** specifies the last time the page bus002.htm was updated. This notifies Internet Explorer when an update is available and a site crawl performed only to download updated content. Use this attribute to speed up updates and help reduce server load from Web crawls.
- **PRECACHE="Yes"** specifies that the news articles are downloaded to the users' hard disk cache for offline browsing. Use "No" to specify that the content for this item is not downloaded to the cache.
- **LEVEL="1"** specifies that Web crawls go one level deep from bus002.htm and download content to the cache. Use this attribute to specify additional content you want to be available for offline browsing. You can also use this attribute with the CHANNEL element. When this attribute is omitted, the default value is LEVEL="0" and Web crawls do not go deeper than the specified URL.
- **ITEM HREF="bus003.htm"** is a second news article. When users click the item, the news article displays in the right pane of the browser.
- **ITEM HREF="control1.ocx"** is an ActiveX control needed for offline browsing. This control is downloaded to the subscribers' cache so the channel will function properly when offline.
- **USAGE VALUE="None"** specifies that control1.ocx does not appear in the Channel bar hierarchy and is used only to download control1.ocx to the subscribers' cache for offline browsing.

In this example, you could easily add more subchannels and news items for more news categories. Although you can add an unlimited number of subchannels and items, you should limit subchannels and items to a manageable number. It's best to create a channel hierarchy that easily fits in the browser window when fully expanded. You should also limit the content that is downloaded to an appropriate amount to conserve bandwidth as well as subscribers' hard disk space.

Keep in mind that the Channel bar displays a navigable outline of the URLs you specify in a CDF file. The exposed hierarchy of the channel helps users efficiently navigate through a channel to find the content that most interests them.

Another Example of a CDF File

This example shows a CDF file for a desktop item. (Another example of a CDF file for a desktop item and more information about creating desktop items is available in the "Developing Active Desktop Items" section of Chapter 14, "Active Desktop.")

```
<CHANNEL HREF="http://www.samplesports.com/samplesports.htm">
<SELF HREF="http://www.samplesports.com/channels/ch1.cdf"/>
<TITLE>SampleSports</TITLE>
<ABSTRACT>The latest in sports and athletics from
    SampleSports</ABSTRACT>
<SCHEDULE ENDDATE="1994.11.05T08:15-0500">
<INTERVALTIME DAY="1" />
<EARLIESTTIME HOUR="12" />
<LATESTTIME HOUR="18" />
</SCHEDULE>
<LOGO HREF="http://www.samplesports.com/images/logo.gif" STYLE="ICON" />
<ITEM HREF="http://www.samplesports.com/articles/a1.html"
    LASTMOD="1994.11.05T08:15-0500">
<TITLE>How to get the most out of your mountain bike</TITLE>
<ABSTRACT>20 tips on how to work your mountain-bike to the bone and come
    out on top. </ABSTRACT>
<LOG VALUE="document:view" />
</ITEM>
<CHANNEL>
<TITLE>SampleSports News</TITLE>
<ABSTRACT>Up-to-date daily sports news from SampleSports</ABSTRACT>
<LOGO HREF="http://www.samplesports.com/images/newslogo.gif"
    STYLE="ICON"/>
<LOGO HREF="http://www.samplesports.com/images/newslogowide.gif"
    STYLE="IMAGE"/>
<ITEM HREF="http://www.samplesports.com/articles/news1.html"
    LASTMOD="1994.11.05T08:15-0500" >
<TITLE>Michael Jordan does it again!</TITLE>
<ABSTACT>Led by Michael Jordan in scoring, the Chicago Bulls make it to
    the playoffs again! </ABSTRACT>
<LOG VALUE="document:view" />
</ITEM>
<ITEM HREF="http://www.samplesports.com/articles/news2.html"
    LASTMOD="1994.11.05T08:15-0500" >
<TITLE>Islanders winning streak ends</TITLE>
<ABSTRACT>The New York Islanders' 10-game winning streak ended with a
    disappointing loss to the Rangers"</ABSTRACT>
<LOG VALUE="document:view" />
</ITEM>
</CHANNEL>
<ITEM HREF="http://www.samplesports.com/animations/scrnsvr.html" >
<USAGE VALUE="ScreenSaver"></USAGE>
</ITEM>
```

```
<ITEM HREF="http://www.samplesports.com/ticker.html"
    LASTMOD="1994.11.05T08:15-0500" >
<TITLE>SampleSports News Ticker"</TITLE>
<ABSTRACT>The latest sports headlines from SampleSports"</ABSTRACT>
<USAGE VALUE="DesktopComponent">
<WIDTH VALUE="400" />
<HEIGHT VALUE="80" />
</USAGE>
</ITEM>
<LOGTARGET HREF="http://www.samplesports.com/tracking" METHOD="POST"
    SCOPE="OFFLINE" >
<PURGETIME HOUR="12"  />
<HTTP-EQUIV NAME="encoding-type" VALUE="gzip" />
</LOGTARGET>
</CHANNEL>
```

Item Usage

Not all tagged ITEM elements appear in the Channel bar. The ITEM elements can include the USAGE element to specify how the item is used. When the <USAGE> tag is omitted, the default is USAGE VALUE="Channel", which specifies that the referenced URL appears in the Channel bar hierarchy.

USAGE VALUE="Email" specifies that the referenced URL is sent as an e-mail to subscribed users whenever content is updated. However, users must choose the e-mail notification as a subscription option before an e-mail can be sent.

USAGE VALUE="None" specifies that the referenced URL does not appear in the Channel bar. The URL is downloaded to the subscribers' cache when PRECACHE="No". Use USAGE VALUE="None" to download Web content such as ActiveX controls or Java applets that you want to download to subscribers' caches to support offline browsing.

Publisher's Schedule

Publishers can specify a schedule using the SCHEDULE element as a child element in the first CHANNEL element. All subchannels and items in a CDF file are updated according to this schedule. It's a good idea to specify schedules that optimize network and server load. It's also helpful to randomize the time that subscribers update the channel, to prevent all subscribers from updating at the same time. The following sample schedule would cause the channel to be updated every day at a random time between 2 A.M. and 6 A.M.

```
<SCHEDULE>
    <INTERVALTIME DAY="1" />
    <EARLIESTTIME HOUR="2" />
    <LATESTTIME HOUR="6" />
</SCHEDULE>
```

Users can accept the publisher's schedule for updates or can specify a custom-update schedule. System administrators, however, can use the IEAK Configuration Wizard and the IEAK Profile Manager to lock down schedule options. For example, you can prevent users from changing the channel schedule and ensure that users use the publisher's schedule that you specify. For more information, see "Locking Down Active Channels and Desktop Items" later in this chapter.

Active Desktop Items

You can designate a specific URL as a desktop item by including the **USAGE VALUE="DesktopComponent"** attribute in the channel's CDF file. This must always be nested within the first ITEM element in the file. You also must close the usage section with a closing tag: **</USAGE>**, as shown in the following example.

```
<?XML VERSION="1.0"?>
<CHANNEL>
        <SCHEDULE ENDDATE="1998.11.05T08:15-0500">
        <INTERVALTIME DAY="1"  />
        <EARLIESTTIME HOUR="12"/>
        <LATESTTIME   HOUR="18"/>
</SCHEDULE>
    <ITEM HREF="http://www.XYZ.com/desktop.asp"
        LASTMOD="1996.11.05T08:15-0500">
        <TITLE>My News Ticker</TITLE>
        <USAGE VALUE="DeskopComponent">
            <OPENAS VALUE="HTML"/>
            <WIDTH VALUE="200" />
            <HEIGHT VALUE="80"  />
            <CANRESIZE VALUE="Yes"/>
        </USAGE>
    </ITEM>

        <ITEM HREF="control2.ocx">
            PRECACHE="Yes">
            <USAGE VALUE="None">
    </ITEM>
</CHANNEL>
```

Use the HEIGHT and WIDTH attributes to specify the size in pixels of the display frame on the user's desktop. By default, users can resize the frames on their desktops. However, you can specify the **CANRESIZE="No"** attribute to prevent users from resizing the frame.

Desktop items can contain unlimited ITEM elements that do not appear in the desktop frame. For example, extra items can reference support content, such as ActiveX controls or audio files, that needs to be downloaded to the cache on the subscriber's hard disk to enable offline use of the desktop item.

For more information about Active Desktop items, see Chapter 14.

Channel Screen Savers

You can designate specific URLs as screen-saver pages by nesting the **USAGE VALUE="ScreenSaver"** attribute within ITEM elements in the top-level channel.

```
<ITEM HREF="http://www.companyXYZ.com/screensaver.htm">
    <USAGE VALUE="ScreenSaver" />
</ITEM>
```

Note If **USAGE VALUE="ScreenSaver"** occurs within an ITEM element within a subchannel, the screen saver will be ignored.

Channel screen savers should download supporting content, such as ActiveX controls and Java applets, needed for the screen saver to function offline. Use additional ITEM elements with the PRECACHE="Yes" attribute and the USAGE VALUE="None" element to download support content to subscribers' caches.

Personalized Webcasts

You can personalize Webcasts by configuring your Web servers to dynamically generate CDF files and HTML pages for individual users or user groups. You need server software, such as Microsoft Internet Information Server, that supports generating dynamic CDF and HTML. You also need a way to identify users; you can use network user profiles or HTTP cookies to do this.

For example, when a user subscribes to a personalized channel, the Web server generates a CDF file based on predefined criteria and user preferences. An HTML form is used to query users for information and preferences during the subscription process. User information and preferences are then stored in a database on the server and in a cookie on the user's computer.

For an intranet news channel, you could configure the Web server so users in different departments receive different CDF files specifying news items related to their department. The URLs contained in the CDF file can also reference dynamically generated and personalized content. For more information about generating dynamic Web content, see the documentation for your Web server software.

In order for the dynamically generated CDF to be recognized, it must have the proper MIME/content type: **application/x-cdf.** To generate a CDF response from Internet Information Server Active Server Pages, the following line of code must be placed at the top of the page that generates the CDF:

```
<% Response.ContentType = "application/x-cdf" %>
```

Without this line, browsers will not perform the expected actions for CDF files. For example, the CDF file could be displayed as text in the browser window rather than launch the channel Subscription Wizard.

Example of a Personalized Channel

As an example, a software company could provide an Active Channel that would allow users to receive the latest information on software products, including support information on these products. In order to be effective, a personalized channel could be employed so that a user gets information only on products about which he or she is interested, rather than every product manufactured by that company.

The following steps illustrate how a software company, such as Microsoft, could create a personalized channel to provide product information, using Active Server Pages (ASP):

1. Create an .htm page that allows users to select the Microsoft products they want information about and save the file as Products.htm.

   Products.htm uses a form to query users which product they want information on. (For simplicity, this example includes only two of the Microsoft products currently available.)

```
<HTML>
    <TITLE>Microsoft Custom Subscription List</TITLE>
    <FONT FACE="VERDANA,ARIAL,HELVETICA" SIZE="2">
    <H1>Microsoft Custom Product Subscription List</H1>

    <FONT FACE="courier" SIZE="3">
    <PRE>
    <FORM METHOD="GET" ACTION="GenCDF.asp">
        <INPUT TYPE="checkbox" Name="IE4"     VALUE="1">Internet
            Explorer
        <INPUT TYPE="checkbox" Name="InetSDK" VALUE="1">Internet
            Client SDK
    </PRE>
    </FONT>
        <INPUT TYPE=SUBMIT Value="Subscribe" Name="Validate" >
    </FORM>
</HTML>
```

2. Create an .asp file that uses the information submitted from the form in Products.htm to dynamically generate the personalized CDF file. Save this file as Gencdf.asp.

   Gencdf.asp uses VBScript to generate the personalized CDF file for each user.

```
<%@ LANGUAGE=VBScript %>
<%
IE4 = Request ("IE4")
InetSDK = Request ("InetSDK")
%>
```

```
<% Response.ContentType = "application/x-cdf" %>

<?XML VERSION="1.0"?>
    <CHANNEL HREF="http://www.microsoft.com">
        <TITLE>Sample MS Custom Channel</TITLE>
    <ABSTRACT>Demo Channel on how to get a customized subscription
     using ASPs</ABSTRACT>

        <SCHEDULE>
            <INTERVALTIME DAY ="1" />
            <EARLIESTTIME HOUR="1" />
            <LATESTTIME HOUR="5" />
        </SCHEDULE>

<% If Not IE4 = 0 Then %>
    <CHANNEL HREF="http://www.microsoft.com/ie/ie40">
        <TITLE>IE4 Support Page</TITLE>
        <ABSTRACT>IE4 Product Page</ABSTRACT>
        <ITEM HREF="http://www.microsoft.com/iesupport">
        <ABSTRACT>This is where you can get all your questions on
         IE4 answered.</ABSTRACT>
    </ITEM>
    </CHANNEL>
<% End if %>

<% If Not InetSDK = 0 Then %>
    <CHANNEL HREF="http://www.microsoft.com/workshop/prog/inetsdk">
        <TITLE>Internet Client SDK</TITLE>
        <ABSTRACT>Internet Client SDK Product Page</ABSTRACT>
        <ITEM HREF="http://www.microsoft.com/support/inetsdk">
            <TITLE>Internet Client SDK Support Page</TITLE>
            <ABSTRACT>This is where you can get all your questions on
             IE4 programmability answered. </ABSTRACT>
        </ITEM>
</CHANNEL>
<% End if %>
</CHANNEL>
```

Again, note that CDF files that are dynamically generated through ASP should have the following line inserted at the top of the CDF file (before any CDF lines):

<% Response.ContentType = “application/x-cdf” %>

This ensures that servers return the MIME/content type "application/x-cdf." Problems with CDF files being displayed inside the browser as HTML text rather than presenting a Subscription Wizard are very likely caused by the server returning an incorrect MIME type of "text/html."

Note As an alternative for IIS 4.0 site authors, the CDF file can be created with a .cdx extension, instead of .asp, and the MIME content type line above can be omitted.

- For simplicity, the <LOGO> tags have been omitted in the sample CDF file. As a result, the default satellite dish logo is displayed in the Desktop Channel bar and in the Channel pane.

3. Place the Products.htm and Gencdf.asp files in the same directory on an IIS Server and then on he Internet Explorer address bar.

When users browse to the Web page (Products.htm), they fill out the form and submit the form to obtain the information they want for the specified product. The ASP page Gencdf.asp generates the personalized CDF file, and the Subscription Wizard opens so users can subscribe to the personalized channel. The personalized channel provides information on the product the user selected.

Design Guidelines for Active Channels

To take true advantage of the Webcasting mechanism available in Internet Explorer 4, channels should be as small as possible to minimize download time and provide compelling content and interactions when used offline. These guidelines will assist you in designing an effective Active Channel:

- Channels intended for offline use should consider the impact of bandwidth limitations. If the target user is a remote user dialing over a modem, the average amount of content delivered per update per channel should be no more than 500 KB to 1 MB, preferably smaller. Under ideal conditions, a megabyte of content can be downloaded via a 28.8K modem in about 6 minutes. In practice, however, depending on line and server conditions, downloading a megabyte takes about 10 minutes.
- Active Channels should be self-contained. The channel must deliver real value to the user while offline. It is likely that users will become annoyed with an offline channel that constantly forces online connections. This means that links to noncached resources should be minimized or hidden when the user is offline. Scripting can be used on an HTML page to determine when the user is offline and re-flow the page accordingly, using Dynamic HTML.

- Channels should be personalized whenever possible. A publisher can provide a personalized channel experience for each user by generating a custom CDF file for each user (based on user preferences stored in a cookie). Channels based on personalized CDF files can be more rapidly delivered, because content that is not important to the user will not be downloaded.

For design guidelines specifically related to Active Desktop items, see "Developing Active Desktop Items," in Chapter 14, "Active Desktop."

Subscribing to Active Channels and Desktop Items

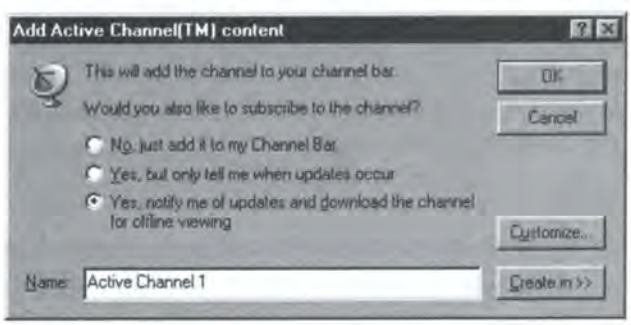
You subscribe to an Active Channel or Active Desktop item by opening the URL for the channel's CDF file. Publishers of Active Channels and desktop items normally provide a link that you can click to subscribe.

▶ **To subscribe to an Active Channel**

1. While browsing the site, click the subscription link.

   (To get to the site, you can enter the URL and CDF file name directly into the Internet Explorer Address bar or, on the **File** menu, click **Open**, and enter the information in the resulting dialog box.)

2. In the **Add Active Channel** dialog box, select one of the following subscription options:
   - **Yes, but only tell me when updates occur** to subscribe, but not download, updated content to the cache.
   - **Yes, notify me of updates and download the channel for offline viewing** to subscribe and download channel content to the cache.



3. To customize subscription options, click **Customize** and use the Subscription Wizard to configure subscription options. For more information, see "Customizing Active Channels and Desktop Items" next in this chapter.
4. Click **OK**.

▶ **To subscribe to an Active Desktop Item**

1. While browsing the site, click the subscription link.

   (If you know the URL and the CDF file name, you can get to the site by entering the URL and file name directly into Internet Explorer's Address bar or, on the **File** menu, click **Open**, and enter the information in the resulting dialog box.)

2. To customize subscription options, click **Customize Subscription**, and use the Subscription Wizard to configure subscription options. For more information, see "Customizing Active Channels and Desktop Items" next in this chapter.
3. Click **OK**.

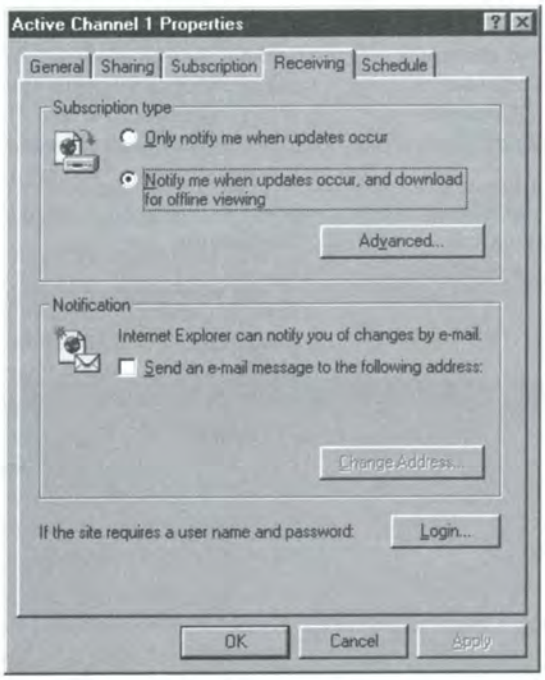
Customizing Active Channels and Desktop Items

You can customize most Active Channel and Active Desktop item options by using the Subscription Wizard when you subscribe to sites. After subscribing to a channel or desktop item, you can use the **Receiving** and **Schedule** tabs in the channel's **Properties** dialog box to configure all subscription options.

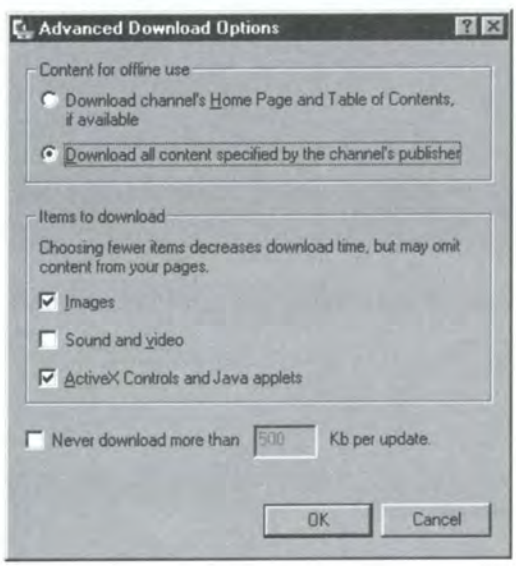
Most of the customization options for channels are the same as the customization options for subscriptions. (For more information, see "Customizing Subscriptions" in Chapter 15, "Basic Webcasting.") However, the content download options for the **Advanced Download Options** dialog box are different for channels and are described below.

▶ **To configure advanced download options**

1. On the **Favorites menu**, click **Manage Subscriptions**.
2. In the **Manage Subscriptions** dialog box, right-click the channel or desktop item to which you want to subscribe, and click **Properties** on the shortcut menu.
3. Click the **Receiving** tab.



4. On the **Receiving** tab, click **Advanced**.
5. In the **Advanced Download Options** dialog box, configure the options.



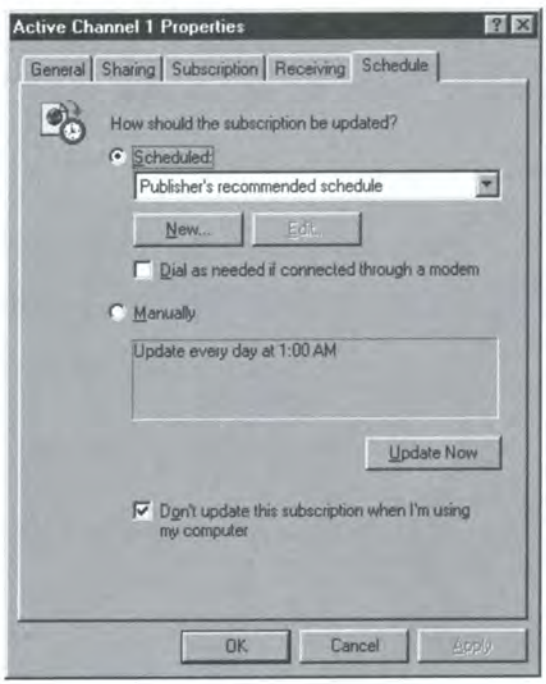
Option	Description
Download channel's Home Page and Table of Contents, if available	Select to download the top-level channel's home page and all subchannel pages as specified in the CDF file. When this option is selected, URLs referenced by ITEM elements are not downloaded, and the Web site may not function properly when browsed offline.
Download all content specified by the channel's publisher	Select to download all channels and items specified in the CDF file.
Images	Clear to prevent downloading graphical images when Internet Explorer updates the selected Web site. If you choose not to download images, the Web site may appear incomplete or not display properly when browsed offline.
Sound and video	Clear to prevent downloading sound and video files when Internet Explorer updates the selected Web site. If you choose not to download these files, the Web site may not function properly when browsed offline.
ActiveX Controls and Java applets	Clear to prevent downloading ActiveX controls and Java applets when Internet Explorer updates the selected Web site. If you choose not to download these programs, the Web site may not function properly when browsed offline.
Never download more than __ Kb per update.	Select this option and enter the number of kilobytes to limit the maximum amount of information that Internet Explorer downloads to your hard disk from this Web site for each update.

6. Click **OK**.

You can also accept the publisher's recommended update schedule using the channel's **Schedule** tab.

▶ **To accept the publisher's schedule**

1. On the **Favorites** menu, click **Manage Subscriptions**.
2. In the **Manage Subscriptions** dialog box, right-click the channel or desktop component to which you want to subscribe, and click **Properties** on the shortcut menu.
3. Click the **Schedule** tab.
4. Select **Scheduled**.



5. Select the publisher's schedule from the list of schedules.
6. Click **OK**.

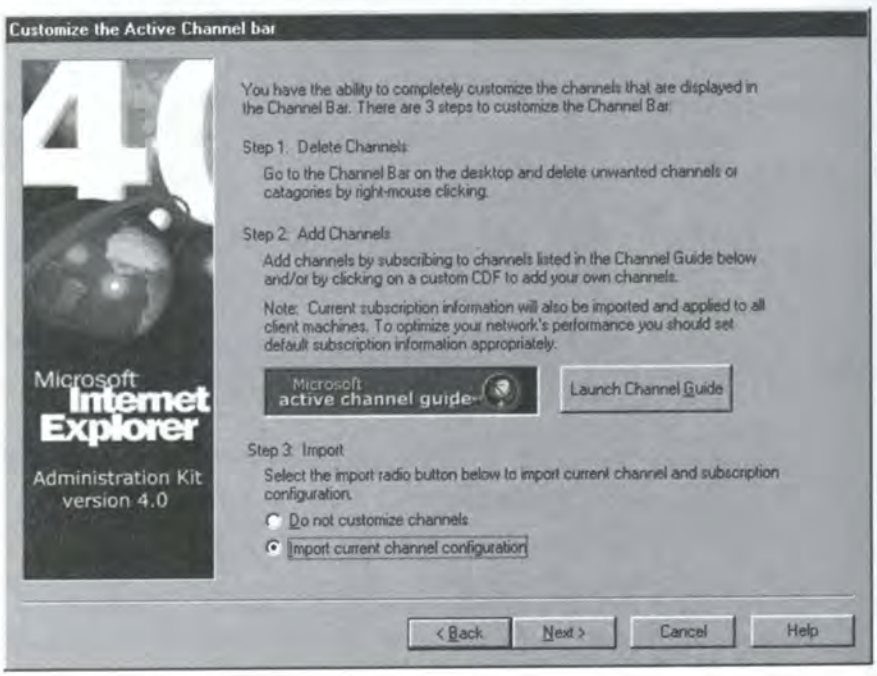
Deploying Custom Packages with Predefined Channels for User Groups

System administrators can use the IEAK 4 Configuration Wizard to build custom packages with pre-configured Active Channels and Active Desktop items for their organizations. Using the IEAK, you can also build custom packages with specific settings for different user groups. (For more information, see "Building Custom Packages" in Chapter 36, "Implementing the Deployment.")

To create a custom package, you configure the Active Channels and Active Desktop items (components) on the computer running the IEAK Configuration Wizard and then import the channel and desktop item settings using the **Customize the Active Channel bar** and **Specify Active Desktop Components** pages of the wizard.

▶ **To import Active Channels**

1. On the **Customize the Active Channel bar** page of the IEAK Configuration Wizard, select **Import current channel configuration**.



2. Click **Next**.

▶ **To import Active Desktop items (components)**

1. On the **Specify Active Desktop Components** page of the IEAK Configuration Wizard, select **Import the current Active Desktop components**.

2. Click **Next**.

Finish building the custom package, as necessary.

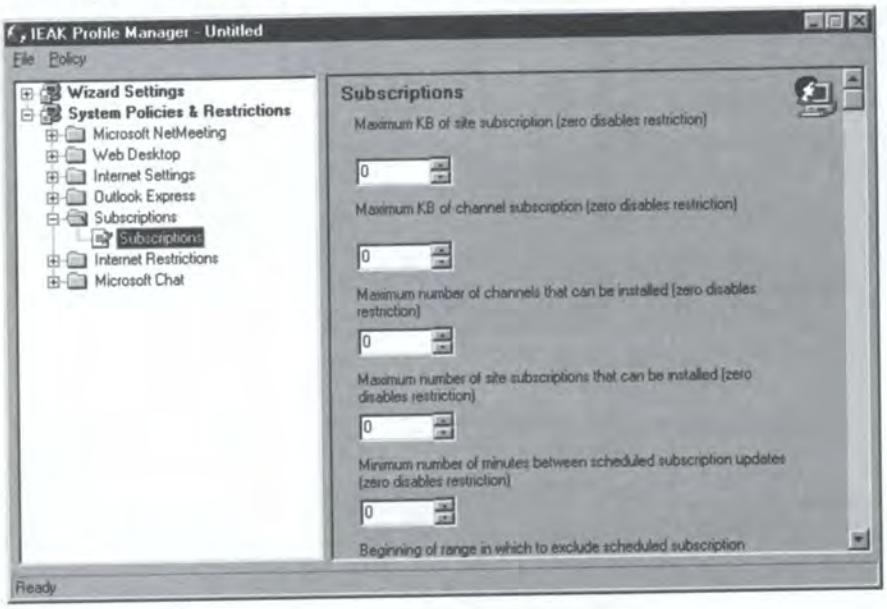
The Active Channel and Active Desktop item configurations you imported from the current desktop are installed on users' desktops when the custom package is deployed. You can also lock down Active Channel and desktop item settings so users cannot modify the settings.

Locking Down Active Channels and Desktop Items

Once you've built your custom packages, you can use the IEAK Configuration Wizard to lock down channel options and restrict users from changing settings. After browsers are deployed, you can use the IEAK Profile Manager to lock down channel options for user groups that have automatic browser configuration implemented. (For more information about automatic browser configuration, see Chapter 37, "Automatic Browser Configuration.")

▶ **To lock down channel options**

1. On the **System Policies and Restrictions** page of the IEAK Configuration Wizard or at the IEAK Profile Manager, in the System Policies & Restrictions hierarchy, select **Subscriptions**, and then select the **Subscriptions** object.

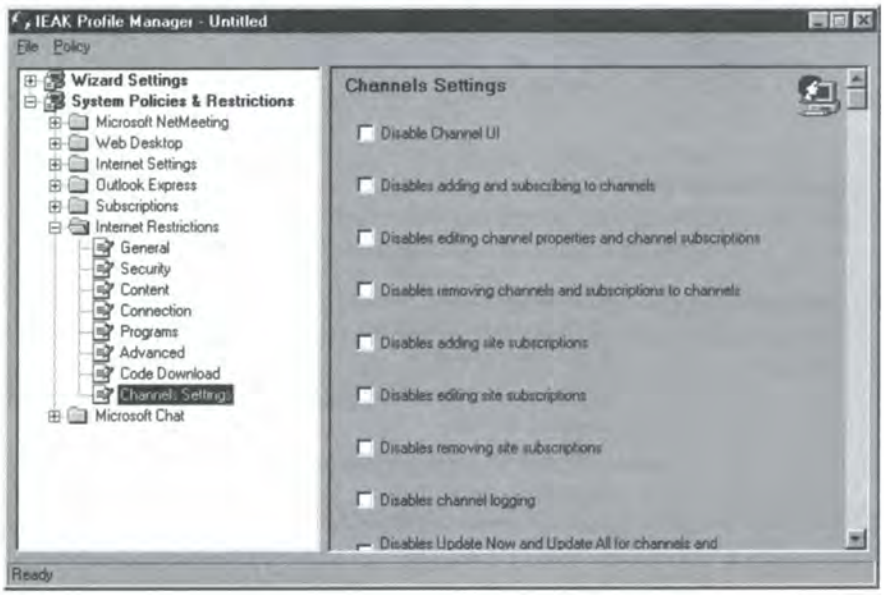


Note The IEAK Profile Manager is shown here, but the **System Policies and Restrictions** page of the IEAK Configuration Wizard works similarly.

2. In the right pane, configure the subscription options.

Option	Description
Maximum KB of channel subscription (zero disables restriction)	Enter the maximum kilobytes that can be downloaded per channel update for each user. Users are prevented from downloading more than the specified number of kilobytes per channel update.
Maximum number of channel subscriptions that can be installed (zero disables restriction)	Enter the maximum number of channel subscriptions that can be installed on users' computers. Users are prevented from installing more than the specified number of channels locally.

3. In the System Policies & Restrictions hierarchy, select **Internet Restrictions**, and then select the **Channels Settings** object.



Note The IEAK Profile Manager is shown here, but the **System Policies and Restrictions** page of the IEAK Configuration Wizard works similarly.

4. In the right pane, configure the channels settings options.

Option	Description
Disable Channel UI	Select to prevent users from using the Active Channel interface. When this option is selected, users cannot add or subscribe to channels, delete or unsubscribe from channels, manually update channels, or change any subscription settings for the subscribed channels.
Disables adding and subscribing to channels	Select to prevent users from adding and subscribing to new Active Channels.
Disables editing channel properties and channel subscriptions	Select to prevent users from editing channel settings using the tabs in the channel's **Properties** dialog box.

Option	Description
Disables removing channels and subscriptions to channels	Select to prevent users from removing channels or unsubscribing from channels.
Disables channel logging	Select to prevent channel logging on users' computers. Publishers can specify channel logging in CDF files. The channel log file resides on subscribers' hard disks and logs information about how the channel is used, even when browsed offline. Select this option if you don't want log files residing on your users' computers.
Disables Update Now and Update All for channels and subscriptions	Select to prevent users from manually updating channels or site subscriptions.
Disables all scheduled channel and site subscriptions	Select to prevent automated Web crawls and updates from users' computers.
Disables unattended dialing by subscriptions	Select to prevent automated dial-up connections to update channel and site subscriptions for users who have dial-up connections.
Disables password caching for channel and site subscriptions	Select to prevent users' authentication passwords from being stored and submitted automatically to channels and sites that require user authentication. When this option is selected, users will be prompted to enter the authentication password each time Internet Explorer connects to secure channels or sites.
Disable downloading of channel subscription content-change notification will still work	Select to prevent the download of channel content to the cache on users' computers. When this option is selected, users cannot browse the channel offline.
Disables editing and creating of schedule groups	Select to prevent users from creating custom automated schedules for updates to subscribed channels and sites.

5. Save your changes to the auto-configuration (.ins) file, or finish building the custom package as appropriate.

Managing Channels for Deployed Browsers

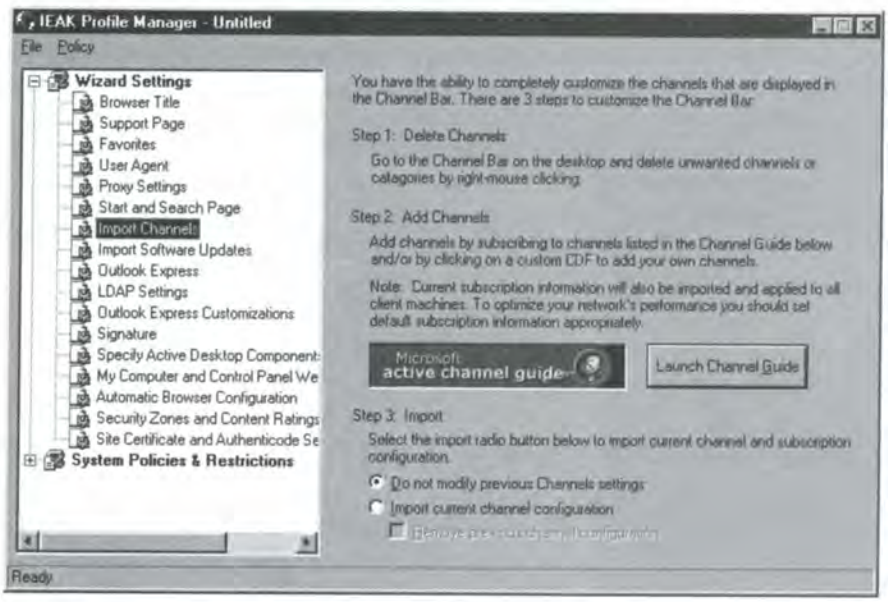
To manage Active Channels and Active Desktop items for deployed browsers, you can use the IEAK Profile Manager in conjunction with automatic browser configuration. (For more information about automatic browser configuration, see Chapter 37, "Automatic Browser Configuration.")

To change Active Channels and desktop items for a group, configure the channels and desktop items on the machine running the IEAK Profile Manager, and then use the Profile Manager to import the configurations and save the new configuration settings to the auto-configuration files for the user group.

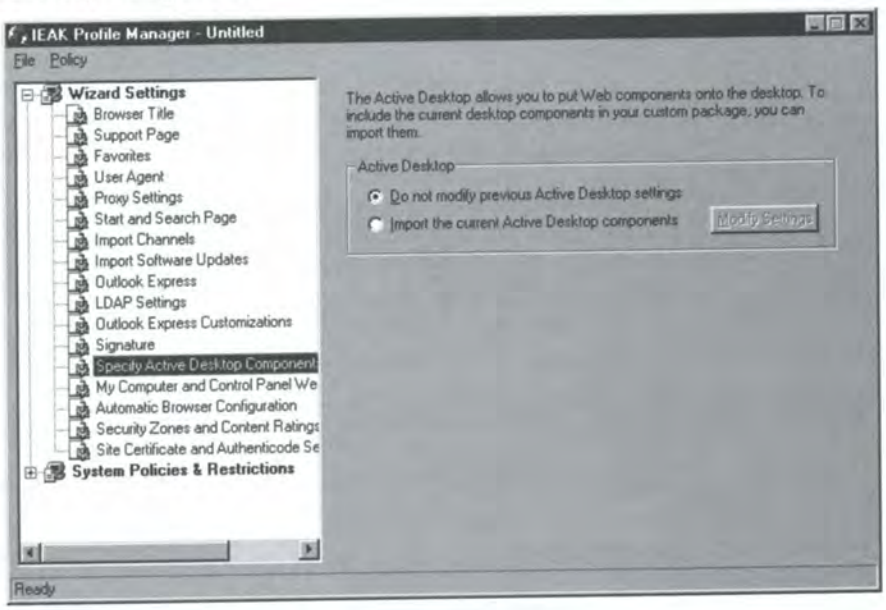
When automatic browser configuration is refreshed at users' computers, users' desktops are configured with the new Active Channel and Active Desktop settings you specify. You can also optionally remove previous Active Channel configurations from users' computers.

▶ **To change Active Channel and desktop-item configurations**

1. Use Internet Explorer to configure channels and desktop items for the current desktop.
2. Run the IEAK Profile Manager.
3. Open the auto-configuration (.ins) file for the user group.
4. In the left pane, select **Wizard Settings**, and then select **Import Channels**.



5. In the right pane, select **Import current channel configuration**.
6. If you want to remove previous channel configurations from users' computers, select **Remove previous channel configuration**.
7. In the left pane, select **Wizard Settings**, and then select **Specify Active Desktop Components**.



8. In the right pane, select **Import the current Active Desktop components**.
9. On the **File** menu, click **Save**.

   Changes are made to users' configurations as scheduled by automatic browser configuration.

CHAPTER 17

True Webcasting

True Webcasting uses true push technology to deliver content to users' desktops while optimizing network bandwidth and performance. Users can subscribe to an information source, and the information is streamed from the server to the users' computers. Applications such as Microsoft NetShow can be used with Internet Explorer 4 to implement information delivery solutions that overcome the limitations of basic and managed Webcasting.

The Webcasting architecture in Internet Explorer 4 allows developers to create add-ons that define new URL transport protocols or provide an alternative delivery mechanism for Active Channels. For more information, see the *Microsoft Internet Client SDK.*

Microsoft NetShow uses the extensible architecture of Internet Explorer 4 to support Internet Protocol (IP) multicast or "true push." By taking advantage of special network hardware, multicast protocols provide bandwidth-efficient streaming of content throughout an intranet. If your network hardware supports multicast protocols, your organization can install and use NetShow servers and client software to deliver rich multimedia content to users' desktops with more bandwidth efficiency and better network performance than is possible using managed Webcasting. For more information about IP multicast technology and Microsoft NetShow, visit **http://microsoft.com/netshow/** and also see Chapter 19, "NetShow."

PART 5

Communication and Collaboration

Collaboration across the Internet has become one of the most exciting, avidly discussed topics today. The Microsoft® Internet Explorer 4 suite of applications provides solutions for all Internet-based communication needs users may have. Using its modular installation program or the Internet Explorer Administration Kit (IEAK), users and administrators can set up only the pieces they need or take advantage of the flexibility of the Internet Explorer suite and integrate it with their existing software solutions.

The Internet Explorer suite allows seamless integration from one application to the next, because the applications are all tightly integrated and have been developed with a common menu and toolbar. This simplifies training, because a user who learns one application in the suite already has a head start in learning the next one.

The Internet Explorer 4 suite contains these components for communication and collaboration across the Internet or local intranet:

- Outlook™ Express for e-mail and newsgroups
- NetShow™ for broadcasting
- NetMeeting™ for conferencing and application sharing
- FrontPage® Express for Web authoring
- Personal Web Server for Web publishing

An organization that uses the Internet Explorer suite does not need to discard its existing tools. For example, a corporation can use its existing messaging solution and still integrate with the Internet Explorer suite. In addition, Internet Explorer offers a scalable solution for users who need to move up to high-end applications. Microsoft Outlook can replace Outlook Express mail for those who need a richer mail client. While FrontPage Express is terrific for creating Web pages, Microsoft FrontPage is the full Web-site development platform. Then, for a true enterprise Web server, users can move from the Personal Web Server to Microsoft Internet Information Server, which has all the security, scalability, and power that a Web site needs for the Internet today. Internet Explorer 4 integrates with these high-end applications just as easily as it does with its built-in components.

Part 5 discusses each of the components in the Internet Explorer suite and explains how you can use it to improve communication and collaboration in your organization.

Chapter 18: Outlook Express

E-mail has become one of the most popular and effective ways for people to communicate, both in business and in their personal lives. Unfortunately, most e-mail has been limited to text-only messages, perhaps with attachments. Internet Explorer 4 allows an entirely new type of standards-based messaging, opening the door to greater richness and detail. Outlook Express supports full Hypertext Markup Language (HTML), so you can create e-mail messages with the color and functionality of Web pages, without knowing how to program HTML code. You can even send full Web pages from the Internet as part of your mail message.

Plus, Outlook Express provides users with powerful mail management features, more efficient e-mail and newsgroup use, enhanced security, and full support for Internet standards and technologies. Outlook Express is flexible enough to meet the diverse e-mail needs of users with dial-up Internet access through an Internet service provider (ISP) and users who work on a local area network based on Internet standards such as Simple Mail Transfer Protocol (SMTP), Post Office Protocol 3 (POP3), or Internet Mail Access Protocol 4 (IMAP4).

Chapter 19: NetShow

NetShow brings the power of broadcasting to the desktop. With NetShow, Web content comes alive with interactive content, including audio, illustrated audio (images and sound), and video. It includes both client and server components to add the power of traditional broadcasting systems—audio and video—to HTTP. NetShow harnesses Internet technologies and the power of Windows NT® server to transform Web communications into a richer and more effective medium: the network show.

Chapter 20: NetMeeting

Microsoft NetMeeting is a powerful tool that enables real-time communications and collaboration over the Internet or intranet, providing standards-based audio, video, and multipoint data conferencing support. From a Windows® 95– or Windows NT 4.0–based desktop, users can communicate over the network with real-time voice and video technology. They can share data and information with many people through true application sharing, electronic whiteboard, text-based chat, and file-transfer features. NetMeeting helps organizations communicate more effectively, thus increasing productivity for their users. Also, NetMeeting helps organizations reduce their support burden by providing real-time application sharing features, so that help-desk staff can remotely resolve user problems more quickly and efficiently.

Designed for corporate communication, NetMeeting supports international communication standards for audio, video, and data conferencing. With NetMeeting, people can connect by modem, ISDN, or local area network using the TCP/IP protocol, and communicate and collaborate with users of NetMeeting and other standards-based, compatible products. In addition, support for system policies in NetMeeting makes it easy for administrators to centrally control and manage the NetMeeting work environment.

Chapter 21: FrontPage Express

While HTML has made it possible for many people to become Web publishers, it is still not a particularly intuitive language. For this reason, Microsoft FrontPage Express has been included as a component of the Internet Explorer suite. Based on the full-featured Microsoft FrontPage 97 Web authoring and management tool, FrontPage Express features a graphical interface that makes creating HTML pages as easy as creating a document in a word processor. FrontPage Express allows users to create their own Web pages in a what-you-see-is-what-you-get (WYSIWYG) environment, without knowing HTML. FrontPage Express includes all the features of the FrontPage 97 editor, except for premium features such as Active Server Pages and some of the WebBot® components that rely on specific server extensions. WebBot components from FrontPage 97, such as Include, Search, and Time-Stamp, are still included, however.

Chapter 22: Personal Web Server

As millions of users flock to the Internet for information and entertainment, many decide that they also want to share information with the world. Personal Web Server offers a way for users and corporations to publish Web pages on their own servers, while the Web Publishing Wizard offers users the opportunity to publish Web pages on their own server or on a third-party server. The simplicity of these tools makes them perfectly designed for use by home users, schools, and corporate workgroups.

CHAPTER 18

Outlook Express

In this chapter

Outlook Express Overview 248
- Setup and Migration Tools 249
- Web Integration 249
- LDAP Support 250
- Security 250
- Support for Internet Standards 251
- Outlook Bar 252
- InBox Assistant 253
- Newsgroup Filters 254
- Subscription and Channel Delivery by E-Mail 254
- Smart Reply 254
- Stationery 255
- Multiple E-Mail and News Server Accounts 256
- Productivity Features 256
- Integration with the Internet Explorer Suite 257

Understanding Outlook Express 258
- Outlook Express Architecture 258
- Outlook Express Flat File Directory 258
- Outlook Mail Data Directory 259
- Outlook News Data Directory 259
- Windows Address Book 260

Configuring and Customizing Outlook Express 261
Specifying Mail and News Servers 261
Importing Messages and Address Book Information 262
Configuring the LDAP Directories 264
Maintaining Directory Service Settings 266
Using the InBox Assistant 267
Customizing the InfoPane 268
Creating a Welcome Message 270
Appending a Standard Signature 270
Customizing the Appearance of Outlook Express 270
Creating Custom Stationery 271
Understanding Security in Outlook Express 274
Using Security Features in Outlook Express 275
Obtaining a Digital ID 275
Sending and Receiving Signed Mail 276
Sending and Receiving Encrypted Mail 276
Changing Security Zones Settings for Outlook Express 278

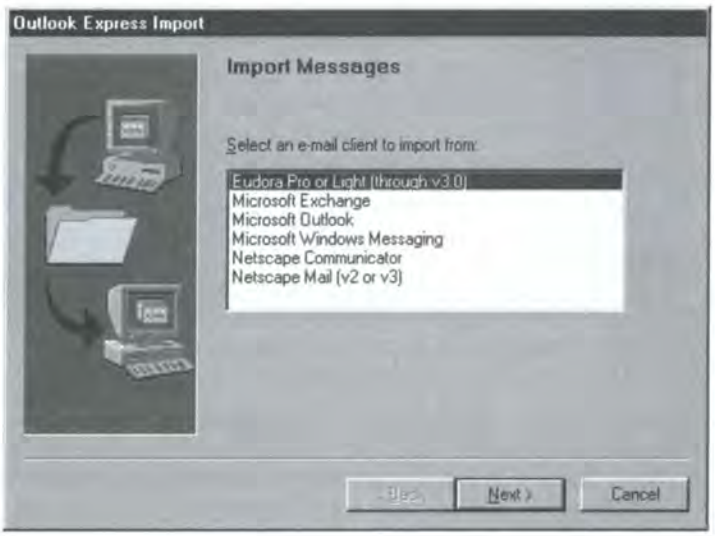
Outlook Express Overview

Outlook Express offers a wide range of new features that make it easy to communicate more effectively with others, whether they're down the hall, across the city, or around the world. Important features include:

- Setup and migration tools
- Web integration
- Support for Lightweight Directory Access Protocol (LDAP)
- Support for Secure Multipurpose Internet Mail Extensions (S/MIME)
- Support for Internet Mail Access Protocol (IMAP4)
- The Outlook bar
- Newsgroup filters
- Subscription and channel delivery by e-mail
- Smart Reply
- Stationery
- Multiple e-mail and news server accounts
- Productivity features
- Integration with the Internet Explorer suite

Setup and Migration Tools

Users can get up and running easily with Outlook Express. The Internet Connection Wizard guides users through each step of establishing new mail, news, and LDAP accounts. Migration is simplified because Outlook Express automatically detects and offers the opportunity to import mail messages, news settings, and address books from Netscape Messenger, Eudora, and Internet Mail, on first boot. Users can also very easily import information from these products at their convenience. Of course, Outlook Express can also import information from Microsoft Exchange and Microsoft Outlook.

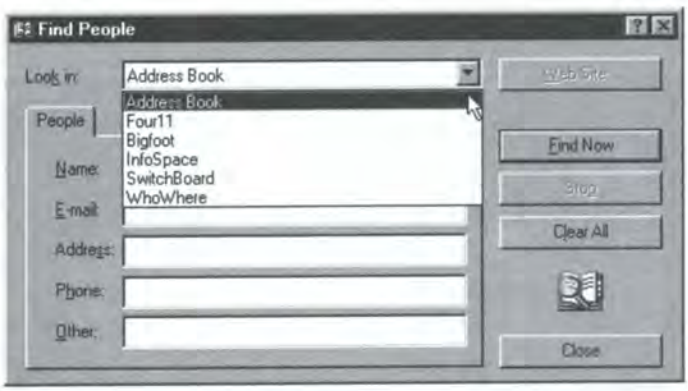


Web Integration

As the successor to Microsoft Internet Mail and News, Outlook Express further integrates e-mail with the Web by using the Internet Explorer Web browser control. Outlook Express supports full Hypertext Markup Language (HTML) as a native message format, so you can create messages in HTML and communicate using the richness of Web pages, without knowing how to program HTML code. In fact, with support for MIME HTML, users can send full Web pages from the Internet or intranet to each other. The **Insert HTML** command also lets users quickly and easily insert content from existing Web pages into messages. To maintain compatibility, messages created in Outlook Express are readable by both HTML-capable and non-HTML-capable e-mail clients. Outlook Express also doubles as a newsreader, so it is not necessary to switch between two separate applications for mail and news.

LDAP Support

Internet Explorer 4 has full support for Lightweight Directory Access Protocol (LDAP) directory services, which provide access to virtual Internet white pages. This makes it easy to find anyone on corporate LDAP servers or to use the built-in support for Four11, InfoSpace, Bigfoot, Switchboard, and WhoWhere? to locate anyone on the Internet. Internet Explorer also includes support for vCard, for exchanging business card information.



Security

Internet Explorer 4 features the latest in state-of-the-art technology for security. Support for the Secure Multipurpose Internet Mail Extensions (S/MIME) standard allows users to encrypt their messages, digitally sign messages, and certify senders with digital certificates. Public key cryptography is used to facilitate the encoding and decoding of encrypted messages. Outlook Express also incorporates Internet Explorer security zones, which protect you as you access certain Web sites and as you receive mail with Web content. (For more information about security zones, see Chapter 27.)

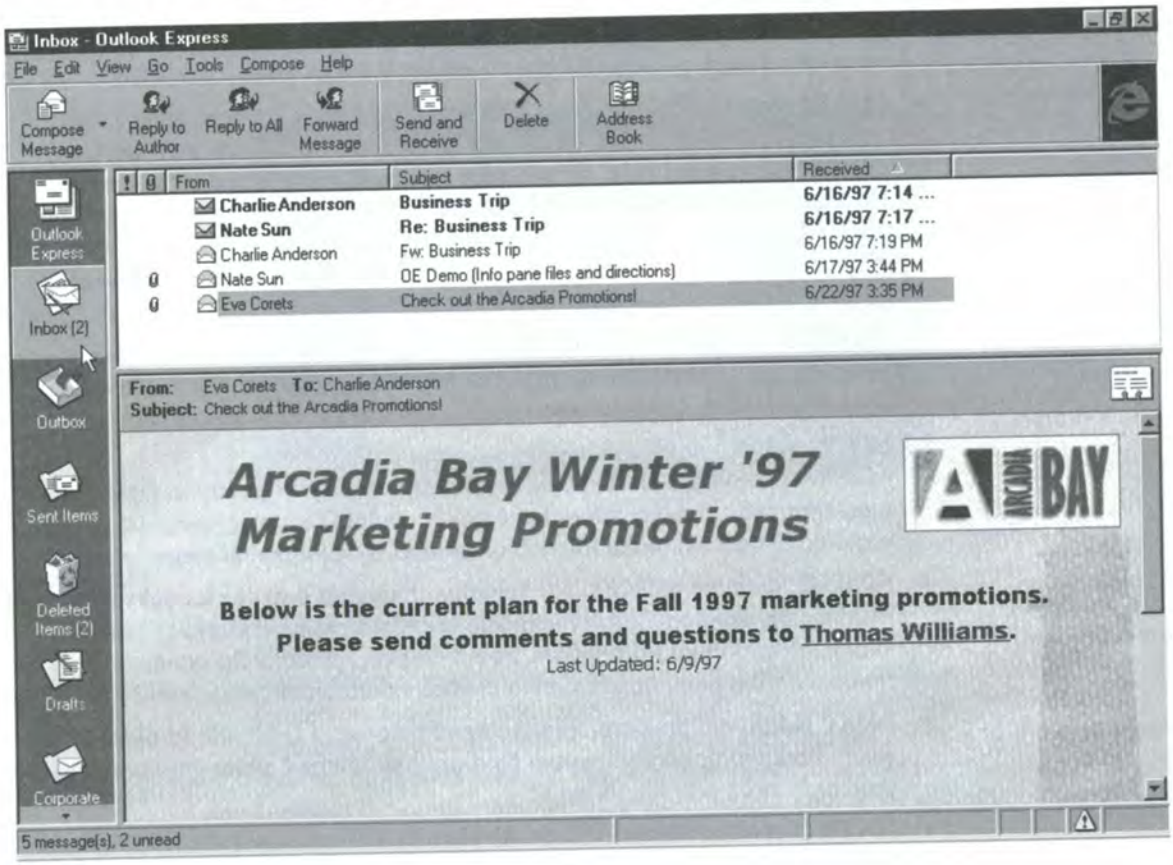
Support for Internet Standards

Outlook Express provides full support for the following Internet standards:

- **SMTP (Simple Mail Transfer Protocol).** This protocol is a server standard for sending e-mail.
- **POP3 (Post Office Protocol 3).** This protocol is a server standard for receiving e-mail.
- **IMAP4 (Internet Mail Access Protocol).** IMAP4 is the next-generation standard for e-mail messaging. It allows users to access their mail from any computer on the network, by allowing messages to be stored on the server. For example, users can access their e-mail from work and from home. Support for IMAP4 also offers improved bandwidth utilization. For example, users can choose to download headers only. Plus, IMAP4 provides central mail-store administration, such as mail-store backup.
- **HTML Mail.** Users can send e-mail messages in HTML format. These messages retain their formatting, even if read by an e-mail client other than Outlook Express. Text and attachments can also be read by non-HTML e-mail clients.
- **MHTML (MIME HTML).** Users can embed images directly in their messages, which makes for richer e-mail content. This way, users are not required to connect to the Internet or intranet to view the message contents.
- **S/MIME (Secure MIME).** The security of a user's e-mail messages is ensured by allowing users to digitally sign and encrypt messages. Digital signatures verify the integrity of the sender, and encryption protects the contents of messages from being read by anyone except their intended recipient.
- **LDAP (Lightweight Directory Access Protocol).** LDAP allows users to easily find people on the Internet by searching Internet white-page services such as Four11 and WhoWhere?
- **NNTP (Network News Transfer Protocol).** This standard allows news clients to read and post to newsgroups. It also enables communication between news servers.

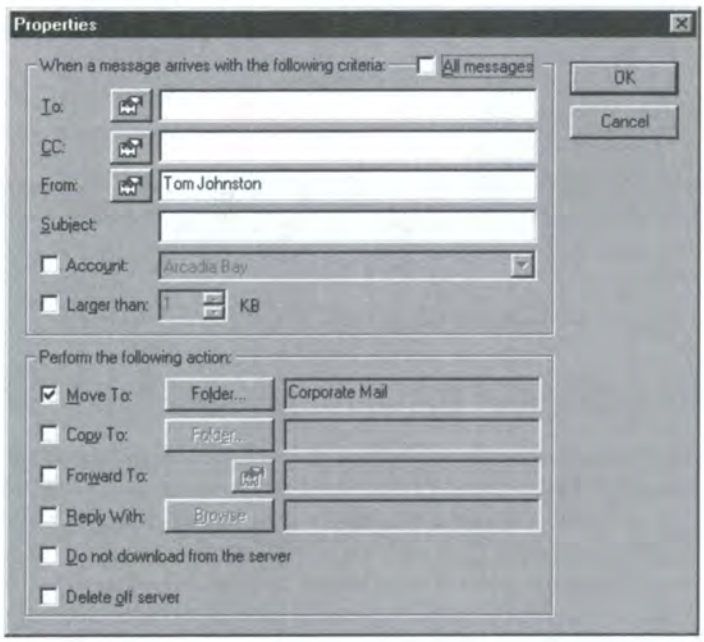
Outlook Bar

Outlook Express offers the same style of navigational tool that Microsoft Outlook 97 introduced. The Outlook bar allows easy access to different folders and modules and has been one of the most popular features of Outlook 97.



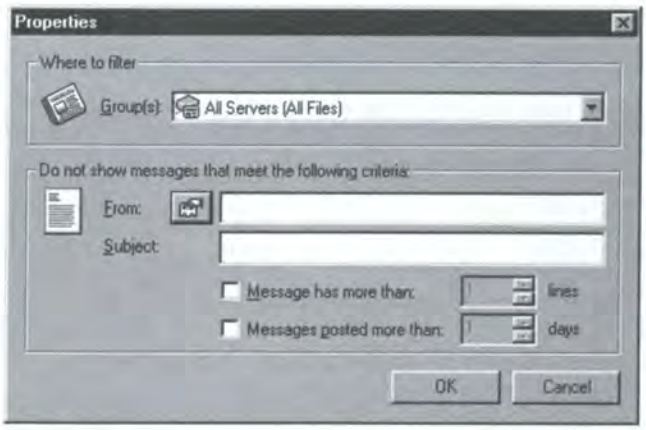
InBox Assistant

Users can develop e-mail rules for their mail to help them save time and stay organized. Filters can also be created to forward, move, or copy messages automatically. Plus, new rules can be applied to existing folders. Now you can use powerful queries to organize the messages that you've already received, in addition to your new incoming mail.



Newsgroup Filters

Users can create filters on an individual-newsgroup basis. Filters can be created to ignore messages based on sender, subject, date posted, or length of message.



Subscription and Channel Delivery by E-Mail

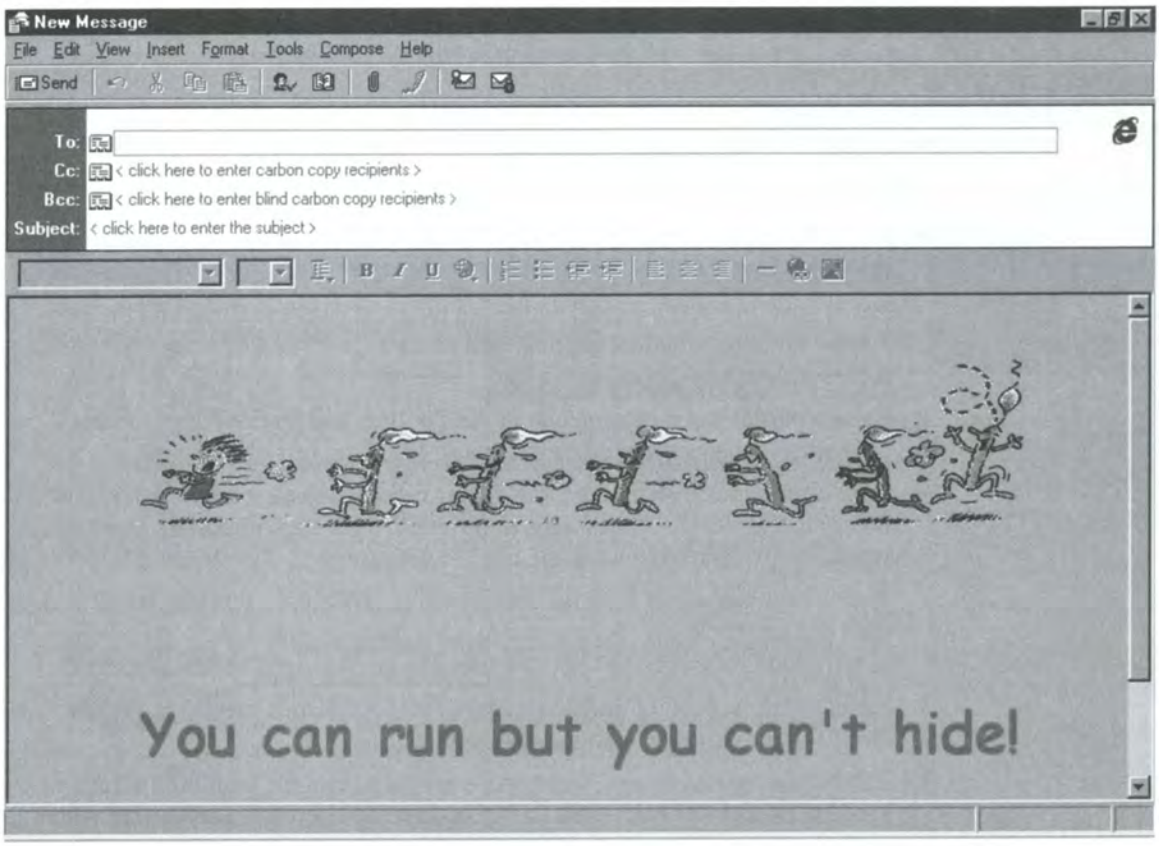
Internet Explorer allows users to subscribe to Web sites in order to be notified of updated information. Users have the option of receiving notification of updated Web site subscriptions through e-mail. Users can also choose to have the updated content automatically delivered by e-mail. This way, when there is new information on a Web site to which you've subscribed, the actual Web page is delivered right to your Inbox. Because Outlook Express supports HTML, it simply displays the page instead of including it as an attachment. (For more information, see Part 4, "Information Delivery: Webcasting.")

Smart Reply

Outlook Express automatically replies to messages in the same format in which they were sent. For example, if you receive HTML mail, Outlook Express responds with HTML. If you receive the message in plain text, Outlook Express does not send HTML mail when replying to that message, and it remembers not to do so for future messages to that recipient.

Stationery

Because Outlook Express supports HTML as a format for its mail messages, you can use HTML stationery and professionally designed stationery from Greetings Workshop to give mail a personal touch.



Multiple E-Mail and News Server Accounts

With Outlook Express, you can access and manage multiple e-mail and news accounts from a single client. You can also send and receive mail from numerous e-mail accounts. Outlook Express keeps track of which account an incoming message is using, so that when you reply to it, you can simply click **Send**, and Outlook Express sends your response back through the appropriate account.

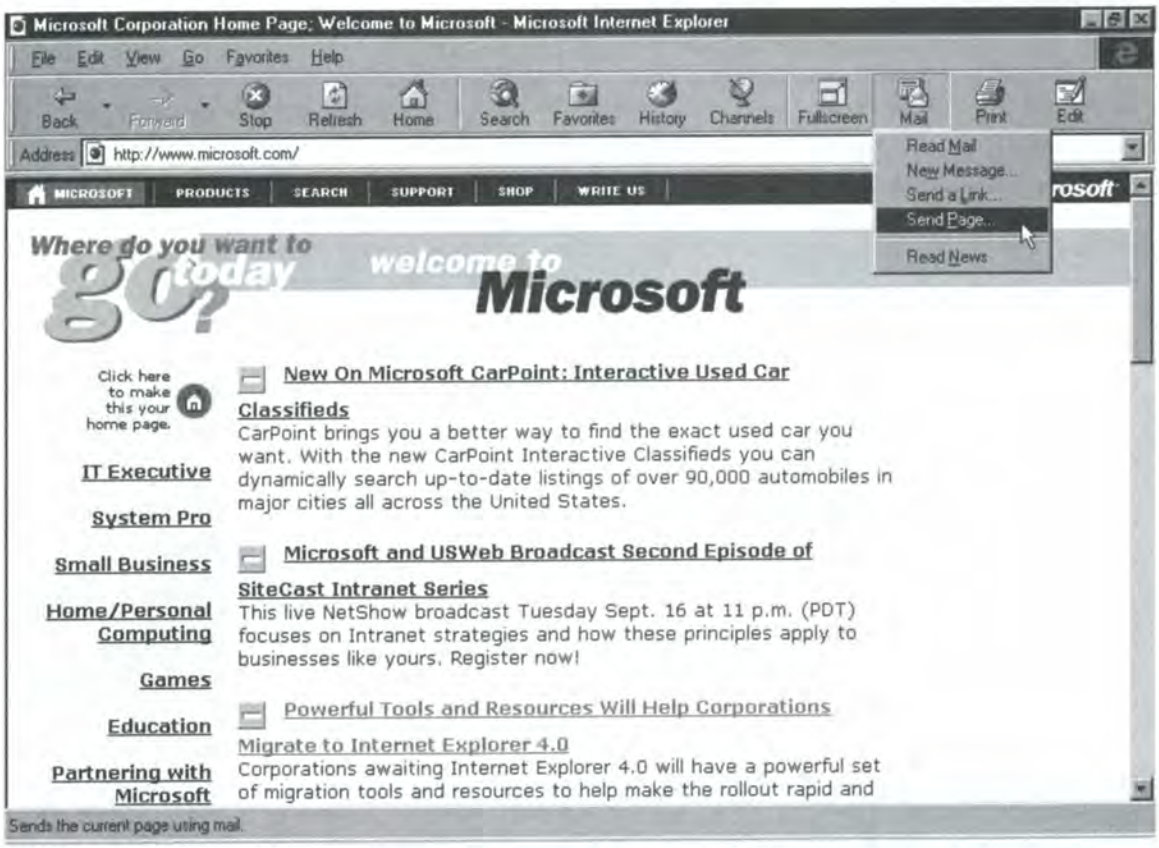
Productivity Features

Numerous features have been added to make Outlook Express easier to use. Now you can do the following:

- Create multiple, hierarchical folders and drag them and their associated messages as needed.
- Easily save important e-mail addresses by using Address Book and the new Auto-Add feature, which automatically adds replied-to addresses to the Address Book.
- Enter a partial name of someone in the To: line, and Outlook Express compares it against your address book to fill in the rest of the name.
- Save e-mail messages in the Draft folder before they are sent, which makes it easy to keep track of messages in progress. This way, important messages won't get overlooked among the mail in your Inbox.
- Execute the **Send** and **Receive** commands separately, so you can spend your time online efficiently. For example, if you're on a slow link, you can choose to only send messages and not download large messages with attachments.
- Receive notification of unread messages and unfinished messages in your Draft folder when you start the application.
- Take advantage of several word processor-like features, including a built-in spelling checker (if Microsoft Office or another Microsoft application with spell checking is installed) and up to 150 levels of Undo.

Integration with the Internet Explorer Suite

Because Outlook Express is tightly integrated with the rest of the Internet Explorer applications, you can easily use them together and switch between applications seamlessly. Not only does Outlook Express share common menus and toolbars with the other suite components, it also lets you send a whole Web page to someone from a single click in the browser. The message embeds the entire page, not just a link to a Web site.



Understanding Outlook Express

Outlook Express is based on a very small, flat file system. When Outlook Express is installed, a .cab file named Mailnews.cab is copied to the Internet Explorer Setup directory. The .cab file is then extracted to give Internet Explorer the Outlook Express files and setup information.

Outlook Express Architecture

The main files used for Outlook Express are:

Wab32.dll	Windows Address Book (WAB) search engine
Msimnui.dll	Dialog boxes and other mail and news functions
Wab.exe	Windows Address Book; launches Wab32.dll
Wldap32.dll	LDAP support file for WAB and LDAP searches
Wabmig.exe	WAB import user interface
Msimnimp.dll	Internet Mail and News setting importer/exporter
Msimn.exe	Outlook Express main program; loads Msimnui.dll
Wabimp.dll	WAB import back end
Wabfind.dll	WAB search user interface
Inetcomm.dll	Parses MIME information
Msoeacct.dll	Account manager .dll file
Msoemapi.dll	Provides Simple MAPI support in Outlook Express
Msoert.dll	Run-time utility functions

To render HTML code in both the Preview pane and the Message window, Msimnui.dll uses the services of Mshtml.dll (the Internet Explorer WebBrowser Control).

Outlook Express Flat File Directory

When Outlook Express downloads mail messages and news articles, it stores them on the local hard disk. The path for Windows 95 varies slightly depending on whether the User Profiles feature is turned on or not. (For more information about user profiles, see Chapter 5.)

Without User Profiles:

Windows\Application Data\Microsoft\Outlook Express

With User Profiles (and on Windows NT):

*%windir%***Profiles***username***Application Data\Microsoft\Outlook Express**

Within this directory are two folders, one for mail and one for news. The news folder has a subdirectory for each news account that is in use.

Outlook Express places data in two places in the registry. User accounts and LDAP services are stored in:

HKEY_CURRENT_USER\Software\Microsoft\Internet Account Manager\Accounts

Setting and configuration options are stored in:

HKEY_CURRENT_USER\Software\Microsoft\Outlook Express

Outlook Mail Data Directory

The Mail data directory has no subdirectories, only a collection of files used to manage mail accounts. The three types of files in the Mail directory are:

- **.mbx files** These files contain the text of all messages stored on the system.
- **.idx files** These files function as indexes for .mbx files. The .idx and .mbx files are always paired.
- **.nch files** These files are used to retain the folder hierarchy.

Each .idx file has text pointers to each message that is stored in its companion .mbx file. The .mbx files store the content of each message, including the addressing information. For HTML messages, the raw HTML source code is visible in the .mbx file, but not on the HTML page.

Outlook News Data Directory

Inside the News data directory, a folder is created for each news account. Three types of files are used to manage newsgroups:

- **.nch files** These files store a cache of articles downloaded from the server.
- **.dat files** These files store two types of newsgroup lists. The names of all current newsgroups are stored in Grplist.dat, while the names of all subscribed newsgroups are stored in Sublist.dat.

Windows Address Book

The Windows Address Book directory service is a local resource provided by Wab.exe. The Wab.exe program creates and searches a local file *username*.wab found in the Application Data directory. The full path varies slightly, depending on whether the User Profiles feature is enabled or not. (For more information about user profiles, see Chapter 5.)

With User Profiles

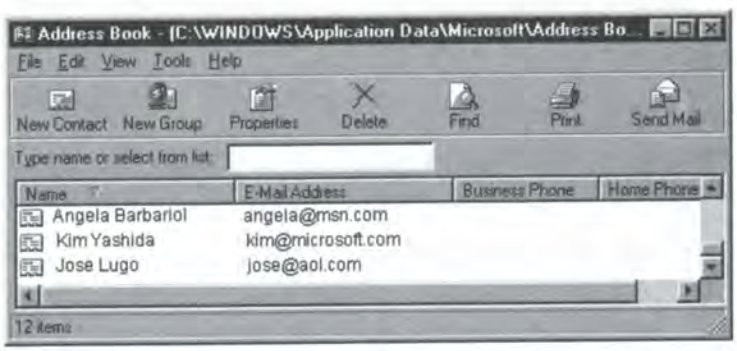
***%windir%*\Profiles*username*\Application Data\Address Book*username*.wab**

Without User Profiles

***%windir%*\Application Data\Address Book*username*.wab**

This file contains a list of people with whom the user has frequent communication. Data from previous address books is imported to the .wab file. The default name of the .wab file is the currently logged-on user's name with a .wab extension. This allows the Windows Address Book to easily support multiple users on the same machine and adds a level of privacy to the content.

When Windows Address Book is highlighted in the **Find People** dialog box, the current user's .wab file is searched for the address. The local .wab file is maintained by Wab.exe as well. The screen shot below shows the standard user interface.



Configuring and Customizing Outlook Express

As the administrator, you can control a wide range of options during initial setup by using the Outlook Express interface and the IEAK Configuration Wizard, and after setup by using the IEAK Profile Manager. These options include:

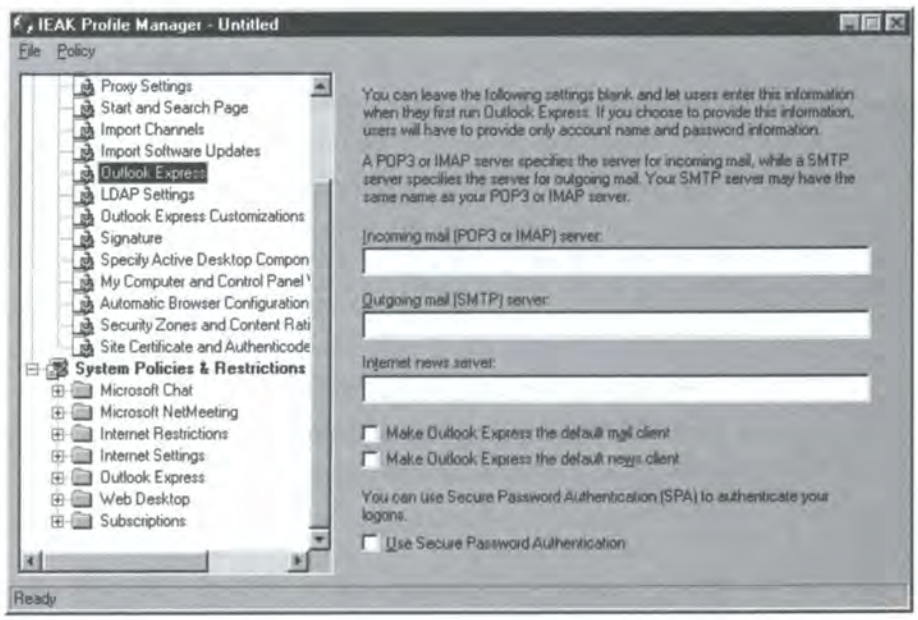
- Specifying the type of incoming mail server (either IMAP4 or POP3).
- Specifying the name of the incoming mail server.
- Specifying the name of the outgoing mail server (SMTP).
- Specifying the name of the Internet news server (NNTP).
- Making Outlook Express the default mail and/or news client.
- Using Secure Password Authentication (SPA) to authenticate users when they log on.
- Importing messages and address book information.
- Specifying an LDAP server to provide Internet directory services for users.
- Setting rules with the InBox Assistant.
- Customizing the Outlook Express InfoPane.
- Creating a welcome message for new users.
- Appending a standard signature to all messages and news postings written in Outlook Express.
- Customizing the appearance of the Outlook Express window.
- Creating custom stationery.
- Configuring Internet security zones for mail and news.

For more information about deploying and maintaining custom packages using the IEAK Configuration Wizard and IEAK Profile Manager, see Part 9, "Deployment and Maintenance."

Specifying Mail and News Servers

When Outlook Express is launched for the first time, the Internet Connection Wizard automatically guides the user through the process of setting up Outlook Express for use with either an ISP or a POP3-based LAN or an IMAP4-based LAN, and an Internet news server. The user also has the option of making Outlook Express the default mail and/or news reader.

Administrators can pre-configure these settings for users by using the **Specify Internet Mail Servers, Domain, and News Server** page of the IEAK Configuration Wizard or the **Outlook Express** page of the IEAK Profile Manager. Pre-configuring these pages is optional, and they can be left blank to let users enter the information themselves when they run Outlook Express for the first time.



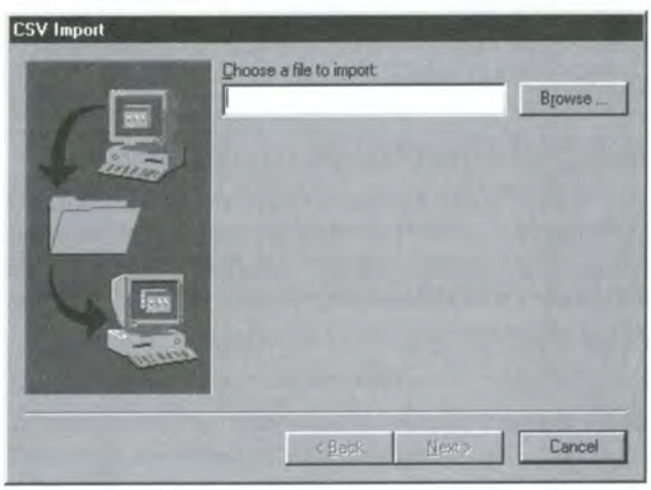
Importing Messages and Address Book Information

Once Outlook Express is set up, the first time users run Outlook Express and open the Inbox, Outlook Express automatically checks to see if they have another messaging application. If so, it immediately offers to import their address book and existing messages. Once that process has taken place, it is up to users to import their existing and future e-mail messages and address book information from the previous e-mail client. They can do so by selecting **Import** on the **File** menu. When messages are imported, the contents of the messages are saved into .mbx files in the Mail data directory. Address information gets copied into the user's Windows Address Book (.wab) file located in the Address Book directory.

▶ **To import address book information**

1. On the **File** menu, click **Import**, and then click **Address Book**. The **Address Book Import Tool** dialog box appears.

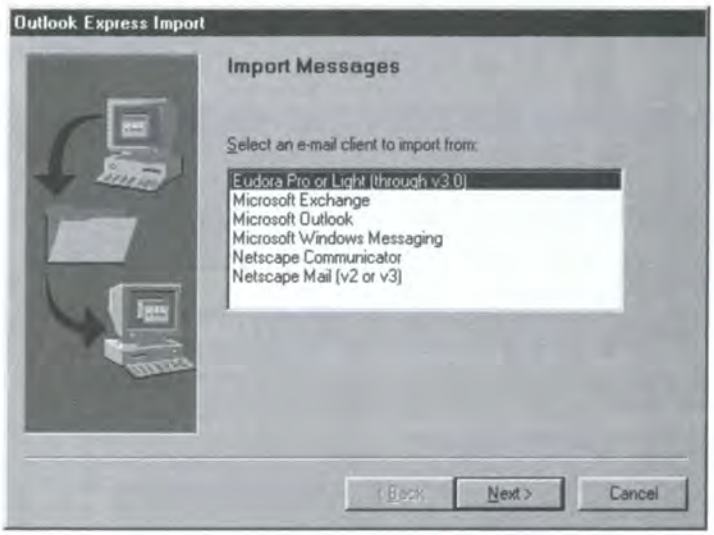
2. Select the format of the address information you want to import, and click **Import**. If you select Microsoft Exchange or Microsoft Internet Mail, you are prompted for a user profile. For other formats, Outlook Express either attempts to auto-detect the existing address book or prompts you for a path to the file's location.



3. The address information is imported to your Windows Address Book file.

▶ **To import messages**

1. On the **File** menu, click **Import**, and then click **Messages**. The **Outlook Express Import** dialog box appears.



2. Select the e-mail client from which you want to import messages. If you select one of the Microsoft clients, you are prompted for a user profile. If you select any other client, you are prompted to enter the path to the current location of the message files.
3. Select a profile from the menu, or enter the path to the message file folder, and then click **OK**.
4. Select the message folders you want to import, and click **Finish** to import the messages to Outlook Express.

Configuring the LDAP Directories

The Address Book in Outlook Express supports LDAP for accessing Internet white-page services. These services give users a way to search for e-mail addresses and names. Profiles for each of these Internet directory search services are installed with Outlook Express:

- Four11
- InfoSpace
- InfoSpace Business
- Bigfoot
- Switchboard
- WhoWhere?
- VeriSign

Directory services maintain a search database of people around the world and information on how to contact them. Normally, if you know simple information, such as the name of a person, and some other piece of data, such as the city they live in, you can successfully search for people.

The LDAP search interface is controlled by Wabfind.dll. This .dll file provides a user interface for searching directory services on the Internet and in the local Windows Address Bookfound in the Windows directory.

You can specify one or more LDAP directories against which to check names and e-mail addresses. This is especially useful if your organization has its own LDAP server with a company-wide address book. Outlook Express checks address information against the user's personal address book first, and then against any LDAP servers that have been selected. If more than one LDAP server is specified, you can designate the order in which they must be checked.

▶ **To specify an LDAP directory for checking addresses**

1. On the **Tools** menu, click **Accounts**.
2. In the **Internet Accounts** dialog box, select the directory service you want to use for checking addresses, and click **Properties**.
3. Select **Check names against this server when sending mail**.
4. Click **OK**.

▶ **To set the order in which LDAP directory services are checked**

1. On the **Tools** menu, click **Accounts**.
2. In the **Internet Accounts** dialog box, click the **Set Order** button. The **Directory Services Order** dialog box opens.
3. Set the order of the listed directory services as desired, and then click **OK**.

Maintaining Directory Service Settings

Because the available directory services are constantly changing, being able to add and remove listings as needed is important. You can maintain your settings using the **Directory Service** tab.

▶ **To update directory services**

1. On the **Tools** menu, click **Accounts**.
2. Click the **Directory Service** tab. From here, you can:
   - Add a new directory service.
   - Remove an existing directory service.
   - Update the properties of an existing directory service.
   - Set the order of servers for checking names. (This option can be used only if you select the **Check Names against this Server** option in the **Directory Service Properties** dialog box.)
3. When you are finished making changes, click **Close**.

Tip You can access the **Directory Service** tab of the **Internet Accounts** dialog box directly from the **Find People** dialog box by right-clicking the **Look In** drop-down box and selecting **Directory Service**.

The configuration data for directory services is stored in the registry at:

HKEY_USERS\.Default\Software\Microsoft\Internet\Account Manager\Accounts

You can specify an additional LDAP service for your users on the **Specify LDAP Server Settings** page of the IEAK Configuration Wizard or the **LDAP Settings** page of the IEAK Profile Manager. If your organization has an LDAP server, it is recommended that you set it up here so it will be available to your users immediately after they are set up.

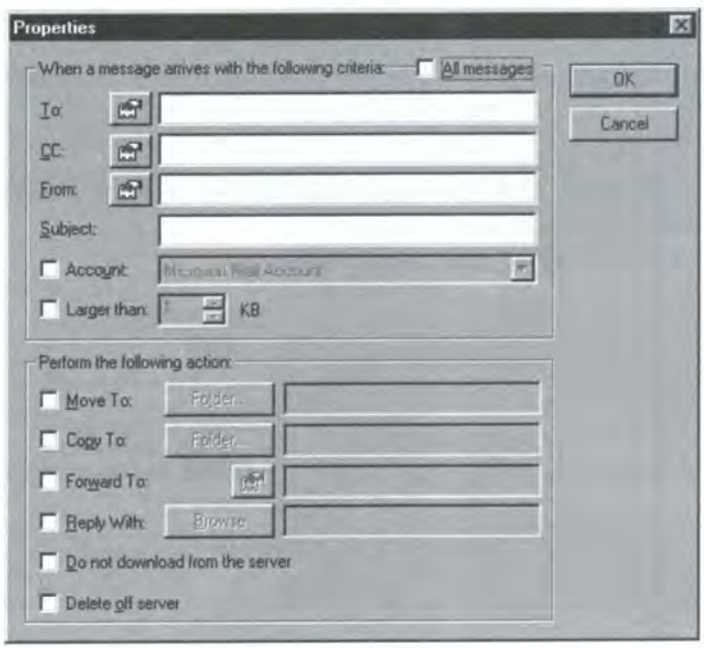
Note Searching a directory service on the Internet requires the use of Windows Sockets. If your users connect to the Internet through a proxy server, Find People can be run only if a Web proxy client is installed.

Using the InBox Assistant

The Inbox Assistant helps users to manage the flow of incoming messages in several ways. Rules can be specified to move, copy, forward, or delete messages, or to send an auto-reply, depending on specified criteria. For example, a user may choose to move all mail received from a particular user to a different folder. The Inbox Assistant makes short work of managing multiple e-mail accounts by providing the ability to sort messages by account. A folder can be created for each separate account, preventing mix-ups. Rules can also be applied to entire folders at once, which is useful for managing messages that have already been received.

▶ **To use the Inbox Assistant**

1. On the **Tools** menu, click **Inbox Assistant**. The **Inbox Assistant** dialog box appears. It is blank until rules are added.
2. To add a new rule, click **Add** to bring up the **Properties** dialog box for the new rule.



3. Create the rule by selecting message criteria and the desired actions to take.
4. Click **OK**.

▶ **To apply an inbox rule to a folder**

1. On the **Tools** menu, click **Inbox Assistant**. The **Inbox Assistant** dialog box appears.
2. Select the rule you want to apply to the folder.

3. Click the **Apply to** button. The **Select Folder** dialog box appears.
4. Select the folder to which you want to apply the rule, and click **OK**.

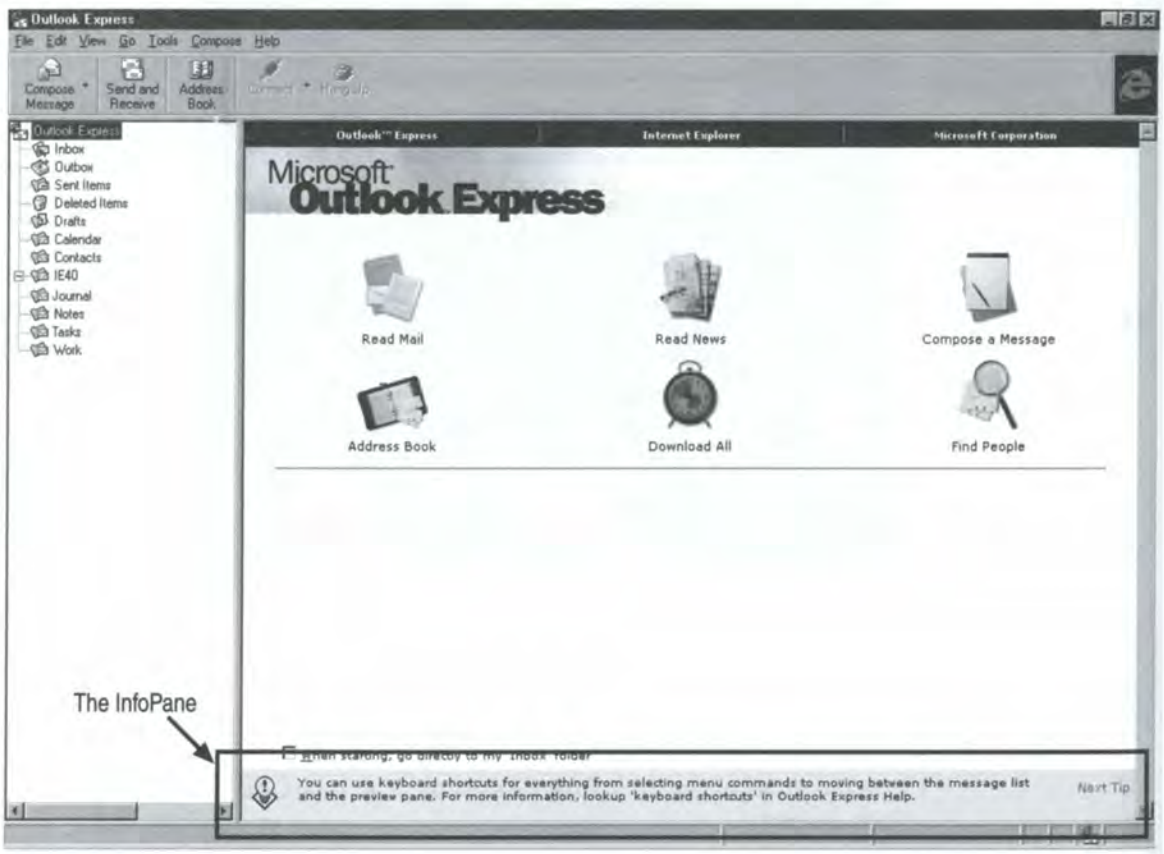
Data for Inbox Assistant rules is stored in the Registry at:

HKEY_CURRENT_USER\Software\Microsoft\Outlook Express\Mail\Inbox Rules

Each rule has its own entry under Inbox Rules.

Customizing the InfoPane

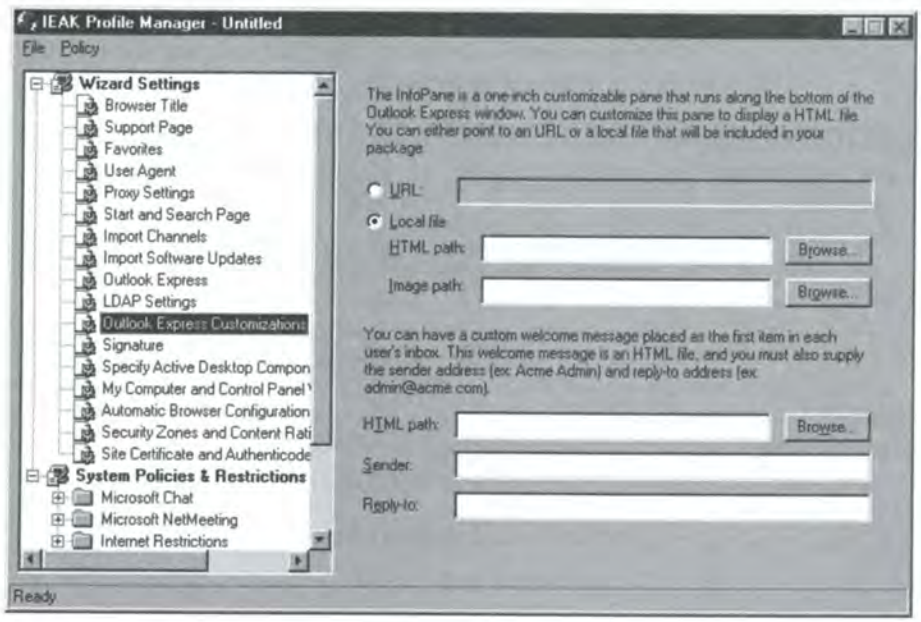
The InfoPane is a one-inch pane that appears at the bottom of the Outlook Express window when you start the program. It is available only if you have customized Outlook Express with the IEAK. You can customize the InfoPane with an HTML file that is either a local file or an Internet address (URL). The InfoPane is a good place to include support numbers, links, FAQs, and other useful resources.



▶ **To customize the InfoPane**

1. Create the HTML file that will go into the InfoPane. Keep in mind that you have a limited amount of space.
2. Save the file.
   - Saving the file to a Web server lets you point to the file's URL. This makes it easy to update the file as needed.
   - Saving the file to a local disk includes the file in your installation package. The files will be installed on users' machines when they install Internet Explorer.

The InfoPane customization options are found on the **Outlook Express Customization** page of both the IEAK Configuration Wizard and the IEAK Profile Manager. You can select an alternate InfoPane page using either administration tool.



After configuring these options, the InfoPane path can be found in the registry at:

HKEY_CURRENT_USER\Programs\Microsoft\Outlook Express\bodybar

Creating a Welcome Message

You can create a welcome e-mail message that new users will see when they launch Outlook Express for the first time. This message can provide information that helps make it easier for new users to get started and directs them to resources they can use. The welcome message must be an HTML file. After the welcome message is created, you must specify it on the **Outlook Express Customization** page of either the IEAK Configuration Wizard or the IEAK Profile Manager.

Appending a Standard Signature

You can include a default signature or disclaimer that will appear in Outlook Express newsgroup or e-mail messages. For example, a disclaimer may be used to show that messages submitted by employees over the Internet do not represent official company policies. The maximum size of the signature is 1K. You can choose to append signatures to newsgroup messages only, to e-mail messages only, or to both. This is done on the **Include a Signature** page of the IEAK Configuration Wizard or the **Signature** page of the IEAK Profile Manager.

Customizing the Appearance of Outlook Express

Consistent with the rest of Internet Explorer, you can customize the toolbars in Outlook Express to display the most frequently used commands. Users can choose to position the toolbar on the top, right, or left side of the screen. These options can be selected by right-clicking the toolbar.

Administrators can further configure the view under **View Customization** on the **System Policies and Restrictions** page of both the IEAK Configuration Wizard and the IEAK Profile Manager. You can:

- Turn on the Outlook bar (it's off by default).
- Turn off the Folder List (tree view of folders).
- Turn on the Folder bar (horizontal line that displays folder names).
- Turn off the Tip of the Day.

Creating Custom Stationery

Stationery templates are files that allow users to personalize the look of the mail they send. Using HTML, stationery files define the background graphic and font colors and styles that appear in the message. Stationery can contain images, script, and other objects, which may be included with the message or referenced with a URL according to your needs. For example, a background graphic of 1K can easily be included with the message, while it would likely be preferable to use a URL to reference a larger image of 100K.

Note Style sheets, objects, and scripting that are referenced by URLs can be included only in stationery; they cannot be entered into the body text of Outlook Express messages.

Outlook Express comes with several user-modifiable default stationery templates, and you can create your own custom templates. Outlook Express reads stationery files and their associated graphics from:

C:\Program Files\Common Files\Microsoft Shared\Stationery

You can save your custom stationery templates and accompanying graphics in this folder, or you can create a subfolder in which to save them. Be sure to save any associated graphics in the same directory as the HTML files to which they belong. Because the path to Outlook Express stationery files cannot be customized, you must make any custom stationery files available for users to download to their computers.

When users create a message using stationery, Outlook Express loads the stationery template they select and inserts the message body directly into the HTML code. When the message is sent, the accompanying graphics are included by MHTML. If a message using stationery is sent to a mail client that is incapable of interpreting HTML, the message body will still be displayed properly, and other objects will be available as an attachment in a standard e-mail message.

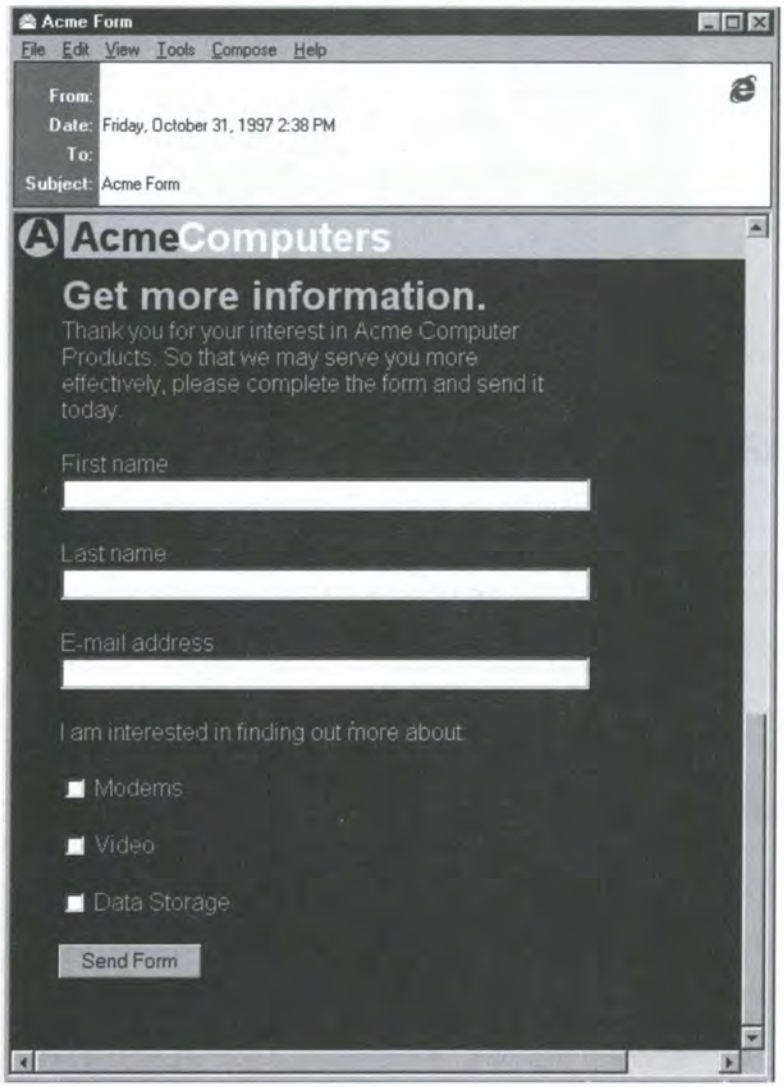
A typical stationery file looks like this:

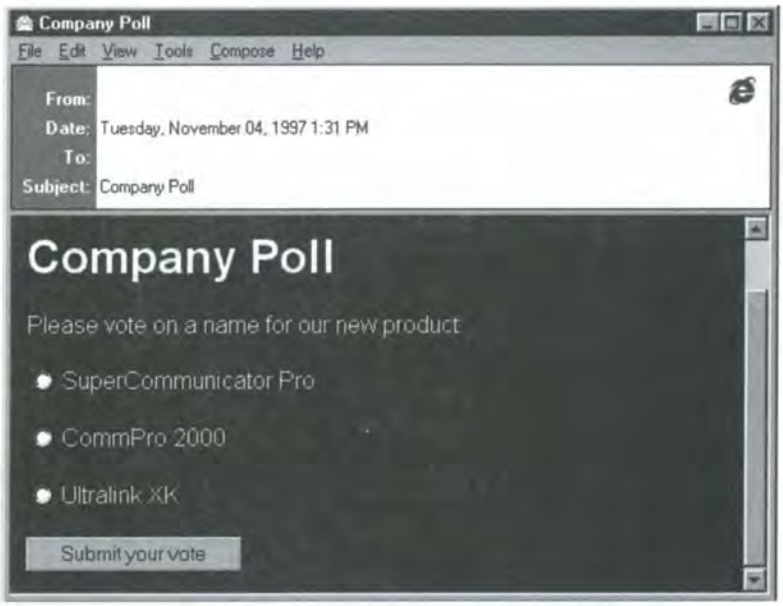
```
<HTML>
<HEAD>
<STYLE>
<!--
BODY {
margin-left: 4em;
color: "#427D64";
font-size: 12pt;
font-weight: regular;
font-family: "Arial";
}
-->
</STYLE>
</HEAD>
<BODY background="Ivy.gif">
 
</BODY>
```

The top of the file contains CSS styles, and the background graphic is defined in the <BODY> tag. Other graphics can be included, as well as scripts, objects, and text. Items that are referenced can be stored in the same folder as the stationery HTML file or in a different directory, as long as the stationery file references the correct path to the file. Stationery templates are created in exactly the same way as ordinary Web pages. They can be as complex or simple as your needs dictate. For more information about building Web pages, visit the Site Builder Workshop at **http://www.microsoft.com/workshop/**.

Tip When writing a message in HTML format, you can add formatting tags from within the New Message window by highlighting the text you want to format, clicking the **Style Tag** button in the toolbar, and choosing the desired style tag.

The integration of HTML with e-mail provides a range of possibilities that were not possible with traditional e-mail clients. Standard HTML, Dynamic HTML, input forms, scripting, and objects can all be incorporated into Outlook Express stationery. Now you can take ordinary messages and make them visually engaging as well as highly functional. The following screens show two examples of stationery that has been designed to gather input from users. Forms such as these could link to Active Server Pages or CGI scripts stored on your Web server.





Understanding Security in Outlook Express

Outlook Express gives users the ability to use encryption and digital signatures for increased security when sending and receiving e-mail. Security is provided through the S/MIME version 2 specification and supports the following encryption algorithms: RC2 (40-bit and 128-bit), DES (56-bit), and 3DES (178-bit). The RC2 40-bit encryption algorithm is the only algorithm available in non-U.S./Canadian versions of Outlook Express. Outlook Express can decrypt 3DES (112-bit) and RC2 (64-bit) encrypted mail but cannot send messages using these algorithms.

Outlook Express uses SHA-1 as the hashing algorithm when signing messages. The bit length of your private key varies, depending on the Certificate Authority (CA) from which you obtain it. A CA that uses the Microsoft Enrollment Wizard generates private keys that are at least 512 bits in length.

By using digital IDs with Outlook Express, you can prove your identity in electronic transactions, a practice similar to showing your driver's license when you cash a check. You can also use your digital ID to encrypt messages to keep them private. A digital ID is composed of a "public key," a "private key," and a "digital signature." When you send your digital ID to others, you are actually giving them your public key so that they can send you encrypted mail, which only you can decrypt and read with your private key.

The private keys are stored on your computer and are only as secure as your computer. Private keys installed using Microsoft cryptographic system components are not transmitted to the CA that issues the digital ID; the keys are not stored in escrow with any government agency.

For more information about digital signatures, digital certificates, and security in Internet Explorer, see Part 7, "Security and Privacy."

Using Security Features in Outlook Express

The security features in Outlook Express allow you to associate your digital ID with a mail account, send and receive digitally signed mail, retrieve digital IDs for other people, send and receive encrypted mail, and configure security zone settings for Outlook Express.

Obtaining a Digital ID

Before encrypted mail can be sent or received, a digital ID must be obtained from an independent Certificate Authority. CAs are third-party companies that have a process for checking and verifying your identity. Once a digital ID has been obtained, it must be associated with the e-mail account for which you want to use it.

▶ **To obtain a digital ID for yourself**

- See the Microsoft Internet Explorer Digital ID Web site at **http://www.microsoft.com/ie/ie40/oe/certpage.htm**.

▶ **To associate a digital ID with a mail account**

1. In Outlook Express, on the **Tools** menu, click **Accounts**.
2. Select the mail account from which you want to send secure mail, and then click **Properties**.
3. On the **Security** tab, check **Use a digital ID when sending secure messages from**, and then click the **Digital ID** button.
4. Select the ID you want to use with this account.

▶ **To send your digital ID to others**

- Simply send them digitally signed e-mail (see below), and Outlook Express automatically includes your digital ID.

Sending and Receiving Signed Mail

Signed e-mail allows an e-mail recipient to verify your identity.

▶ **To send signed mail**

- On the **Tools** menu in your Message window, click **Digitally Sign** (or use the button on the Message toolbar).

  To send signed mail, you must have a digital ID of your own (see above).

Signed e-mail from others allows you to verify the authenticity of a message—that the message is from the supposed sender and the message has not been tampered with during transit.

▶ **To receive signed mail**

- Signed e-mail messages are designated with special, signed e-mail icons. You receive and read them as you would any other e-mail message. Any problems (described by Outlook Express security warnings) with signed e-mail that you receive could indicate that the message has been tampered with or was not from the supposed sender.

Sending and Receiving Encrypted Mail

Encrypting an e-mail message prevents other people from reading it when it is in transit. To encrypt an e-mail message, you need the digital ID of the person to whom you are sending the e-mail. The digital ID must be part of the person's entry in the Address Book. Outlook Express is the first e-mail program to allow you to retrieve digital IDs from directory services.

▶ **To retrieve a digital ID from someone else**

1. On the **Edit** menu, click **Find People**.
2. Select a directory service that has digital IDs (such as VeriSign), enter the recipient's name or e-mail address in the appropriate search field, and click **Find**.
3. Select a listing from the Results pane, and click **Add to Address Book**.

Tip Another way to get someone's digital ID is to have that person send you signed mail. To add the digital ID from a piece of signed mail to your Address Book, on the **File** menu, click **Properties**. Click the **Security** tab, and click the **Add Digital ID to Address Book** button.

Note When you add a digital ID to your Address Book, it has a trust status associated with it that indicates whether you trust the individual, group, or corporation to whom the digital ID was issued. If a digital ID owner warns you that he or she suspects that the digital ID's private key has been compromised, you may want to change the trust status to "Explicitly Distrust." For more information, look up "Trust Status of a Digital ID" in the Outlook Express Help Index.

▶ **To send encrypted mail**

1. Create your mail message as you would an ordinary piece of e-mail.
2. On the **Tools** menu in your Message window, click **Encrypt** (or click the button on the Message toolbar).

   When sending encrypted mail, you must have the recipient's digital ID in your address book (see above) so that they can decrypt the message when they receive it. Your e-mail address for replies must be the same as the account from which you're sending the digitally signed mail. Otherwise, message recipients won't be able to use the ID to reply with encrypted mail. (The e-mail address to which they reply won't match the address on the digital ID.)

When you receive an encrypted e-mail message, you can be reasonably confident that the message has not been read by anyone else.

▶ **To receive encrypted mail**

- Outlook Express automatically decrypts e-mail messages, provided that you have the correct digital ID installed on your computer.

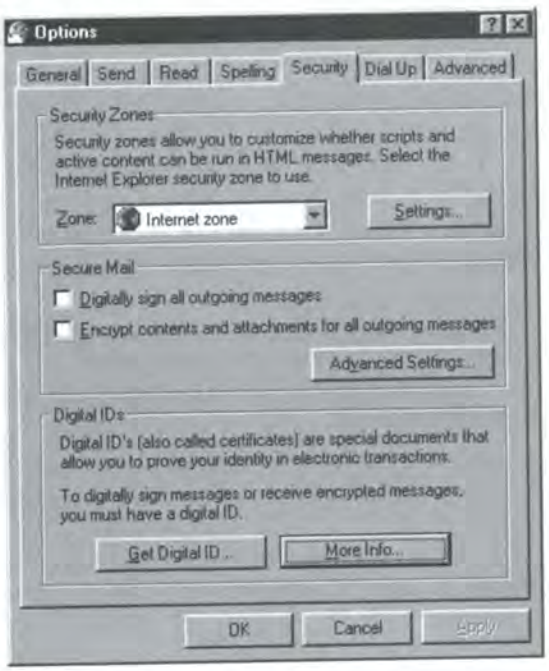
For others to be able to send you encrypted mail, they need your digital ID.

Changing Security Zones Settings for Outlook Express

Outlook Express uses Internet Explorer's security zones technology to ensure that active Web content such as Java applets and ActiveX™ controls do not run inappropriately. Users can designate all incoming mail as part of the Internet zone (under which users are prompted before any active content runs) or the Restricted Sites zone (under which no active content can be run at all). In addition, for each security zone, you can choose a High, Medium, Low, or Custom security level setting. For more information, see Chapter 27, "Security Zones."

▶ **To change your security zone settings for Outlook Express**

1. On the **Tools** menu, click **Options**, and then click the **Security** tab.
2. You can select which security zone to use for mail and news. Click **Advanced** to change the security settings for zones. Changing the settings for the Internet zone or Restricted Sites zone also changes this setting for Internet Explorer, and vice versa.



Administrators can set and lock down security options for users by using the **System Policies and Restrictions** and **Security Zones and Content Ratings Customization** pages in both the IEAK Configuration Wizard and the IEAK Profile Manager. For more information, see Part 7, "Security and Privacy."

CHAPTER 19

NetShow

In this chapter

NetShow Overview 281
- Streaming Audio and Video 282
- Rich, Synchronized Multimedia 283
- Compatibility with Streaming Technologies 283
- Live Broadcast of Audio and Video 283
- Content Scheduling 284
- Easy User Access to Broadcasts 284
- Multicast File Transfer 284
- Multiple Network Transport Protocols 285
- Server Integration 285
- Scalability 286
- Distributed Delivery of Content Streams 286
- Web-Based Administration 286
- Tools for Monitoring and Logging Events 287
- Firewall Support 287

Components of NetShow 288
- NetShow Player 288
- NetShow Server 289
- NetShow Administrator 289
- NetShow Authoring and Production Resources 289
- NetShow Software Development Kits 290
- NetShow Theater Server 291

Webcasting with NetShow 291
- True Webcasting Application Scenarios 291
  - Training 291
  - Corporate Communications 292

Entertainment and Information 292
Advertising and Retailing 292
Understanding NetShow Technology 293
Understanding Streaming Media 293
Understanding Network Bandwidth 293
Understanding Content Compression 293
Understanding Codec Independence 294
Understanding the Active Streaming Format 295
Understanding Error Correction and Masking 296
Understanding Unicasting and Multicasting 296
Understanding Unicasting and Multicasting with NetShow Server 300
NetShow Announcements 300
HTTP Versus MMS 301
NetShow Channels 302
NetShow Streams and Programs 303
NetShow Multicast and Unicast Scenarios 304
Multicasting Live ASF Streams 304
Multicasting Stored ASF Streams 306
Unicasting a Live ASF Stream 307
Unicasting a Stored ASF Stream 307
Unicasting Stored ASF Streams Using a Channel 308
Setting Up ASF Stream Distribution Servers 309
Multicasting File Transfers 310
Using HTTP Streaming 311
Developing Content for NetShow 313
A NetShow Scenario 313
Developing Live NetShow Content 314
Developing On-Demand NetShow Content 315
Obtaining Content 315
Selecting a Bandwidth 317
Building the ASF File 318
Selecting a Tool 318
Adjusting Your Content for the Bandwidth Using Codecs 320
Streaming the ASF File 322
Using the NetShow Player 323
Using the Stand-Alone Player 323
Launching the Player from a Web Page 324
Embedding NetShow Player in a Web Page 325

NetShow Overview

Microsoft NetShow 2.1 delivers a complete, high-performance system for broadcasting live and on-demand audio, video, and multimedia. NetShow is a platform for streaming multimedia over networks that range from low-bandwidth, dial-up Internet connections to high-bandwidth, switched local area networks (LANs). Using NetShow, organizations can offer new streaming content for applications such as training, corporate communications, entertainment, and advertising to users all over the world. NetShow is a powerful broadcast system that is easy to set up and operate. It empowers organizations to offer rich, high-quality interactive content over today's networks.

Only a few years ago, the business world was full of textual documents, and adding graphics to them was not easy. Now, the race is on to offer Web sites that attract and retain visitors through the latest in graphics and animation. Audio-enabled and video-enabled Web sites and applications are the next wave. The integration of audio and video into applications such as online training, corporate communications, customer and sales support, news and entertainment services, and product promotions will provide individuals and organizations with new and exciting ways to communicate.

Microsoft NetShow provides a complete platform for integrating audio and video into online applications, bringing the vibrant power of networked multimedia to the Internet and corporate intranets. With its leading-edge live and on-demand media-streaming technology, Microsoft NetShow allows users to receive audio and video broadcasts from their personal computers. It uses client/server architecture and sophisticated compression and buffering techniques to deliver live and on-demand audio, video, and illustrated audio (synchronized sound and still images) to users of the NetShow Player. The NetShow Player continuously decompresses and plays the content in real time. Users can listen and watch live audio and video programs or navigate on-demand audio and video content.

Building upon the open, standards-based platform established with NetShow networked multimedia software version 1.0, NetShow 2.1 provides an easy, powerful way to stream multimedia content across intranets and the Internet. NetShow allows developers and Web professionals to add production-scale audio and video broadcasts to any Web application or site.

NetShow reduces the total cost of streaming multimedia, because of its tight integration with Microsoft Internet Information Server (IIS), the Microsoft Windows NT Server network operating system, and the Internet Explorer browser. Standard multimedia authoring and HTML programming tools can be used to create and host content.

Some of the important features and functionality of NetShow include:

- Streaming audio and video
- Rich, synchronized multimedia
- Compatibility with streaming technologies
- Live broadcast of audio and video
- Content scheduling
- Easy user access to broadcasts
- Multicast file transfer
- Multiple network transport protocols
- Server integration
- Scalability
- Distributed delivery of content streams
- Web-based administration
- Tools for monitoring and logging events
- Firewall support

Streaming Audio and Video

NetShow streams audio and video to users on demand, on an intranet or on the Internet, without waiting for content to download and without consuming the user's disk space. Streaming delivers content to the client as a continuous flow of data with little waiting time before playback begins, and data is never actually stored on the user's computer. NetShow provides an array of the best compression/decompression algorithms, or *codecs*, making NetShow the most comprehensive platform available today for delivering high-quality streaming multimedia content over a wide range of networks.

NetShow provides broadcast-quality audio and includes an audio codec from Fraunhofer Institut Integrierte Schaltungen that delivers mono and stereo audio at 28.8 kilobits per second (Kbps) and scales to near-CD-quality audio over ISDN, ADSL, cable modems, and LANs. In addition, NetShow includes several Voxware codecs, which are excellent for voice-only audio tracks and are able to compress audio smaller than any of the other codecs installed with NetShow. Plus, NetShow provides a premier implementation of the forthcoming, leading-edge, MPEG-4 video standard. Microsoft, working closely with the MPEG-4 standards committee, is implementing one of the first video codecs based on the emerging MPEG-4 video standard. At 28.8 Kbps, this codec produces market-leading video quality, that is becoming ever more impressive as network bandwidth increases. NetShow also includes Duck TrueMotion and Vivo video codecs.

For more information about streaming media in NetShow, see "Understanding Streaming Media," "Understanding Content Compression," and "Understanding Codec Independence" later in this chapter.

Rich, Synchronized Multimedia

Unlike most audio or video streaming products, NetShow enables content providers to generate sophisticated productions in which graphics, slides, photographs, and URLs can be synchronized with the audio and video stream. You can create multimedia programs that use live and on-demand streaming of audio, illustrated audio, and video synchronized with pictures (.jpeg, .bmp, and .gif files), as well as script (JScript™ and ActiveX script), URLs, and text. The result is a compelling mixed-media experience and the ability to deliver multimedia applications over low-bandwidth networks such as the Internet.

Compatibility with Streaming Technologies

NetShow supports the industry-leading Active Streaming Format (ASF) and the RealAudio and RealVideo formats by RealNetworks. By adding the ability to create, serve, and play live and on-demand ASF and RealAudio/RealVideo content, NetShow can now access more than 95 percent of the streaming media content on the Web, and content providers can use NetShow to provide streaming media for the majority of users on the Web. Thus, NetShow, coupled with industry efforts to establish ASF as the common streaming format, will help ensure the continued growth of the streaming media market. For more information about ASF, see "Understanding the Active Streaming Format" later in this chapter.

Live Broadcast of Audio and Video

NetShow 2.1 enables the delivery of live, real-time encoded audio, video, and illustrated audio content. By allowing many users to "tune into" a single broadcast transmission, network managers can dramatically reduce the load that would otherwise be placed on their networks when large numbers of users listen to or watch live events.

This represents a major improvement, as NetShow 1.0 only allowed the live delivery of audio over multicast-enabled networks, thus limiting the delivery of live events. NetShow 2.1 allows the delivery of live content, as well as on-demand content, over any network, independent of the transmission techniques and network transports used. This means that people using NetShow can broadcast live and on-demand content using either unicast or multicast techniques, depending on their needs and the available network infrastructure. To further facilitate delivery, NetShow can automatically switch from multicast to unicast to accommodate networks that are not multicast-enabled or to adjust to other delivery needs. For more information, see "Understanding Unicasting and Multicasting" later in this chapter.

Content Scheduling

NetShow 2.1 opens vast opportunities for Web-based broadcasts similar to the delivery of content over current television and radio stations. With NetShow, you can configure and schedule content streams for later delivery. You can also generate announcements to notify users of upcoming and in-progress sessions. This allows audio and video content to be organized into separate channels, each programmed with one or more individual programs. Each NetShow Server can have multiple channels.

Consider this analogy: A cable company manages multiple channels that are organized in a numeric sequence to make television surfing easy. For example, NBC might be channel 5, CNN channel 7, and so on. Each channel offers time-based programs, such as the 10:00 P.M. news, or a movie at 11:00 P.M. Users can easily find their favorite programs, because they know when and where to look. Following this analogy, NetShow makes it possible to organize audio and video content into separate channels, each of which has scheduled programs and announcements of current or upcoming sessions.

Easy User Access to Broadcasts

NetShow 2.1 brings the power of broadcasting to Microsoft Internet Explorer and other browsers such as Netscape Navigator, enabling browsers to receive a new class of innovative, interactive content and extending the browsers' strength for one-to-many communications. Because the NetShow Player is an integral part of browsers, users can easily view NetShow-hosted broadcasts. NetShow also adds NetShow Players for Windows 3.1, the Macintosh®, and UNIX operating systems.

Multicast File Transfer

NetShow supports multicast file transfer, which provides another way for network managers to save bandwidth when large quantities of data have to be simultaneously distributed to many users. Multicast file transfer provides a reliable mechanism to distribute any type of file. It can be used, for example, to dynamically change files on Web sites in order to broadcast a variety of Web content.

Multiple Network Transport Protocols

NetShow 2.1 supports a variety of network transport protocols, including IP multicast, RTP, UDP/IP, TCP/IP, and HTTP. As a result, NetShow users can distribute multicast audio to MBONE-compatible applications and can listen to MBONE multicasts (VAT-compatible on client and server). The broad range of network transport protocols supported by NetShow allows content providers to choose the protocol most effective for delivery of their content and the network infrastructure available.

NetShow is optimized to reduce content delivery losses due to changes or obstacles in the network infrastructure through its intelligent fail-over feature; it uses IP multicast on multicast-enabled networks and, if necessary, falls back to unicast UDP/IP traffic, then to unicast TCP/IP, and finally to HTTP traffic if unicast UDP/IP cannot be delivered.

The intelligent fail-over feature enables content providers to use NetShow to broadcast content, using either unicast or multicast techniques, without worrying about the nature of the available network infrastructure. Intelligent fail-over also allows content to be sent through firewalls, allowing intranet users to view NetShow content as easily as other users, without compromising the firewall security features.

Server Integration

NetShow and its streaming services are tightly integrated with Windows NT Server 4.0 to provide an efficient, reliable, scalable, and secure platform for delivering audio, illustrated audio, and video content over corporate intranets and the Internet. NetShow uses all key Windows NT Server administration features, including the graphical administration console, performance monitoring, the integrated directory and security model, and the event log.

NetShow also fully supports Windows NT connectivity, including network environments such as IP, IPX, 14.4/28.8 POTS, ISDN, Ethernet, and others. Sharing the same user interface, APIs, services, and tools means that end users and computer professionals don't have to learn different interfaces or tools. An integrated solution offers easier management, better connectivity, and lower support costs, and because NetShow and Microsoft IIS are designed to work together, they combine to deliver a complete, well-tested, and high-performance system for broadcasting multimedia.

Scalability

Broadcasts of audio and video on the Web need a server that can scale up to thousands of streams for enabling full-scale, one-to-many communication. The NetShow Server takes advantage of the scalability of Windows NT Server to provide both small-scale implementation and easy scaling to higher-performance systems. Thus, NetShow can easily adapt to an organization's growth.

NetShow scales to very high-performance, enterprise-class multiprocessor systems with many gigabytes of memory and terabytes of storage; it exploits the ability of Windows NT Server to run on leading-edge microprocessors such as Intel Corporation's Pentium and Pentium Pro and Digital Equipment Corporation's Alpha. Microsoft internal tests have shown that a NetShow Server can handle more than 1,000 streams of data at 28.8 Kbps on an industry-standard, Pentium Pro–based uniprocessor computer with 64 MB of RAM, with no impact on the quality of the content and services delivered to clients.

Distributed Delivery of Content Streams

With NetShow 1.0, all server components were installed on a single Windows NT Server. NetShow 2.1 allows the system to be divided into several components that can be installed on separate Windows NT Server computers. This increases the overall number of simultaneous streams that can be delivered, to provide services to an otherwise unserviceable number of clients. In addition, distributed delivery enables a geographically diverse distribution of NetShow servers, in order to optimize how content is distributed from the source to clients. The array of servers, in turn, can consist of many small, inexpensive computers, rather than just a few powerful but costly platforms.

Using Windows NT Server DNS, a cluster of NetShow servers can be configured to deliver content to a greater number of clients. For example, if you are broadcasting a live event, you can perform real-time encoding at the event site, using the NetShow Real-Time Encoder. The encoded content is transmitted in real time to a cluster of NetShow servers that then deliver the content to their respective connected clients.

Web-Based Administration

NetShow 2.1 administration is provided through the NetShow Administrator, an intuitive, Web-based interface that enables both local and remote administration of NetShow servers. Web-based administration allows administrators to centrally manage server installations and make configuration changes, without having to physically access each server.

Network managers can set the maximum content throughput per file and per server to control network bandwidth utilization and to monitor the state of all key NetShow Server services. The NetShow Administrator runs on both Windows NT and Windows 95 operating systems. Web-based administration enables faster deployment of information delivery solutions and reduces administrator training and support costs.

Tools for Monitoring and Logging Events

NetShow includes tools for monitoring and logging events. The Monitor Events Log is used to display events, and the NetShow Server Event Log is used to store events in a database. Both tools allow you to filter events by type: server, client, administrative, and alert. Server events indicate server status, such as whether a server is online or offline. Client events are logged, for example, when a client connects or disconnects. Administrative events are logged when configuration settings are modified, such as when the maximum number of clients or maximum bandwidth is modified. Alerts are logged when server configuration settings are exceeded, such as when the maximum number of clients is exceeded.

In addition, you can use all standard Windows NT Server administration features, such as the Windows NT Event Viewer and Performance Monitor, to monitor the number of streams sent from the server, the percentage of CPU time being taken by the NetShow services running, and other performance-related data.

Firewall Support

To prevent unauthorized access to their networks, corporations deploy firewalls. Most firewalls are based on packet filtering. In packet filtering, the computer examines the source and destination IP addresses of a packet and forwards only those packets for which access has been granted. The content streamed from Microsoft NetShow can stream through a number of firewalls from vendors such as Ascend Communications Inc., CheckPoint Software Technologies Ltd., Cisco Systems, Computer Software Manufacturer, Cycon Technologies, LanOptics Inc., Firewall System, Microsoft Corporation, Milkyway Networks, Secure Computing, and Technologic Inc. In addition, several other vendors are working to add NetShow support to their products. For a complete list of all firewall products that currently support NetShow, visit the NetShow Web page at **http://www.microsoft.com/netshow/**.

For firewalls that do not support streaming content from NetShow Server, you can stream NetShow content using an HTTP server. Firewalls can process HTTP streamed content; however, some of the functionality of ASF is not supported when streaming uses HTTP servers. For more information, see "Using HTTP Streaming" later in this chapter.

Components of NetShow

NetShow 2.1 includes the following software components:

- NetShow Player
- NetShow Server
- NetShow Administrator
- NetShow content authoring and production resources
- NetShow Software Development Kits (SDKs)
- NetShow Theater server

The NetShow components are available for free download from Microsoft. For more information about each of the NetShow components, visit the Microsoft NetShow Web page at **http://www.microsoft.com/netshow/**. To download components, visit **http://www.microsoft.com/netshow/software.htm**.

NetShow Player

The NetShow Player allows intranet and Internet users to play audio, illustrated audio (synchronized sound and still images), and full-motion video files. By simply activating a link to a file, the player launches automatically and begins playing the requested file. Within seconds, the content starts playing, with no download required. The player enables the same functions as a regular VCR, allowing the user to stop, pause, and start content.

The NetShow Player is provided as an optional component of the Internet Explorer 4 installation package. The NetShow Player is typically installed when you choose the Full Installation option during setup for Internet Explorer 4.

The NetShow Player comprises a stand-alone player and an ActiveX control that can be embedded in Web pages or in applications that support embedded controls. The stand-alone player can be used even if no browser is installed. NetShow Player can be run independently of the user's browser by launching it in a helper application that runs on top of the page. The advantage of playing content in the external player is that it can be launched by means of a small redirector in an e-mail message, launched from a shortcut to a location on a corporate network, or played in the background while the user surfs to other pages on the Web. The NetShow Player is available cross-platform, providing access to NetShow content for users of the Macintosh, UNIX, Windows 3.1, and Windows NT 3.51 platforms.

The ActiveX control allows you to embed the NetShow Player in Web pages to play NetShow content within the browser window of browsers that support ActiveX controls, such as Internet Explorer, versions 3 and 4. The ActiveX control also allows developers to embed NetShow content in applications that support ActiveX controls (including Visual Basic®, Visual C++®, and many third-party tools and applications).

The NetShow Player is also provided as a Netscape Navigator Plug-in, enabling users of Navigator to receive NetShow broadcasts directly within their browsers.

NetShow Server

The NetShow Server consists of a set of services running on Windows NT Server 4 that allow users to unicast and multicast audio, video, and other files to client computers. The NetShow Server can deliver both live and on-demand content. To deliver live, real-time content, the server works in conjunction with the NetShow Real-Time Encoder, which compresses the audio and/or video feed in real time and passes it to the NetShow Server for delivery to the network. Delivery of on-demand content requires that it be stored on a server's hard disk and passed to the network by the NetShow Server.

NetShow can restrict access and control client and server connections to the NetShow Server and the Real-Time Encoder based on the IP addresses of the clients or servers attempting to connect. Access is controlled by creating a list of addresses in the Windows NT registry. By default, all connections are allowed.

NetShow Administrator

The NetShow Server includes the NetShow Administrator, a set of documentation and tools that allow administrators to manage, configure, and monitor NetShow services. For example, you can use the NetShow Administrator to set the maximum number of concurrent streams the server can run, and you can limit the amount of data the server can send at one time. The NetShow Administrator also allows you to configure and set up channels and programs for multicasting content at specific times. The *NetShow Server Administrator Guide* provides detailed information on how to use the NetShow Administrator.

NetShow Authoring and Production Resources

Microsoft provides many resources to help you author both real-time (live) and on-demand (stored) NetShow content. The foundation for NetShow content creation is the Active Streaming Format (ASF). (For more information, see "Understanding the Active Streaming Format" later in this chapter.)

The NetShow authoring and production resources include:

- **Authoring tools for on-demand content.** NetShow provides simple ASF authoring tools to convert .avi and QuickTime® files to .asf or .wav files. Also provided is a simple ASF editor called the NetShow ASF Editor, which allows you to synchronize images to .wav audio files and add URL flip commands, markers, and script events. The NetShow tools also include the new NetShow Presenter, which allows a person giving a Microsoft PowerPoint® presentation to stream his or her video and slides to ASF.

- **Real-Time Encoder.** The Real-Time Encoder is a set of tools for encoding and authoring live multimedia streams. These tools allow content authors to encode live audio and video feeds and add them to dynamic mixtures of other media, such as audio, video, text, URLs, and script commands. These tools are used to indicate what media is to be injected into a live stream, and when. The appropriate media components are synchronized, compressed, augmented with error-correction information, and transmitted over a network.
- ***NetShow Content-Creation Authoring Guide.*** This HTML document provides detailed information and guidelines for creating live and on-demand ASF content. The document is updated frequently with new information. Be sure to download the latest version.
- **NetShow Encoder with RealVideo.** This tool converts video files into the RealVideo format for use in RealVideo programs.

NetShow Software Development Kits

Microsoft provides several SDKs for NetShow:

- ***NetShow Client SDK.*** This kit documents the interfaces to the player. This gives developers of Web pages and client applications all the information they need to integrate NetShow functionality into their products. Developers have complete control over audio and video rendering, along with the ability to interpret script commands transmitted as part of the NetShow stream.
- ***NetShow Server SDK.*** This kit documents the interfaces to the three administration controls. This gives application authors complete local or remote control over a network of NetShow Servers. For example, NetShow administration can be integrated into an overall, enterprise management system. The *NetShow Server SDK* supports the creation of NetShow Channel Manager administrative applications with the Channel Manager ActiveX control and the creation of NetShow unicast administrative applications with the Unicast ActiveX control. Independent software vendors and solution providers can build customized administration utilities using this SDK.
- ***Real-Time Encoder SDK.*** This kit documents the interfaces to the Real-Time Encoder itself. This gives application developers the ability to automatically insert script commands into the NetShow stream alongside audio and video. For example, a closed-caption or subtitling application can feed its output directly to the NetShow Real-Time Encoder by using these interfaces. The PowerPoint NetShow Presenter add-in for the PowerPoint presentation graphics program is an application written using the *Real-Time Encoder SDK.*

NetShow Theater Server

NetShow Theater Server adds a new streaming media engine to the NetShow Server capabilities. NetShow Theater Server is an optional feature of NetShow. It provides a powerful, scalable, and cost-effective solution for the delivery of MPEG-quality, full-motion, full-screen video across high-bandwidth corporate networks and dedicated video LANs.

NetShow Theater Server delivers a complete, high-performance system for streaming MPEG video that runs on the Windows NT Server platform. It's a distributed, fault-tolerant, real-time video server designed to provide MPEG video streams at a constant rate to large numbers of PC clients.

NetShow Theater Server's highly scalable and fault-tolerant architecture makes it ideal for video-on-demand applications that require large numbers of high-bandwidth video streams from 1 to 10 megabits per second (Mbps) or more.

Webcasting with NetShow

You can use NetShow 2.1 to provide true Webcasting solutions that push information to end users' computers. NetShow uses advanced streaming technology to deliver multimedia content optimized for network bandwidths from 14.4 Kbps to 6 Mbps.

True Webcasting can be used to overcome the limitations of basic Webcasting and managed Webcasting. For more information about basic Webcasting and managed Webcasting, see Part 4, "Information Delivery: Webcasting." For more information about using NetShow for true Webcasting, see the *NetShow Content Creation Authoring Guide*, the *NetShow Server Administration Guide*, and the NetShow SDKs.

True Webcasting Application Scenarios

Multimedia allows people to communicate more expressively and more effectively. The richer data structures of sound and moving images transcend the communicative power of text and graphics. NetShow provides a complete, true Webcasting platform for integrating audio and video into Web-based applications. These Web-based applications often deliver information solutions for training, corporate communications, advertising, and entertainment applications.

Training

Many organizations devote substantial resources to training. Using NetShow to extend the reach of professional instructors through intranets allows organizations to maximize the value of this investment. NetShow makes it easy for trainers to generate the content and for users to receive the training whenever and wherever they need it.

For example, a recorded speech in conjunction with the speaker's slides can form a NetShow broadcast. Training material can be provided for all divisions and subsidiaries of an organization so that all employees have the advantage of hearing the speech as it was delivered by the speaker.

Delivering multimedia information and presentations over the intranet or the Internet can help organizations save substantially on the costs required to distribute and maintain up-to-date training materials on hard copy or compact discs. It's easy to update NetShow content on the network and provide immediate, up-to-date information to users.

Corporate Communications

With NetShow, everyone in an organization, regardless of their geographic location, can hear important organizational briefings—such as internal presentations or presentations for the press or analyst—live. These same presentations can also be captured for later playback. The presentations can also grow into a library of available-on-demand information for reference and training. NetShow can reduce the burden of requiring all employees to fly into a central location for an announcement or training session, by allowing companies to send the content over the network to each employee's desktop or to a portable PC for those traveling.

Entertainment and Information

NetShow makes Web sites come alive with interactive audio and video content, from musical events and similar entertainment to news and other broadcasts. Organizations that already provide entertainment content on the Web can use NetShow to add visual content to their sites. When combined with products such as Microsoft Site Server, Enterprise Edition, organizations can offer NetShow-based entertainment content for fun and help justify its use by selling products over the Web, using the Site Server.

Advertising and Retailing

Advertising a product or service with audio and video can be much more compelling than doing it with static Web pages. For example, audio commentary can be used with images on a Web site to guide users through demonstrations of a product, a process, or the site itself. The synchronized sound and images of NetShow illustrated audio provide a rich environment for advertising and facilitate showing off a product or concept to best advantage, such as in a media catalog.

Organizations with external Web sites can promote and sell products on the Web using NetShow. Advertisers can use the NetShow streaming technology to deliver their messages without requiring users to wait for a movie or audio clip to download to their hard disks—which means users get the messages without frustrating delays.

Understanding NetShow Technology

NetShow provides a wide variety of features that allow customization of the system. How you use NetShow depends largely on the types of media you want to stream and the characteristics of the network used to deliver the data. Understanding the basic concepts discussed in this section will help you determine how to deploy NetShow for your particular application.

Understanding Streaming Media

Most of the audio and video content currently hosted on intranets and on Internet sites is downloadable. This means that the multimedia content must be copied to the user's local hard disk before it can be played. NetShow uses a client/server streaming architecture to deliver multimedia content to clients. *Streaming content* is digitized content that has been compressed or encoded into a format that the server can break down into packets and then stream across a network to a client player. Streaming is a significant improvement over the download-and-play approach to multimedia file distribution, because it allows content to be delivered to the client as a continuous flow of data, with little waiting time before playback begins. The content arrives, is buffered briefly, plays, and is discarded. It is never actually stored on the user's computer. NetShow users benefit by experiencing instant play, and they don't have the frustration of waiting for content to download before determining whether it meets their needs or interests. (For more information, see "NetShow Streams and Programs" later in this chapter.)

Understanding Network Bandwidth

Any computer network connection (Internet or intranet) has an upper limit on the amount of data that can pass through it in a given second. This data limit is called its *bandwidth*. The *data rate* (also *bit rate*) of an audio and video file is the amount of data that must transmit in a given second for the whole file to be heard or viewed in its entirety. To transmit a content file completely and smoothly, its data rate must be less than the available bandwidth of its target network.

Understanding Content Compression

Because today's networks are usually bandwidth-constrained, audio and video files must be compressed to reduce their data rates. To compress them, we apply mathematical compression/decompression algorithms called *codecs* that analyze the audio and video and decide what bits of data can be removed or merged with minimum impact on what the human ear can hear or eye can see. By nature, however, applying compression to an audio or video clip results in some loss of quality.

The level of quality and fidelity you can deliver in sound and video files depends primarily on how much bandwidth you have available and whether you have authored the content appropriately for that available bandwidth. For example, imagine that the bandwidth available is like an empty pipe. You decide to fill that pipe with audio only. If your pipe is large (that is, you are on a corporate network), you can author that audio to be only slightly compressed, so it has very high quality. If your pipe is small (you are targeting users on 28.8-Kbps dial-up connections), you have to use a codec to compress the audio to fit in such a small pipe, and as you would expect, it has a lower quality than audio authored for higher bandwidths. Now, imagine that you want to add images or video to that audio content. In order to make room in that pipe, you need to compress the audio even more. You also need to compress the images or video significantly. The end result is highly compressed multimedia that can play at bandwidths that previously supported only still images.

Understanding Codec Independence

The quality of streaming content for a given bit rate is largely determined by how good the codec is that you use to compress it. Advances in codec technology are happening practically monthly, and for this reason, NetShow was developed to be codec-independent. This means NetShow includes the best variety of codecs for creating content in a wide range of application and bandwidth scenarios. Content providers can choose the best compression scheme for their particular type of application and content, and users can decode without hassles.

NetShow offers the content author and end user a well-rounded suite of bundled and stand-alone third-party codecs for voice, music, images, and video, at a variety of bit rates. This gives you more options, more flexibility, higher-quality content, and faster innovation, because the codecs can be upgraded independent of NetShow. The codecs provided with NetShow include Microsoft MPEG-4, Vivo G.723 and H.263, Lernout and Hauspie (L&H), Fraunhofer Institut Integrierte Schaltungen (FhG) MPEG Layer-3, Voxware MetaSound, VDONet, and Duck TrueMotion RT. The inclusion of these new audio and video codecs makes NetShow 2.1 the most comprehensive platform available today for delivering high-quality, streaming multimedia content over a wide range of networks.

For more information, see "Using Codecs" later in this chapter, and visit **http://www.microsoft.com/netshow/codecs.htm**.

Understanding the Active Streaming Format

The foundation for all of the NetShow content-creation components is the Active Streaming Format. ASF is an open, standards-based file format that prepares multimedia content for streaming and adds error correction and other features necessary for streaming. ASF also enables the synchronization of different data types on a common timeline, enabling, for example, .jpg images, bitmaps, or .wav files to be synchronized with each other. Multimedia content must be converted to ASF before delivery over a network. ASF content can be hosted on a local hard disk, an HTTP server, or a specialized media server such as NetShow. This hosting flexibility gives extraordinary freedom to the content creator: the same content can be played locally from a compact disc or a hard disk, or played from a remote location hosted on an HTTP server or a specialized media server. ASF allows content and tool developers to work to a shared specification that supports the authoring, combining, archiving, annotating, and indexing of synchronized media objects, without regard to original media formats or underlying transports.

Multimedia content is stored in ASF as objects. Multimedia objects can include audio, video, still images, events, URLs, HTML pages, script commands, and executable programs. You can easily combine existing multimedia objects into a single ASF multimedia stream. ASF retains the form of each object's media (including audio and video compression), as well as optional synchronization information. Therefore, when the file is played over a network, users see and hear the file exactly as intended.

ASF data objects are stored within .asf files as packets. Each packet is designed to be directly inserted "as is" into the data field of data communication transport protocols. These packets are streamed across a network at a specific bandwidth or bit rate, so users can use, or "play," the multimedia data as it arrives. ASF supports a number of standard graphics, video, and audio file formats. You can easily convert other formats to a format that is supported by ASF.

ASF data can be tailored to satisfy a wide variety of differing network requirements. The data in each .asf file is optimized to stream at a distinct bit rate. You can specify ASF streaming bit rates from 14.4 Kbps to 6 Mbps. ASF content can thus be flexibly targeted for specific network environments with distinct capacity requirements.

For more information about ASF, see the Active Streaming Format white paper, available at **http://microsoft.com/netshow/about/whtepprs/**.

Understanding Error Correction and Masking

As files stream across a network, packets of information can be lost, particularly on the Internet. When packets are lost, users can experience choppy play, loss or gaps in the audio or video, and even loss of entire images. If packets stop being delivered, the NetShow Player stops rendering until it can rebuild its buffer.

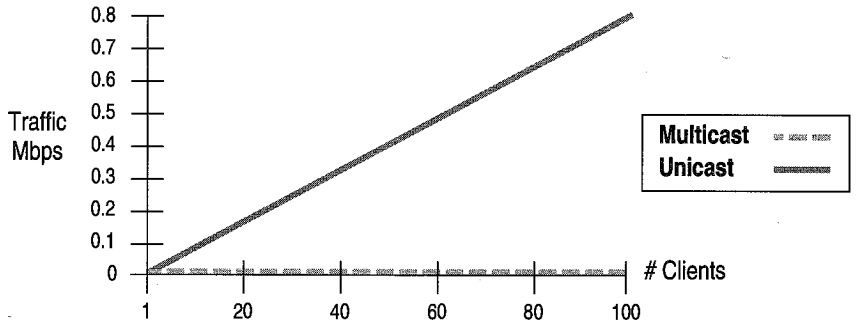
To improve performance when packets of information are lost during transmission, you can use the NetShow error correction and masking feature of the ASF file format. For example, the ASF Editor and ASF Real-Time Encoder can optionally include redundant error-correction data in the stream, which enables the NetShow Player to correct for certain levels of packet loss. If packet loss is extreme (to the point where error correction is not possible), the player masks the errors. For audio masking, the player uses neighboring audio information to conceal the loss of data. For image masking, NetShow supports conversion of images into loss-tolerant .jpg images, which the NetShow Player can then use to minimize the loss of data packet, as opposed to discarding the entire image.

Understanding Unicasting and Multicasting

Unicasting refers to networking in which computers establish two-way, point-to-point connections. The bulk of the traffic on today's networks is unicast: A user requests a file, and a server sends the file to that client only. The disadvantage of unicasting is that each client that connects to the server receives a separate stream, which rapidly uses up network bandwidth. Networks also support *broadcasting*. When data is broadcast, a single copy of the data is sent to all clients on the network.

When the same data needs to be sent to only a portion of the clients on the network, both unicasting and broadcasting waste network bandwidth. Unicasting wastes bandwidth by sending multiple copies of the data. Broadcasting wastes bandwidth by sending the data to the whole network whether or not the data is wanted.

Multicasting avoids the weaknesses of both of these approaches. Multicasting sends a single copy of the data to those clients who request it. Multiple copies of data are not sent across the network, nor is data sent to clients who do not want it. Multicasting allows the deployment of multimedia applications on the network, while minimizing their demand for bandwidth. The following graph compares the network load per client, when unicasting and multicasting an 8 Kbps PCM audio stream, and shows how a multicast saves bandwidth.

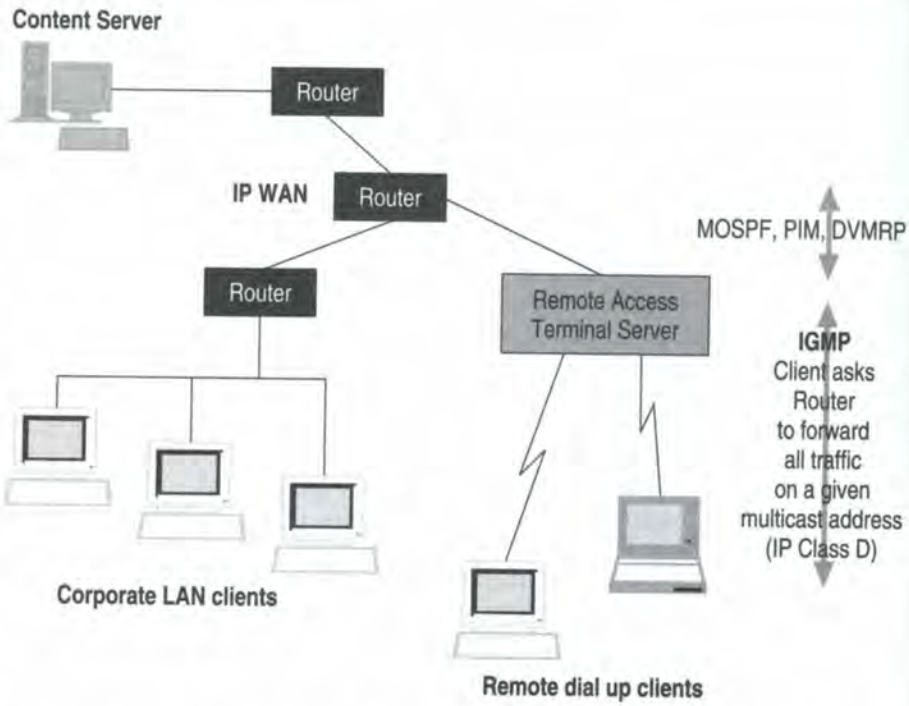


Unicast content is stored on a server, and users request the stream by opening a URL for the unicast. With unicasting, the content is sent over the network to each user on demand. When streaming multimedia over a network, the advantage of unicasting is that the client computer can communicate with the computer supplying the multimedia stream. With Microsoft NetShow, for example, the NetShow server can provide a unicast video stream to a client, and the client can take advantage of the VCR controls in the NetShow Player to ask the server to pause the stream or to skip backward or forward to a marker in the stream. As you can see, however, you pay a high price in bandwidth for this control.

Multicast content is broadcast from a server over the network, and users "tune in" to receive the content. With multicasts, the network bandwidth consumed is the same whether 1 user or 1,000 users receive the content stream. However, as with tuning in to a live television program, the user loses the ability to fast-forward and rewind.

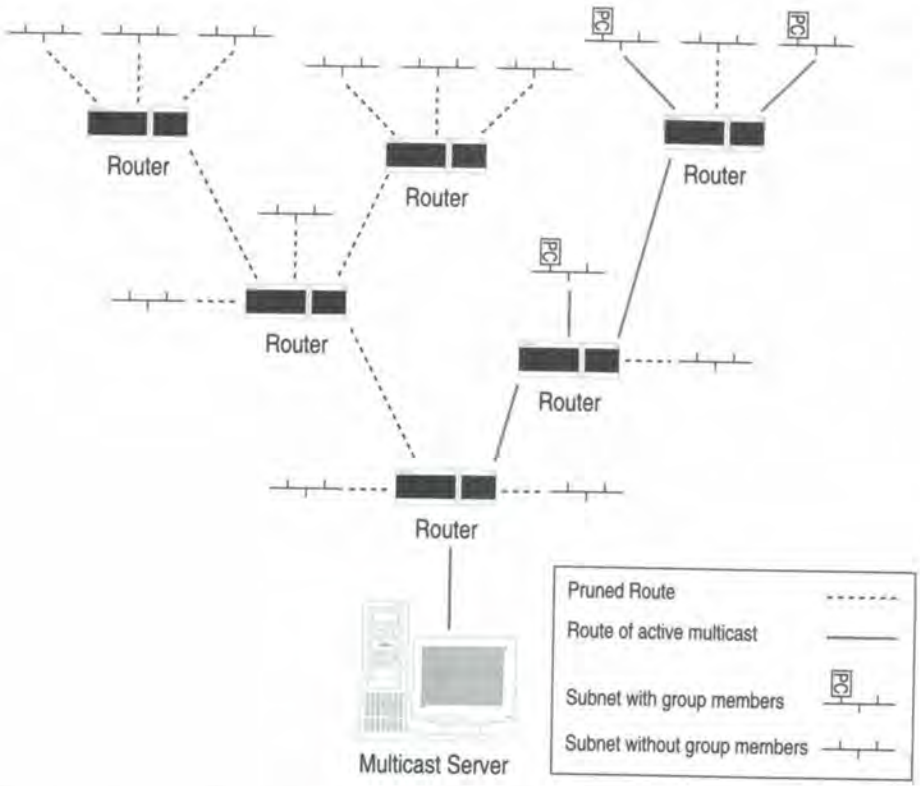
Multicasting can provide bandwidth-efficient Webcasts when tens, hundreds, or thousands of individuals need the same information. For example, sales trends for the week can be presented to all regional sales managers by means of a multicast. Events such as a product introduction or an important press conference can also be multicast.

Multicasting follows a "push" model of communications. That is, like a radio or television broadcast, those who want to receive a multicast tune their machines to the channel they want to receive. In the case of multicasting, the user is simply instructing the computer's network card to listen to a particular IP address for the multicast. The computer originating the multicast does not need to know who has decided to receive it.



Multicast is sometimes called *IP multicast* because it requires a network with routers that support IP multicasting protocols. Two years ago, major router manufacturers began adding multicasting capability to their routers. Multicasting can be enabled on such routers by simply updating their software and adding memory.

Three multicast routing protocols are in use today: Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF) protocol, and Protocol-Independent Multicast (PIM). The task of these protocols is to create efficient multicast delivery paths through the network. Multicast routing protocols use varying algorithms to achieve efficiency.



An efficient delivery path ensures that multicast data travels only to those users who want to receive it and takes the shortest path to those clients. If data travels elsewhere through the network, bandwidth goes to waste. You can visualize the network as a tree structure. The source of the multicast sends data through the branches of the tree. The routers are responsible for sending data down the correct branches to other routers and to the subnetworks, where members of a group are waiting for data. Routers prune off branches where no one wants data and graft branches back to the tree when a client in a new subnetwork joins the group. Routers can also stop data that is not wanted from traveling to their own subnetworks.

Multicasts can also support bidirectional communication. For example, individuals in widely dispersed locations can use Microsoft NetMeeting to set up a live conference that includes audio, video, and a whiteboard. For more information, see Chapter 20, "NetMeeting."

Understanding Unicasting and Multicasting with NetShow Server

NetShow Server is a service, similar to an HTTP server, that runs on a Windows NT Server. When NetShow Server is installed, a publishing point is established as the directory where the server will look to find the files that it will stream. When you place an .asf file on a NetShow Server for streaming, you must put that file in the NetShow publishing point; otherwise, NetShow Server won't know where to find the .asf file. When you enter a URL to access or play an .asf file, you specify the name of the NetShow Server and then the path to the .asf file. NetShow Server immediately begins looking in the NetShow publishing point for the requested .asf file. NetShow Server streams .asf files to NetShow Players.

To play an .asf file from a NetShow Server, you can open the .asf file using the stand-alone NetShow Player. You can also open .asf files using the browser, and the streamed content will automatically play using the integrated NetShow ActiveX control, which functions in the same way as the stand-alone player. For more information, see "Using the NetShow Player" later in this chapter.

NetShow Announcements

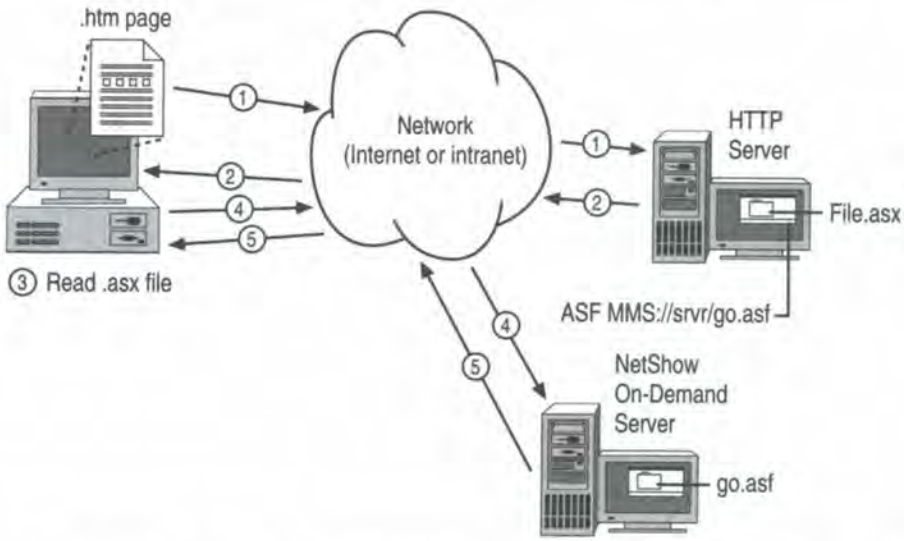
You can use a NetShow announcement (.asx) file to point users to a multicast channel or to the source of a unicast stream. The .asx file contains the information that NetShow Player needs to locate and play unicasts and multicasts.

There are two types of .asx files: those that are encrypted and those that aren't. Encrypted .asx files are used for multicasts. You use the NetShow Administrator Channel Manager to create .asx files for multicasts. Nonencrypted .asx files are used for unicasts. You can use a text editor to create .asx files for unicasts. For more information on creating .asf files, see the *NetShow Server Administration Guide* and the *NetShow Content Creation Authoring Guide*.

To notify users of a unicast or multicast, you can:

- Send .asx files to users by e-mail.
- Add the .asx file as a link reference in a Web page.
- Add the .asx file to a public directory on a network share.

The most common method for using an .asx file is as a link in a Web page. The following illustration shows what happens when a user clicks an .asx file on a Web page.



In this illustration, the user is browsing the Web page, clicks the .asx link, and the NetShow Player is called to play the .asf file. The basic process follows:

1. The user clicks the link—the .asx file.
2. The HTTP server sends the .asx file to the requesting browser.
3. The browser reads the .asx file and determines which NetShow server to contact and which .asf file to request.
4. The browser launches the NetShow Player to play the requested .asf file.
5. The browser contacts NetShow Server and requests the specified .asf file.
6. The NetShow Server begins streaming the .asf file to the NetShow Player.

HTTP Versus MMS

After you create .asf files, in order to stream them, you must place them on the NetShow Server or on an HTTP server. It is recommended that you place .asf files on the NetShow Server, because only there do you receive all the ASF functionality. HTTP servers can stream .asf files; however, an HTTP server cannot duplicate all the functionality of a NetShow Server.

In addition, when identifying the path to an .asx or .asf file, you must specify the protocol used to deliver the .asf file, either HTTP or the MMS protocol. For example:

```
mms://netshowserver/file.asf
```

```
http://webserver/file.asf
```

You normally want to use the MMS protocol when specifying URLs to stream .asf files to users. MMS streams content more reliably and delivers better streaming performance than HTTP. However, since some firewalls may not allow for streaming content that uses the MMS protocol, it may be necessary for some users to stream .asf files from an HTTP server. For more information, see "HTTP Streaming" later in this chapter.

NetShow Channels

When multimedia or files are multicast using NetShow, a channel is first set up. You use the NetShow Administrator Channel Manager to create and administer channels. The channel can be populated with programs, which are the content for that channel. Programs can be live feeds or on-demand content scheduled to be played at specific times. The channel establishes the communication between the server and the multicast clients. Much like television, what is seen on your screen may be live, or it may be played from a tape at a specific time. Also like television, you can't skip ahead or back; you get the broadcast and if you don't like it you either turn it off or change channels.

A NetShow channel specifies the destinations for ASF streams and provides the information that clients need to listen to the channel, including information on the IP address and port for the multicast and the format of the stream itself. NetShow channels are analogous to broadcast television channels. Just as television programs are assigned to channels, programs in NetShow are assigned to channels by specifying that the streams in the program use a particular channel.

Channels provide the flexibility to stream data using multicast and unicast methods to fit a variety of situations:

- To distribute streams by multicasting, specify a multicast address.
- To distribute streams by unicasting, specify a unicast server to which clients can connect. This allows clients that are not able to receive the multicast to receive the data.
- To distribute data to additional NetShow servers, set up a channel for distribution.

A channel is required when multicasting ASF streams. When unicasting ASF streams, using a channel is optional.

Although the primary use of a channel is to provide address and stream information to users, channels can also:

- **Roll Over to Unicast.** You can set up a channel so that clients unable to receive the multicast can automatically connect to the NetShow Server and receive a unicast of the same information. This feature can be used to reach

- **Stream Distribution.** You can set up channels to supply secondary NetShow servers with streams, which they can, in turn, multicast. For example, you can distribute a stream to NetShow servers located on separate local area network segments. Each server can then multicast to clients on its segment. This feature is useful on intranets with routers that are not configured for multicasting.

You use the NetShow Administrator Channel Manager to create channels. After you create a channel, export the channel (.asc) file to a location where clients can reach the file. Typically, this is a publishing point on your HTTP server. However, it can also be a public share on your network.

NetShow Streams and Programs

A NetShow *stream* refers to the data transmitted across the network and to any properties associated with the data. A NetShow *program* consists of a group of one or more streams. You use the NetShow Administrator Program Manager to create and administer programs. You use a program to organize the streams you want to multicast. Starting a program starts the multicast of the streams in the program. A program plays its streams one at a time, in order. When you add streams to a program, you can specify how long each stream plays. Information about programs, such as what streams to play and how long to play them, is stored in program (.nsp) files on the NetShow Server.

The term *stream* also refers to the way in which data is transmitted across the network. When data is streamed across the network, clients can render the data as they receive it. For example, the client can play an audio stream as it arrives rather than storing it in a file and playing it only after the entire file has been received.

NetShow supports the following stream types:

- **Active Streaming Format (ASF).** ASF supports the greatest variety of data, including video, audio, images, URLs, and scripts. You can generate live ASF streams that contain audio and video using the NetShow Real-Time Encoder, or use the tools provided with NetShow or those from third parties to create and store ASF files that you can stream. An ASF stream can combine different types of data. For example, if you want to present a lecture that includes slides, you can stream audio while also streaming .gif files of the slides.

  By using NetShow channels, ASF provides the greatest flexibility for various network situations. NetShow can multicast or unicast ASF streams. When multicasting an ASF stream, you can also configure NetShow to provide a unicast source for the stream so that clients that are unable to receive the multicast can automatically receive it as a unicast.

- **RTP Live Audio.** RTP Live Audio streams audio that is fed into the server's sound card. NetShow supports RTP Live Audio multicasts, but not unicasts. Multicasts that stream RTP Live Audio are easy to set up. However, because channels do not support RTP Live Audio streams, connection information must be coded in the embedded control that clients use to receive and render the multicast.
- **RTP WAV Audio.** RTP WAV Audio streams audio recorded as .wav files. NetShow supports RTP WAV Audio multicasts, but not unicasts. Multicasts that stream RTP WAV Audio are easy to set up. However, because channels do not support RTP WAV Audio streams, connection information must be coded in the embedded control that clients use to receive and render the multicast.
- **File transfer.** File transfer supports the streaming transfer of directories and files. NetShow supports file transfer multicasts, but not unicasts. Multicasts that stream file transfers are easy to set up. However, because channels do not support file-transfer streams, connection information must be coded in an embedded ActiveX control that clients use to receive and render the multicast.

NetShow Multicast and Unicast Scenarios

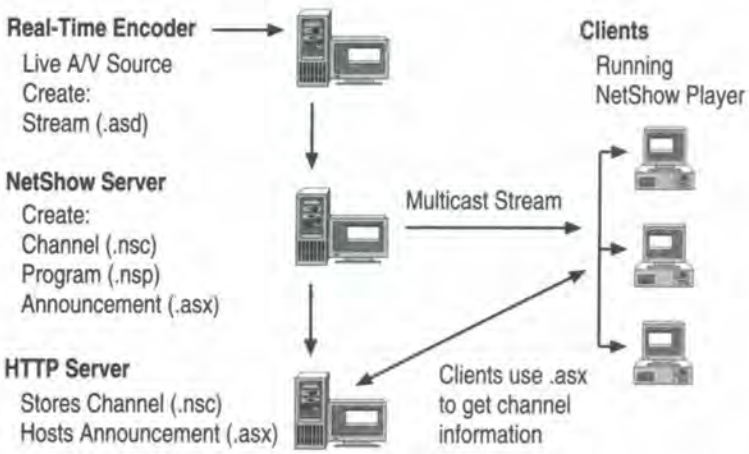
You can set up and run NetShow 2.1 to provide a wide variety of true Webcasting solutions. The possibilities are nearly unlimited, but the following sections suggest some of the ways you can use NetShow to provide Webcasting solutions. (For more information about application scenarios, see "Webcasting with NetShow" earlier in this chapter.)

Multicasting Live ASF Streams

You can use ASF streaming to deliver live multimedia content to users' desktops. For example, you can multicast a CEO's speech over the corporate network to all employees. The only limitations on live ASF streaming are those presented by the network bandwidth.

You multicast audio and video by feeding the signal into the audio or video capture card on a server with the Real-Time Encoder. The Real-Time Encoder server delivers the encoded information to the NetShow Administrator Channel Manager on the NetShow Server, which multicasts a live ASF stream. As with .asf files, live ASF streaming must be restricted to the bandwidth of the network. In the case of live ASF streaming, the Real-Time Encoder uses codecs during the encoding process to make the outgoing ASF stream small enough to stream across the network.

The following illustration shows the Real-Time Encoder, NetShow Server, and HTTP server on separate computers. While all software can run on a single computer, in most cases you want to run the system as shown. Because the Real-Time Encoder places heavy demands on the processor, dedicating a computer to run it can ensure optimum performance.



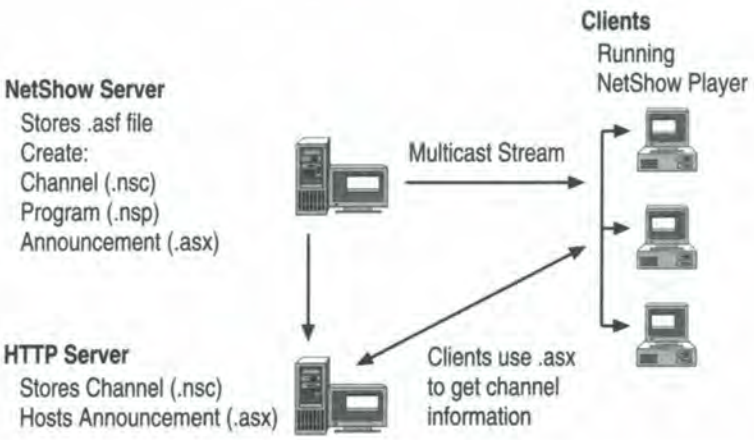
The Real-Time Encoder encodes live or stored audio and video into an ASF stream (a temporary .asd file), an .asf file, or both. The Real-Time Encoder is an extremely useful NetShow content-creation tool for two reasons: it is the only tool that can accept a live source, and it is the only tool that can compress the video source (live or stored) so that the resulting ASF stream or .asf file fits in a target bandwidth. All other tools require you to edit the video before you convert it to an ASF stream or .asf file. You can even use the Real-Time Encoder to convert .avi or .wav files into .asf files, without doing any preproduction work. Because the Real-Time Encoder can use video as a source, the encoder lets you set the size of the display window and choose a codec to compress the output ASF. Even if you are using live video and compact disc–quality audio as input, the ASF Real-Time Encoder can create an ASF stream with a 28.8-Kbps bit rate.

The Real-Time Encoder delivers the ASF stream that it creates to NetShow Server, which can then multicast or unicast the ASF stream over the network. If the Real-Time Encoder is used to create an .asf file, that file is saved to the directory that is specified during the configuration process. To play back an .asf file created by the Real-Time Encoder, you use NetShow Player to open the .asf file. For more information, see the *NetShow Server Administration Guide*.

The computer that houses the Real-Time Encoder must contain a sound card to encode live audio and a video capture card to encode video; you don't need either of these cards to encode an .avi or .wav file to an .asf file. The ASF Real-Time Encoder uses the inputs on the audio and video cards to receive the live source. For example, to multicast a live speech, you would plug the output from a video camera into the video card's input jack, and plug the output from the microphone into either the video card's audio input or the input on the sound card (whichever input the encoder is set to use). The encoder acknowledges these input sources, receives the input, and encodes the audio and video into an ASF stream that is then delivered to NetShow Server.

Multicasting Stored ASF Streams

You can multicast ASF streams stored as .asf files on your server. The following illustration shows how NetShow may be set up to multicast ASF streams.



All software can run on a single computer; however, if you plan to use NetShow Server's unicast HTTP streaming, you must run your HTTP server on a separate computer.

Basically, you store the ASF files you want to stream in the publishing point on the NetShow Server. You generate a channel (.nsc) file for each stream and place the .nsc files at a location on the network where clients can reach the files. For example, if you are multicasting over a local area network to clients running the stand-alone NetShow Player, you can store the file in a public directory. If you are multicasting over an intranet or the Internet to clients running NetShow Player embedded in a Web page, you can store the file on the HTTP server.

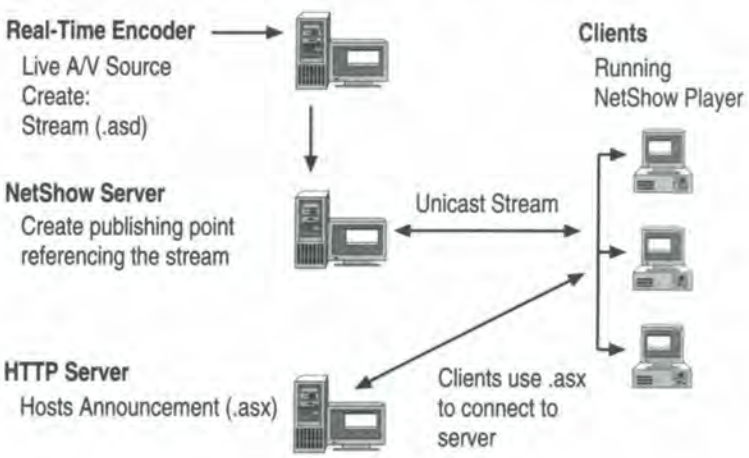
You use the NetShow Administrator Program Manager to configure programs and create announcements (.asx files) for the programs. Announcements provide the information needed to locate the .nsc file that supplies channel information for the program. You can provide the announcement as a link on a Web page, or you can send the announcement to users in an e-mail message.

You use the Program Manager to start the program to begin the multicast. Users can now listen to the multicast, using their NetShow Player.

Unicasting a Live ASF Stream

You can also unicast a live ASF stream from the Real-Time Encoder. Setting up a unicast in this way allows users to receive one ASF stream. If you want clients to receive a series of streams, set up the unicast using a channel.

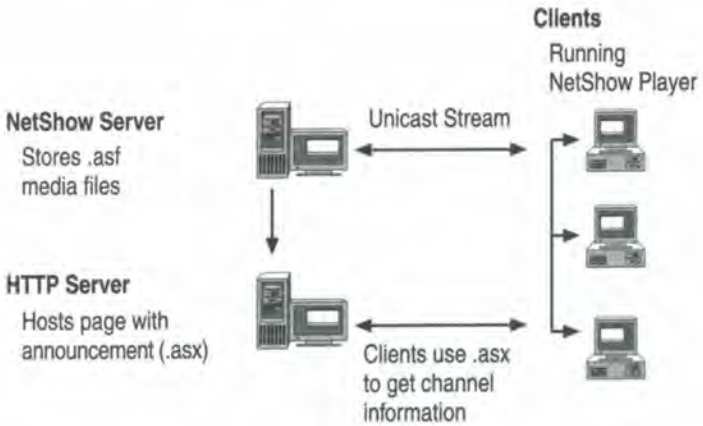
The following illustration shows the Real-Time Encoder, NetShow Server, and the HTTP server on separate computers. While all software can run on a single computer, in most cases you want to run the system as shown. Because the Real-Time Encoder places heavy demands on the processor, dedicating a computer to run it can ensure optimum performance.



Unicasting a Stored ASF Stream

You can unicast an ASF stream stored as an .asf file on your server. When unicasting a stored ASF stream, users play only the one .asf file. If you want users to play a series of files, set up the unicast using a channel (as described in the next section).

The following illustration shows NetShow Server and the HTTP server on separate computers. All software can run on a single computer; however, if you plan to use NetShow Server's unicast HTTP streaming, you must run your HTTP server on a separate computer.

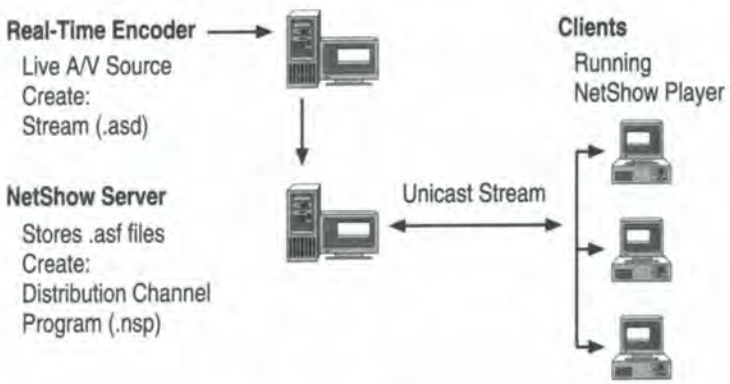


Basically, you store the .asf files you want to stream in the publishing point on the NetShow Server. You then create an announcement (.asx) file for users and store the .asx file where users can reach it. You can provide a link to the .asx file on a Web page. When users click the .asx link on the Web page, their browsers launch the NetShow Player to play the .asf file.

Unicasting Stored ASF Streams Using a Channel

You can use a channel to unicast a program that contains a series of ASF streams. The program can contain both live streams from the Real-Time Encoder and streams stored as files on your server.

The following illustration shows the Real-Time Encoder and NetShow Server on separate computers. If you are only unicasting stored streams, you do not need the Real-Time Encoder server.



While all software can run on a single computer, in most cases you want to run the system as shown. Because the Real-Time Encoder places heavy demands on the processor, dedicating a computer to run it can ensure optimum performance.

Basically, you store the .asf files you want to stream on the NetShow Server. You generate a channel (.nsc) file for each stream and place the .nsc files at a location on the network where clients can reach the files. For example, if you are multicasting over a local area network to clients running the stand-alone NetShow Player, you can store the file in a public directory. If you are unicasting over an intranet or the Internet to clients running NetShow Player embedded in a Web page, you can store the file on the HTTP server.

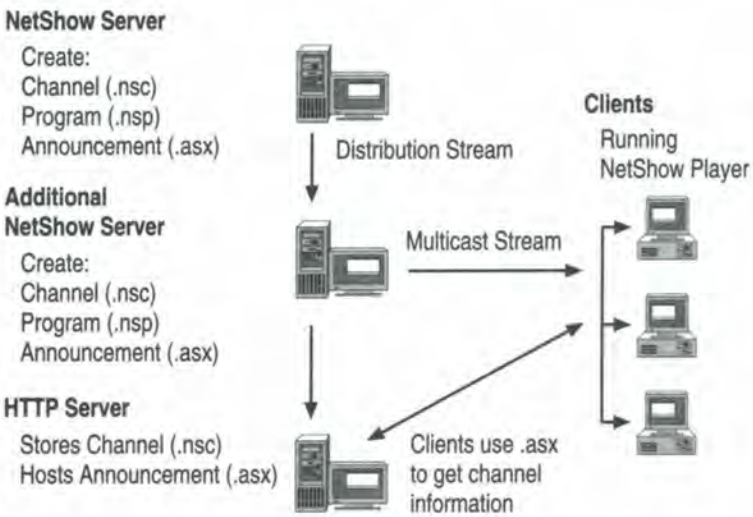
You use the NetShow Administrator Manage Programs tool to configure programs and create announcements (.asx files) for the programs. Announcements provide the information needed to locate the .nsc file that supplies channel information for the program. You can provide the announcement as a link on a Web page, or you can send the announcement to users in an e-mail message.

You use the Manage Programs tool to start the program to begin the multicast. Users can now listen to the multicast using their NetShow Player.

Setting Up ASF Stream Distribution Servers

You can set up a NetShow Server to feed additional NetShow Servers. Such an arrangement can be useful when your network routers are not multicast-enabled. For example, you can set up NetShow Server to originate a live stream, unicast it to additional NetShow Servers located on subnetworks, and then multicast to the clients on those subnetworks.

The following illustration shows how you can set up a NetShow Server to feed a second NetShow Server, which provides a unicast stream to clients on a subnet.



On the second NetShow Server, in NetShow Administrator, you use the Configure Channels tool to create a channel for the stream. The channel must match the channel you created when setting up the first server, except that the IP address and port used for the multicast must be changed. You can create the new channel by importing the original channel's .nsc file and modifying it.

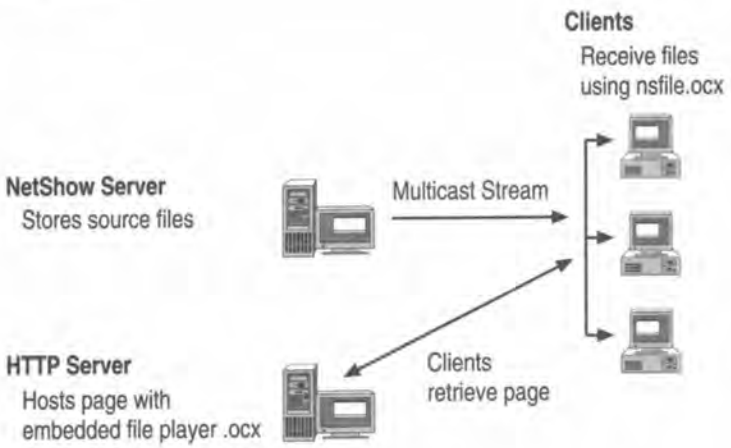
On the second NetShow Server, you use the NetShow Administrator Manage Programs tool to configure programs and create announcements (.asx files) for the programs. Announcements provide the information needed to locate the .nsc file that supplies channel information for the program. You can provide the announcement as a link on a Web page, or you can send the announcement to users in an e-mail message.

You use the Manage Programs tool to start the program on the second NetShow Server to begin the multicast. Users can now listen to the multicast from the second server using their NetShow Player.

Multicasting File Transfers

You can use NetShow multicasts for transfer of large files over the network. Multicast file transfer provides a reliable mechanism for distributing any type of file. With multicast file transfer, delivery is guaranteed only within the limits of the forward-error-correction scheme used in NetShow.

The following illustration shows NetShow Server and the HTTP server on separate computers. When multicasting files, all software can run on a single computer.



Basically, you store the files to multicast on the NetShow Server, and then use the NetShow Administrator Program Manager to create a program that organizes one or more media streams for multicast. You add a file to the program by specifying its file name and properties, such as the file location, the amount of bandwidth to use for the multicast, the destination IP address and port, and the scope of the multicast.

You must also create a Web page that includes the file-transfer control Nsfile.ocx and the multicast address where the client computer listens to receive the file. You store the Web page in the appropriate publishing point on the HTTP server to make it available to users. For more information, see the *NetShow Client SDK*.

You use the Program Manager to start the program on the NetShow Server to begin the multicast. Users can retrieve the Web page with the embedded Nsfile.ocx control to begin the multicast file transfer.

Using HTTP Streaming

You can unicast .asf files using the HTTP protocol with NetShow to stream content through firewalls that don't support MMS. Using HTTP streaming allows clients to receive streams across a network firewall; however, HTTP streaming does not perform as well as MMS streaming, and you cannot use certain features of the .asf file, such as jumping to markers.

Note All server software can run on a single computer; however, if you use unicast HTTP streaming, you must run your HTTP server on a separate computer.

By default, HTTP streaming is disabled. To enable HTTP streaming, you must edit the Windows NT system registry to change the setting for HTTP streaming and then stop and restart the NetShow unicast service. You must also add two MIME types for the .asf and .asx extensions on the HTTP server computer.

Caution Before editing the system registry, be sure to back up the registry. Incorrect changes or deletions made to the registry can damage the Windows operating system and prevent the system from running normally.

▶ **To enable HTTP streaming**

1. On the **Start** menu, click **Run**.

   The **Run** dialog box is displayed.

2. In the **Open** box, type **regedit**, and then click **OK**.

   Registry Editor starts.

3. In Registry Editor, use the tree view to navigate through the following path:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nsunicast\parameters
4. Click the **parameters** folder.
5. In the **Name** column, click **EnableHTTPStreaming**.
6. On the **Edit** menu, click **Modify**.
7. In the **Edit DWORD Value** dialog box, type **1** in the **Value data** box, and then click **OK**.
8. On the **Registry** menu, click **Exit** to close Registry Editor.
9. On the **Start** menu, point to **Settings,** and then click **Control Panel**.
10. In Control Panel, double-click the **Services** icon.
11. In the **Services** dialog box, click **NetShow Unicast Manager**, and then click **Stop**.
12. In the **Stopping** dialog box, click **OK**.

    Both the NetShow Unicast Manager service and the NetShow Program Manager service are stopped.
13. In the **Services** dialog box, click **NetShow Unicast Manager**, and then click **Start**.
14. In the **Services** dialog box, click **NetShow Program Manager**, and then click **Start**.

▶ **To set MIME types for Microsoft Internet Information Server**

1. On the **Start** menu, click **Run**.

   The **Run** dialog box is displayed.
2. In the **Open** box, enter **regedit**, and then click **OK**.

   Registry Editor starts.
3. In Registry Editor, use the tree view to navigate through the following path:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\MimeMap
4. Click the **mimemap** folder.
5. On the Edit menu, click **New**, and then click **String Value**.
6. In the right pane, enter **video/x-ms-asf,asf,,5** in the **mimemap** folder.
7. On the **Registry** menu, click **Exit** to close Registry Editor.
8. Restart the Internet Information Server.

▶ **To set MIME types for non-Microsoft HTTP servers**

1. Refer to your HTTP server documentation and follow instructions for adding the following:

   MIME type: video/x-ms-asf
   Extensions: asf,asx

2. If necessary, restart the HTTP server.

Developing Content for NetShow

Microsoft NetShow delivers live and on-demand multimedia content over intranets and the Internet. NetShow can transfer live multicast audio and files and can play on-demand streamed audio, illustrated audio, and video.

For more information about developing content for NetShow, see the *NetShow Content Creation Authoring Guide*, and visit the NetShow How-To Web page at **http://microsoft.com/netshow/howto.htm**.

A NetShow Scenario

Suppose that you work for ABX Compute Corporation, and your company is getting ready to release the newest version of your product, the Widget. ABX Compute is going to have a release party to promote the new Widget, and it is your job to design a way for all company employees, partners, and potential customers to share in the festivities, informational talks, and training that will go on.

At the party, the company president will give a keynote speech, and then several vice presidents will give demonstrations on how to use the Widget. ABX Compute always videotapes release parties. You decide to use NetShow, because you know that you can stream audio and images across the Internet; but your boss isn't quite convinced that it's the tool to use. He asks:

- *How do we deliver rich multimedia to Internet customers who are using 28.8-Kbps modems?*

  You explain that with NetShow, you can create and stream audio and video while controlling the bandwidth. NetShow is codec-independent, and because all NetShow installations include the VDONet VDOWave codec, you can compress the video for streaming over a 28.8-Kbps bandwidth network.

- *If lots of people around the company are running multimedia files across the network, won't that bog down all communications on the network?*

  Multicasting a file sends a single file to multiple users across the network. In the case of streaming files, a system administrator designates a maximum amount of data that can be streamed across the network at one time. This keeps the network from being bogged down or from interfering with other communication on the network.

- *How do we create the training and presentation files without spending a lot more money?*

  NetShow comes with several tools that allow you to turn various sources into .asf content. You can use the videotape to capture frames for illustrated audio .asf files and for Internet- and intranet-bandwidth video .asf files. You can use the existing video editing tools to capture frames and to edit the video.

- *ABX Compute has licensed a new video codec from Awesome Computers. Can we use this codec with NetShow?*

  NetShow is codec-independent. You can use any PCM/ACM codec for audio or video as long as the user's computer has the same codec.

- *All company employees are going to want to be part of the release party, even if they can't attend.*

  With NetShow, you can multicast the audio live from the release party to every employee, partner, or customer desktop. All any user needs to receive the multicast audio is the NetShow Player.

- *But what about viewing the demonstrations and training?*

  You can use the demonstration and training portion of the videotapes to create various NetShow .asf files. As opposed to forcing users to download huge video files, NetShow streaming allows users—even during peak viewing hours—to watch video files with no download time. The NetShow Player is free, so you don't have to worry about the user not having the player.

When streaming audio and video over a network, bandwidth is your most valuable resource, and knowing how to deal with it is your top priority. In most cases, you want to deliver the richest multimedia content possible, while using the minimum network bandwidth.

Developing Live NetShow Content

Because you don't have to convert any of your source material to a different format, it's easy to create live NetShow content. Just plug a compact disc player, tape player, audio output from a VCR or laser disc player, or any other audio source into the input on your NetShow Real-Time Encoder server's sound card.

In the case of ABX Compute Corporation, you would feed the live audio from the keynote speaker's microphone into the NetShow Real-Time Encoder server's sound card. The speech will be multicast to all listeners.

As long as you can input an audio source to your sound card through your computer's bus or input jack, you can multicast that audio. (Be sure you aren't violating any copyright or ownership laws.)

When choosing an audio source, be sure that you have complete access to the entire source. Using a live source is fine, but do not use a streaming or multicast audio file as your input source.

Developing On-Demand NetShow Content

NetShow on-demand content is provided in .asf files. You can deliver multimedia content such as audio, illustrated audio, or video at various data rates, including 28.8 Kbps. For more information, see "Adjusting Your Content for the Bandwidth Using Codecs" later in this chapter.

You can do more with .asf files than just deliver audio and video content. Within an .asf file, while it is playing, you can open Web pages and deliver scripting commands to the user's computer to create an entire user experience.

Using .asf files, you can, for example, deliver a training video, a PowerPoint presentation, or audio with synchronized images. With the ability to embed scripting commands, you can extend the ability of the .asf files so they can invoke URLs, process feedback from the user, and execute client-side commands.

NetShow streams .asf files, as well as RealAudio and RealVideo. You can place practically any type of video, image, or audio into an .asf file and then include URL and script commands.

Obtaining Content

Your first step in creating an .asf file is to obtain the video, image, and/or audio content you want to use. When creating .asf files, always start with the highest-quality source files. Then, as network bandwidth dictates, you can reduce the quality of your source files. In the ABX Compute Corporation example, you can use any videotape that was shot at the release party. If anyone used PowerPoint to create presentation slides, you can use the presentation files, too. The following sections briefly describe how to turn this content into .asf files.

Video

Video is an excellent source, because the audio and visual elements are already combined. To create a video source, convert (digitize) any beta, VHS, 8mm, Hi-8, laser disc, or other video content into .avi or .mov format. Once this is done, you can use the NetShow tools to convert the video source into .asf files.

For our example, let's say you've got the Widget release party speeches and presentations on VHS videotape. The first thing you must do is convert the video format from VHS tape to digital .avi or .mov format.

To digitize your video content and place it on a hard disk or network share, use a video editing tool such as VidEdit, VidCap, or Adobe Premiere. For this conversion, you need a computer that, along with the proper software, also contains a high-resolution video card with audio and video inputs, such as the Truevision Targa 1000 Pro, Miro Video DC20, or Digital Processing Systems Perception Video Recorder.

If you have digital video content that is not in .avi or .mov format, use a video conversion tool, such as those mentioned, to convert the video into .avi or .mov format. If you want to create illustrated audio, all you have to do is make a copy of the video's audio track and then synchronize frame captures to it.

Images

You can use any image in the .bmp, .dib, .rle, .jpg, or .gif format to create .asf files. Begin with a source image from a digital camera, video frame capture, PowerPoint slide, scanned image, or any other digital art. Once the image is in a digital format, use any image-processing software to convert it to one of the appropriate formats.

If you want to use video frame captures as images in an .asf file, you need the software and hardware discussed in the "Video" section above. If you already have your video in .avi or .mov format, use a video-editing tool such as VidEdit or Adobe Premiere to capture frames.

If the video is not in a file on a shared or local drive, use a video player to play the video into your video capture board, and capture and save the video frames in a usable format to your hard disk.

If you use Microsoft PowerPoint to create presentations, you can easily convert these slides into the JPEG (.jpg) format, using the Internet Assistant for Microsoft PowerPoint add-on tool. For information on how to download and use the tool, see **http://www.microsoft.com/powerpoint/Internet/IA/default.htm**.

Audio

Only .wav files can be used in an .asf file. The .wav file can use any ACM codec; that is, any .wav file you can play on your computer can be used in an .asf file. (Your Windows 95 or Windows NT operating system contains these codecs.)

If you want to convert live audio, a compact disc, or any other external source of recorded audio into a .wav file, use sound-editing software, such as SoundRecorder or Sonic Foundry's Sound Forge, to record the audio through your sound card's input or line-in jack, and then save it as a .wav file.

URLs

You can include any URL in an .asf file. The NetShow Player does not check, locate, or open the URLs. When the NetShow Player comes to a URL in an .asf file, it sends the URL to the default browser, which then opens the URL. If the computer that is playing the .asf file does not contain a browser or for some reason cannot access the URL, the URL is ignored.

Script Commands

Script commands are client-side code that you include in the .asf file. The NetShow Player does not run or execute script commands. Instead, as the NetShow Player receives the .asf file, the script commands are passed to a client-side application that handles the commands and executes the actions.

Selecting a Bandwidth

After you have your source, you need to decide for what size network bandwidth you are going to create the .asf file. Bandwidth limits the size of an .asf file. You can use NetShow tools to create an .asf file for any size bandwidth, but you may not be able to stream it across your network.

There are three classes of bandwidths: Internet, intranet, and specialized networks. Each class, and the type of .asf file that will stream across it, is briefly described in the following sections. In order to stream an .asf file over a limited bandwidth such as 14.4 or 28.8 Kbps, you need to reduce or compress your source files.

Internet (14.4-Kbps and 28.8-Kbps Modems or 64-Kbps ISDN Modem)

Internet bandwidth is equivalent to what most phone lines carry. 14.4-Kbps and 28.8-Kbps modem speeds are used by the majority of Internet computers, while only a few have access to ISDN modems. With a 14.4-Kbps modem, you can receive streaming audio with only a few images (if the streaming file is long). With a 28.8-Kbps modem, you can receive streaming audio and illustrated audio. With a 64-Kbps ISDN modem, you can receive extremely high-quality streaming audio, illustrated audio (with many more images than with a 28.8-Kbps modem), and reduced-quality video.

Intranet (100 to 200 Kbps or 128-Kbps ISDN Modem)

Intranet bandwidth of 100 to 200 Kbps is typical among business networks. With an intranet or a 128-Kbps ISDN modem, you can typically stream compact disc–quality audio, illustrated audio (with compact disc–quality audio and lots of images), and medium-quality video.

Specialized Networks (1 to 6 Mbps)

Many organizations are using new technology to upgrade their local area networks to support increased bandwidths from 1 to 6 Mbps. With 1 to 6 Mbps of effective bandwidth, a network can easily stream true video, as well as compact disc–quality audio.

Building the ASF File

After you have chosen source files and a bandwidth, you are ready to build .asf files. Because the source files used to create an .asf file are different, while you are creating the .asf file you need to make sure that all source files that will make up an .asf file can fit within the designated bandwidth. However, you may find that once you've built the .asf file, you don't like it. If so, edit and adjust the source files, and build it again. As you build .asf files, you will see that there are three main types: audio, video, and illustrated audio. The type that you build depends on the NetShow tool that you use.

Selecting a Tool

When you installed the NetShow tools, you installed everything that you need to convert audio, video, and illustrated audio into .asf files. Each of the tools described in the following sections is used to convert a specific file type or combination of file types into an .asf file.

For more information about content development tools, see "NetShow Authoring and Production Tools" earlier in this chapter.

WavToAsf

Use WavToAsf to convert .wav files to .asf files. The size of the .asf file that WavToAsf creates is directly proportional to the audio source. If you use a compact disc–quality .wav file to create an .asf file, you will get a compact disc–quality .asf file. Even though this may be what you think you want, realize that you will be streaming this file over a network. Many bandwidths cannot support the streaming of compact disc–quality audio. To build an audio-only .asf file for your chosen bandwidth, build the .asf file, and then test it to see whether it can stream over the network bandwidth. If it cannot, you need to compress the audio source and rebuild the .asf file.

The WavToAsf command-line tool is not an editor. If you need to do any editing or if you want to compress the .wav file, you need to use a third-party tool such as SoundRecorder or Sound Forge 4.0a to edit the audio source and then rebuild the .asf file. Even though you must run WavToAsf from the command line, WavToAsf allows you to create a full .asf file complete with markers, URLs, and script commands.

To return to the ABX Compute release party example, you can use WavToAsf to turn the multicast audio into a richer .asf file. Using WavToAsf, you can add script commands or invoke URLs at points in the audio where the additional Web page or .asf file information add to the audio-only .asf file.

VidToAsf

VidToAsf is a command-line tool that converts .avi or .mov files into .asf files. As with WavToAsf, VidToAsf is not an authoring or editing tool, it only converts video files into .asf files. The size of the video file that you use as a source directly correlates to the size of the .asf file. If you need to edit or compress the video source in any way before converting it to an .asf file, use a video editor. Use VidToAsf to turn the videotape from the ABX Compute release party into an .asf file.

When you run the command to convert the video source to an .asf file, you can also specify standard .asf options such as including markers, URLs, or script commands.

ASF Editor

ASF Editor is the most robust of the .asf creation tools. With ASF Editor, you can combine and synchronize audio, images, and script commands, and then output these into an .asf file.

To help you create .asf files that are suited for limited bandwidths, ASF Editor allows you to specify the bandwidth and also furnishes you with tools for resizing and compressing the audio and images.

In ASF Editor, you design the .asf files by placing the audio and image source files in a time line. ASF Editor gives you a graphic representation of how the source files fit in the specified bandwidth. From this representation, you will be able to tell whether your files can be built into an .asf file, whether you have attempted to put too much image or audio information into the bandwidth, or whether you have room left in the bandwidth for more information.

Unlike VidToAsf or WavToAsf, ASF Editor allows you to specify a bandwidth, which sets a limit on the size of your .asf file. Based on your chosen limit, as you build your .asf file, you may be forced to edit or discard some of the source files in order to correctly build an .asf file. If you want to edit the content of your audio or image sources, you need to use a third-party tool. If the image or audio won't fit in the bandwidth, use the ASF Editor compression tools to force the source into the designated bandwidth. If you like how the audio and image files fit together, build and then view your .asf file.

In the ABX Compute scenario, you can use ASF Editor to combine portions of the release party multicast audio, audio clips from the videotapes, frames captured from the videotapes, scanned pictures taken at the party, and slides taken from presentation files into an .asf file. You can determine the bandwidth that the .asf file will stream over.

NetShow Presenter

NetShow Presenter is a PowerPoint add-in tool that lets you synchronize a PowerPoint presentation into an ASF stream. You can use NetShow Presenter to output a PowerPoint presentation directly to an .asf file that's ready to stream. NetShow Presenter enables PowerPoint to:

- Connect with the Real-Time Encoder.
- Send URL script commands to the Real-Time Encoder.
- Export presentation slides.
- Create and control a PowerPoint presentation that remote users can watch over the network on their own desktop.

For example, suppose a company's CEO is giving a presentation to the heads of several companies. Because of time constraints and other commitments, many employees who would like to attend the presentation cannot. With NetShow and NetShow Presenter, the CEO can provide a live video feed of himself speaking along with synchronized images of his presentation slides. NetShow Server delivers the live video feed, and NetShow Presenter allows the CEO to synchronize the delivery of the presentation slides to specific points in the live video feed. Employees who wanted to watch the presentation can download the presentation Web page and receive the presentation as if they were in same room as the presenter.

ASFChop

ASFChop is a simple .asf file editor. Use ASFChop to delete pieces of an .asf file. ASFChop also makes an .asf file seekable; that is, if an .asf file is longer than 10 seconds in length, ASFChop creates an index for that file so that you can seek (similar to fast forwarding) through the file. This is a useful feature, because the Real-Time Encoder does not create seekable .asf files. You can use ASFChop to create an index in an .asf file without cutting any information.

Adjusting Your Content for the Bandwidth Using Codecs

What makes NetShow such an effective tool for creating streaming network multimedia files is its capability to adjust the source content files. NetShow is codec-independent. Other technologies use a proprietary codec, which means the codec can be used only by that one application, and other codecs cannot be applied.

NetShow lets the .asf author and end user use a well-rounded suite of bundled and stand-alone third-party codecs for voice, music, images, and video for various bandwidths. This results in a greater number of options, more flexibility, higher-quality content, and because the codecs can be upgraded independent of NetShow, faster innovation.

After choosing a bandwidth, reduce the size of your source content. You can reduce the size of your source content two ways: adjust the size (height and width) of the source of image or video files, or apply a codec. Most of the time you will want to apply a codec.

Adjusting Image Size

To adjust the size of an image or video file, use an image editor (Adobe Photoshop, CorelDRAW, or Microsoft Imager) or a video editor (AVIEdit, Adobe Premiere). You cannot adjust the size of an audio file without changing the content.

Using Codecs

When you install NetShow, you also install a variety of codecs designed specifically for image, audio, and video. These codecs reside on your computer and are available to the various editing tools that you will use to edit your audio, video, and image sources. Most codecs are designed for a specific purpose.

For example, VDONet VDOWave is a good video codec, but it is not good for compressing images. It's better to use JPEG compression for images. Before you build your .asf file, experiment with the various codecs to reduce the size of your source files.

Because the codecs are installed on your machine, you can use the codecs in any application that supports codecs. For example, to compress an .avi file, you can open Adobe Premiere on your computer and apply the VDONet VDOWave codec to the .avi file. Of the three NetShow tools (VidToAsf, WavToAsf, and ASF Editor) that are installed on your computer, only ASF Editor allows you to associate codecs with source files. You must use video or audio editing tools to associate codecs with the source files before using VidToAsf or WavToAsf to create .asf files.

Once you have the video content in .avi or .mov format, compress the files before you convert them to .asf. Because most compression algorithms are specifically designed for audio or video, you want to compress both the audio and video portions of your .avi file. Use tools such as Sonic Foundry's Sound Forge or Adobe Premiere to apply compression algorithms to the .avi.

NetShow supplies you with the codecs that you will want to use. For example, if you are creating a video .asf file of the ABX Compute release party for use over the Internet (28.8-Kbps bandwidth), you'll want to use the VDOWave codec for the video portion and the L&H codec for the audio portion. If you're creating a video .asf file for use over the ABX Compute internal network (100-Kbps bandwidth), you may want to use the Indeo 4.1 codec for the video and the Truespeech codec for the audio.

For more information about the codecs that are included with NetShow, see **http://www.microsoft.com/netshow/codecs.htm**.

Streaming the ASF File

After creating an .asf file, you will want to play it. You can double-click the .asf file to play it locally, but how do you enable others to stream the .asf file to their desktops? First, place the .asf file on a NetShow Server, and then create a way for others to access the .asf file.

Posting ASF Files on the NetShow Server

To post .asf files on a NetShow Server, simply copy the .asf files to the NetShow publishing point on that server.

Posting ASX Files on the HTTP Server

For users to play an .asf file on a Web page, the Web page must contain an .asx file (similar to a hypertext link that points to the .asf file), or the NetShow Player must be embedded as an OBJECT element in the Web page.

Active Stream Redirector (.asx) files are necessary for playing .asf files from a Web page that does not contain an embedded NetShow Player. An .asx file provides the information that the NetShow Player needs to connect to a server. An .asx file is a one-line text file whose contents look like this:

```
ASF protocol://servername/virtualdir/filename.asf
```

An .asx files specifies:

- The protocol used (either HTTP or MMS)
- The name of the server
- A virtual directory (if necessary)
- The .asf file

Only one .asf file can be specified per .asx file. Place your .asx files in a directory on an HTTP server. It is a good idea to create a virtual directory for the .asx files so that it is easy to create, remember, and possibly edit the path to these files.

For more information, see "NetShow Announcements" earlier in this chapter.

Using the NetShow Player

You can use the NetShow Player to listen and watch streaming multimedia programs. The NetShow Player can be used to play programs three ways:

- As a stand-alone player.
- Launched from the browser, using a link on a Web page.
- Embedded as an ActiveX control in a Web page, Visual Basic script application, or other ActiveX container applications.

You can also configure the NetShow Player for a variety of options. For more information, see Help for NetShow Player.

Using the Stand-Alone Player

When you use NetShow Player as a stand-alone application, you decide how you want to access the .asf files. If you play an .asf file locally, you are not streaming the file. NetShow Player reads the .asf file from the hard disk and then plays it. If you play an .asf file from a remote location, it streams to your computer.

When you play an .asf file, you must specify a protocol (either HTTP or MMS), a server, and an .asf or .asx file. The following instructions explain how to run NetShow Player and use it to play .asf files locally and from remote locations.

▶ **To open NetShow Player**

1. On the **Start** menu, select **Run**.

   The **Run** dialog box opens.
2. In the **Open** box, type **nsplayer.exe**.
3. Click **OK**.

▶ **To open a local .asf file**

1. On the **File** menu, select **Open File**.

   The **Open** dialog box opens.

2. Type the file name, or browse and select the .asf file you want.
3. Click **OK**.

   The NetShow Player buffers the .asf file and then begins playing it.

▶ **To open an .asf file in a remote location**

1. On the **File** menu, select **Open Location**.

   The **Open Location** dialog box opens.

2. Enter the URL of the .asf or .asx file; for example:

   ```
   mms://XYZcompany.com/products/widgets.asx
   ```

3. Click **OK**.

   The NetShow Player buffers the .asf file and then begins playing it.

Launching the Player from a Web Page

To enable launching NetShow Player from a Web page, create a link in the HTML code as shown in the following example:

```
<A HREF=mms://server/file.asx>
```

When users click on the link from their browsers, the NetShow Player is launched to play the ASF stream. For more information about the process used to launch the NetShow Player and play the ASF stream, see "NetShow Announcements" earlier in this chapter.

Embedding NetShow Player in a Web Page

An easy way of providing the NetShow Player to people who don't already have it on their computers is to embed a NetShow ActiveX control (Nsplay.ocx) as an OBJECT element in a Web page. When a user accesses the page, the player is downloaded to the user's computer and used as an embedded control in the browser window. In the script commands that you use to embed the player, you can set the property that identifies the .asf file to play as well as how to play it. You can even set one of the player's properties to check the version of the player on the computer. If the player is outdated, the computer downloads the newest version of the player.

The Netshow player control is available on Microsoft's Web site, packaged as a .cab file (**http://www.microsoft.com/netshow/download/en/nsasfinf.cab**). The .cab file contains the .ocx file for the control, along with the codecs needed to support the control. An .inf file included in the cabinet specifies the files that need to be downloaded and performs the setup for the control to run.

The following example shows the OBJECT element used to embed the NetShow ActiveX control. This example would embed the control and show .asf content in the browser window for either Internet Explorer or Netscape Navigator.

```
<OBJECT ID="NSPlay"
    WIDTH=160 HEIGHT=128
    CLASSID="CLSID:2179C5D3-EBFF-11CF-B6FD-00AA00B4E220"

    <!--Code to go get NetShow player-->

    CODEBASE="http://www.microsoft.com/netshow/download/en/nsasfinf.cab#
        Version=2,0,0,912">

    <PARAM NAME="FileName" VALUE="http://server/path/myvideo.asx">

    <EMBED type="video/x-ms-asf-plugin"
        Src="http://server/path/myvideo.asx"
        FileName="http://server/path/myvideo.asx"

    PluginsPage="http://www.microsoft.com/netshow/download/player.htm"
        ControlType=1
        Width=290 height=250>
    </EMBED>
</OBJECT>
```

Use of the CODEBASE attribute is recommended, because it contains a URL pointing to where the Nsplayer.ocx control can be found if it is unavailable on a user's system. Besides the address of the player object, the CODEBASE attribute may also optionally specify a version number of the control.

How you set the parameters determines how the NetShow Player will work. For example, ControlType=1 specifies that the embedded control appears in the browser window with a full set of NetShow controls. The FileName parameter identifies the .asf file that will play. You set the value for this parameter to be a URL just as if you were going to play the .asf file from the **Open Location** option on the NetShow Player **File** menu.

When a user who doesn't have NetShow installed accesses the Web page, the player control downloads and is used to embed the player in the browser window and to play the .asf file (as specified by Myvideo.asx in the previous example).

For users who have downloaded and installed NetShow, the player control, Nsplay.ocx, is already located in the C:\Program Files\Microsoft NetShow\Player folder. Internet Explorer uses the installed version of the control unless a newer version is available. If a newer version of the embedded control exists, as indicated by the CODEBASE Version attribute, Internet Explorer installs and uses the newer version.

For more information about the NetShow ActiveX control and its properties, see the *Microsoft NetShow Client SDK*.

CHAPTER 20

NetMeeting

In this chapter

NetMeeting Overview 329
- Internet Phone/H.323 Standards–Based Audio Support 330
- H.323 Standards–Based Video Conferencing 330
- Intelligent Audio/Video Stream Control 331
- Multipoint Data Conferencing 331
- The Internet Locator Server 332
- LDAP Standard Support 333
- Support for Windows NT Workstation 4.0 333
- System Policies 333
- User Interface Enhancements 333
- NetMeeting Mail Extension 334
- The Software Development Kit 335
- Compliance with Communications and Conferencing Standards 335

What's New in NetMeeting 2.1 336
NetMeeting Application Scenarios 337
NetMeeting Architecture Components 339
- Transport Layer 340
- Core Standard and Application Components 340
- User Interface 342
- ActiveX Conferencing Platform 342

Understanding Conferencing Standards 343
- Standards Organizations 344
  - International Telecommunications Union 344
  - Internet Engineering Task Force 344
  - International Multimedia Teleconferencing Consortium 344

The T.120 Standard 344
Benefits 345
Interoperability 345
T.120 Topologies 345
How NetMeeting Uses the T.120 Standard 347
The H.323 Standard 348
Benefits 348
Interoperability 349
How NetMeeting Uses the H.323 Standard 349
The Lightweight Directory Access Protocol 350
LDAP Version 2 and Version 3 350
Benefits 350
Interoperability 351
LDAP Topology 351
How NetMeeting Uses the LDAP Standard 352
Understanding Product Interoperability 353
Interoperability: The Big Picture 354
Compatible Products and Services 355
T.120 Conferencing Products and Services 355
H.323 Conferencing Products and Services 356
LDAP Conferencing Servers 357
Directory Servers 357
The Internet Locator Server 358
For More Information 359
Firewall Configuration 359
Components of a Secured System 359
NetMeeting and Firewalls 360
Establishing a NetMeeting Connection with a Firewall 361
Firewall Limitations 362
Security and Policy Concerns 362
System Policies and Restrictions 363
Issues to Consider Before Setting Policies 363
Customizing System Policies and Restrictions 363
NetMeeting System Policies and Restrictions Options 365
Common System Policies and Restrictions Scenarios 368
Using NetMeeting on Intranet Web Pages 368
The NetMeeting ActiveX Control 369
Intranet Scenarios 370

NetMeeting Overview

Microsoft NetMeeting conferencing software helps small and large organizations take full advantage of the global reach of the Internet or corporate intranet for real-time communications and collaboration. Connecting to other NetMeeting users is made easy with the Microsoft Internet Locator Server (ILS), which enables participants to call each other from a dynamic directory within NetMeeting or from a Web page. While connected on the Internet or corporate intranet, participants can communicate with both audio and video, work together on virtually any Windows-based application, exchange or mark up graphics on an electronic whiteboard, transfer files, or use the text-based chat program.

NetMeeting also provides an open development environment that supports international communications and conferencing standards and enables interoperability with products and services from multiple vendors.

NetMeeting 2.1 is fully compatible with NetMeeting 1.0 and 2.0, and with applications and solutions that use the *Microsoft NetMeeting Software Development Kit (SDK)* for the Windows 95 operating system. With its first release, NetMeeting 1.0 transformed the everyday telephone call into a richer and more effective tool. For the first time, people could use voice communication to interact and collaborate over the Internet using multipoint data conferencing capabilities based on the International Telecommunications Union (ITU) T.120 standard. NetMeeting 2.0 integrated a number of new features, as well as improving functionality and enhancing the user interface.

The following key features distinguish NetMeeting 2.1:

- Internet phone/H.323 standards–based audio support
- H.323 standards–based video conferencing
- Intelligent audio/video stream control
- Multipoint data conferencing
- The Internet Locator Server (ILS)
- Lightweight Directory Access Protocol (LDAP) standard support
- Support for Windows NT Workstation 4.0
- System policies
- User interface enhancements
- NetMeeting Mail extension
- The Software Development Kit
- Compliance with communications and conferencing standards

Internet Phone/H.323 Standards–Based Audio Support

Real-time, point-to-point audio conferencing over the Internet or corporate intranet enables you to make voice calls to associates and organizations around the world. NetMeeting audio conferencing offers many features, including half-duplex and full-duplex audio support for real-time conversations, automatic microphone-sensitivity-level setting to ensure that meeting participants hear each other clearly, and microphone muting, which lets users control the audio signal sent during a call. This audio conferencing supports network TCP/IP connections.

Support for the H.323 protocol enables interoperability between NetMeeting and other H.323-compatible audio clients. The H.323 protocol supports the ITU G.711 and G.723 audio standards and Internet Engineering Task Force (IETF) real-time protocol (RTP) and real-time control protocol (RTCP) specifications for controlling audio flow to improve voice quality. On MMX-enabled computers, NetMeeting uses the MMX-enabled audio codecs to improve performance for audio compression and decompression algorithms. This results in lower CPU use and improved audio quality during a call.

H.323 Standards–Based Video Conferencing

With NetMeeting, a user can send and receive real-time visual images with another conference participant using any video for Windows-compatible equipment. Participants can share ideas and information face to face, and use the camera to instantly view items, such as hardware or devices, that the user chooses to display in front of the lens. Combined with the audio and data capabilities of NetMeeting, a user can both see and hear the other conference participant, as well as share information and applications. This H.323 standard–based video technology is also complaint with the H.261 and H.263 video codecs.

NetMeeting video conferencing includes the following features:

- Participants can switch audio and video to another person during a meeting. This makes it easy for users to communicate with many different people.
- During a meeting, participants can remotely adjust the video image quality, balancing the need for higher quality or faster performance.
- Users can dynamically change the size of the video window to reduce or enlarge the image being sent to another person.
- In the NetMeeting main window, the video Preview and Receive windows are integrated on the Current Call window. Users can view these video windows from Current Call, or drag them to a different location on the desktop.
- Users can choose whether or not to transmit video immediately when a call starts. Also, they can pause or resume sending or receiving video by pressing a button in the video window frame.

- NetMeeting automatically balances the performance and quality of video during a meeting based on the speed of the network connection, thus providing the highest-quality, lowest-bandwidth video capabilities.
- Administrators can control access to video features using NetMeeting system policies. For more information about setting system policies, see "System Policies" later in this chapter.
- On MMX-enabled computers, NetMeeting uses the MMX-enabled video codecs to improve performance for video compression and decompression algorithms.
- Support for H.323 conference servers and gateways (currently being developed by leading vendors) enables NetMeeting users to participate in meetings with multiple audio and video connections.

Intelligent Audio/Video Stream Control

NetMeeting provides intelligent control of the audio and video stream, which automatically balances the load for network bandwidth, CPU use, and memory use. This intelligent stream control ensures that audio, video, and data are prioritized properly, so NetMeeting maintains high-quality audio while transmitting and receiving video and data during a call. Through system policy features, IS organizations can configure the stream control services to limit the bandwidth used for audio and video on a per-client basis during a meeting. For more information about limiting bandwidth usage, see Chapter 10, "Network Bandwidth Considerations," in the *Microsoft NetMeeting 2.1 Resource Kit.*

Multipoint Data Conferencing

Two or more users can communicate and collaborate as a group in real time. Participants can share applications, exchange information through a shared clipboard, transfer files, collaborate on a shared whiteboard, and use a text-based chat feature. Also, support for the T.120 data conferencing standard enables interoperability with other T.120-based products and services.

The following features make up multipoint data conferencing:

- **Application sharing.** A user can share a program running on one computer with other participants in the conference. Participants can review the same data or information in the program and see the actions as the person sharing the application works on the program (for example, edits content or scrolls through information). Participants can share Windows-based applications transparently, without any special knowledge of the application capabilities.

  The person sharing the application can choose to collaborate with other conference participants, and they can take turns editing or controlling the application. Only the person sharing the program needs to have the given application installed on his or her computer.

- **Shared clipboard.** A user can exchange the contents of a shared Clipboard with other participants in a conference, using familiar cut, copy, and paste operations. For example, a participant can copy information from a local document and paste the contents into a shared application as part of a group collaboration.
- **File transfer.** With the file transfer capability, a user can send a file in the background to one or all of the conference participants. When one user drags a file into the main window, the file is automatically sent to each person in the conference, who can then accept or decline receipt. This file transfer capability is fully compliant with the T.127 standard.
- **Whiteboard.** Multiple users can simultaneously collaborate using the whiteboard to review, create, and update graphic information. The whiteboard is object-oriented (versus pixel-oriented), enabling participants to manipulate the contents by clicking and dragging with the mouse. In addition, they can use a remote pointer or highlighting tool to point out specific contents or sections of shared pages.
- **Chat.** A user can type text messages to share common ideas or topics with other conference participants, or record meeting notes and action items as part of a collaborative process. Also, participants in a conference can use chat capabilities to communicate in the absence of audio support. A new "whisper" feature lets a user have a separate, private conversation with another person during a group chat session.

The Internet Locator Server

Replacing the NetMeeting 1.0 User Location Service (ULS), the ILS for NetMeeting 2.1 expands existing server technology to provide more advanced directory services, higher scalability, and better performance standards (LDAP). The ILS enables users to locate each other for conferencing. Users can view the ILS directory from within NetMeeting or a Web page and review a list of people currently running NetMeeting. Then, they can choose to connect to one or more of the listed users or select another user by entering the user's location information. For more information about the ILS, see the Web site at **http://www.backoffice.microsoft.com/**.

NetMeeting can detect whether a server is available and automatically attempt to log on in the background, without user intervention. If a user disconnects and then logs on again later, NetMeeting automatically logs on to the specified ILS.

LDAP Standard Support

The Lightweight Directory Access Protocol is an Internet standard that defines the protocol for directory access. NetMeeting uses LDAP to access the ILS and perform server transactions, including logging on and off, creating a directory listing of all available users, and resolving a particular user's address information, such as the Internet Protocol (IP) address. LDAP facilitates interoperability and allows organizations to implement compatible servers.

Support for Windows NT Workstation 4.0

Users of computers running the Windows NT operating system can communicate and collaborate with each other and with users of NetMeeting on Windows 95. The functionality of NetMeeting, including audio, video, and multiuser data conferencing for electronic whiteboard, text-based chat, and file transfer, is supported for Windows NT. Windows NT Service Pack 3 is required for a Windows NT user to share applications.

System Policies

Administrators can implement NetMeeting system policies to control user and computer privileges. Using system policies, they can predefine settings and restrictions, such as preventing the use of audio features and provide standard configurations for their user community. New system policies include the ability to limit the network bandwidth for audio and video streams. For more information about setting system policies, see "System Policies" later in this chapter.

User Interface Enhancements

One of the goals of NetMeeting is to enhance the existing user interface. Functions are easier to locate, view, and use. Enhancements focus on these areas:

- Call windows within the NetMeeting main window make it easy to connect to other users and participate in the current call. Four windows let users see a variety of information, such as the directory of all people currently logged onto the directory server and their audio/video capabilities, SpeedDial entries and their status, the users participating in the current call, or a history log of all received calls.
- Users can filter the directory entries to more easily find and connect with people. For example, users can filter the entries to identify only people currently participating in a call or only people with audio and video capabilities. Also, users can choose one of three user categories—personal, business, or adults-only—as an additional filter to show only people who selected the same user category.

- To more fully integrate with Internet Explorer, NetMeeting now includes a new **Go** menu for accessing Web-based directory information, as well as Internet Mail and News. Users can connect to other people from a Web-based directory view by entering the user's e-mail address, specifying the user's IP address, or entering the user's machine name.
- Refined NetMeeting Options tabs and wizards make it easier to set up and configure the NetMeeting environment. A new **Calling** tab lets a user choose directory and SpeedDial options. Additionally, an H.323 gateway calling option lets a NetMeeting user connect to a person by using a telephone number.
- A graphical interface similar to the one in Internet Explorer, including a common toolbar, makes it easier to move between applications that are part of the Internet Explorer suite. The toolbar is context-sensitive, displaying the most appropriate options based on the active window.
- The host computer lets the meeting originator hang up on one or more meeting participants, so people can be removed from the call more easily within conference groups.
- E-mail messaging gives users the option of sending e-mail to people who are not available for conferencing. NetMeeting uses MAPI to start a mail client of choice, automatically adds the subject information, and then includes a SpeedDial shortcut so that a person can easily call back later.

NetMeeting Mail Extension

NetMeeting includes a mail extension that works with Outlook and Exchange mail clients, which enables a user to place a call directly from a menu in the mail client based on entries in the mail address book. This feature gives people the flexibility to use their e-mail client to send a mail message or invoke a real-time meeting from the same mail address book. A NetMeeting system policy enables system administrators to specify the Exchange attribute for the NetMeeting address. For more information about setting this system policy, see "System Policies" later in this chapter.

The Software Development Kit

The *Microsoft NetMeeting 2.1 SDK* enables developers to integrate NetMeeting conferencing functionality directly into their applications or Web pages. Developers can extend the functionality of NetMeeting to add capabilities, such as a business card exchange, or to add conferencing functions to existing applications.

In addition, the *Microsoft NetMeeting 2.1 SDK* includes an ActiveX control for conferencing, allowing Web-site creators to integrate this functionality directly into their pages. Also, they can integrate the ActiveX control for conferencing with other ActiveX scripting solutions, such as JavaScript and Visual Basic Scripting Edition.

The *Microsoft NetMeeting 2.1 SDK* is available for download from the Web site at **http://www.microsoft.com/netmeeting/**.

Compliance with Communications and Conferencing Standards

NetMeeting is based on standards from the ITU, the same group that set standards for modems and the global telephone system. As a result, NetMeeting users can communicate with people using products on different platforms and from different vendors. In contrast, some Internet products are proprietary, and they leave their users stranded and able to talk only to people with the same product.

Broad support of standards ensures interoperability among solutions from different vendors. For example, the telephone industry uses a set of international standards to ensure that a person making a telephone call can connect to another person, regardless of what manufacturer produced either phone handset. Support for audio, video, and data conferencing standards provides the same transparent interoperability that customers expect.

NetMeeting supports the following industry standards that the ITU or IETF have ratified or proposed:

- **T.120.** ITU set of protocols for transport-independent, multipoint data conferencing. This standard also integrates with H.320 and H.324 protocols.
- **H.323.** ITU set of protocols for audio, video, and data conferencing over TCP/IP networks. This standard includes RTP and RTCP, and integrates with T.120, H.261, H.263, G.711, and G.723 protocols.
- **RTP/RTCP.** IETF packet format for sending real-time information across the Internet.
- **LDAP.** IETF set of protocols for directory access.

For more information about the standards that NetMeeting supports, see "Conferencing Standards" later in this chapter.

What's New in NetMeeting 2.1

As the leading Internet conferencing solution, NetMeeting has become the key building block for vendors of conferencing products and services. NetMeeting 2.1 is designed to support new technology featured in Windows 98, including DirectX® 5, universal serial bus (USB) video cameras, and the new video device driver model. NetMeeting 2.1 is packaged as part of Windows 98, but it is also designed to run as a stand-alone product on computers running Windows 95 and Windows NT 4.0 with Service Pack 3.

NetMeeting 2.1 includes the following enhancements:

- **Enhanced interoperability with H.323 devices.** Changes to the NetMeeting 2.1 infrastructure improve interoperability with new H.323 devices, including gateways and Multipoint Conferencing Units (MCUs). These changes include the ability to initiate a call using the H.323 calling model instead of the T.120 calling model.
- **Support for Windows 98 application sharing.** NetMeeting 2.1 enhances application-sharing functions to ensure support for Windows 98 device driver model changes and special multiple monitor support.
- **DirectX 5 support.** DirectX is a set of technologies that enable faster access to hardware in Windows. DirectX 5 is the latest version of DirectX. Installing the DirectSound® API component of DirectX 5 on your computer (with a compatible audio device) significantly reduces the latency of sending or receiving audio over the Internet using NetMeeting.

  DirectSound modifies the multimedia APIs in Windows 95, replacing your existing sound card driver with a new DirectX driver. The new DirectX driver supports DirectSound record and playback APIs (and also supports the existing driver functions). Users can install the DirectSound component of DirectX 5 from the NetMeeting Web site at **http://www.microsoft.com/netmeeting/**.

  Because DirectX drivers are often installed with games, many users may have DirectX capabilities already. Users must upgrade to DirectX 5, because older versions of DirectX do not support NetMeeting 2.1. To identify your DirectX version, double-click the **DirectX** icon in Control Panel.

  Note Some DirectX sound drivers do not support full-duplex audio. When you upgrade your existing driver to a DirectSound driver, you may lose this capability. Removing DirectX restores your original configuration.

- **Outlook bar.** A new Outlook bar gives NetMeeting an appearance and functionality consistent with Microsoft Outlook 97 and Outlook Express. This Outlook bar provides easy access to frequently used NetMeeting functions, including Directory, SpeedDial, Current Call, and History.

NetMeeting Application Scenarios

NetMeeting is changing the way people communicate and collaborate on the Internet and corporate intranets by expanding the situations in which the PC is used. With its rich set of audio, video, and data conferencing features, this product has proven its effectiveness for management groups, developers, and user communities worldwide.

The following table describes several scenarios that demonstrate how you can use NetMeeting to work and communicate more efficiently.

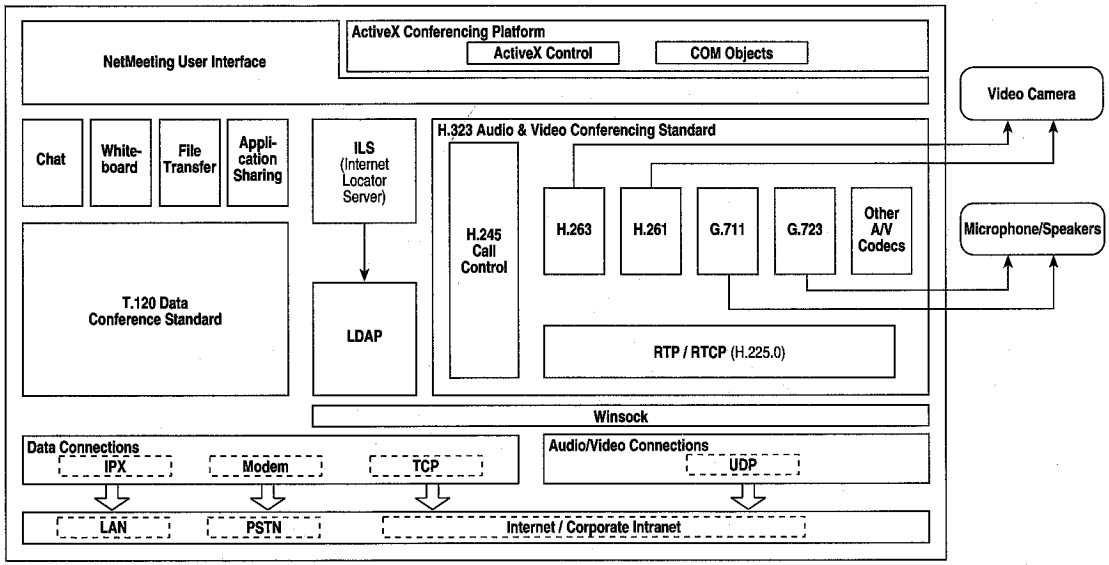
In the past	Today, with NetMeeting
Virtual Meetings	
A problem with a part on a manufacturing line could have taken several days to resolve. An engineer might have needed to fly to the factory, inspect the part, and diagnose the problem.	*Participants can conference remotely or from different locations and conduct meetings as if everyone were in the same room.* Engineers and factory representatives can hold a conference without traveling to another location. They can view the part on camera, share engineering specifications, and discuss the problem. If an administrator disables audio features using NetMeeting system policies, other support engineers can participate in the conference using the company's existing PBX phone system for audio support.
Document Collaboration	
Negotiating a contract might have taken days, weeks, or even months, as companies faxed legal documents back and forth with their changes. Valuable time was spent as accounting, legal, and management representatives each provided their input to the negotiation.	*Two or more participants can work together and collaborate on documents or information in real time.* With application sharing, accounting, legal, and management representatives can instantly collaborate on the contract. Each person can provide input, update text, or graphics, and discuss contract details.
Customer Service	
If a customer wanted more information about a Web site but had only one telephone line, the customer would have to close the Internet connection before dialing.	*From a Web site, customer service representatives can communicate directly with their user community and share important information.* The ActiveX control enables developers to add conferencing directly to Web pages. With the click of a button, a customer can reach a service representative in person. The representative can share information about products and services and promote sales opportunities.

In the past	Today, with NetMeeting
Telecommuting	
On business travel to conferences and client sites, corporate managers who needed to convey vital information to their teams would have relied on telephone calls, faxes, and e-mail.	*While on the road or in remote locations, data conferencing capabilities enable users to extend beyond voice or e-mail.* A manager can participate in a virtual meeting, give a presentation, or share applications without being in the office. For example, a manager can remotely share a PowerPoint presentation. Because NetMeeting interoperates with other T.120 products, a manager can interact with team members at worldwide locations using existing T.120 MCUs.
Distance Learning	
For specialized training, an employee might have to travel to a corporate learning center or other distant location. Companies incurred the cost of traveling, training, and staff hours.	*Corporations can disseminate presentations or educational information over the Internet or intranets.* Students can participate in an interactive learning session. A teacher can present the class and then hold discussion groups or interact with individual students. Training can incorporate video instruction for complex processes and application sharing for additional class demonstration. To facilitate large conference groups, NetMeeting can also interoperate with standards-based conferencing bridges.
Technical Support	
A help-desk representative might have spent hours with a customer attempting to understand and diagnose a problem. The customer would have explained the symptoms, and the representative would have to duplicate the customer's scenario without actually "seeing" the problem.	*Support organizations can view a situation on a remote computer and correct a problem during the support call without traveling to a remote location.* Help-desk personnel can use the application-sharing features of NetMeeting to remotely observe a user's computer and take corrective actions in real time.

NetMeeting Architecture Components

NetMeeting is both a client and a platform. The NetMeeting client enables users to experience the benefits of a real-time, multipoint communication and collaboration program. The NetMeeting platform enables third-party vendors to integrate conferencing features into their own products and services. To support this dual purpose, Microsoft implemented NetMeeting capabilities using an open architecture of interworking components. Each component communicates with and passes data to and from the component layer above and below it. This open architecture means that vendors can easily develop products and services that build on the NetMeeting platform and interoperate with NetMeeting client conferencing features.

At the core of the NetMeeting architecture is a series of audio, video, and data conferencing and directory service standards. These standards work together with transport, application, user interface, and ActiveX conferencing components to form the NetMeeting architecture, shown in the following diagram.

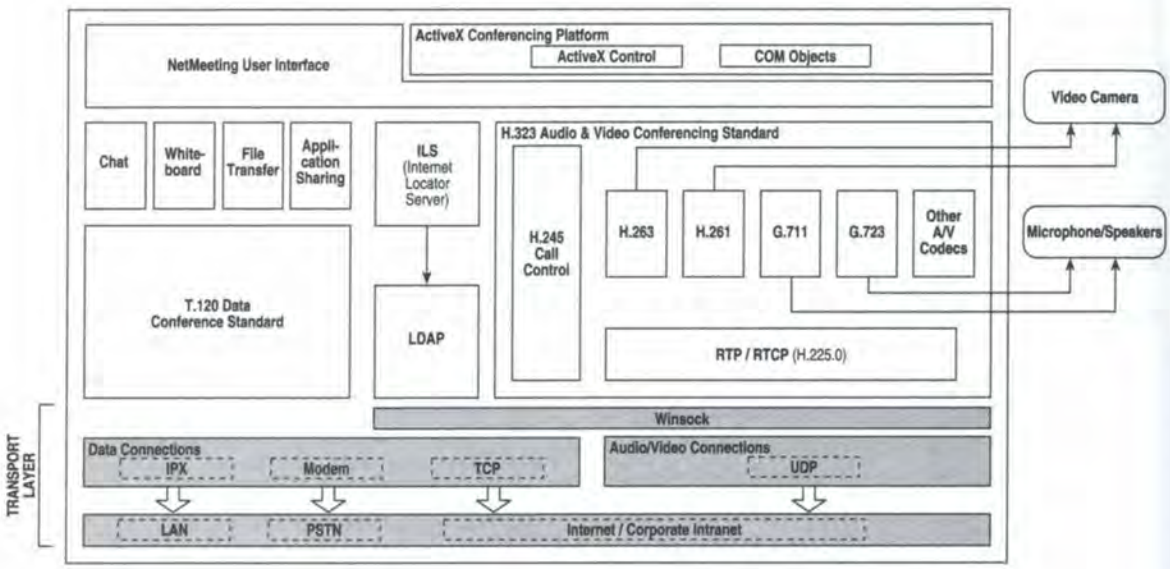


The following sections describe the key components that compose the NetMeeting architecture, beginning from the bottom of the diagram.

Transport Layer

At its lowest level, standards are responsible for translating, sending, and receiving NetMeeting information. The NetMeeting architecture includes protocols for null modem and network TCP/IP connections. The null modem protocol supports data-only conferencing connections; only TCP/IP connections support NetMeeting audio and video.

Information can be transported over the Internet or corporate intranet using transport control protocol (TCP) and user datagram protocol (UDP) connections. TCP is used primarily for data transport and call control, while UDP represents secondary connections for sending and receiving NetMeeting audio and video. Winsock provides the interface to the network stack and maps information between the applications and the network. The following diagram highlights this transport layer.

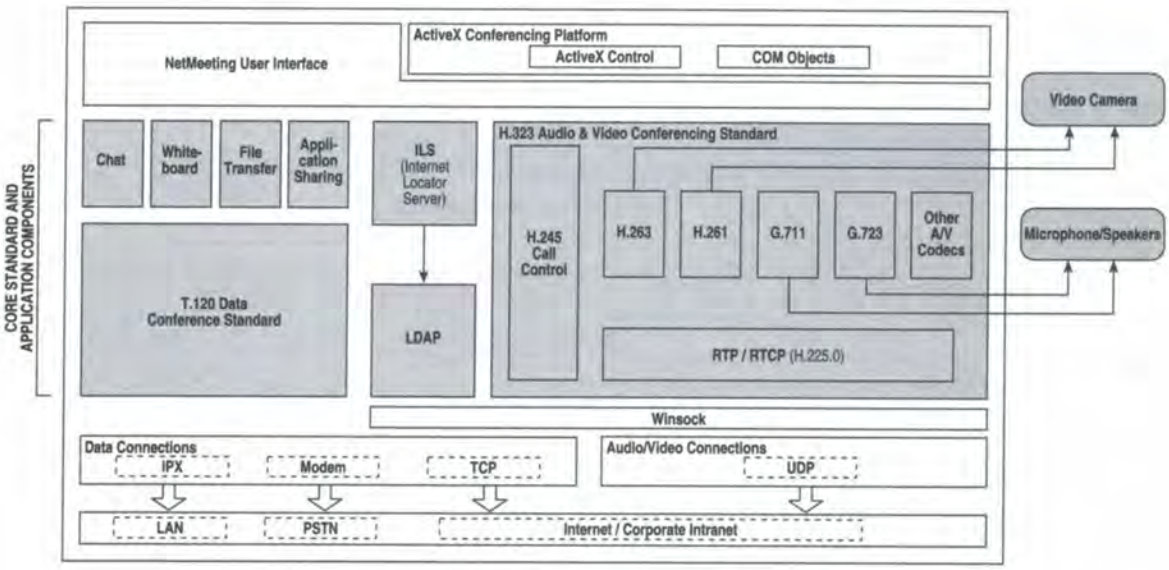


Core Standard and Application Components

The NetMeeting architecture is based on the following industry standards:

- The ITU T.120 standard for data conferencing
- The ITU H.323 standard for audio and video conferencing
- The IETF LDAP standard for directory services support

These standards provide the framework for managing NetMeeting connections, data conferencing, audio and video capabilities, and directory server access, highlighted in the following diagram. By supporting industry standards, NetMeeting enables audio, video, and multipoint data conferencing with standards-based products from other vendors.



Each of these standards is summarized in the following paragraphs. For more detailed information, see "Conferencing Standards" later in this chapter.

T.120. NetMeeting contains a wealth of collaborative data capabilities, including chat, whiteboard, file transfer, and application sharing. At the heart of these capabilities is the ITU T.120 standard for multipoint data conferencing. T.120 provides the protocols for establishing and managing NetMeeting data flow, connections, and conferences.

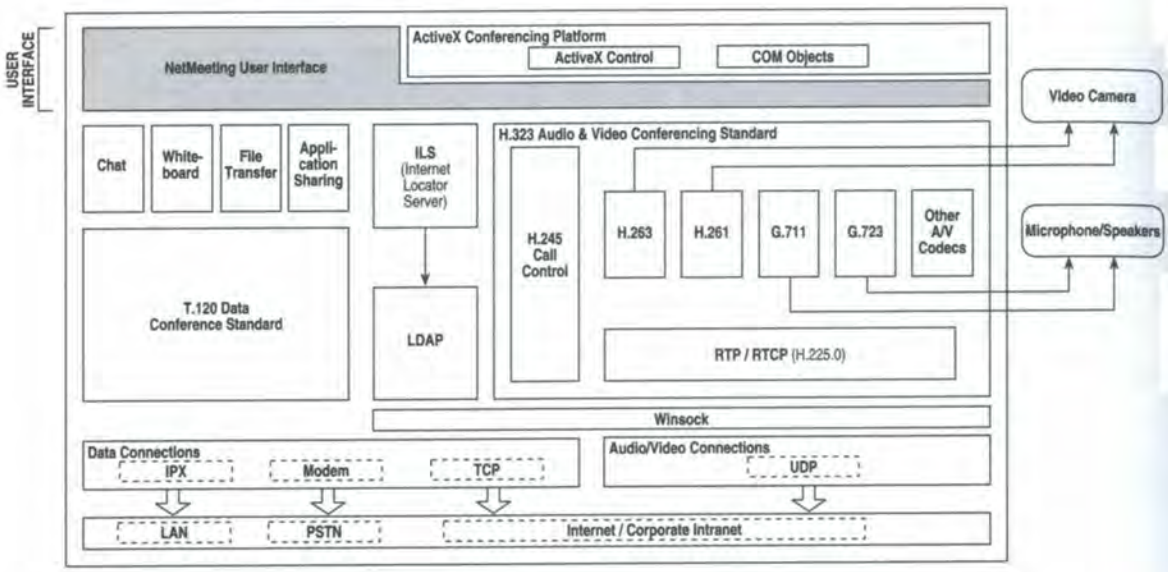
H.323. NetMeeting includes audio and video codecs, as well as framing and call control protocols. H.323 codecs provide the format for audio and video that is transmitted over various connection rates. NetMeeting supports a suite of codecs appropriate for many different modes of Internet telephony.

ITU H.323. These protocols enable NetMeeting to send and receive audio and video information between NetMeeting and H.323-compatible nodes. A microphone, speakers, and video camera provide the necessary input/output devices for these audio and video capabilities.

LDAP. Directory services for NetMeeting use the LDAP standard. Internet Locator Servers use the LDAP interface to create directories of current NetMeeting users that people can call and communicate with over TCP/IP connections.

User Interface

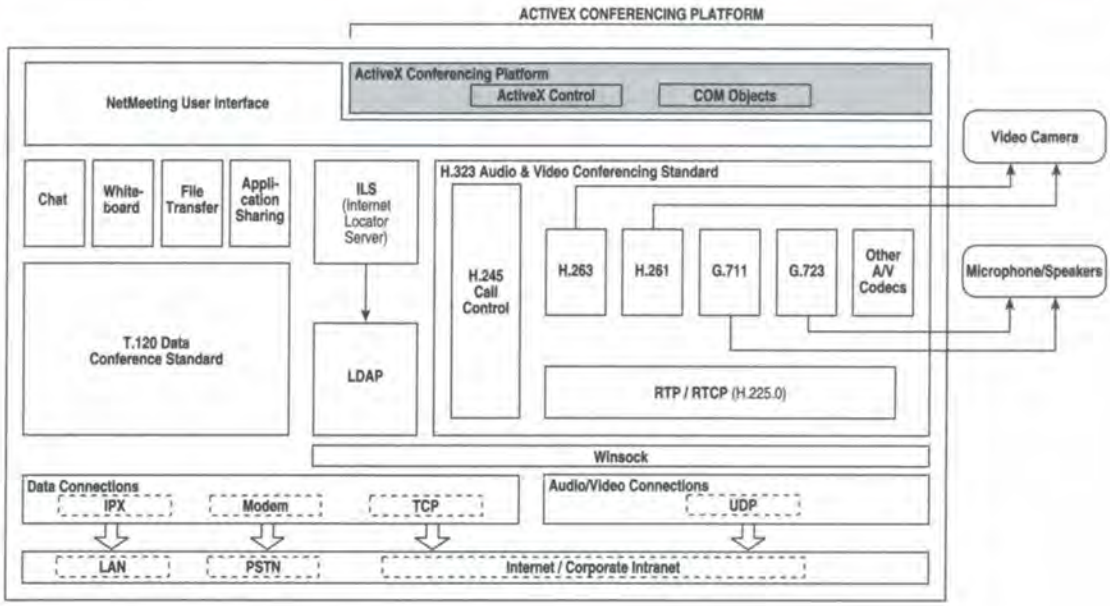
NetMeeting audio, video, and data conferencing capabilities are available to end users through the NetMeeting user interface. This interface presents NetMeeting features in an easy-to-use and understandable format of visual icons, "hot" buttons, simple adjustment bars, and tabbed windows and call directories. The following diagram highlights this level of the NetMeeting architecture.



ActiveX Conferencing Platform

The NetMeeting platform enables developers to incorporate NetMeeting features, such as chat and whiteboard, in their applications and Web pages. This platform is based on ActiveX conferencing, which includes the following components:

- A set of objects based on the OLE Component Object Model (COM) for adding conferencing support to COM applications.
- A set of COM objects for building LDAP directory support into applications, including support for LDAP-based directories (ILSs) used by NetMeeting.
- ActiveX controls for adding conferencing support to Web pages, Visual Basic applications, and OLE-enabled documents and applications.
- A set of COM objects for installing custom audio and video codecs that NetMeeting can leverage during calls.



ActiveX conferencing is available in the *Microsoft NetMeeting 2.1 SDK*.

Understanding Conferencing Standards

Standards are critical to achieving the vision of NetMeeting: to create and popularize a new type of multimedia communications. To achieve this vision, a real-time communications and conferencing product must be developed with open standards so that users can connect with each other as easily and reliably as using a telephone. Consumers of this technology expect and demand that all products operate error-free and that every connection succeed. Standards ensure this experience.

With standards, a product from one vendor can provide a guaranteed level of compatibility with products from other vendors. Vendors can continue to build compatible add-on products that successfully interoperate with different real-time communications and conferencing products. Depending on the standards that these products support, users can potentially share applications and information, see each other with video, talk to one another, or perform all of these functions simultaneously.

Standards Organizations

The NetMeeting team is an active member of organizations that promote industry standards. The following organizations are at the forefront of standards development for communications and conferencing products:

International Telecommunications Union

Governments and the private sector coordinate global telecommunications networks and services through the ITU, which develops, regulates, and standardizes global telecommunications and organizes regional and world events. For more information about the ITU standards, including T.120 and H.323, visit the ITU home page at **http://www.itu.ch/**.

Internet Engineering Task Force

The IETF is the protocol engineering and development arm of the Internet. This organization is a large, open, international community of network designers, operators, vendors, and researchers concerned with the evolution of Internet architecture, as well as the smooth operation of the Internet. The IETF maintains working groups for research and technical study. For more information about the IETF standards, including drafts for LDAP specifications, visit the IETF home page at **http://www.ietf.org/**.

International Multimedia Teleconferencing Consortium

The IMTC is a nonprofit corporation founded to promote the creation and adoption of international standards for multipoint document and video teleconferencing. The IMTC provides a forum for its worldwide members to develop product specifications and educate others on standards-based development. The IMTC and its members promote a "Standards First" initiative to guarantee interoperability for all aspects of multimedia teleconferencing. For more information about the IMTC and to obtain information on standards, visit the IMTC home page at **http://www.imtc.org/**.

The T.120 Standard

The ITU T.120 standard is made up of a suite of communications and application protocols developed and approved by the international computer and telecommunications industries. These protocols enable developers to create compatible products and services for real-time, multipoint data connections and conferencing. T.120-based applications enable many users to participate in conferencing sessions over different types of networks and connections.

Depending on the type of T.120 product, they can make connections, transmit and receive data, and collaborate using compatible data conferencing features, such as application sharing, conferencing whiteboard, and file transfer. Microsoft and more than 100 other major companies support the development of products and services using the T.120 standard.

Benefits

The demand for standards-based products and services is based on the many benefits that standards provide to consumers. The following list describes some of the primary benefits of developing under the T.120 standard:

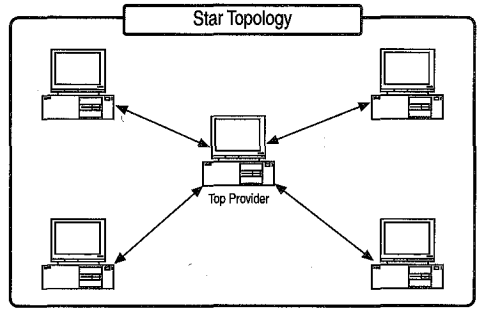
- T.120 ensures that many participants can send and receive data in real time without any errors in data transmission. Users can expect this reliability over many types of supported connections, including TCP/IP and null modem.
- For multipoint data conferencing, the T.120 standard supports a variety of common topologies, including star, daisy-chain, and cascaded connections. To learn more about these topologies, see "Topologies" later in this section.
- Developers can create applications with T.120 alone or in combination with other ITU standards, such as the H.323 standard for audio and video conferencing.

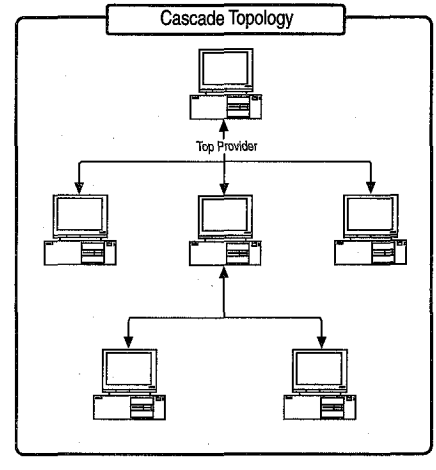
Interoperability

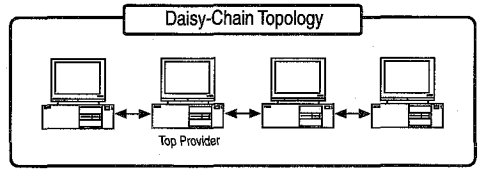
One of the most important features of the T.120 infrastructure is the interoperability of products and services that support the standard. This interoperability extends both to networking and applications. For more information about T.120 interoperability, see "Understanding Product Interoperability" later in this chapter.

T.120 Topologies

The T.120 architecture supports several topologies that define how users connect to a conference and transmit data during the conference. The following diagram illustrates three common topologies: star, daisy-chain, and cascaded. These topologies represent the types of connections typical of NetMeeting conferences.





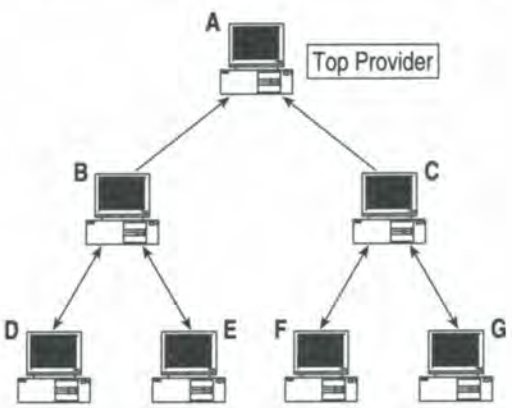


One of the participants in the conference is assigned as the top provider (conference host). The top provider's application is responsible for handling the global resources in the conference and also for sequencing data, if necessary. The top provider is determined within the initial connection between the first two participants. The location of the top provider and how the other participants are connected to the top provider differentiates the various topologies.

With NetMeeting, the person initiating the call is the top provider. The previous diagram illustrates the relationship of this top provider to other NetMeeting nodes in the various conference topologies; many other combinations of top provider and conference nodes are possible. NetMeeting actually initiates the call and determines whether a conference is already running. If no conference is running, NetMeeting creates one locally, with the person initiating the call as the top provider. If a conference is running, NetMeeting notifies the person, who then has the option of joining the remote conference. Also, NetMeeting provides a Host Meeting feature (**Call** menu), which determines the top provider automatically, based on the selected conference host rather than on the caller order.

Data flows according to the conference topology, which is determined by how each connection in the conference is established ("who calls whom"). For example, in the diagram, the following order of calls establishes the conference topology:

- A (top provider) initiates calls to B and C.
- Then, B calls D and E (or D and E each call B).
- C calls F and G (or F and G each call C).



A conference can have only one top provider. After the top provider (A) is established, that computer remains the top provider throughout the conference. Note that two conferences cannot be joined together. Therefore, if C called F and G first, it would not be possible for them to join the conference with A, B, D, and E.

During data conferencing, if B shared an application with the other conference participants, data would flow simultaneously to the adjoining connections (A, D, and E). Then, data would continue to flow outward to the remaining connections. Also, any two connections within the conference can initiate audio and video conferencing. NetMeeting enables audio and video switching so that participants can switch the pair sharing audio and video.

If B hangs up or is removed from the conference, D and E are also removed. D and E may remain conferenced together, though, if they were connected with audio and video in the original session. D and E would not have data conferencing, however, because this function is removed when B hangs up.

How NetMeeting Uses the T.120 Standard

Microsoft developed NetMeeting data conferencing features based on the T.120 infrastructure, enabling NetMeeting to interoperate with other T.120 standards-based products. T.120 protocols are the building blocks for the following NetMeeting functions:

- **The ability to establish and maintain NetMeeting data connections.** Two or more participants can establish a NetMeeting connection. Within the conference, T.120 protocols manage the sequencing and flow of data transported by NetMeeting connections. T.120 ensures that data is accurately and reliably transmitted between conference nodes.
- **Built-in data conferencing applications.** NetMeeting conference participants can use T.120-based applications (which sit on top of T.120), such as file transfer based on the T.127 standard and application sharing based on T.128. Note that NetMeeting does not support T.126 currently; this support is expected in a future version of NetMeeting.
- **Support for multiuser conferences.** Many people can join a NetMeeting data conference for communication and collaboration. The conference host (top provider) in NetMeeting provides the node that manages participants and their applications.
- **Capability to interoperate across networks and platforms.** T.120 interfaces enable NetMeeting and other standards-based products to communicate across common networks using TCP/IP without platform restrictions.

Microsoft continues its efforts to enhance and extend the application of T.120 protocols. The standards groups that define T.120 are now working on the next generation of the protocols, and Microsoft is very active in these efforts.

The H.323 Standard

H.323 is an ITU standard that sets forth a specification for terminals (PCs), equipment, and services for multimedia communication over networks that do not provide a guaranteed quality of service. H.323 terminals and equipment can carry real-time audio, video, and data, or any combination of these elements. This standard is based on the IETF real-time protocol (RTP) and real-time control protocol (RTCP), with additional protocols for call signaling and data and audiovisual communications.

Products that use H.323 for audio and video enable you to interconnect and communicate with other people over the Internet, just as people using different makes and models of telephones can communicate over public switched telephone network (PSTN) lines. H.323 defines how audio and video information is formatted and packaged for transmission over the network. Standard audio and video codecs encode and decode input/output from audio and video sources for communicating between nodes.

Benefits

Leading the audio and video teleconferencing industry, H.323 products and services offer many benefits to consumers. The following list describes some of the benefits that distinguish H.323:

- Microsoft and more than 120 leading companies have announced their intent to support and implement H.323 in their products and services. This broad support establishes H.323 as the leading solution for audio and video conferencing over the Internet.
- Products and services developed by multiple vendors under the H.323 standard can interoperate without platform limitations. H.323 conferencing clients, bridges, servers, and gateways will support this interoperability.
- H.323 provides multiple audio and video codecs that format data according to the requirements of various networks, with different bit rates, delays, and quality options. Users can choose the codecs that best support their computer and network selections.
- The addition of T.120 data conferencing support to the H.323 specification means that products developed under H.323 can offer a full range of multimedia functions, with both data and audiovisual conferencing support.

Interoperability

Just as telecommunications standards enable people with telephones from different vendors to call anyone around the world, the H.323 standard will enable the same level of interoperability for audio and video conferencing on the Internet. For more information about H.323 interoperability, see "Understanding Product Interoperability" later in this chapter.

How NetMeeting Uses the H.323 Standard

Microsoft developed NetMeeting audio and video conferencing features based on the H.323 infrastructure, enabling NetMeeting to interoperate with other H.323 standards-based products. H.323 codecs and protocols are the building blocks for these NetMeeting functions:

- **The ability to establish and maintain audio and video connections.** Two participants can establish a NetMeeting conference with audio and video over a TCP/IP connection. H.225.0 enables multiple audio and video streams to transport inbound and outbound NetMeeting information. H.323 protocols enable NetMeeting to communicate with and transmit data to other compatible audio or video clients. Also, conferencing servers and bridges enable multiuser audio and video conferences between multiple NetMeeting clients, as well as other H.323-compatible products.
- **Audio and video codecs that optimize Internet connections.** NetMeeting supports a suite of codecs, operating between 4.8 Kbps and 64 Kbps, that support various computer and connection types.
- **Support for T.120 data communications.** NetMeeting creates the association between T.120 and H.323 during a NetMeeting conference. This association allows the NetMeeting call to be completed in two phases, one each for T.120 and H.323, but appear as a single call.
- **Support for H.323 Gateways.** Although Microsoft does not provide a gateway, NetMeeting will support third-party H.323 gateways when they become available. See "H.323 Conferencing Products and Services" later in this chapter.
- **Support for H.323 Gatekeepers.** NetMeeting does not support H.323 gatekeepers. Instead, NetMeeting uses system policies, Windows NT Domain Control, and LDAP-based directories to perform gatekeeper functions. This enables customers to leverage their investments on other standards-based management tools.

The standards groups that define H.323 are now working on the next generation of the protocols, and Microsoft is very active in these efforts. The H.323 group is incorporating its experience from the first generation of H.323 and is also adding new features, such as strong security and interoperability with streaming media servers.

The Lightweight Directory Access Protocol

The Lightweight Directory Access Protocol (LDAP), as its name implies, is a standard method by which application clients query for and access information stored on directory servers over TCP/IP connections. Typically, a published directory contains data about people or other entities that users can access. The most common examples of paper-based directories are the yellow and white pages of telephone books, which enable a customer to look up the telephone number for a person or company. An example of an electronic directory is a published e-mail address book, which enables an e-mail user to look up a person's e-mail address and other details, such as office location and internal phone extension.

LDAP Version 2 and Version 3

Most Internet applications currently support LDAP Version 2, which provides static directory services for storing user information that is constant. Traditionally, directories contain data that rarely changes. Telephone books are published once a year because customers do not change numbers frequently. E-mail directories are also very stable, typically requiring changes only when new users are added or outdated users are removed. One result is that a directory is read very often, but seldom written.

LDAP Version 3 is currently being developed. This version includes specifications for dynamic directory services, which store user information that changes frequently. Dynamic information can expire and must be updated periodically to remain up-to-date (for example, IP addresses through the dynamic host configuration protocol [DHCP]). Ratification of this new version is expected in the near future.

Benefits

LDAP has several benefits that give this standard an advantage over other directory server protocols:

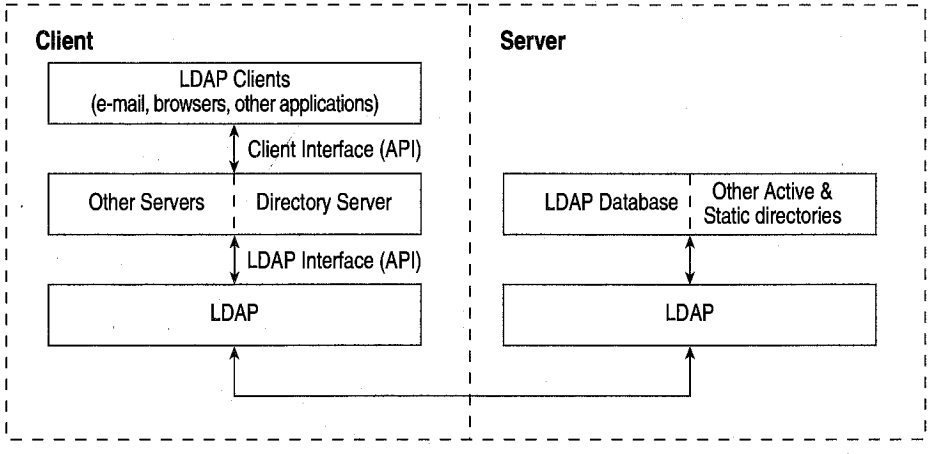
- With LDAP, Internet clients, servers, and applications have a common method for accessing and using directory information.
- LDAP provides a simple search operation that reduces the amount of input and output processing. This is important for Internet implementations that favor more simplistic processing methods for slow Internet connections.
- By using TCP/IP connections, LDAP can be implemented relatively easily on most platforms.
- Directory servers developed under LDAP can support many users and their associated address information.
- With dynamic directory services, LDAP can track more volatile information that changes frequently and must be updated on a regular basis. This is a requirement for dynamic call directories that need to be continually updated to show the most current directory of users.

Interoperability

Many current servers use LDAP Version 2 technology, and developers can easily create new servers that provide dynamic directory services (required for NetMeeting). For more information about LDAP interoperability, see "Understanding Product Interoperability" later in this chapter.

LDAP Topology

The IETF Web site documents the LDAP architecture. For detailed information, visit **http://ietf.org/ids.by.wg/ldapext.html**. The following diagram illustrates the LDAP architecture. LDAP clients, such as e-mail programs, Web browsers, and external applications, can execute directory service transactions against the LDAP server. Directory services define the operations for locating and connecting people who want to join a conference. This capability may include simple search and update transactions, such as logging on and off, creating a directory listing of all available users, and resolving a user's address information.



Through an LDAP interface, the server communicates with the LDAP database to retrieve directory information. LDAP servers can be multithreaded to support many LDAP databases simultaneously. The LDAP database stores and retrieves information based on a hierarchical structure of entries, each with its own distinguishing name, type, and attributes. The attributes (properties) define acceptable values for the entry. An LDAP database can store and maintain entries for a large number of users.

Databases can include static entries, dynamic entries, or both. A static entry contains information that remains accurate for a long period of time. On the other hand, dynamic entries (objects) and their attributes have a defined life expectancy and will expire if they are not updated by the client. The LDAP server is responsible for removing expired entries.

How NetMeeting Uses the LDAP Standard

Traditional static directories contain information that is presumed to have value until it is removed. In contrast, the NetMeeting directory information quickly degrades over time. Because NetMeeting is an IP-based conferencing application, the most important attribute is the computer's network or IP address. With the widespread use of DHCP, each time a user logs on to a LAN or ISP, his or her machine is assigned a different IP address, which means that the directory information needs to be frequently updated. Therefore, a traditional static database approach—to write seldom and read often—cannot be used.

NetMeeting directory requirements resulted in the Microsoft ILS, a new breed of LDAP-based *dynamic* directory services. ILS information can be updated as frequently as desired, because storage is not based on SQL or file-based databases; instead, the ILS stores directory data in memory to allow frequent changes. Even though the directory stores information differently, access to the information is still the same as traditional static directories. Therefore, NetMeeting uses the LDAP protocol to access the ILS.

NetMeeting and the ILS support the LDAP standard for the following functions and capabilities:

- **Using the LDAP Client/Server Infrastructure.** Based on the LDAP infrastructure, the ILS consists of both a client and a server. NetMeeting uses the client to access the server through an LDAP interface. The ILS provides a publishing mechanism that enables NetMeeting to publish information about its users for other people to view and search. For example, if NetMeeting publishes a user's name, online status, and audio/video capability, a potential caller can then identify the party to call, whether the user has video capability, and whether the user is currently on a call.
- **Support for Dynamic Objects (LDAP Version 3).** The ILS defines two dynamic object types: the **User** object and **MeetingPlace** object. The **User** object defines the properties of a NetMeeting user and is used to publish information about that user. The **MeetingPlace** object defines the properties of an ongoing conference. Each object consists of attributes, or properties. For example, the **User** object includes attributes, such as *First Name* and *Country*, and the **MeetingPlace** object includes attributes, such as *Attendee* and *Description*.
- **Dynamic Entry Expiration and Update Operations.** An important concept for dynamic directories is entry lifetime—each entry is valid only as long as the NetMeeting client is active. If the client computer goes offline (for example, the computer connection is terminated by another incoming call) the ILS entry is no longer valid and will be removed by the server as soon as possible. Each NetMeeting entry in the ILS directory has a lifetime, which is typically

a specific number of minutes. NetMeeting regularly updates its entries in the ILS, maintaining only those entries that are still online and ready to accept a call. If a NetMeeting entry fails to update itself within the allotted time, the server removes the entry to keep the entire directory listing as accurate as possible.

NetMeeting and ILS 1.0 use LDAP Version 2, the most current, approved version of the LDAP specification. However, the concept of a dynamic entry lifetime and update is not defined in LDAP Version 2. Microsoft and other IETF members have proposed a dynamic entry lifetime and update for LDAP Version 3.

- **Developing Compatible Servers.** As described in this section, NetMeeting's unique capabilities impose new requirements for LDAP-based directory services that are unique to real-time conferencing applications. The ILS provides new services that were previously unavailable with LDAP Version 2. Because of this innovation, no other servers are currently compatible with NetMeeting.

  Compatible server products, though, can be engineered without a high development learning curve. The ILS uses the well-known industry standard LDAP for its access protocol, and these servers are close cousins of traditional static LDAP servers. ILS functionality and transactions are open and published; therefore, expertise in developing traditional LDAP directories can be easily applied toward building ILS-like directories. Because LDAP is the standard for directory protocols, any infrastructure or tools, such as proxy servers, firewalls, or networking monitoring tools, are completely applicable with it. Also, the ILS can easily be integrated with static directories that use a common LDAP protocol.

Understanding Product Interoperability

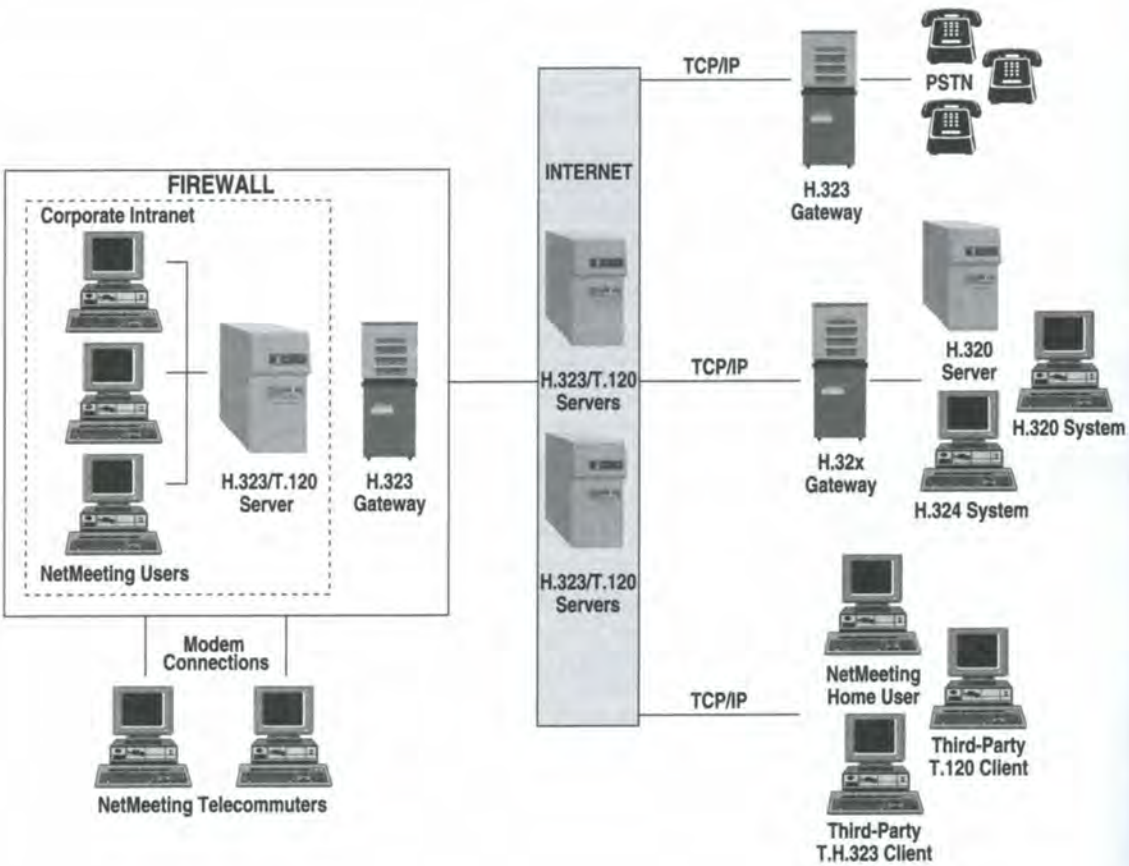
Product interoperability is important for several reasons:

- To instill customer confidence, products must be based on compatible rather than proprietary technologies.
- Particularly in the corporate environment, communication between local and remote locations is a necessity; to support these user communities, products must interoperate across platforms and networks.
- Products of varying levels of functionality must be able to communicate; standards enable compatibility of applications with a variety of functions and features.
- As more vendors develop standards-based products, customers benefit from competitive pricing, improved quality, and product upgrades.

The following sections describe the specific types of products and services, including clients, servers, and gateways, that interoperate with NetMeeting. Interoperability testing is described for two ITU standards that NetMeeting currently supports: T.120 for data conferencing and H.323 for audio and video conferencing. This discussion includes the elements that make T.120 and H.323 standards-based products interoperate.

Interoperability: The Big Picture

The following diagram shows how NetMeeting can interoperate with T.120- and H.323-compatible products over the Internet and corporate intranet using TCP/IP network connections. For corporate and home users, many interoperability scenarios are possible between NetMeeting and compatible, standards-based clients, servers, bridges, and gateways.



The diagram shows the following interoperability scenarios:

- Within an organization, NetMeeting users can connect with each other over the corporate intranet. A T.120 or H.323 conferencing server can provide inbound and outbound connectivity with compatible clients for data, audio, or video conferencing.
- An H.323 gateway can be used to bridge internal and external networks over a corporate firewall, supporting connections for audio and video conferencing.
- NetMeeting users can initiate multipoint connections with third-party T.120 clients. T.120 servers can provide administration services for this data conferencing scenario.
- NetMeeting users can initiate point-to-point connections with H.323 clients. T.120 data conferencing may be supported in conjunction with H.323 audio and video conferencing. H.323 servers can provide administration services for this conferencing scenario.
- Through an H.32*x* gateway, NetMeeting users can connect to H.320 and H.324 systems. Also, an H.320 server can be used for connectivity with multiple H.320 systems.
- An H.323 gateway also enables NetMeeting users to connect to people over PSTN lines.

Compatible Products and Services

For information about NetMeeting-compatible products and services from leading vendors, see the NetMeeting Web site at **http://www.microsoft.com/netmeeting/**. Some of these products are mentioned in the following sections.

Note Information in this directory has been submitted by third-party vendors. Microsoft has not tested or certified all the products and services for interoperability. Contact the appropriate vendor for more information.

T.120 Conferencing Products and Services

Because T.120 is purely a data standard, T.120-compliant products and services that operate with NetMeeting provide data conferencing and collaboration features rather than audio or video support. NetMeeting can potentially interoperate with any T.120 product or service that uses the same TCP/IP transport. These products and services fall primarily into the following categories:

Data conferencing clients. The T.120 infrastructure gives products and services the ability to connect and send data. This means that NetMeeting is capable of connecting to any data conferencing client that supports ITU T.120. Also, these clients can potentially interoperate, depending on the compatibility of application layer protocols supported by the third-party product (any incompatibility in protocols would be at the application layer, and not with the T.120 protocols).

Conferencing servers or bridges. Otherwise known as multipoint control units, or MCUs, these central computing facilities host multipoint data conferences that might also be held in conjunction with audio or video conferencing. The MCU has the necessary components to handle multiple nodes from multiple computers in a single conference. For the MCU provider, conferencing services require an investment in server hardware, software, and technical support. Customers benefit from guaranteed performance and scalability that can cross platforms, vendors, and networks.

Note Using a bridge to host a meeting between computers with conferencing products from different vendors does not guarantee compatibility, particularly in the case of products that do not support the same application layer protocols. The conferencing bridge does not provide a compatibility layer that enables different products to interoperate.

Vendors provide multipoint bridging and management services that enable many sites to collaborate over the Internet. Also, these services enable users to schedule, create, administer, join, and track conferences all in one convenient place. Note that NetMeeting hardware requirements apply through a conferencing bridge, which limits the conference to the speed of the slowest connection.

H.323 Conferencing Products and Services

NetMeeting can potentially operate with any H.323 conferencing product or service over TCP/IP connections. H.323-compatible products and services fall primarily into these categories:

Audio and video conferencing clients. NetMeeting is capable of connecting to any audio or video conferencing client that supports the H.323 specification. Depending on the availability of compatible audio and video codecs (G.723.1, G.711, H.263, and H.261 are supported by NetMeeting), H.323 products can potentially interoperate and exchange audio and video information. For example, NetMeeting is fully interoperable with the Intel® Internet Video Phone, providing high-quality, face-to-face video and audio conferencing from a local desktop.

Conferencing servers and bridges. Conference servers and bridges can host multipoint audio and video conferencing for H.323-compatible products. Some servers and bridges may also host audio and video conferencing in conjunction with data conferencing. Service providers will offer an advanced network of audio, video, and data conferencing bridges, professional conference administrators, and special features such as password security. Connecting multiple sites in the United States or worldwide, these services will assist with equipment, reservation, scheduling, and customized reporting needs.

Gateways. Currently, several companies are developing H.323 gateways. These gateways will enable NetMeeting users to bridge to other types of networks, such as an H.320 ISDN video conference or a regular telephone on a public network.

Gatekeepers. NetMeeting does not support the gatekeeper function. Instead, NetMeeting uses system policies, Windows NT Domain Control, and LDAP-based directories to perform gatekeeper functions.

LDAP Conferencing Servers

ILS supports the IETF LDAP Version 2 standard for NetMeeting directory services. LDAP servers support the same LDAP protocol, but each server extends LDAP for a particular purpose. For example, ILS applies LDAP for use with dynamic records. LDAP Version 3 designers propose these dynamic directory services as part of the LDAP protocol. When Version 3 is finalized and implemented within NetMeeting, developers can access standards information from the IETF Web site for developing compatible servers. Currently, vendors can develop interoperable servers for NetMeeting by obtaining information about the LDAP extension from Microsoft.

In addition, many people use the ULS, developed for NetMeeting 1.0, for locating and connecting to other NetMeeting users. Third-party vendors have developed many interoperable ULSs, such as **uls.four11.com** and **uls.sitebuilder.net**, which users can log on to from NetMeeting.

Directory Servers

ILS, an optional component of the Internet Information Server, provides directory services that enable NetMeeting users to locate each other on the Internet or corporate intranets. These servers create a directory of NetMeeting users. From this directory, users can select participants for real-time conferencing and collaboration. The ILS product replaces ULS technology, which Microsoft originally developed for NetMeeting 1.0. ILS provides all of the ULS functionality, as well as introducing advanced server technology not previously available. Users can benefit from enhanced features and functions, better performance, and higher scalability to support more NetMeeting users.

Note The ILS is not required to use NetMeeting and connect with other users. You can connect directly with other people by entering their e-mail address in the NetMeeting Call window (although you must log on to the ILS for this connection method to work). Instead, the ILS simplifies the process of finding and connecting users by providing a directory listing of people who are running NetMeeting.

The Internet Locator Server

The ILS offers a standards-based, dynamic directory solution to the user location problem on the Internet. It provides organizations with a directory server for NetMeeting users. Like the ULS, the ILS provides a memory-resident database for storing dynamic directory information. This database enables users to find dynamic information, such as an IP address, for people currently logged on to an Internet service or site. The ILS database maintains the entries, which clients update periodically. This process ensures that clients can always access the most current information about each user's Internet location.

The following features distinguish the ILS:

Support for industry-standard protocols. The ILS provides both an LDAP interface for NetMeeting support and a proprietary ULP interface for legacy support of NetMeeting 1.0. Through these built-in protocols, the ILS provides directory server support for NetMeeting. These interfaces allow NetMeeting to access the server for dynamic directory information and facilitate point-to-point Internet communication sessions.

Note Other clients can access ILS through the LDAP interface; see the *Microsoft NetMeeting 2.1 SDK*. All client applications must migrate to LDAP to access dynamic directory information.

Performance monitoring. The ILS supports Windows NT Server administration features—including performance monitoring (Perfmon counters), Simple Network Management Protocol (SNMP) monitoring, and event logs—to measure activity and system performance. Operators enjoy administration features, such as transaction logs that collect usage statistics, track messages and transactions, and allow administrators to examine usage patterns. These tools enable administrators to proactively monitor server performance and identify potential problems.

Stable, robust server capabilities. This standards-based Internet directory server was designed to provide stable, robust directory services. The ILS uses thread pooling and connection management to enable more efficient handling of system resources. ILS users will experience better performance, because the ILS uses binary data packets to optimize performance. Also, the server uses a spanning tree architecture to support many concurrent users in a single server configuration.

Customization through Active Server Pages. With Active Server Pages, administrators can combine HTML and scripting components to customize their ILS interface. They can create scripts to display a specific group of NetMeeting users currently online, enable user searches, and initiate real-time communication sessions with other users. For example, administrators can create "buddy pages" that limit the directory to a list of specific users that share a common interest.

Easy setup and administration. The ILS provides a graphical setup program so that you can install server components quickly and easily. Then, administrators can set options for user logon, security, and server access through the Microsoft Internet Service Manager.

Microsoft product support. Microsoft provides worldwide product support through the Microsoft Support Network. ILS users can choose from standard or priority support.

For More Information

For information about setting up and implementing an ILS, refer to the *Microsoft Internet Locator Server Operations Guide* or to its companion, the *Microsoft Internet Locator Service Operations Reference*. Both documents are packaged with the product, or you can download the ILS and the accompanying documents from the Web site at **http://www.backoffice.microsoft.com/** (click **Product Information**, select the **Microsoft Commercial Internet System** product, and then click **Internet Locator Server**).

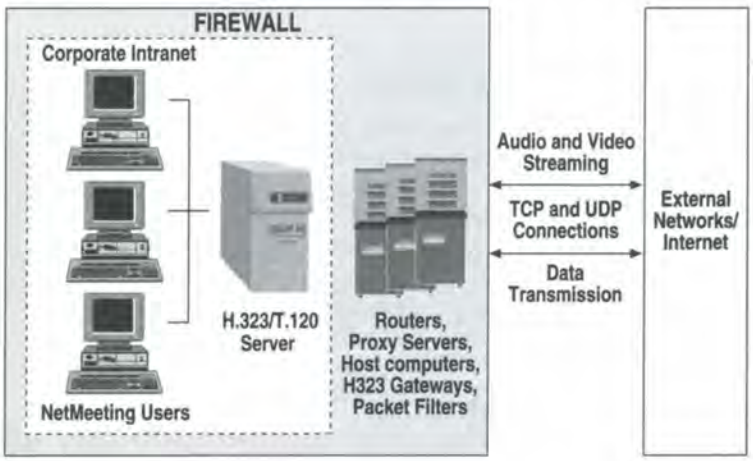
Firewall Configuration

Microsoft NetMeeting can be configured to work with most organizations' existing firewall security. However, because of limitations in most firewall technology, few products are available that allow you to securely transport inbound and outbound NetMeeting calls containing audio, video, and data across a firewall. You want to consider carefully the relative security risks of enabling different parts of a NetMeeting call in your firewall product. Especially, you must consider very carefully the security risks involved when modifying your firewall configuration to enable any component of an inbound NetMeeting call.

Components of a Secured System

A firewall is a set of security mechanisms that an organization implements, both logically and physically, to prevent unsecured access to an internal network. Firewall configurations vary from organization to organization. Most often, the firewall consists of several components, which can include a combination of routers, proxy servers, host computers, gateways, and networks with the appropriate security software. Very rarely is a firewall a single component, although a number of newer commercial firewalls attempt to put all of the components in a single box.

For most organizations, an Internet connection is part of the firewall. The firewall identifies itself to the outside network as a number of IP addresses—or as capable of routing to a number of IP addresses—all associated with domain name service (DNS) entries. The firewall might respond as all of these hosts (a virtual machine) or pass on packets bound for these hosts to assigned computers. The following diagram shows a firewall configuration.



NetMeeting and Firewalls

You can configure firewall components in a variety of ways, depending on your organization's specific security policies and overall operations. While most firewalls are capable of allowing primary (initial) and secondary (subsequent) TCP and UDP connections, they might be configured to support only specific connections based on security considerations. For example, some firewalls allow only primary TCP connections, which are considered the most secure and reliable.

To enable NetMeeting multipoint data conferencing (application sharing, whiteboard, chat, file transfer, and directory lookups), your firewall only needs to pass through primary TCP connections on assigned ports. For NetMeeting to make calls with audio and video conferencing, your firewall must be able to pass through secondary TCP and UDP connections on dynamically assigned ports. Some firewalls can pass through primary TCP connections on assigned ports, but cannot pass through secondary TCP or UDP connections on dynamically assigned ports.

Note NetMeeting audio and video features require secondary TCP and UDP connections. Therefore, when you establish connections through firewalls that accept only primary TCP connections, you are not able to use the audio or video features of NetMeeting.

Establishing a NetMeeting Connection with a Firewall

When you use NetMeeting to call other users over the Internet, several IP ports are required to establish the outbound connection. If you use a firewall to connect to the Internet, it must be configured so that the following IP ports are not blocked.

This port	Is used for
389	Internet Locator Server (TCP)
522	User Location Service (TCP)
1503	T.120 (TCP)
1720	H.323 call setup (TCP)
1731	Audio call control (TCP)
Dynamic	H.323 call control (TCP)
Dynamic	H.323 streaming (RTP over UDP)

To establish outbound NetMeeting connections through a firewall, the firewall must be configured to do the following:

- Pass through primary TCP connections on ports 389, 522, 1503, 1720, and 1731.
- Pass through secondary TCP and UDP connections on dynamically assigned ports (1024-65535).

The H.323 call setup protocol (over port 1720) dynamically negotiates a TCP port for use by the H.323 call control protocol. Also, both the audio call control protocol (over port 1731) and the H.323 call setup protocol (over port 1720) dynamically negotiate UDP ports for use by the H.323 streaming protocol, which is the RTP. In NetMeeting, two UDP ports are determined on each side of the firewall for audio and video streaming, for a total of four ports for inbound and outbound audio and video. These dynamically negotiated ports are selected arbitrarily from all ports that can be assigned dynamically.

NetMeeting directory services require either port 389 or port 522, depending on the type of server you are using. ILSs, which support LDAP for NetMeeting, require port 389. ULS, developed for NetMeeting 1.0, requires port 522.

Firewall Limitations

Some firewalls cannot support an arbitrary number of virtual internal IP addresses or cannot do so dynamically. With these firewalls, you can establish outbound NetMeeting connections from computers inside the firewall to computers outside the firewall, and you can use the audio and video features of NetMeeting. Other people, though, cannot establish inbound connections from outside the firewall to computers inside the firewall. Typically, this restriction is due to limitations in the network implementation of the firewall.

Note Some firewalls are capable of accepting only certain protocols and cannot handle TCP connections. For example, if your firewall is a Web proxy server with no generic connection-handling mechanism, you will not be able to use NetMeeting through the firewall.

Security and Policy Concerns

Some organizations may have security or policy concerns that require them to limit how fully they support NetMeeting in their firewall configuration. These concerns might be based on network capacity planning or low confidence in the firewall technology being used. For example, security concerns might prohibit an organization from accepting any inbound or outbound flow of UDP data through its firewall. Because these UDP connections are required for NetMeeting audio and video features, disabling this function will exclude audio and video features in NetMeeting for calls through the firewall. The organization can still use NetMeeting data conferencing features—such as application sharing, whiteboard, and chat—for calls through the firewall by allowing only TCP connections on ports 522 and 1503.

A useful reference for firewall design, including policy and security considerations, is *Building Internet Firewalls* (D. Brent Chapman and Elizabeth D. Zwicky, 1995, Sebastopol, California: O'Reilly & Associates, Inc.).

System Policies and Restrictions

You can use the NetMeeting policies and restrictions to restrict most of the features that users can access within NetMeeting. For example, you can preset the NetMeeting work environment to control audio and video access, or to prevent users from sending files to or collaborating with other users during application sharing. You can also specify several NetMeeting settings, including the default directory server.

Issues to Consider Before Setting Policies

Before implementing system policies, you must consider the following issues:

- What types of restrictions and settings would you like to define and manage centrally? For example, do you want to limit access to NetMeeting file transfer, application sharing, or audio and video features?
- Do you want to use one set of standard system policies for all users and computers, or do you want to customize settings by groups of users? Also, do you want to maintain some individual settings for specific users and computers?
- Would it be useful to customize settings by groups, such as Accounting group and Marketing group? A small group of individuals, such as administrators, can have unique privileges set for their specific needs.

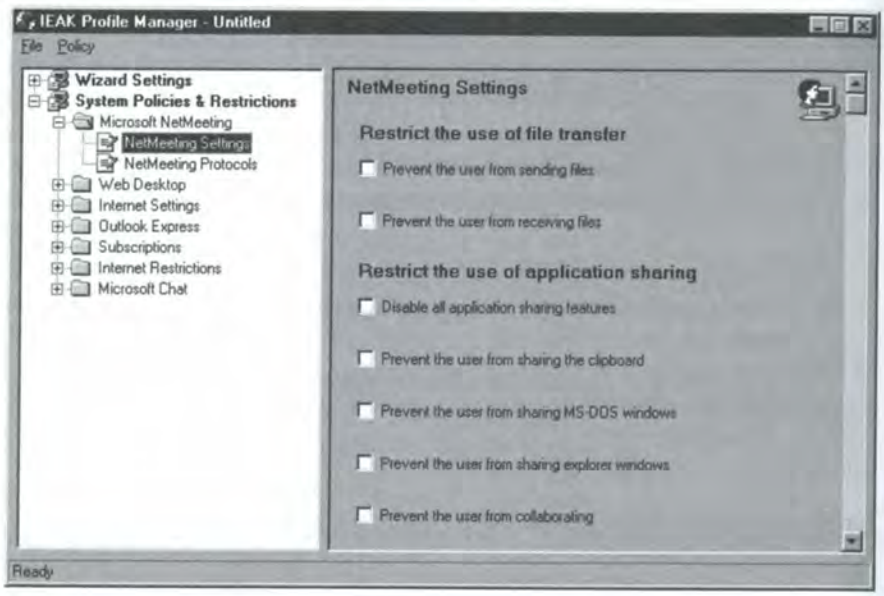
Customizing System Policies and Restrictions

You can use the Internet Explorer Administration Kit (IEAK) Configuration Wizard **System Policies and Restrictions** page to deploy Internet Explorer and NetMeeting to users with pre-configured system policy and restriction settings. For more information on how to use the IEAK Configuration Wizard to build custom packages, see "Building Custom Packages" in Chapter 36, "Implementing the Deployment."

When automatic browser configuration is enabled, you can use the IEAK Profile Manager to update and maintain users' system policies and restrictions for NetMeeting. For more information on automatic browser configuration, see Chapter 37, "Automatic Browser Configuration."

▶ **To change policies and restrictions for NetMeeting**

1. Perform one of the following:
   - Run the IEAK Configuration Wizard, and navigate to the **System Policies and Restrictions** page for the custom package you want to build.
   - Run the IEAK Profile Manager, and open the auto-configuration (.ins) file you want to change.
2. In the **System Policies & Restrictions** hierarchy, select **Microsoft NetMeeting**, and then select **NetMeeting Settings**.



Note The IEAK Profile Manager is shown here, but the **System Policies & Restrictions** page of the IEAK Configuration Wizard works similarly.

3. In the right pane, select the options you want.

   For more information, see the next two sections: "NetMeeting System Policies and Restrictions Options" and "Common System Policies and Restrictions Scenarios."
4. Save your changes to the auto-configuration (.ins) file, or finish building the custom package, as appropriate.

NetMeeting System Policies and Restrictions Options

The following table describes the available NetShow system policies and restrictions options.

Option	Description
Restrict the use of file transfer	
Prevent the user from sending files	Prevents the user from sending files to other participants in a meeting.
Prevent the user from receiving files	Prevents the user from receiving files from other participants in a meeting.
Restrict the use of application sharing	
Disable all application sharing features	Disables the **Share Application** option on the **Tools** menu and the **Share Application** button.
Prevent the user from sharing the clipboard	Prevents the user's Clipboard contents from being accessible to other participants in a meeting.
Prevent the user from sharing MS-DOS windows	Prevents the **MS-DOS windows** option from appearing on the **Share Application** menu. **Note** If you choose this option, you must also click **Prevent the user from sharing explorer windows**. These two options work only if they are both selected. If you do not choose **Prevent the user from sharing explorer windows**, the user will be able to share MS-DOS windows from Microsoft Explorer.
Prevent the user from sharing explorer windows	Prevents the **Microsoft Explorer windows** option from appearing on the **Share Application** menu.
Prevent the user from collaborating	Prevents a user from collaborating with other users with whom he or she is application sharing.
Restrict the use of the option dialog	
Disable the 'General' options page	Removes the **General** tab from **Options** (on the **Tools** menu) to prevent the user from changing general, network bandwidth, and file transfer features.
Disable the 'My Information' options page	Removes the **My Information** tab from **Options** (on the **Tools** menu) to prevent the user from changing user address and user category (personal, business, or adults-only) information.
Disable the 'Calling' options page	Removes the **Calling** tab from **Options** (on the **Tools** menu) to prevent the user from changing Directory features and SpeedDial options for automatically adding people and updating the SpeedDial list.

Option	Description
Restrict the use of the option dialog *(continued)*	
Disable the 'Audio' options page	Removes the **Audio** tab from **Options** (on the **Tools** menu) to prevent the user from changing general audio, microphone sensitivity, and H.323 gateway features.
Disable the 'Video' options page	Removes the **Video** tab from **Options** (on the **Tools** menu) to prevent the user from changing options for sending and receiving video, or from specifying send image size, video quality, and video camera properties.
Disable the 'Protocols' options page	Removes the **Protocols** tab from **Options** (on the **Tools** menu) to prevent the user from changing the selected TCP/IP or null modem protocol.
Prevent the user from answering calls	Prevents the user from being notified of incoming calls.
Prevent the user from using audio features	Prevents the user from using the audio features to talk to or hear other participants in a meeting. **Note** **Disable the 'Audio' options** page is selected automatically when you choose this option.
Restrict the use of video	
Prevent the user from sending video	Prevents the user from transmitting video images to other participants in a meeting. **Note** **Disable the 'Video' options** page is selected automatically when you choose both **Prevent the user from sending video** and **Prevent the user from receiving video**.
Prevent the user from receiving video	Prevents the user from receiving video images from other participants in a meeting. **Note** **Disable the 'Video' options** page is selected automatically when you choose both **Prevent the user from sending video** and **Prevent the user from receiving video**.
Prevent the user from using directory services	Prevents the user from accessing all servers and server functions, including logging on, logging off, resolving addresses, and viewing directories. Users can still access Web directory services using the **Go** menu or using their Web browser. **Note** You may also want to choose **Disable the 'Calling' options** page.
Set the default Directory Server	
Directory Server	Sets the default Internet Location Server (ILS) to which the user logs on when starting NetMeeting.

Option	Description
Set Exchange Server Property for NetMeeting Address	
Exchange Server Property	Specifies the Exchange attribute that the Exchange Extension looks at to determine the NetMeeting address. Press the DOWN ARROW, and select one of the listed Attributes. **Note** If you are unable to use one of the listed attributes, you must set the registry key manually. To set the registry keys manually, use the administration tool to set the attribute in Exchange. Export the following values from the address book: Object Class, Directory Name, Extension Attribute *number*, and SMTP or Exchange Address. Write a script to set the value of Extension Attribute *number* for each person (for example, *<servername>/<email address>*). Import the revised file into the address book. If you are specifying attributes for a large number of people, use several different servers. For additional information, see the Microsoft Exchange reference documents.
Preset User Information Category	
Category	Identifies your category and purpose for using NetMeeting as either Personal, Business, or Adult. Category selections no longer appear in the IEAK Configuration Wizard and are dimmed on the **My Information** tab. Your name will be listed only in the directory for the category that you select. **Note** You can set this policy only prior to installing NetMeeting. If NetMeeting is already installed, you must remove the application, and then set the policy in order for it to take effect.
Select enabled set of T120 protocols	
Enable TCP/IP T120 protocol	Enables a user to place and receive calls using a TCP/IP connection.
Set the NetMeeting home page	
URL	Defines the specific URL for the Web page that users see when they select **Product News** (on the NetMeeting **Help** menu, click **Microsoft on the Web**, and then click **Product News**).
Set limit for audio/video throughput	
Average audio/video throughput limit (in bps)	Sets the average bandwidth limit for combined audio and video throughput over the network. For example, if you are concerned about excessive audio and video bandwidth utilization over your network, you may set a limit to restrict the average throughput that is permitted. This limit applies to audio and video transmission only, and does not affect data.

Common System Policies and Restrictions Scenarios

Typically, organizations may want to implement several system policies and restrictions for their user communities. The following table lists some common scenarios and the associated system policy name that organizations would set.

If you want to	Set this system policy
Disable all audio features	1. **Prevent the user from using audio features** 2. **Disable the 'Audio' options page**
Disable all video features	1. **Restrict the use of video** 2. **Prevent the user from sending video** 3. **Prevent the user from receiving video** 4. **Disable the 'Video' options page**
Limit the network bandwidth used by audio/video	**Set limit for audio/video throughput**
Prevent access to ULS or ILS directory services	1. **Prevent the user from using directory services** 2. **Disable the 'Calling' options page**
Specify the Web page that users see when they select **Product News**	**Set the NetMeeting home page**

Using NetMeeting on Intranet Web Pages

The NetMeeting platform enables third-party vendors to integrate conferencing features into their own products and services. The NetMeeting platform is available with the *Microsoft NetMeeting 2.1 SDK*, which includes the following components:

- A set of objects based on the OLE Component Object Model (COM) for adding conferencing support to COM applications. The COM interfaces are the same as the interfaces provided with NetMeeting 1.0.
- Online HTML Help files located in the \Nm21sdk\Doc\ folder, which is downloaded with the SDK. These Help files have been enhanced for NetMeeting 2.1.
- Sample files that demonstrate how to use the ActiveX control and COM objects, and detailed information about how to install and use each sample.
- A set of COM objects for supporting LDAP-based directories (ILSs) in your application.

- An ActiveX control for adding conferencing support to Web pages (using JavaScript-compatible languages or Visual Basic Scripting Edition), Visual Basic applications, and other OLE-enabled documents and applications.
- A set of COM objects for installing custom audio and video codecs that NetMeeting can leverage during calls.
- New tutorials and sample scripts that demonstrate how to use the *Microsoft NetMeeting 2.1 SDK*.

To download a copy of the *Microsoft NetMeeting 2.1 SDK*, see the NetMeeting Web site at **http://www.microsoft.com/netmeeting/** under Authors & Developers. Note that you must install NetMeeting 2.1 to use the *Microsoft NetMeeting 2.1 SDK*.

The NetMeeting ActiveX Control

With the ActiveX control, organizations can easily integrate the NetMeeting conferencing capabilities into their intranet Web pages. Web site creators can take advantage of established protocols for conferencing over the Internet and corporate intranet. They can create a Web page with NetMeeting conferencing support using JavaScript-compatible languages (JScript) or VBScript. Then, people can join a conference from the Web page.

The NetMeeting control runs on Windows 95 and Windows NT 4.0 platforms, using Microsoft NetMeeting 1.0, 2.0, or 2.1 (NetMeeting 2.0 or later is required for audio and video capabilities). The SDK provides sample HTML files for adding ActiveX control technology to Web pages. For more information, see the *Microsoft NetShow SDK*.

Note If you use Web pages with ActiveX conferencing, you must use a browser that supports ActiveX controls, such as Internet Explorer.

Alternatively, Web site creators can add a "callto" statement using the NetMeeting protocol handler so that people can place a NetMeeting call directly from a Web page. For more information, see the *Microsoft NetMeeting 2.1 SDK*.

Intranet Scenarios

NetMeeting expands the power of the corporate intranet as a tool for communication, information, and learning. Corporate users experience many benefits from these communication and collaboration tools; employees are more productive—they can find information quicker, resolve problems faster, and communicate more effectively. The following scenarios provide examples of how organizations can use the NetMeeting ActiveX control in the workplace.

Note In some cases, an alternative solution to using the NetMeeting ActiveX control would be to create NetMeeting SpeedDial numbers and add them to the intranet Web pages. The SpeedDial numbers enable people to connect using the NetMeeting application. This chapter, however, focuses on using the NetMeeting ActiveX control.

Help Desk: An Employee Needs Help with a Hardware or Software Problem

Scenario

An employee experiences a hardware or software problem and needs to reach a help-desk or PC repair representative within the company as soon as possible. The representative needs to resolve the problem quickly so that the employee can continue working with a minimum loss of productivity.

Solution

A company can use the NetMeeting ActiveX control to develop a PC Support page that provides an in-person response to hardware and software problems. An employee can go instantly to the page on the corporate intranet when they have a problem. In conjunction with NetMeeting support, the page can include a troubleshooting area. From this area, the employee can make selections that narrow down the type of hardware or software problem. For example, the employee may select an application, such as Microsoft Word, from a list. Then, he or she may select a specific problem area, such as Fonts, from a list of possible problem areas. Based on these selections, the server can connect the employee with an expert in the specific problem area.

After submitting the problem report, the server notifies the employee about how many other employees are also waiting for assistance and approximately how long it will be before an engineer can call back. The engineer then initiates a NetMeeting conferencing call with the employee to discuss the difficulty. The employee can use the NetMeeting application-sharing feature to share the Word application with the engineer, who can quickly locate the problem as the employee watches.

Virtual Meeting: A Manager Wants to Set Up a Conference with Employees at Local and Remote Locations

Scenario

A manager calls an emergency meeting with 10 team members who are located at both local and out-of-state offices. Team members need a simple, convenient method for meeting, because time does not permit traveling to a remote meeting location. Also, the manager needs to distribute pertinent information to all team members.

Solution

The NetMeeting ActiveX control makes virtual meeting setup simple. A company can create a Virtual Meeting page on its corporate intranet. The manager can send e-mail messages to attendees that gives the page URL and the time of the meeting. Attendees can then access the Web page prior to the conference to view the agenda, and download files pertinent to the meeting.

At the scheduled time, all the team members can open the intranet Web page and join the conference with data conferencing features, such as application sharing, chat, and file transfer. Attendees can follow along as the manager shares a presentation with the group using the NetMeeting application-sharing feature. During the meeting, one person takes notes and action items using the NetMeeting chat program. In addition, the manager can use the NetMeeting file transfer capability to send a copy of the presentation to each meeting attendee. The team completes the meeting's objectives successfully.

Group Collaboration: Employees at Different Locations Need to Collaborate for Information and Problem Solving

Scenario

Employees need an effective method for collaborating with each other, whether they are separated by a hallway, a building, or a continent. For example, within a manufacturing organization, engineering facilities, factories, and suppliers may need to coordinate their efforts to produce an assembled part. Delay in communication among these groups may slow the manufacturing process.

Solution

A company can use the NetMeeting ActiveX control to develop interactive Web pages for selected departments, such as Accounting, Engineering, and Human Resources. Using these intranet Web pages, the groups can quickly connect and communicate information and ideas with each other. If the factory needs to consult with an engineer because a part is not working correctly, and they cannot interpret the engineering blueprints, they can access an Engineering Web page from the factory floor. Then, they can initiate an ActiveX conferencing call with the appropriate engineer.

A series of events can occur, all made possible by the ActiveX control. The factory can show the part to the engineer, using a camera and NetMeeting video capabilities. The engineer can share the existing blueprints with the factory using the NetMeeting application-sharing feature. Then, the engineer can modify the part and call the supplier that builds the device. The engineer can use NetMeeting's file transfer capability to send the new blueprints to the supplier. The supplier can correct the problem and send the new part back to the factory, with only a minimum delay in the manufacturing process.

Benefits Information: Employees Want to Learn About Company Benefits

Scenario

Employees often have questions about the benefits provided by their company. "When can I enroll in the 401(k) plan?" "How do I purchase stock options?" They need a simple, easy method for locating benefits information and for contacting a human resources representative who can respond directly to questions.

Solution

Using the ActiveX control, companies can create a home page on the intranet where employees can go to learn more about benefits and policies. From the home page, employees can select options to get information about the 401(k) plan, medical and dental insurance, stock options, payroll, and vacation time. Employees can read fact sheets and responses to frequently asked questions about the selected benefit option. Then, they can click an icon to initiate a conference with a representative to get additional information. With the ActiveX control, the representative can transfer files with pertinent benefit information. Employees can easily stay informed about company benefits and policies.

Corporate Sales: Employees Want to Purchase Company Products and Services

Scenario

A company store or sales office may offer products and services for employee purchase. Employees need a simple method for getting product information and ordering a desired product or service that doesn't require time away from the office for ordering and delivery.

Solution

A company can develop a product and service catalog on the corporate intranet. Employees can access the Web catalog and select from a list of options to obtain the following types of information:

- Product specifications
- Availability of products
- Order forms to create and submit an order
- Payment information
- Delivery times and request forms

An employee can enter credit-card information directly on the intranet Web page. Optionally, the employee can click an icon to initiate ActiveX conferencing with a sales representative who can accept the payment details in person and also provide additional information about a product. By streamlining this process, employees can easily purchase company products and services from their desktops.

Discussion Groups: Employees with Common Interests Can Participate in Group Discussions

Scenario

An organization wants to provide a mechanism that enables employees to hold technology discussions and share other common interests from their local desktops.

Solution

Web pages with the ActiveX control can facilitate information sharing among employees. Employees with common interests can visit a Web page to share their knowledge and provide insightful information to other employees who want to learn about a new technology or topic. Organizations can build work-related communities based on these topics of interest.

For example, NetMeeting enthusiasts can go to a Web page and join a conference to talk with other NetMeeting users. During group discussions, experts can join the call to respond to questions or issues. Optionally, an employee can initiate a one-on-one discussion through ActiveX conferencing with a NetMeeting expert who is currently online.

CHAPTER 21

FrontPage Express

In this chapter

FrontPage Express Overview 376
- Flexible Authoring Environment 376
- Table Creation and Editing 377
- Forms Creation 377
- Templates and Wizards 378
- Support for Web Technologies 378

Using FrontPage Express 379
- Creating and Editing Web Pages 379
- Creating a Custom Template 381
- Inserting WebBot Components 381
- Inserting Scripts and Objects 382

FrontPage Express Overview

The features that distinguish FrontPage Express include the following:

- Flexible authoring environment
- Table creation and editing
- Forms creation
- Templates and wizards
- Support for Web technologies

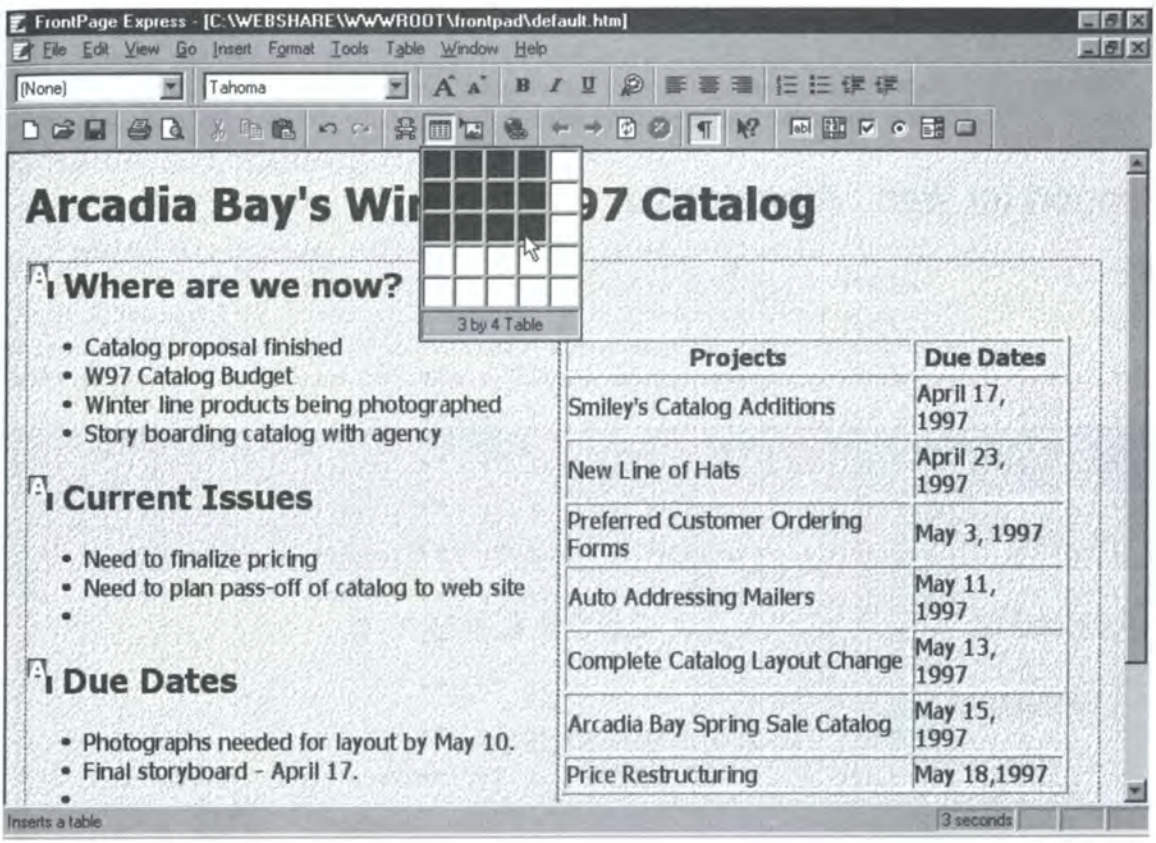
Flexible Authoring Environment

FrontPage Express offers two ways to create and edit Web pages. The what-you-see-is-what-you-get (WYSIWYG) editor displays a graphical representation of what the Web page will look like when it's published. Users can type the contents directly onto the page and highlight text to change the color, size, and style. Things such as background colors and graphics, text and link colors, margins, and base location can be easily specified in the **Page Properties** dialog box. Generating tables is effortless with the Insert Table feature. FrontPage Express creates all the HTML code automatically, based on user input.

FrontPage Express also allows authors to view the actual HTML that is being generated. It adds color codes, indents, and formats to the HTML so it is easy to read. In fact, authors who are comfortable working with HTML can even edit in this mode and instantly see the results in the FrontPage Express application.

Table Creation and Editing

With FrontPage Express, it's easy to insert a table into a Web page and then edit either the entire table or individual cells. This feature also makes it easy to create nested tables without knowing the HTML that it normally requires.



Forms Creation

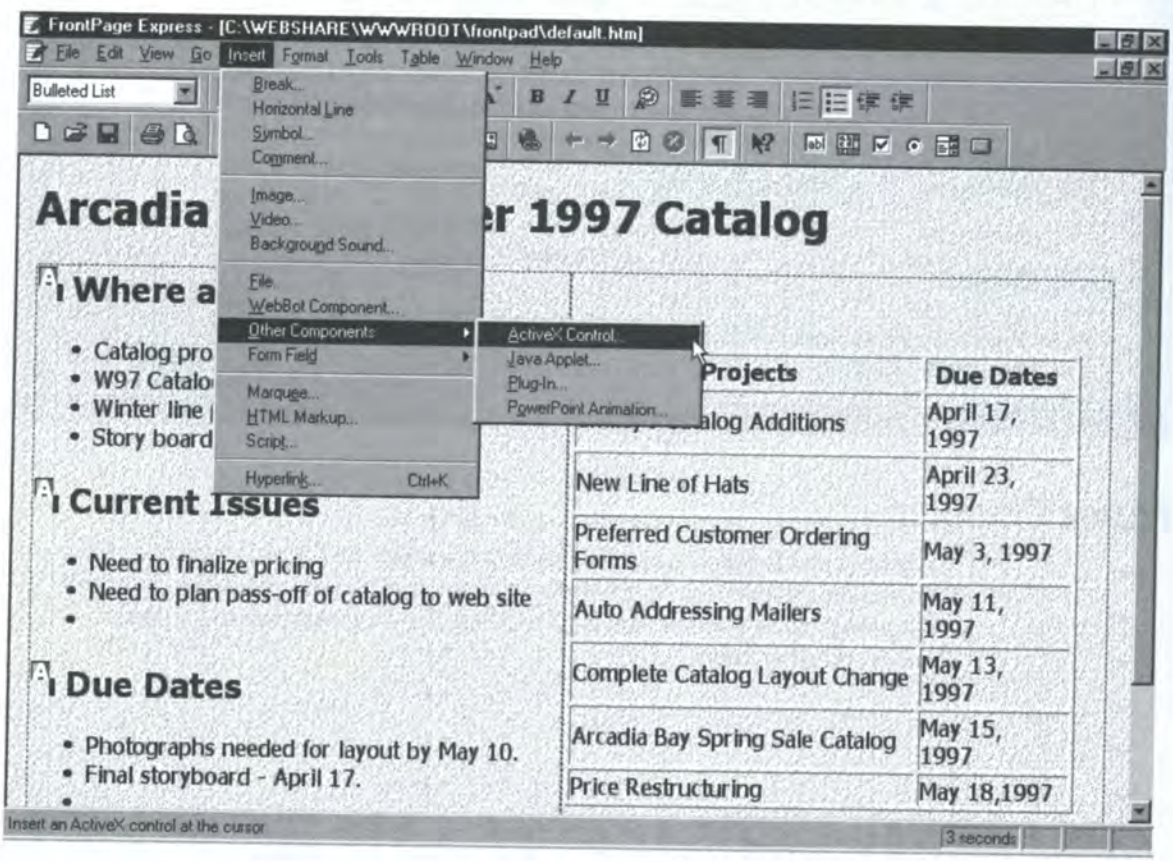
You can add forms to your Web page that people can fill out and return. Your forms can include text boxes, drop-down menus, images, and more. You must be connected to a server running FrontPage server extensions to use these form-related features.

Templates and Wizards

If you're connected to a server running FrontPage server extensions, you can also use form-related templates and wizards in FrontPage Express. Templates are preformatted Web pages that can be used as a guide in creating your own pages and Web View folders. Wizards walk users through the step-by-step process of creating a Web page. Templates and wizards let you create the forms you want simply by selecting the types of information you need. You can also create a survey to collect information from readers and store it on your Web server. Or you can create a page to acknowledge that you have received the reader's input.

Support for Web Technologies

FrontPage Express supports top-standard Internet technologies to make your pages more engaging, without requiring programming knowledge. FrontPage Express supports the insertion of JavaScript, Java applets, VBScript, and ActiveX objects. Form elements, such as text boxes, buttons, and drop-down menus, can be added to pages directly from the toolbar, and users can easily edit their properties.



Using FrontPage Express

Creating and editing Web pages with FrontPage Express is easy. Its editing window shows you exactly what your page will look like, and all you need to do is add the text, links, graphics, and other elements to the page. After FrontPage Express is installed, an **Edit** button is added to the Internet Explorer toolbar so that you can launch FrontPage Express directly from the browser window to edit the current Web page.

Creating and Editing Web Pages

▶ **To start a new page**

1. Start FrontPage Express. A new blank page is displayed by default. You can start working on this page, or continue to step 2.
2. On the **File** menu, click **New**. The **New Page** dialog box appears.
3. Select a template or wizard from the menu. Selecting **Normal Page** gives you the same blank page that comes up when FrontPage Express is started.
4. Set the page properties by right-clicking anywhere on the page and clicking **Page Properties** on the shortcut menu. Here you can choose the background color or graphic, define colors for the various types of text on the page, set the margins, and specify header information.
5. To enter text, click your cursor on the page, and start typing. Editing the text is done the same way as with a word processor—just highlight the text you want to change, and choose the style from the toolbar or from the **Format** menu.
6. Use the **Insert** menu for inserting graphics, video clips, horizontal lines, scripts, and other items.
7. Click the **Save** button to save your new file.

▶ **To edit an existing page**

1. Start FrontPage Express.
2. On the **File** menu, click **Open**, navigate to the file you want to edit, select the file, and click **OK**. To open and edit a Web page from within Internet Explorer, click the **Edit** button on the Internet Explorer toolbar.
3. Make the desired changes to the Web page, and save your changes.

You can edit the registry to make Internet Explorer use a different HTML editor. The registry key that needs to be changed is:

HKEY_CLASSES_ROOT\htmlfile\shell\edit\command

The value for FrontPage Express is:

C:\Program Files\FPXpress\bin\fpexpress.exe %1.

For Notepad, this value can be changed to:

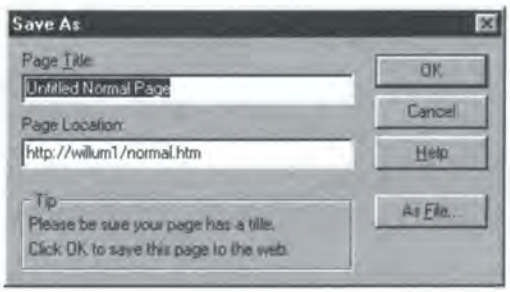
C:\Windows\Notepad.exe %1

For FrontPage 97 or FrontPage 98, this value can be changed to:

C:\Program Files\Microsoft FrontPage\bin\fpeditor.exe %1

When you open a Web page in FrontPage Express, it displays icons to represent certain items in the graphics display. Scripts, objects, and unrecognized HTML markup are each represented by their own special icon. These items can be edited by double-clicking the icon.

Note When you save a page in FrontPage Express, you can save the page as a file or add the page to a Web site using the Web Publishing Wizard. If you're using Microsoft Personal Web Server, save the page as a file to a location under Webshare\wwwroot. Use the Web Publishing Wizard to publish your Web files to an Internet service provider (ISP) or other file server. The Web Publishing Wizard is automatically activated unless you click the **As File** button in the **Save As** dialog box. For more information, see "Understanding the Web Publishing Wizard" in Chapter 22, "Personal Web Server."



Creating a Custom Template

You can create your own custom page templates for use with FrontPage Express. This is useful if you frequently need to create a certain type of page. Template files are stored in the C:\Program Files\Microsoft FrontPage Express\Pages folder by default.

▶ **To create a custom template**

1. Open the Pages directory and create a new folder. When naming the folder, be sure to include a .tem file name extension so FrontPage Express recognizes it as a template folder. The template folder will contain two files: an .htm file that contains the template, and an .inf file that tells FrontPage Express what to display in the menu.
2. Create the HTML file for your template and save it in the Pages directory.
3. Use Notepad to create the .inf file. It should look something like this:

   ```
   [info]
   title=Normal Page
   description=Create a blank Web page.
   ```

   The title shows up in the menu of the **New Page** dialog box. The description shows up when a user clicks the template title in the **New Page** dialog box.
4. Store the HTML and .inf files in the .tem folder you created in the Pages directory. The next time you click **New** on the **File** menu, you will see your new template in the menu along with the default templates and wizards.

Inserting WebBot Components

WebBot components let you add functionality to your Web pages. FrontPage Express offers three WebBot components:

- The **Include** component lets you insert the body of another HTML page into the current page. This is useful for serial elements, such as headers and footers, that appear on multiple pages. Instead of having to edit every instance of a serial element, all you have to change is the included file.

- The **Search** component provides a convenient way to create a search page for users to search the contents of your Web site.
- The **Time Stamp** component automatically inserts the date and time a page was last updated.

▶ **To insert a WebBot component**

1. Position the blinking cursor where you would like the WebBot component to appear.
2. Click the **WebBot** button in the toolbar; or, on the **Insert** menu, select **WebBot Component**.
3. Select the WebBot component you would like to insert, and click **OK**. The **Properties** dialog box appears.
4. Enter the properties for the WebBot component, then click **OK**.

Inserting Scripts and Objects

Scripts and objects can help you bring your Web pages to life by adding a new level of functionality and interactivity. FrontPage Express makes it possible to insert client-side scripts, ActiveX objects, and Java applets directly into your Web pages.

▶ **To insert a script**

1. On the **Insert** menu, click **Script**. The **Script** dialog box appears.
2. Select the option for the type of script you are using, and paste the body of the script into the text window.
3. Click **Extended** to add attributes to the <SCRIPT> tag.
4. Click **OK** when you are finished. An icon representing the script appears on the screen. To edit the script at any time, double-click the icon.

▶ **To insert an ActiveX object**

1. On the **Insert** menu, click **Other Components**, then click **ActiveX Control**. The **ActiveX Control Properties** dialog box appears.
2. Select the control you would like to insert from the drop-down list box.
3. Under **Layout,** select the object's alignment, size, and border thickness. Click **Properties** to edit the properties of the control.
4. Click **OK** when you are finished.

CHAPTER 22

Personal Web Server

In this chapter

Personal Web Server Overview 384
Integration with Windows 384
Ease of Installation, Use, and Management 385
Standards-Based Technology 386
Expanded Programmability Features 386
Understanding Personal Web Server 387
Using Personal Web Server 388
Administering Personal Web Server 389
Adding Web Pages and Files 390
Using an ASP Redirect Page 390
Personal Web Server Scenarios 391
A Simple Document-Sharing Solution 391
A Custom Information-Exchange Solution 392
A Web Application-Testing Solution 392
Understanding the Web Publishing Wizard 393

Personal Web Server Overview

Microsoft Personal Web Server turns any Windows 95– or Windows NT 4.0–based computer into a Web server, enabling easy publication of personal Web pages. Easy to install and administer, Personal Web Server simplifies the sharing of information on corporate intranets or the Internet. Individual users can serve Web pages directly from their desktops, and workgroups can share information with each other or expose their projects to a wider audience.

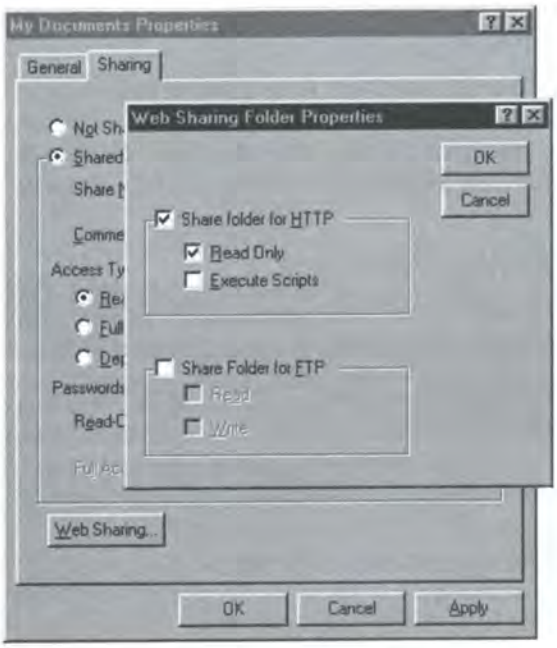
Because Personal Web Server supports Active Server Pages, it can be used as a development and testing platform for Web sites. This lets Web developers test every aspect of a site directly on their local machine prior to posting it on an Internet or intranet server. Personal Web Server also includes support for all Internet Server API (ISAPI) extensions and CGI scripts, which offers the benefit of faster development. Personal Web Server is optimized for interactive workstation use and does not have the system requirements of a full Web server such as Internet Information Server (IIS).

The features and benefits that distinguish Personal Web Server include the following:

- Integration with Windows operating systems
- Ease of installation, use, and management
- Standards-based technology
- Expanded programmability features

Integration with Windows

Personal Web Server turns a Windows 95–based personal computer into a low-volume Web server, making it as easy to share HTML and FTP files over intranets and the Internet as it is to share and print document files over a network.



The software is fully integrated into the Windows 95 taskbar and Control Panel, allowing users to start and stop services whenever needed, administer the server, or change general options. Personal Web Server was designed to be complementary to larger Web server products such as Windows NT IIS and Peer Web Services included with Windows NT Workstation version 4.0.

Ease of Installation, Use, and Management

Personal Web Server installs in minutes and includes the Personal Web Manager, an intuitive, HTML-based administration utility that supports full remote administration. You can easily start and stop the service, view connection statistics, and manage virtual directories. The Home Page Wizard provides an easy way for novice users to get started creating Web pages, and the Publishing Wizard automates the process of publishing Web files to an Internet Service Provider (ISP). Personal Web Server supports both user-level and local security, ensuring flexible and effective protection of sensitive corporate information. Users can set up Personal Web Server to support Windows NT Challenge/Response encrypted-password transmission.

Standards-Based Technology

Personal Web Server 4 supports Active Server Pages and existing standards such as CGI, and includes the open ISAPI extension to the Win32® API, which is up to five times faster than CGI-based applications. This enables any user to take advantage of Active Server Pages, ISAPI, and CGI scripts.

Expanded Programmability Features

The following components can be installed to further extend the potential for developing Web-based applications using Personal Web Server:

- **Microsoft Transaction Server (MTS).** MTS is a component-based transaction processing system. It is used for developing, deploying, and managing high-performance, scalable, and robust server applications for the enterprise, Internet, and intranet. MTS defines an application programming model for developing distributed, component-based applications. It also provides a run-time infrastructure and a graphical tool for deploying and managing these applications. For more information about Microsoft Transaction Server, see the Personal Web Server online documentation, or visit **http://www.microsoft.com/transaction/**.
- **Microsoft Data Access Components.** These components facilitate database access with support for ActiveX Data Objects (ADO) and the Microsoft Access driver. Also included is Microsoft Remote Data Service (RDS), a fast and efficient data connectivity and data publishing framework for applications hosted on Internet Explorer. For more information about ADO, see the Personal Web Server online documentation, or see **http://www.microsoft.com/data/ado/**.
- **Microsoft Message Queue Server.** MSMQ makes it easy for application programs to communicate with other application programs quickly, reliably, and asynchronously by sending and receiving messages. The key features of MSMQ, such as ActiveX support, comprehensive security controls, powerful administration tools, an extensive feature set, and integration with strategic Microsoft products such as Internet Information Server and MTS, make MSMQ the message-queuing product of choice for applications running on Windows 95, Windows 98, and Windows NT operating systems. The Personal Web Server package includes the MSMQ Dependent and Independent Clients. For more information about MSMQ, see the Personal Web Server online documentation, or see **http://www.microsoft.com/ntserverenterprise/** (look under Product Guide, Features).

- **Internet Connection Services for Remote Access Systems (RAS).** This is a collection of software applications designed to help corporations and ISPs build comprehensive Internet access solutions, including dial-up Virtual Private Networks (VPN). Internet Connection Services for RAS let administrators quickly implement a custom remote access network. For more information about ICS for RAS, see the Personal Web Server online documentation.

Understanding Personal Web Server

When Personal Web Server is installed, a variety of files and directories are added to your computer. The C:\Inetpub directory is of primary concern to users, because its subdirectories store all files related to publishing on the site.

The subdirectories of C:\Inetpub are as follows:

- The Iissamples directory stores all of the Web files that are used by the Home Page Wizard.
- The Scripts directory stores all scripts used on the Web site.
- The Webpub directory stores all documents that are published on the site (for example, Word or Excel files). It effectively performs the same role as the FTP service in earlier versions of Personal Web Server.
- All Web files are served from the Wwwroot directory, which is the home directory by default. Users can add subdirectories to the Wwwroot directory as needed.

The URL for the server is taken from the **Identification** tab of the computer's Network control panel. For example, a computer named "Arnold" would have a URL of **http://Arnold/**. If the computer's name is changed, Personal Web Server automatically updates the URL to reflect the change.

Using Personal Web Server

By default, Personal Web Server runs automatically at startup with the HTTP service turned on. The Personal Web Server icon is visible in the taskbar when it is running. All Web server settings can be accessed through the Personal Web Manager.

▶ **To access the Personal Web Manager**

- Double-click the **Personal Web Server** icon in the taskbar, or right-click the icon and select **Properties**. The Personal Web Manager is started. The **Main** window of the **Personal Web Manager** dialog box tells you whether the server is running or not and lets you start or stop the Web server as needed. Connection statistics are displayed in the lower half of the window under **Monitoring**.

Administering Personal Web Server

Personal Web Server is administered on the **Advanced Options** page of the **Personal Web Manager**. Here, users can create or delete virtual directories, edit directory properties, and set directory browsing and logging options. Virtual directories are displayed in a tree-view menu.

▶ **To Administer Personal Web Server**

1. Double-click the **Personal Web Server** icon in the taskbar, or right-click the **Personal Web Server** icon, and select **Properties**. The Personal Web Manager is started.
2. Click the **Advanced** icon in the left pane to open the **Advanced Options** window. From here, you can:
   - Create, edit, and delete virtual directories. A virtual directory is a directory that is not physically contained within the home directory, but which appears as though it were to users who visit your Web site. Virtual directories give you the ability to publish information located on your hard disk. For example, if you created a virtual directory for C:\Documents\Reports and gave it an alias of /Reports, users could view the directory using **http://*<servername>*/Reports/**.

- Set the default document(s) for your Web site. Default documents are the first pages that users see when they access your site, or a subdirectory on your site. Each directory on your site can have a default document.
- Allow users to browse your site by directory. Directory browsing is an alternative way to provide navigation for visitors coming to your site. Instead of providing links on a home page to documents in the home directory, the default document is disabled, and a list of all files in the directory is presented. This is particularly useful if you intend to share documents in their native format.
- Save an Activity log for your Web site. This log file lists everyone who has visited your site and what files they accessed while they were there.

For more information, access the online documentation from the **Help** menu.

Adding Web Pages and Files

For files to be accessible through Personal Web Server, they must be stored in the home directory, or in a virtual directory that you create. You can add new Web files by copying them to the home directory, to a subdirectory of the home directory, or to a virtual directory.

Using the Publishing Wizard, you can publish other types of files (such as word-processed documents and spreadsheets) on your site. This allows users to view the files in their native format. Directory browsing must be enabled for this to work properly.

Using an ASP Redirect Page

An ASP redirect page can be used to redirect users to another location on your site. This can come in handy for certain situations. For example, if you need to take your site down for maintenance, you can redirect users to an alternate page (which is friendlier than just leaving them with a 404 error). Or, if most of the files you want to publish are stored in a certain directory, you can use an ASP redirect page to automatically send users to that directory, eliminating the need to change your default home directory.

▶ **To use an ASP redirect page**

1. Open the **Personal Web Manager**, click the **Advanced** icon, and create a virtual directory for the directory to which you would like to point.
2. Using Notepad, create the .asp file containing the redirect script, and save it to the root of the home directory. The contents of the file should look something like this:

```
<%@ LANGUAGE="VBSCRIPT" %>
<%
response.redirect "/<virtual directory alias>/default.htm"
%>
```

3. Go back to the **Advanced** page of the **Personal Web Manager**, and type the name of the .asp file in the **Default Document(s)** field. Use a comma to separate it from any other entries.

Now when users access your site, the ASP redirect script points them to the specified location.

Personal Web Server Scenarios

A Simple Document-Sharing Solution

Scenario

A construction company executive organizes the computer files by project. Each project contains building specification files, that the executive needs to share with the engineers. Each project also contains sensitive financial information to which the engineers should not have access. Because the project directories contain information the engineers should not access, the specification files must be copied to another directory. As changes are made to the specifications by customers, however, the copy available to the engineers must be updated.

Solution

Because the project directories contain information to which the engineers should not have access, the specification files are copied to the publishing directory. The engineers have the same software on their computers to view the files as the executive used to create the files; therefore, the documents do not need to be converted to HTML to be shared.

The executive creates a simple home page by using the Personal Web Server Home Page Wizard. He uses the Publishing Wizard to place copies of the files in the publishing directory. The wizard automatically adds links to the documents to the home page. The files can be easily updated by using the File Refresh feature built into the wizard. The executive sends the Web site address in an e-mail message to the engineers.

By using browsers such as Internet Explorer, the engineers are now able to view the specification files. The files are viewed in their native formats. The engineers are not able to view other documents in the project directories, because the project directories are not shared. Because the publishing directory folder has read permissions only, the files cannot be changed by the engineers.

A Custom Information-Exchange Solution

Scenario

The director of human resources at a small advertising firm wants to know how satisfied employees are with the current health-insurance carrier before spending time looking for potential new carriers. The director wants to keep the responses confidential, and doesn't want to compile results from a number of paper surveys. She would like to use an online survey; however, the entire graphic art department works on Macintosh computers, while the remainder of the employees have Windows-based computers.

Solution

Using the Home Page Wizard, the director builds an HTML-based Web page that includes a link to a survey form. She follows the cut-and-paste directions in the Personal Web Server documentation topic "Constructing an Interactive Form" to produce the survey form. She then sends out the Web address of the home page in an e-mail message.

Because the form is HTML-based, the employees are able to use their browsers to view and respond, regardless of the operating system they are using. The director has all the responses in a database file, which can be easily used to compile the results.

A Web Application-Testing Solution

Scenario

A Web application developer works as a private contractor, selling Web applications to various companies. He uses his home computer for much of the work and then sends the applications either to a company's Web server or to an Internet service provider. The present contract calls for the development and testing of a unique calendar component to include on a client's home page.

Solution

The developer has a choice of creating a calendar component using Microsoft Visual Basic or any other language program, or using an off-the-shelf, third-party component. Because the customer has requested something extraordinary, the developer elects to create a component using Visual Basic.

Once the component is finished, the developer registers it using the Microsoft Transaction Server Explorer. The component is then included when building the Web application, easily done with the Microsoft Visual InterDev™ Web development system.

As a final step, the developer loads the application on a sample Web site, hosted on Personal Web Server, and tests the calendar. When the developer is satisfied that the application is working properly, the page is sent to the company's Web server.

Understanding the Web Publishing Wizard

The Web Publishing Wizard (not to be confused with the Publishing Wizard feature of Personal Web Server) makes it easy to post Web pages on the Internet or an intranet by automating the process of copying files from your computer to a Web server or an ISP.

The Web Publishing Wizard can automatically post to a variety of Web servers and offers support for:

- Standard protocols: FTP, UNC, and HTTP Post.
- Third-party services, such as AOL, GNN, and Sprynet Primehost.
- System-independent protocols, such as CRS and FrontPage Extended Web.

The Web Publishing Wizard also supports the following ISPs:

- CompuServe
- America Online (AOL)
- Sprynet
- GNN
- America Online Primehost
- Microsoft FrontPage

ISPs that have their own protocol scheme for uploading files to their Web servers can write a custom WebPost Provider .dll file and distribute it from the Microsoft Web site at **http://www.microsoft.com/windows/software/webpost/**. For details on this procedure, send e-mail to **WebPost@lists.msn.com** expressing your interest in writing a provider .dll file. Code for a sample WebPost provider is included in the *Microsoft ActiveX SDK*.

The Web Publishing Wizard connects to the ISP, determines the protocol needed to copy the files, and then uploads the files to the appropriate directory on the ISP computer. Before publishing a Web site with the Web Publishing Wizard, users need to have the following information:

- The local path to the Web content that the user wants to publish (such as C:\WebFiles).
- The protocol used by the ISP to upload files. Users can choose from FTP, HTTP Post, or CRS. (FTP is the most commonly used protocol.)
- The URL to which files are to be uploaded (for example, **ftp://my_isp.com/** if the ISP requires users to post files using FTP).

- The URL for the user's root on the Web server. This is often the URL for the ISP, plus the username (for example, **http://www.my_isp.com/*username/***).
- The user name and password for the account.

The Web Publishing Wizard guides you through the steps of connecting to your ISP or intranet site and automatically uploads Web files from a directory you specify. Because it saves all the information that you enter, you won't be asked to enter it during subsequent publishing efforts, making future publishing efforts quick and convenient.

PART 6

Authoring for Internet Explorer 4

Microsoft Internet Explorer 4 supports a wide range of new features that publishers can use to create innovative and interactive Web pages based on the newest Internet technologies and standards. Using this set of technologies, Web authors can produce more attractive content while taking up less bandwidth. They can also develop consumer and business applications that deliver more value and create a unique, fun, and more rewarding Web experience.

Chapter 23: Dynamic HTML

Limitations of current browser technology often require authors using Hypertext Markup Language (HTML) to choose between interactivity and speed. Once a page is loaded, changing the display or content of the page typically requires that the entire page be reloaded. Or, depending on how extensive the changes are, additional pages may have to be retrieved from the server. To address that limitation, Internet Explorer 4 includes several new features collectively called Dynamic HTML. These features provide the client-side intelligence and flexibility to enable HTML authors to create interactive pages that dynamically change the display and content of a page entirely on the client machine, without requiring additional server resources. Dynamic HTML enables HTML authors to create innovative Web sites with interactive pages without having to pay a performance penalty. In addition, by reducing requests from the server, Dynamic HTML also reduces server load, thus improving performance.

Chapter 24: ActiveX

For those who need more power than Dynamic HTML offers, Internet Explorer 4 provides enhanced support for ActiveX™ controls. ActiveX controls are software components that provide interactive and user-controllable functions. Small and versatile, they open a wide range of opportunities for creating attractive and effective Web content. Internet Explorer 4 presents a set of new opportunities for control and script engine developers. Most notably, Internet 4 is an OC96-compliant host. Controls can now be faster, smaller, and more integrated than ever before.

Chapter 25: Java

You can also use Java to extend the capabilities of Dynamic HTML and create even more interactive, exciting Web content. While Internet Explorer 3 brought the fastest, most reliable, and fully compatible Java support to the market, Internet Explorer 4 goes even further, taking Java support to the next level. For example, you can use Microsoft Java multimedia class libraries to provide true multimedia effects in your Web pages. Java applets not only run faster than in Internet Explorer 3, but you can optimize Java applets to take full advantage of all the features and speed of the Windows® 95 and Windows NT® operating systems. Plus, Internet Explorer 4 introduces a Java security model that allows you to run powerful, interactive Java applications without harming your organization's computers or threatening its privacy.

Chapter 26: Active Scripting

With its support for active scripting, Internet Explorer 4 provides fast, comprehensive, language-independent script-handling capability. Users can view Web pages that use any popular scripting language, including Visual Basic® Scripting Edition (VBScript) and JScript™ development software. Internet Explorer scripting support allows authors to develop pages that can ask questions, respond to queries, check user data, calculate expressions, link to other programs, and connect to ActiveX controls, Java applets, or anything else you specify. In addition, Internet Explorer 4 complies with the new Internet scripting standard specified by the European Computer Manufacturers Association (ECMA). Compliance to Internet standards helps ensure that the Web content and active scripting you develop for Internet Explorer 4 will run seamlessly across a wide range of platforms and products.

CHAPTER 23

Dynamic HTML

In this chapter

Dynamic HTML Overview 398
Dynamic HTML and Internet Standards 399
Understanding the Document Object Model 400
Using Dynamic HTML 401
- Using Dynamic Styles 401
  - Using Style Sheets 401
  - Using Font Embedding 403
  - Positioning HTML Elements 405
  - Using Visual Filters and Transitions 406
- Using Dynamic Content 406
- Using Data Binding 407
- Using Scriptlets 409
  - Creating Scriptlets 410
  - Adding Scriptlets to Applications and Web Pages 411
  - Understanding Scriptlet Security 412

Dynamic HTML Overview

The most significant authoring innovation introduced by Internet Explorer is Dynamic HTML, a set of features that make it easy for developers to build interactivity into applications without burdening their servers. Dynamic HTML extends the Hypertext Markup Language (HTML) standard, giving developers the ability to control virtually anything on a Web page at any time.

You can use Dynamic HTML to create fast, compact, and interactive Web pages without the need to load new content from the server or use complex ActiveX controls and Java applets. Using only HTML and a few lines of script, you can change any element on a Web page based on user actions, and Internet Explorer will immediately display the changed content and reflow unchanged content as needed, without having to reload the page from the server. To optimize browser performance and speed, Internet Explorer redraws only those portions of the browser window that are affected by the change.

With previous browser technology, changing the display or content of a Web page typically required the entire page to be reloaded from the server. You could use scripts to change page display and content on clients, but only while the Web page loaded. Only a few, very limited elements in HTML could be changed after Web pages had loaded. When designing Web content, Web publishers had to choose between interactive pages and speed. Now, with Dynamic HTML, you can create dynamic, interactive pages without having to pay a performance penalty. You can also use Dynamic HTML to reduce Web server load and optimize server performance.

The following sections provide an overview of each of the key features of Dynamic HTML. Additional information about Dynamic HTML is available from a variety of resources, including:

- *Inside Dynamic HTML* (Microsoft Press, 1997) and the companion Inside Dynamic HTML Web page at **http://www.insidedhtml.com/**, which provide a comprehensive programmers reference for Dynamic HTML as well as online tips, articles, and samples showing how to use the full power of Dynamic HTML to create Web content.
- *Microsoft® Internet Client Software Developers Kit (SDK)*, which includes an extensive section on Dynamic HTML that provides a comprehensive Dynamic HTML language reference, design guidelines, and samples. The *Microsoft Internet Client SDK* is available at **http://www.microsoft.com/msdn/sdk/inetsdk/**.
- Microsoft Site Builder Network (SBN), which provides a wide range of resources to help you understand and use Dynamic HTML as well as other technologies for creating state-of-the-art Web sites. You can visit SBN at **http://www.microsoft.com/sitebuilder/** and join to become a member.

- SBN Workshop (**http://www.microsoft.com/workshop/**) provides technical workshops on authoring, designing, and programming Web sites. Each workshop provides links to tutorials, tools, and other resources available from Microsoft and from third parties. The workshops on authoring and design provide extensive information on creating and designing Web content using the features of Dynamic HTML.
- *Site Builder Magazine* (**http://www.microsoft.com/sitebuilder/columnists/**) provides up-to-date articles written by industry-leading experts on Web authoring, design, programming, and technology.

For information about troubleshooting authoring for Internet Explorer and for Frequently Asked Questions (FAQs) about Dynamic HTML, see Appendix G and I, respectively.

Dynamic HTML and Internet Standards

The implementation of Internet standards is crucial to enhancing the Web experience. Now, users can view and listen to real-time broadcasts, watch videos, run ActiveX controls and Java applets, and play interactive, multiplayer games over the Web. All of this can be experienced in Internet Explorer 4 due to the browser's extensive support for Internet standards and because it includes support for innovative, underlying technologies, such as Dynamic HTML, ActiveX, and Java.

Microsoft is committed to working with the World Wide Web Consortium (W3C) and other standards bodies to make the Web a better open environment for building efficient and interactive multimedia content. Microsoft is also committed to providing the best standards-based solutions. Dynamic HTML represents the next step in that commitment. Previously, Microsoft participated in the W3C's efforts to define and promote the use of the <OBJECT> tag as an extensible way for adding objects in an HTML page. That standard resulted in the use of ActiveX controls—objects that authors can insert in HTML pages and that users can view and run in a safe and seamless way. Microsoft is now working with the W3C to extend that paradigm to standard HTML.

Dynamic HTML is based on HTML extensions proposed for HTML 4.0 by the W3C, including the Document Object Model, positioning in Cascading Style Sheets (CSS), and Web font embedding. The extensions to HTML used for Dynamic HTML conform to the latest working drafts and recommendations of the W3C.

Microsoft worked closely with the W3C when developing Dynamic HTML and has submitted Dynamic HTML to the W3C for inclusion in HTML 4.0. The specification for HTML 4.0 is currently a work in progress and requires final ratification by the W3C. Microsoft is committed to providing future enhancements to Internet Explorer to conform to all specifications recommended by the W3C.

For more information about Internet standards, visit the Microsoft Specifications and Standards page at **http://www.microsoft.com/standards/** and the W3C home page at **http://www.w3c.org/**.

Since Dynamic HTML is an extension to the HTML standards, you can design Web pages for Internet Explorer 4 that offer rich, interactive content but that *gracefully degrade* in browsers that don't support the Dynamic HTML extensions. Graceful degradation ensures that older versions of browsers can display standard HTML elements while ignoring any Web page features that use Dynamic HTML. For more information about graceful degradation, see the *Microsoft Internet Client SDK*.

Understanding the Document Object Model

Dynamic HTML uses the Document Object Model, an open, language-independent object model for giving scripts access to all elements and attributes of standard HTML. The Document Object Model extends the object model used in Internet Explorer 3 to enable scripts to dynamically change the style, content, and structure of a Web page after the page has loaded. Basically, the Document Object Model exposes all HTML elements and browser window objects to scripts, so scripts can be used to manipulate everything that makes up the content displayed in the browser window. The Document Object Model also enables you to use events, such as user mouse clicks, to control dynamic changes to styles.

With previous object models for browsers, scripts could only be used to change a Web page as the content of pages was loaded. Now, using extensions to the Cascading Style Sheets and a few lines of script such as JScript or VBScript, you can design Web pages that change any element or attribute in an HTML file at any time, without loading any new content from the server. You can also control scripts based on a full set of bubble-up events; for example, when users click a hot spot or move the mouse cursor over an area of the page.

You can use extensions to Cascading Style Sheets and a few lines of script to change the entire makeup and appearance of a Web page, including changing HTML tags, the exact position of elements on the page, and the text that appears on the page. When Web pages change, the changes are rendered immediately in the browser window.

For more information about the Document Object Model, see "Document Object Model" in the *Microsoft Internet Client SDK*, and visit the W3C Document Object Model Web page at **http://w3c.org/DOM/**.

Using Dynamic HTML

Dynamic HTML is an umbrella term for several improvements in Internet Explorer 4. These enhancements fall into the following categories:

- Dynamic styles
- Font embedding
- Positioning
- Filter, transition, and animation controls
- Dynamic content
- Data binding
- Scriptlets

Using Dynamic Styles

Dynamic styles give you comprehensive control over the styles in Web pages. Dynamic HTML extends Cascading Style Sheets (CSS) to add several powerful new features. You can use dynamic styles to:

- Position HTML elements at absolute or relative positions on the page.
- Create multimedia-style effects using standard HTML elements with minimal or no graphics using filters and transitions.
- Create interactive Web features and dynamically change the appearance of HTML elements without needing to reload Web pages from the server or use complicated controls.

The following sections provide an overview of dynamic styles. For more information, see "Dynamic Styles" in the *Microsoft Internet Client SDK*.

Using Style Sheets

Cascading Style Sheets are design templates that help you control the presentation and layout of HTML elements. They allow you to separate the way you design information from the HTML content.

This section provides a brief look at how Cascading Style Sheets work. A number of good resources are available to help you understand and use style sheets. Two of the most comprehensive resources on style sheets are available on the Web:

- W3C Web Style Sheets site at **http://www.w3.org/style/**
- *Web Review* Style Sheets Reference Guide site at **http://style.webreview.com/**

Each of these Web sites provides a wealth of information on style sheets, including tutorials, online examples, book reviews, and lists of additional resources.

Using style sheets, you can create Web pages with minimal graphics, and therefore, much smaller downloads. Style sheets also provide you with a higher level of typographic control, and they enable you to make changes to an entire site through the use of linked style sheets.

You can specify the styles for any HTML element in three ways:

- Link the HTML file to external CSS files.
- Include STYLE element blocks within the HTML file.
- Include a STYLE attribute in individual HTML elements in the HTML file.

You can specify an external style sheet by including a LINK element in the Head section of the HTML file. The following example shows how you could use Sheet1.css to specify the default styles for the Web page.

```
<HEAD>
<LINK HREF="Sheet1.css" REL=Stylesheet>
</HEAD>
```

You can specify STYLE blocks that apply only to the current HTML file. The following example shows a STYLE block that specifies that the text for the P elements on the Web page should display using the 10-Point Verdana font, overriding the external style sheet, Sheet1.css.

```
<HEAD>
<STYLE>
<!--
    P   {font-family: Verdana;
        font-size: 10pt}
-->
</STYLE>
</HEAD>
```

You can specify styles for individual elements using the STYLE attribute. The following example shows how you could specify one instance of the P element to display as 12-point text, overriding the STYLE block.

```
<P STYLE="font-size: 12pt">This is 12-point text</P>
```

Styles are additive. For example, you can use a CSS file to specify the size of all P elements as 12 points; you can use a STYLE block to specify the font face for all P elements in the HTML file as New Times Roman; and you can use a STYLE attribute to specify that an instance of the P element is italicized. If styles conflict, the closest specification takes precedence, as in the previous examples.

The new Document Object Model enables you to access and change all style attributes in an HTML file. Using inline script events and scripts, you can change HTML style attributes based on user actions or *events* such as a mouse click or a response to an HTML form. You cannot use scripts to change the contents of external style sheets (CSS files), but you can use scripts to change all style attributes specified in the HTML file.

Creating dynamic styles is very easy. Once you understand CSS basics, all you need to know is some script syntax—and you can have any HTML tag change dynamically by applying CSS attributes to it.

For example, you can embed *inline script events* to dynamically change the style attributes of an HTML element when users touch it with the mouse (**onmouseover**) or move the mouse off of it (**onmouseout**), as shown in the following HTML code:

```
<H4 onmouseover="javascript:this.style.color= '#6633FF'"
    onmouseout="javascript:this.style.color='#000000'">
This headline will change to purple when the mouse touches it,
and back to black when the mouse moves off of it</H4>
```

When the mouse touches the headline (H4 element), it changes to purple. It also changes back to black when the mouse moves off of it.

Dynamic HTML also gives you several powerful new attributes that you can use to create dynamic content: absolute and relative positioning, and visual filters and transitions. You can now place HTML elements anywhere on a page, specify how the elements are layered, and create multimedia-style effects without having to load bandwidth-consuming controls or graphics files.

Using Font Embedding

The FONT element and the CSS font family attribute enable you to specify the exact font attributes to be used to display your Web pages. If the font is not installed on the user's computer, however, the browser displays the Web page using a default or secondary font face. To address this issue, Internet Explorer 4 supports font embedding, which gives Web designers total control over how their Web pages are displayed. Using font embedding, Web pages can link to compressed font objects, which Internet Explorer will download, decompress, and temporarily install. Thus, Web designers can specify fonts using Cascading Style Sheets or the <FONT FACE=> tag without having to worry about them not being installed.

Internet Explorer uses the Microsoft OpenType Font Embedding technology. Currently, the industry has no standard for font embedding. Microsoft has submitted the OpenType font-embedding technology for consideration by the W3C.

You can use the Microsoft Web Embedding Fonts Tool (WEFT) to embed fonts in your HTML files. WEFT analyzes HTML pages for you, creates an Embedded OpenType (EOT) font object file, and then adds the proper CSS font-embedding information to the HTML pages. For more information, visit the Microsoft WEFT Web page at **http://www.microsoft.com/typography/web/embedding/weft/**.

Font embedding is specified in HTML files using the @font-face definition and standard CSS syntax. The following example shows an HTML file STYLE block that was created using the WEFT. The tool also creates the associated EOT file (the font object), which is a subset of the font that will be temporarily installed on the user's computer to view your Web page.

```
<STYLE>
<!--
@font-face {
    font-family: Mistral;
    font-style: normal;
    font-weight: 700;
    src: url(MISTRAL.eot); }
-->
</STYLE>
```

Font download can be controlled in the **Custom Security Settings** dialog box, available from the **Security** tab of the **Internet Options** dialog box. (On the **View** menu, select **Internet Options**, click the **Security** tab, and then click the **Custom** button.)

TrueType fonts have embedding permissions encoded within them that determine how they can be used for embedding. There are four levels of embedding:

- **No-embedding** prevents fonts from being embedded. The creators of these fonts have decided not to allow embedding. Some foundries set their fonts to no-embedding but offer upgrades to embeddable versions. If you come across a no-embedding font that you would really like to use, contact the supplier and ask about an upgrade.
- **Print and preview** fonts can be embedded, but only in pages with static fonts. If you want to enable dynamic styles so the font can change, use **editable** or **installable**.
- **Editable** fonts can be embedded without the restrictions imposed on **print** and **preview** fonts. These fonts are installed in users' fonts folder.
- **Installable** fonts are treated as editable fonts by Internet Explorer 4, but installable fonts will not be installed in a user's fonts folder. This helps prevent users' fonts folders from becoming filled with fonts that they do not want.

You can use the WEFT to check the permission level of TrueType fonts installed on your system. You can also check the embedding permissions of fonts using the free Microsoft Font Properties Extension, which is available for download at **http://www.microsoft.com/typography/property/property.htm**.

Positioning HTML Elements

Dynamic HTML enables you to position HTML elements in the x- and y-coordinates as well as overlap elements in planes along the z-axis. You can place elements—DIV elements, images, controls, or text—exactly where you want them on the page. You can use positioning to help create 3-D effects. You can use scripts to change the position of elements and animate the page. Browsers that don't support positioning ignore the positioning and display the HTML in its natural flow.

You can also specify absolute and relative positions. Relative positions are measured from where the element would naturally appear on the page. The absolute position is measured from the upper-left corner of the browser window.

The following example shows how you could specify Graphic1.gif to appear starting at 100 pixels from the top of the browser window and 200 pixels from the left side of the browser window.

```
<IMG STYLE="position:absolute; top:100; left:200; z-index: -1"
    SRC="Graphic1.gif">
```

The **z-index** property can be used to layer objects on the page. By default, objects are layered bottom-to-top in the order that they appear in the HTML source. Higher-valued z-index objects are positioned above lower-valued z-index objects. Two elements with the same z-index are layered according to HTML source order. The z-index value can be any positive or negative integer. In the previous example, z-index: -1 ensures that the graphic appears below overlapping text that appears on the page.

The DIV element is often used with positioning because the DIV element can be used to enclose other HTML elements to define distinct areas on the page. Using the DIV element and positioning, you can create a wide variety of spectacular effects on Web pages.

You can use positioning to create 3-D effects by varying the x- and y-axis coordinates and the z-index property of objects. You can also use scripts to dynamically change the positioning of elements on Web pages to create spectacular animation effects. For more information, see "Positioning" in the *Microsoft Internet Client SDK*.

Using Visual Filters and Transitions

Dynamic HTML provides visual filters and transitions that you can use to apply multimedia-style effects to standard HTML elements, including divisions, text, and images. Browsers that don't support filters and transitions simply ignore the filters and transitions.

A *filter* is used to apply a special graphic transformation or effect to an element. For example, you can use a filter to create a mirror image or a drop shadow for an element. A *transition* is a type of filter that enables a transition from one visual state to another. For example, you can use a transition to make each slide in a slide show gradually fade in or out. You can use scripts and transitions to control the way that animated elements appear on the page.

The following example shows a filter that displays a vertical mirror image of Graphic1.gif.

```
<IMG STYLE="filter: flipv" SRC= "Graphic1.gif">
```

Transitions are divided into two types: blend and reveal. *Blend* makes the element gradually blend in or out of the surroundings, while *reveal* makes the element gradually appear or disappear in one of 23 predefined patterns. A primary difference between filters and transitions is that transitions must take place between two states, such as an invisible image and a visible one.

You can use a few lines of script combined with filters and transitions and positioning to provide spectacular animated and 3-D effects. For more information, see "Filters and Transitions" in the *Microsoft Internet Client SDK*.

Using Dynamic Content

Dynamic styles change the appearance of HTML elements on Web pages, but don't actually change the content of the HTML elements. You can use the dynamic content features of Dynamic HTML to change the content of most HTML elements. You can change an element to another element as well as change the text that appears in the element. For example, you could replace a text heading with new text or with a graphic.

The new Document Object Model enables scripts to change text content and HTML elements by manipulating four properties of most visible HTML elements: **innerText**, **outerText**, **innerHTML**, and **outerHTML**. The **innerText** and **outerText** properties specify plain text in elements (any HTML tags appear as verbatim text in the browser window and are not processed as HTML). The **innerHTML** and **outerHTML** properties specify text and HTML tags that are processed in the browser window as HTML. Scripts can also add new text content and new HTML elements to Web pages using the insertAdjacentText and insertAdjacentHTML methods that are supported by most visible HTML elements.

Most dynamic content needs can be met using the **innerText**, **outerText**, **innerHTML**, and **outerHTML** properties, and the insertAdjacentText and insertAdjacentHTML methods. However, you may want to perform advanced text manipulation using the **TextRange** object.

TextRange objects are an advanced feature of Dynamic HTML that you can use to perform useful tasks related to dynamic content, such as searching for and selecting text. **Text range** objects let you selectively pick out characters, words, and sentences from a document.

You can use the dynamic content features of Dynamic HTML to create a wide range of dynamic applications on your Web pages. For example, you could use a script to scan the elements of a page and use dynamic content to insert a table of contents at the beginning of the page. You can control the level of detail in the table of contents based on user actions such as mouse clicks or input from an HTML form. You could also use a script to create dynamic tables of data that a user can sort based on the user's preferences.

You can create many powerful interactive applications using dynamic content. For more information, see "Dynamic Content" in the *Microsoft Internet Client SDK* and visit the SBN Workshop at **http://www.microsoft.com/workshop/**.

Using Data Binding

Data-driven business applications are increasingly taking advantage of the Web, accessing data stored in server databases. These applications often require Web publishers to construct many Web pages for users to navigate through to view and manipulate the data. This can frustrate users as well as increase server loads.

Dynamic HTML provides data-binding features that you can use to integrate data with HTML elements on the Web pages downloaded to clients. You can use data binding with the dynamic content features of Dynamic HTML to create applications that display, sort, filter, and update data on the client with minimal visits to retrieve data from the server.

Dynamic HTML uses HTML tags to define a template that is populated with data on the client, and data source objects to supply data to the page. No specific data protocol or database management system (DBMS) is required to use these data-binding features.

Data source objects supply the data to the Web page. Data source objects can be either Java applets or ActiveX controls. The following data source objects are included with the standard installation of Internet Explorer 4:

- **Tabular Data Control.** The **Tabular Data Control** supplies data that is stored in a delimited text file to the page. The control uses a URL to locate the text file, enabling the data to be transmitted using standard Internet protocols. The file format is easily generated by most DBMS systems and applications. Since the text data files are cached on the client, the control can supply data in offline browsing scenarios.
- **Remote Data Service (RDS), previously known as Advanced Data Connector (ADC).** This data object enables publishers to access data from and transmit data to Open Database Connectivity (ODBC)-compliant databases. **RDS** provides direct, real-time access to server data. **RDS** also transmits data using the HTTP protocol, so **RDS** can work over the Internet and through firewalls.
- **JDBC Data Source Object.** This data source object is a Java applet that implements a JavaBeans-compliant interface. Data is retrieved through the Java Data Base Connectivity (JDBC) data source object and stored in a local cache implemented by the applet. Even though JDBC provides a forward-only cursor, the applet can provide random access to data when used with data binding.
- **Extensible Markup Language (XML) Data Source Object. XML** provides a standards-based way to represent structured data in a text format. The data-binding functionality of Dynamic HTML supports the display of **XML** data within a Web page by using a data source object that enables the **XML** data to be displayed in nested, repeated tables.
- **HTML Data Source Object.** In addition to using external components as data source objects, Web authors can define their data sets within an HTML document and use the Internet Explorer 4 HTML engine to provide read-only data to a page. This enables parts of one HTML document to be included in another HTML document, and all data processing takes place on the client rather than on the server.

As additional data source objects become available, Microsoft will add them to the Data Source Object Gallery at **http://www.microsoft.com/gallery/files/datasrc/**. You can also develop data source objects for use with data binding. For more information, see the *Microsoft Internet Client SDK*.

You specify the data source object to use for data binding by including an <OBJECT> tag in the HTML file. The data supplied by a data source object is bound to HTML elements within the page, using four new attributes that Microsoft has proposed to the W3C:

- DATASRC
- DATAFLD
- DATAFORMATAS
- DATAPAGESIZE

You can bind data to the following tagged HTML elements: <A>, <APPLET>, <BUTTON>, <DIV>, <FRAME>, <IFRAME>, <IMG>, <INPUT>, <LABEL>, <MARQUEE>, <OBJECT>, <SELECT>, <SPAN>, and <TEXTAREA>.

With data binding, you can present data as follows:

- **Repeated tables.** You can merge data with an HTML table on the client, which serves as a template. The content of the table is repeated once for each record in the data set, and headers and footers can be displayed as well. HTML elements within the cells of the table can then be bound to columns from the data set.
- **Current record binding.** You can merge data from the columns of the current record of the data source with HTML elements on the client. The current record can be moved through the data set by calling methods in the Active Data Objects (ADO) model, supplied automatically to every data source object.

For more information, see "Data Binding" in the *Microsoft Internet Client SDK*, and visit the SBN Workshop at **http://www.microsoft.com/workshop/**.

Using Scriptlets

A scriptlet is a Web page based on Dynamic HTML that you can use as a control in any application that supports controls. The scriptlet includes information that allows you to work with it as a control. You can get and set a scriptlet's properties, call its methods, and so on.

Before scriptlets, reusing HTML and script often meant cutting, pasting, and then customizing existing code. With scriptlets, you can author content once, using the HTML and script you know today, and then let others quickly reuse your content as scriptlets in their Web pages and applications. You create scriptlets by following simple conventions to expose well-defined interfaces. Through these scriptlet interfaces, others can customize and reuse existing Web content without having to understand the implementation details.

Since scriptlets consist only of HTML and script, they are easy to create and maintain. Scriptlets are small and efficient, and they quickly download and run on users' computers. Scriptlets can:

- Enable you to make optimal use of your existing knowledge of HTML and script to create reusable components.
- Make Dynamic HTML even easier to use. You need not understand scriptlet details, only the interfaces required to customize the scriptlet. You can take advantage of this existing content in other Web pages and in any application supporting the component object model (COM).
- Increase Web application performance because scriptlets download and run quickly and are stored in the cache on the end user's computer.
- Enable you to create reusable user interface components without having to harness the full power of C, C++, or other control-building environments.
- Enable you to change the appearance and behavior of Web pages from within the host environment. For example, you can use a Visual Basic application to read information from files and then write the information into a scriptlet that is used to change the Web page.
- Enable you to use the features built into Web pages in development environments that support ActiveX controls, such as Visual Basic and Visual InterDev. For example, you could use a scriptlet to provide the graphical and hypertext capabilities of a Web page as the visually rich interface for an application.
- Enable you to quickly prototype controls you intend to write in other environments. Since it's easy to put together a scriptlet, you can create scriptlets to validate your designs quickly. Then, after the design is validated, you can re-implement the control in another environment, such as C++, Visual Basic, or Visual J++™.

Creating Scriptlets

A scriptlet can be either an HTML page or an Active Server Page (ASP). You can use any HTML authoring environment to create a scriptlet, such as the Microsoft FrontPage® Web authoring and management tool or the Visual InterDev integrated Web application development system. Of course, if your authoring environment does not support scripting, you may have to manually insert the scripting code after first composing the HTML display elements.

You use scriptlets like a standard ActiveX control. In the scriptlet, you create any properties, methods, or events that you want by simply creating scripts in common active scripting languages such as JScript or VBScript. The scripts rely on the scripting capabilities of Dynamic HTML, which gives you a complete object model for the elements in the scriptlet.

For example, a scriptlet might be a Web page that contains animation based on Dynamic HTML that moves and resizes text on the page. You can write scripts to expose properties that allow another application to set the text, speed, and direction of the animation text, and to expose methods that allow another application to start, stop, and pause the animation.

You define the scriptlet interface using one of the following methods:

- Use an **ECMAScript Public_Description** object to define explicitly what properties and methods the scriptlet will make available. Any behavior that is not explicitly declared using the **Public_Description** object is not available. For more information, see "Creating a Public_Description Object" in the *Microsoft Internet Client SDK*.
- Use default interface descriptions. You do not explicitly declare properties or methods. Instead, any variables and functions that follow certain naming conventions become available as properties and methods This convention can be used to create scriptlet properties and methods in an active scripting language other than ECMAScript. For more information, see "Using Default Interface Descriptions" in the *Microsoft Internet Client SDK*.

Using a **Public_Description** object has several advantages. You can use any names for variables and functions that you want to expose as properties and methods, because you assign them public names in the **Public_Description** object. In addition, using the **Public_Description** object provides you with a convenient way to summarize and document the properties and methods that the scriptlet exposes.

In contrast, if you use the default interface descriptions, you must use the public_ prefix on any name that you want to expose. If a scriptlet already has a variable or function with the public_ prefix, it will be exposed, whether you want it to be exposed or not.

When you create the properties, methods, and events for the scriptlet, you can make full use of the Document Object Model to display text, animate HTML elements, change colors, or define any other behavior that you want the control to have. You can also use extensions to the Document Object Model to manipulate properties and methods that are unique to scriptlets. For more information, see "Scriptlet Model Extensions" in the *Microsoft Internet Client SDK*.

Support for scriptlets is built into Internet Explorer 4. Use the standard <OBJECT> tag to include the scriptlet in other Web pages.

Adding Scriptlets to Applications and Web Pages

You can use scriptlets in applications that support ActiveX controls and in Web pages.

Applications that support ActiveX controls

To use a scriptlet in applications that support ActiveX controls, you use a control called the *scriptlet container object,* which hosts the scriptlet. The host application, such as Visual Basic, uses the scriptlet container object to create a window for the scriptlet and provide a way for the host application to specify what scriptlet to use, where it displays, at what size, and so on. The scriptlet container object also provides the interface for you to set and get the scriptlet's properties, call its methods, and respond to its events. For more information, see "Adding Scriptlets to Your Application" in the *Microsoft Internet Client SDK.*

Web pages

To use scriptlets in Web pages, you insert an OBJECT element in the HTML file as shown in the following example:

```
<OBJECT width=200 height=123
    TYPE= "text/x-scriptlet"  DATA="Scriptlet.htm">
</OBJECT>
```

In this example, the OBJECT element references the scriptlet file **Scriptlet.htm**. Internet Explorer 4 recognizes the object as a scriptlet by the MIME type **"text/x-scriptlet"**. Note that there is no CLSID in the <OBJECT> tag. Internet Explorer 4 supports this MIME type on all platforms. Because of the open nature of the architecture, competing vendors will be able to implement support for scriptlets as well. Nothing in the nature of scriptlets restricts users to using Microsoft platforms.

Understanding Scriptlet Security

Scriptlets are as secure as Dynamic HTML and ECMAScript itself. A scriptlet recognizes when it is placed in a secure container, such as Internet Explorer, and obeys the security policies of its container.

In general, to operate correctly, a scriptlet must be loaded from the same Web server as its container page (the same as for Java applets). In a security-aware host application such as Internet Explorer, the scriptlet and any controls contained within it are subject to Internet Explorer 4 security.

Internet Explorer 4 uses the security settings for ActiveX controls to enforce security for scriptlets based on the zone of the server where the scriptlet originates. If the scriptlet contains controls that reside on a server in a more restrictive zone, the security settings for the more restrictive zone are enforced. For more information about security settings for ActiveX controls, see "Customizing Security Zones" in Chapter 27, "Security Zones."

For more information, see "Microsoft Scriptlet Technology" in the *Microsoft Internet Client SDK,* and visit the SBN Workshop at **http://www.microsoft.com/workshop/**.

CHAPTER 24

ActiveX

In this chapter

ActiveX Overview 414
Using ActiveX Controls 414
Optimizing ActiveX for Internet Explorer 4 415
Using ActiveX Controls in Web Pages 416
Listing and Removing ActiveX Controls 416
Signing ActiveX Code 419
Using Multimedia Controls 419

ActiveX Overview

ActiveX enables software components to interact with each other in a networked environment, regardless of the language in which they were created. Embracing both Java and Microsoft industry-standard component object model (COM) technology, ActiveX makes writing Internet applications as easy as writing applications for the operating system. You can create ActiveX components that provide standard application features and functions and then easily reuse the components in current networking and stand-alone applications.

The following sections describe how you can use ActiveX with Internet Explorer 4. For more information, see the *Microsoft Internet Client Software Developers Kit (SDK)* and visit the Microsoft Developer Network at **http://www.microsoft.com/msdn/** as well as the SBN programming workshop at **http://www.microsoft.com/workshop/**.

For information about troubleshooting authoring for Internet Explorer, see Appendix G.

Using ActiveX Controls

ActiveX controls are components (or objects) you can insert into a Web page or application to provide sophisticated formatting features and animation. ActiveX controls are small and versatile, and open limitless opportunities for creating spectacular Web content. ActiveX controls can be written in most programming languages, including Java, C++, and Microsoft Visual Basic. ActiveX controls can be easily reused by others in their Web pages and applications.

The open, component architecture of ActiveX enables you to create dynamic Web content and applications using proven COM technology, scripts, and software components (including Java applets). This architecture enables stand-alone applications to interact with each other. You can run ActiveX components in Internet Explorer that interact with other applications and components. For example, you could use a database access applet that interacts with a script to display multimedia graphics of data in Internet Explorer.

Because ActiveX controls are based on COM, the controls are language-independent. Java is an excellent programming language for both implementing and using COM objects, and Java and COM integrate seamlessly. COM makes Java a distributed language. Every public Java class is a COM object and can be called remotely just like any other COM object. COM gives Java direct access to native code. Any COM object appears to the programmer as a Java object. There is no need for large class libraries that wrap existing objects; the existing objects can be called directly.

Hundreds of ActiveX controls are available today with functionality ranging from a timer control (which simply notifies its container at a particular time) to full-featured spreadsheets and word processors. If you can imagine it, you can do it with an ActiveX control.

Optimizing ActiveX for Internet Explorer 4

Internet Explorer 4 complies with the OLE Controls 1996 (OC96) specification for host containers. Therefore, ActiveX control developers can now create faster, smaller, and more integrated controls that include:

- **Windowless controls.** These controls allow creation of transparent and nonrectangular controls, which combined with 2-D placement, enable Web publishers to overlap controls on a page. The multimedia controls included with Internet Explorer 4 are all windowless and take advantage of this functionality.
- **Apartment model controls.** Internet Explorer is a threaded container. For better performance, controls used inside Internet Explorer 4 should be marked as Apartment or Free Threaded.
- **Quick Activation.** The complex "QI Dance" that control writers needed to write is greatly simplified with Quick Activation. In most cases, a single call can initialize most controls.
- **SBindHost service.** Internet Explorer's support for the SBindHost service allows controls to download additional data asynchronously. This results in better performance for controls that download images or other complex data.

Controls written for previous versions of Internet Explorer work in Internet Explorer 4, but don't take advantage of these new OC96 enhancements. You can rewrite older controls to take advantage of the OC96 enhancements, but the control will not work in previous versions of Internet Explorer. Of course, you can always provide separate controls for browsers that don't comply with OC96.

With Internet Explorer 4, ActiveX controls have full access to the Document Object Model of Dynamic HTML. A control can now read its host page and modify all elements in the page at any time. Anything you can do in a script, you can do from an ActiveX control. You can use ActiveX controls to extend the capabilities of Dynamic HTML and create a wide range of interactive and animated Web page applications.

For example, you can apply filters and transitions to windowless controls, such as Microsoft DirectAnimation™ controls, to easily create spectacular multimedia effects. For more information, see "Multimedia Controls" later in this chapter.

Using ActiveX Controls in Web Pages

You can add ActiveX controls to your Web pages by using the standard HTML OBJECT element. The OBJECT element includes a set of PARAM elements that you use to specify which data the control should use and to control the appearance and behavior of the control.

To use an ActiveX control in a Web page, you refer to its Class ID Number as shown in the following example:

```
<OBJECT CLASSID="CLSIDXXXXXxxx..." ID="the name of the control"
  CODEBASE="URL of .ocx or .CAB file" #="version number">
<PARAM NAME="parameter name" VALUE="value">
<PARAM NAME="parameter name" VALUE= "value">
</OBJECT>
```

The CLASSID and version number of the control are generated when the control is compiled.

When downloading ActiveX controls is enabled in the security options, Internet Explorer 4 downloads controls and stores the controls on users' computers. Installed controls are used for Web pages that reference the control. Internet Explorer compares the version number of the installed control to the version number specified on the Web page. If a new version of a control is available, Internet Explorer downloads, stores, and uses the new version.

Listing and Removing ActiveX Controls

Internet Explorer 4 stores ActiveX controls in the following special Windows directory:

C:\Windows\Downloaded Program Files

You can open this directory using Windows Explorer to list installed ActiveX controls, to view information about the installed controls, to update the control with the latest version, or to remove controls.

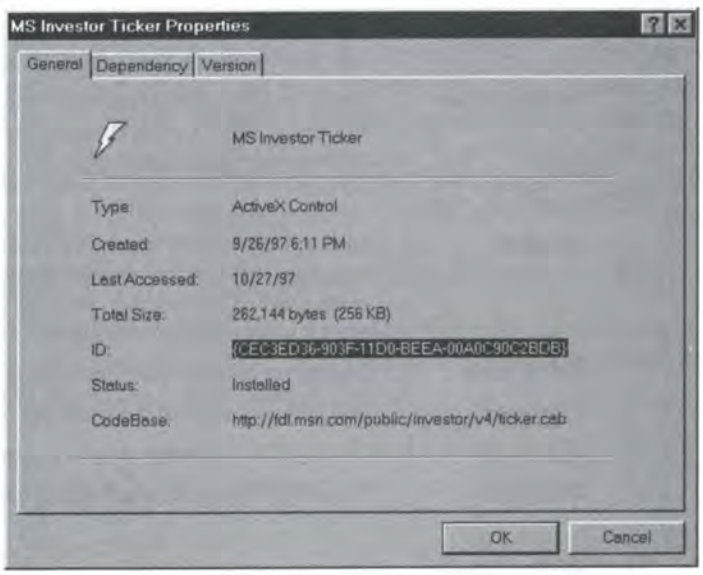
Windows Explorer lists the following properties for installed controls:

- Program File
- Total Size
- Status
- Creation Date
- Last Accessed
- Version

You can use the control's **Properties** dialog box to view more detailed information about the controls.

▶ **To view detailed information about a control**

1. From Windows Explorer, perform one of the following:
   - Select the control, and on the **File** menu, click **Properties**.
   - Right-click the control, and then click **Properties**.
2. Select the **General**, **Dependency**, or **Version** tab to view information about the control.



Option	Description
General Tab	
Type	Displays the type of control. If the type is Shortcut, you are viewing the properties for the shortcut to an item, not for the original item.
Created	Displays the date that the control was created.
Last Accessed	Displays the date that the control was last opened.
Total Size	Displays the size of the control files.
ID	Displays the Class ID (CLSID) for this control. This is the same ID that appears in the OBJECT element in the Web page.

Option	Description
General Tab *(continued)*	
Status	Displays whether the control is installed, damaged, or unplugged. Damaged means that one or more of the dependency files has been deleted. Unplugged means that an installation program has registered the .ocx file in a location different from where the code download installed it.
CodeBase	Displays the URL from which the control was installed. Internet Explorer uses this URL to update the control.
Dependency Tab	
File name	Lists the files on your computer, which this control requires to run. If any of these files are missing, the control will not work.
Package name	Lists any Java packages on your computer, which this object requires in order to run. If any of these packages are missing, the control will not work.
Version Tab	
Version	Displays the version number of this control. Internet Explorer compares this version number to the version number appearing in the OBJECT element in the Web page to determine if a more current version is available.
Description	Displays a description of the control.
Company	Displays the name of the publisher of the control.
Language	Displays the language version of the control.
Copyright	Displays the copyright information for the control.

▶ To update a control

- From Windows Explorer, right-click the control, and then click **Update**.

  Internet Explorer downloads the latest version of the control from the Web.

▶ To remove a control

- From Windows Explorer, perform one of the following:
  - Right-click the control, and then click **Remove**.
  - On the **File** menu, click **Remove Program File**.

  Internet Explorer removes the control and all dependency files.

Signing ActiveX Code

In Internet Explorer 4, you can set several security options for ActiveX controls and plug-ins to restrict the download of ActiveX controls for each security zone. For example, you can prevent the downloading of unsigned ActiveX controls from sites located in the Internet zone, but enable the downloading of unsigned ActiveX controls in the Local Intranet zone. For more information about setting security options for ActiveX controls, see "Customizing Security Zones" in Chapter 27, "Security Zones."

You should obtain a digital certificate from a Certificate Authority (CA) and digitally sign ActiveX controls consistent with your organization's security policy. You could decide not to sign controls for your Local Intranet zone, because you trust all of the sites. However, keep in mind that if you allow unsigned ActiveX controls to download automatically with no user intervention, you are also trusting any unsigned controls users may obtain from the Internet and reuse on your intranet. At a minimum, it is recommended that you sign ActiveX controls used for Web pages on the Internet to assure users of the integrity of your controls. For more information about digital certificates, see "Understanding Digital Certificates" in Chapter 28, "Digital Certificates."

You apply a digital signature to an ActiveX control at the time it is compiled. For more information, see "Signing Code with Microsoft Authenticode Technology" in the *Microsoft Internet Client SDK*.

Using Multimedia Controls

For Web publishers who want more multimedia effects than Dynamic HTML offers, Internet Explorer 4 includes the run-time version of the Microsoft DirectAnimation multimedia controls. A component of the DirectX® set of APIs, DirectAnimation provides unified, comprehensive support for such diverse media types as 2-D vector graphics, 3-D graphics, sprites, audio, and video, and a uniform time and event model.

Because DirectAnimation is a COM API with an underlying run-time engine, the DirectAnimation functionality can be accessed—in different ways—by a wide variety of programmers and authors:

- HTML authors can integrate animation using the DirectAnimation controls.
- Authors using VBScript or JScript, or Java applets, can program Web-page animation with Dynamic HTML.
- Programmers using Java, Visual Basic, and Visual C++® development systems can develop ActiveX controls or full applications that require multimedia support and interactive features.

The DirectAnimation run-time controls provide Internet Explorer 4 with access to many of the DirectAnimation scripting functions and libraries to implement additional multimedia and animation effects. The run-time controls allow you to deliver impressive vector graphics, image, and animation content over the Web with little code and short download times. The run-time controls include:

- **Sequencer.** Easily controls timing of events on pages.
- **Structured graphics.** Provides high-quality, lightweight, scalable, rotatable graphics. Internet Explorer 4 provides programmatic access so you can dynamically add new graphic elements through script. For example, you can create HTML applications that read values from a data-bound control and then create appropriate graphs.
- **Effects.** Alters any item on a Web page by applying a graphic filter. For example, you can add a glow filter and button bevel. The button bevel lets you dynamically create in/out beveled buttons from images and text.
- **Transitions.** Alters any item on a page, or the page itself, over time.
- **Behaviors.** Applies high-level behaviors to controls and Dynamic HTML elements.
- **Sprite.** Creates animated images.
- **Sprite buttons.** Creates animated multistate buttons.
- **Path.** Easily moves objects across a two-dimensional path.
- **Mixer.** Mixes multiple WAV files together dynamically.
- **Hot spot.** Establishes regions of the screen that can process mouse clicks.

All of these controls are transparent, windowless, and can be seamlessly integrated with Dynamic HTML. You can easily add DirectAnimation content to your Web pages by using the DirectAnimation controls as OBJECT elements and setting parameters (PARAM elements) on these controls—without programming at all.

Because DirectAnimation is integrated with Dynamic HTML, it is especially suited to adding compact animation effects to Web pages. For example, you can include DirectAnimation controls as windowless controls on the page (overlaying other elements such as text) and apply filters and transitions to DirectAnimation controls.

For more information, see the *DirectAnimation SDK* (included with the *Microsoft Internet Client SDK*), and visit the SBN Workshop at **http://www.microsoft.com/workshop/**.

CHAPTER 25

Java

In this chapter

Java Overview 422
Java Improvements for Internet Explorer 4 423
How Java Works 424
Microsoft Virtual Machine for Java 425
  Just In Time Compiler 425
  J/Direct 427
Java Security 428
Using Java Applets in Web Pages 428

Java Overview

Java is a programming language developed by Sun Microsystems and is a popular language for developing cross-platform-compatible Internet applications.

Internet Explorer 4 includes the Microsoft virtual machine for Java. It supports all functions of the Java Development Kit (JDK) Version 1.1, except for the following:

- **Java Native Interface (JNI).** Sun's proposed interface for linking Java code to platform-specific code.
- **Remote Method Invocation (RMI).** Sun's technology to allow Java objects to communicate over a network.
- **Digital signatures in Java Archive (.jar) files.** Sun's proprietary certificate format for digitally signing Java packages.

In Internet Explorer 4, Microsoft provides a native code interface called Raw Native Interface (RNI) as well as support for the component object model (COM) and the new J/Direct technology for native code access. COM and J/Direct make integration with native code significantly easier than either JNI or RNI. Overall, JNI is slower than RNI and requires more development work than J/Direct.

Developers looking for a robust, secure, high-performance solution for remote object interoperability should consider Distributed COM (DCOM), which is supported by Java today with the Microsoft virtual machine and allows both Java-to-Java communications and Java-to-non-Java communications.

Internet Explorer 4 does not support signed .jar files, because they are a Sun proprietary technology. However, Internet Explorer does support unsigned and compressed .jar files. Internet Explorer also supports .cab files, which offer significantly better compression than .jar files. Internet Explorer 4 allows the signing of Java .cab packages with industry-standard x.509v3 certificates. For more information, see "Signing Code with Microsoft Authenticode Technology" in the *Microsoft Internet Client SDK*.

Internet Explorer 4 makes it easier for developers to create more full-featured Java applications for the Web. Java performance has been improved, making Java applications run even faster than in Internet Explorer 3. In addition, the Java security model has been greatly expanded and improved.

The following sections describe significant Java improvements available for Internet Explorer 4. For more information, see the *Microsoft Internet Client SDK* and the *Microsoft SDK for Java 2.0*, and visit The Microsoft Java Technologies Web page at **http://www.microsoft.com/java/**, the Microsoft Developer Network at **http://www.microsoft.com/msdn/**, and the SBN programming workshop at **http://www.microsoft.com/workshop/**.

For information about troubleshooting for Internet Explorer and for Frequently Asked Questions (FAQs) about .cab files in Java, see Appendix G and H, respectively.

Java Improvements for Internet Explorer 4

The Java improvements for Internet Explorer 4 include:

- **Speed.** The Internet Explorer 4 Just in Time (JIT) compiler provides the fastest way to run Java applications.
- **Full component integration.** ActiveX controls can be accessed as Java Beans, and Java Beans accessed as ActiveX controls. (Java Beans is a specification for the creation of components referred to as "beans" in Java.) The Microsoft virtual machine for Java is the only virtual machine that has this automatic, seamless, bidirectional component support. In addition, debugging is seamless between VBScript, JScript, and Java.
- **New object model.** The Document Object Model is exposed through Java libraries. This allows Java developers to dynamically manipulate Web pages.
- **Improved Abstract Windows Toolkit (AWT).** The AWT has been improved to reduce overhead and improve performance.
- **New application foundation classes (AFC).** AFC provides the easiest way for Java developers to build rich, commercial-quality applications quickly. They are written entirely in Java and are intended to be compatible with all high-quality implementations of Java. The advantages of AFC include:
  - The GUI libraries include more than 30 prebuilt, extensible components for common UI features such as toolbars, tree controls, and dialog boxes.
  - The FX libraries include textures, support for wide-brush pens, and more. AFC is written in Java and runs on all standard Java platforms.
  - AFC provides a flexible look. Developers can use the standard Windows look, the look of another platform, or they can subclass to provide their own custom look.
  - AFC is available on all platforms for Internet Explorer 4, including Windows NT, Windows 95, Windows CE, Macintosh®, and select commercial versions of UNIX.
  - AFC Enterprise Libraries provide Java developers access to open, standards-based enterprise services, such as directory, management, database, distributed object, and transaction services.
- **New multimedia class libraries.** All the functionality of DirectX media and the DirectX foundation is provided as cross-platform Java class libraries, enabling developers to manipulate and animate a full set of media types.

- **Internationalization support.** Unicode support simplifies developing worldwide applications, because it supports multilingual display and input, and its easy resource format facilitates localization.
- **New Package Manager.** The Package Manager is a component of Internet Explorer 4 that manages the storage of Java class libraries. A database is used to associate properties and data with packages. When a component (the VM, development tool, and so on) wants to get class information, it calls a package manager API to get the information, instead of having to understand the complicated class path rules. The Package Manager works with .cab or .jar files to support the efficient download and storage of multiple .class files.
- **Complete Win32® integration.** With Internet Explorer 4, Java developers can leverage the power of the Win32 platform by utilizing the rich Win32 API set natively from their Java code.
- **Trust-based security for Java.** Internet Explorer 4 includes trust-based security for Java, which is a cross-platform security model for Java that provides fine-grained management of the permissions granted to Java applets and libraries.

How Java Works

Java is unlike other programming languages in that a program can be written without consideration for the platform on which it will be run. A Java applet can run on a variety of dissimilar platforms.

Java requires a client-side interpreter or virtual machine (VM) for running Java code. Platform-specific considerations are built into the virtual machine, thus allowing the Java applications to be platform independent.

Java uses classes to call functions that exist on the system in the form of Java class libraries. These libraries are installed when Internet Explorer is installed. System classes are always required to run Java applets. Without system classes, no Java applet could run on a system.

The Package Manager loads classes using the following registry key:

HLM\Software\Microsoft\Code_Store_Database\Java_Packages

The class libraries are stored in the C:\Windows\Java\directory.

Because Java was developed by the UNIX industry, applets use case-sensitive syntax and long file name extensions such as .class (the 8.3 equivalent of .class is .cla).

Basically, you write Java code, perhaps using Microsoft Visual J++ development software, and then save the noncompiled Java source code as a file with a .java or .jav file name extension. You then compile the source code to byte code, and the file is renamed with a .class (.cla) file name extension. The end users' Java VM compiles the byte code .class (.cla) files into machine-specific code, so applets can run on the end users' computers.

Microsoft Virtual Machine for Java

The Microsoft VM for Java provided in Internet Explorer 4 is the Win32 Reference Implementation of Sun's Java VM. Microsoft's implementation of the VM provides a number of benefits, including:

- High-performance native code interface
- Separation of name spaces
- Integration between Java and COM
- Efficient, controllable garbage collection (GC)
- Efficient internal object model, allowing better performance (in the VM, GC, JIT, and so on)

The Microsoft VM for Java includes the following innovations:

- **Integrated debugging** and the industry's fastest Just-in-Time compiler for the fastest run-time performance of Java applets.
- **Cab file compression** using Microsoft's new, cross-platform, open, backward-compatible Cabinet (.cab) packaging technology, which allows Java applets to download faster than ever before and allows users to install class libraries locally, avoiding repeated downloads across the Internet.
- **COM integration**, which allows Java applets to communicate with the universe of existing software based on the industry-standard COM, including ActiveX, Automation, and Object Linking and Embedding.
- **Unicode support** for development of international applications.

Just In Time Compiler

The Microsoft VM for Java includes a new JIT that speeds up the Java compiling process, so Java applets run much faster in Internet Explorer 4 than in version 3. The JIT compiles the Java byte code to native code (platform-specific) during run time and ensures maximum speed and efficiency.

You can turn the JIT off by accessing the **Advanced** tab in the **Internet Options** dialog box in Internet Explorer 4. You can also use the **Advanced** tab to turn Java logging on, to record Java activity.

▶ **To change Java VM settings**

1. On the **View** menu, click **Internet Options**.
2. Click the **Advanced** tab.
3. Configure the **Java VM** options you want.

Option	Description
Java JIT compiler enabled	Clear to turn off the JIT compiler. The JIT is on by default.
Java logging enabled	Select to create a log of all Java program activity, which is useful for security and troubleshooting. Java logging is off by default.

J/Direct

A new feature of the Microsoft VM for Java, called J/Direct, allows Java code running on the Microsoft VM for Java to directly call Win32 functions. All you need to know is how to make the call in the C programming language.

The new Java packages and classes, such as AFCs and multimedia classes, that Microsoft provides in the *Microsoft SDK for Java 2.0* are one way to access the Win32 API. These new Java classes let you use Windows functionality from Java without writing any C code, but they do not cover all aspects of the Win32 API.

J/Direct, by providing a way to call functions exported from *any* Windows DLL, addresses the need of Java developers to access the full Win32 API directly from their Java code.

Using the Win32 API directly from Java, however, can cause sticky problems. Windows was written to be accessed from C, and because of the many fundamental differences between C and Java data types, it's tricky to build parameter lists and receive return values. There are compatibility problems: Java does not have pointers or a representation for unsigned numbers; Java strings don't look like C strings; and Java doesn't have an analog for the C structure data type. In addition, there are problems in calling conventions—such as how to invoke Win32 functions that use callbacks, or who allocates memory buffers. To help with these problems, J/Direct provides a set of guidelines and helper classes that outline the best ways to interact with the Win32 API from Java code.

If you need to access functionality provided by an existing Windows DLL, and you already know how to use that DLL, J/Direct lets you get the job done. J/Direct enables Java code to call functions exported from any Windows DLL. The code in the DLL does not need to be aware of the Java environment. You can use J/Direct to call Win32 APIs or APIs in third-party DLLs. J/Direct can speed up Java applications on Win32 platforms.

Java Security

Internet Explorer 4 introduces a new Java security model called trust-based Java security. The original Java security model limited Java applets to running in a "sandbox" that prevented applets from accessing a computer or network resources, but that also greatly restricted what applets could do.

Trust-based security for Java is a cross-platform security model for Java that provides fine-grained management of the permissions granted to Java applets and libraries. It uses Java Cabinet (.cab) signing technology to ensure that an applet can only perform actions for which it specifically requests permissions. With trust-based security for Java, developers can precisely limit the section of code where permissions have been granted, and administrators can have flexible control over the permissions granted to local classes. When combined with Internet Explorer zones, trust-based security can also simplify security decisions for end users.

For more information about trust-based security, see Chapter 29, "Trust-Based Security for Java," and visit the Microsoft Security Advisor Web page at **http://www.microsoft.com/security/**.

Using Java Applets in Web Pages

To use a Java applet in a Web page, you use the APPLET element as shown in the following example:

```
<APPLET CODE="loading class" CODEBASE="relative path to the file">
<PARAM NAME="parameter name" VALUE="value">
</APPLET>
```

Note If a Java applet is compressed as a .cab file, the OBJECT element can be used.

For more information about using Java applets in Web pages, see the *Microsoft Internet Client SDK* and the *Microsoft SDK for Java 2.0*.

CHAPTER 26

Active Scripting

In this chapter

Active Scripting Overview 430
ECMAScript: The Internet Standard 431
JScript 3.0 in Internet Explorer 4 432
JScript 3.0 Availability 435
Differences between JScript 3.0 and JavaScript 1.2 435

Active Scripting Overview

With the increasing popularity of Web applications, many Web developers are augmenting their HTML authoring skills by learning a scripting language. With the next generation of Web browsers, however, developers need to decide which scripting language to learn and which scripting features to take advantage of. Before making this decision, it is important to understand which scripting languages are supported by each browser and each browser's corresponding level of compliance with standards.

Internet Explorer 4 supports both JScript and Visual Basic Scripting Edition (VBScript). The modular script architecture of Internet Explorer is designed so that any third party can add scripting-language support to Internet Explorer using the Active Scripting interface. This enables developers to use the languages they know today to create Web pages and ensures that they have the most up-to-date and complete scripting language support available. For more information, see "ActiveX Scripting" in the *Microsoft Internet Client SDK*.

In addition, Microsoft is working with other industry leaders to ensure a broad level of scripting interoperability among third-party products. Consistent with this effort, Internet Explorer includes the JScript 3.0 scripting engine, the industry's first implementation of the new ECMA-262 specification.

For information about troubleshooting authoring for Internet Explorer and for Frequently Asked Questions (FAQs) about scripting, see Appendix G and I, respectively.

ECMAScript: The Internet Standard

The need for a scripting standard is clear. A scripting standard ensures that Web developers understand the direction of the industry and that they can write code consistent with this direction. It also allows developers to author script that will run seamlessly across a variety of products.

The European Computer Manufacturers Association (ECMA) scripting language committee was formed last year with the charter to standardize "the syntax and semantics of a general-purpose, cross-platform, vendor-neutral scripting language." Along with companies such as Borland, IBM, Netscape, and Sun Microsystems, Microsoft has participated actively in the development of the ECMAScript standard. A joint specification from Borland, Microsoft, and Netscape was recently ratified by the committee and published as the ECMA-262 specification.

With products such as Internet Explorer 4 and Internet Information Server 4.0, Microsoft is delivering JScript 3.0, the first scripting language to conform fully to ECMA-262. Microsoft is continuing to work with the ECMA scripting committee to ensure the advancement of the ECMA Internet Scripting Standard.

Basically, the ECMA-262 specification describes a Web scripting language that can enrich and enliven Web pages in a Web browser. More specifically, the ECMA-262 specification outlines an object-oriented programming language for performing computations and manipulating objects within a host environment, such as the browser. The complete ECMA-262 specification can be found at **http://www.ecma.ch/stand/ecma-262.htm**.

From a client perspective, a Web browser provides an ECMAScript host environment for client-side computations. These computations can include objects that represent windows, menus, pop-up menus, dialog boxes, text areas, anchors, frames, history, cookies, and input/output. In addition, the Web browser provides a means to attach scripting code to events such as change of focus, form submission, mouse actions, page and image loading, and others. The scripting code appears in-line with the HTML, and the rendered page is a combination of HTML elements and scripting engine computations.

JScript 3.0 in Internet Explorer 4

Microsoft JScript 3.0 is the first completely compliant industry implementation of the ECMA-262 specification. For more information about the feature/keyword syntax, visit the JScript Web site at **http://www.microsoft.com/jscript/**.

Microsoft has also submitted additional language functionality to the ECMA scripting committee for the next version of the ECMA specification. These features are being delivered in JScript 3.0, to provide the most advanced functionality for Web scripting available.

The following table describes the scripting features available in the ECMA-262 specification and Internet Explorer 4.

Category	Feature/*Keyword*
Array Handling	**Array** *object*, **join** *method*, **length** *property*, **reverse** *method*, **sort** *method*
Assignments	Assign (=) *operator*, Addition (+=) *operator*, Bitwise AND (&=) *operator*, Bitwise OR (|=) *operator*, Bitwise XOR (^=) *operator*, Division (/=) *operator*, Left Shift (<<=) *operator*, Modulus (%=) *operator*, Multiplication (*=) *operator*, Right Shift (>>=) *operator*, Subtraction (-=) *operator*, Unsigned Right Shift (>>>=) *operator*
Booleans	**Boolean** *object*
Comments	/* .. */ or // *statements*
Constants/Literals	**NaN** *property*, **null**, **true**, **false**, **Infinity** *property*, **undefined**
Control Flow	**break** *statement*, **continue** *statement*, **for** *statement*, **for...in** *statement*, **if...else** *statement*, **return** *statement*, **while** *statement*
Dates and Time	**Date** *object*, **getDate** *method*, **getDay** *method*, **getFullYear** *method*, **getHours** *method*, **getMilliseconds** *method*, **getMinutes** *method*, **getMonth** *method*, **getSeconds** *method*, **getTime** *method*, **getTimezoneOffset** *method*, **getYear** *method*, **getUTCDate** *method*, **getUTCDay** *method*, **getUTCFullYear** *method*, **getUTCHours** *method*, **getUTCMilliseconds** *method*, **getUTCMinutes** *method*, **getUTCMonth** *method*, **getUTCSeconds** *method*, **setDate** *method*, **setFullYear** *method*, **setHours** *method*, **setMilliseconds** *method*, **setMinutes** *method*, **setMonth** *method*, **setSeconds** *method*, **setTime** *method*, **setYear** *method*, **setUTCDate** *method*, **setUTCFullYear** *method*, **setUTCHours** *method*, **setUTCMilliseconds** *method*, **setUTCMinutes** *method*, **setUTCMonth** *method*, **setUTCSeconds** *method*, **toGMTStrong** *method*, **toLocaleString** *method*, **toUTCString** *method*, **parse** *method*, **UTC** *method*

Category	Feature/*Keyword*
Declarations	**function** *statement*, **new** *operator*, **this** *statement*, **var** *statement*, **with** *statement*
Function Creation	**Function** *object*, **arguments** *property*, **length** *property*
Global Methods	**Global** *object*, **escape** *method*, **unescape** *method*, **eval** *method*, **isFinite** *method*, **isNaN** *method*, **parseInt** *method*, **parseFloat** *method*
Math	**Math** *object*, **abs** *method*, **acos** *method*, **asin** *method*, **atan** *method*, **atan2** *method*, **ceil** *method*, **cos** *method*, **exp** *method*, **floor** *method*, **log** *method*, **max** *method*, **min** *method*, **pow** *method*, **random** *method*, **round** *method*, **sin** *method*, **sqrt** *method*, **tan** *method*, **E** *property*, **LN2** *property*, **LN10** *property*, **LOG2E** *property*, **LOG10E** *property*, **PI** *property*, **SQRT1_2** *property*, **SQRT2** *property*
Numbers	**Number** *object*, **MAX_VALUE** *property*, **MIN_VALUE** *property*, **NaN** *property*, **NEGATIVE_INFINITY** *property*, **POSITIVE_INFINITY** *property*
Object Creation	**Object** *object*, **new** *operator*, **constructor** *property*, **prototype** *property*, **toString** *method*, **valueOf** *method*
Operators	Addition (+) *operator*, Subtraction (-) *operator*, Modulus arithmetic (%) *operator*, Multiplication (*) *operator*, Division (/) *operator*, Negation (-) *operator*, Equality (==) *operator*, Inequality (!=) *operator*, Less than (<) *operator*, Less than or Equal to (<=) *operator*, Greater than (>) *operator*, Greater than or Equal to (>=) *operator*, Logical And (&&) *operator*, Or (||) *operator*, Not (!) *operator*, Bitwise And (&) *operator*, Bitwise Or (|) *operator*, Bitwise Not (~) *operator*, Bitwise Xor (^) *operator*, Bitwise Left Shift (<<) *operator*, Bitwise Right Shift (>>) *operator*, Unsigned Right Shift (>>>) *operator*, Conditional (?:) *operator*, Comma (,) *operator*, **delete** *operator*, **typeof** *operator*, **void** *operator*, Decrement (--) *operator*, Increment (++) *operator*
Objects	**Array** *object*, **Boolean** *object*, **Date** *object*, **Function** *object*, **Global** *object*, **Math** *object*, **Number** *object*, **Object** *object*, **String** *object*
Strings	**String** *object*, **charAt** *method*, **charCodeAt** *method*, **fromCharCode** *method*, **indexOf** *method*, **lastIndexOf** *method*, **split** *method*, **toLowerCase** *method*, **toUpperCase** *method*, **length** *property*

The following table describes those additional JScript 3.0 features submitted by Microsoft to ECMA but not included in the ECMA-262 specification.

Category	Feature/*Keyword*
Array Handling	**concat** *method*, **slice** *method*, **VBArray** *object*, **dimensions** *method*, **getItem** *method*, **lbound** *method*, **toArray** *method*, **ubound** *method*
Conditional Compilation	**@cc_on** *statement*, **@if** *statement*, **@set** *statement*, **@_win32** *variable*, **@_win16** *variable*, **@_mac** *variable*, **@_alpha** *variable*, **@_x86** *variable*, **@_mc680x0** *variable*, **@_PowerPC** *variable*, **@_jscript** *variable*, **@_jscript_build** *variable*, **@_jscript_version** *variable*
Control Flow	**do...while** *statement*, **Labeled** *statement*, **switch** *statement*
Dates and Time	**GetVarDate** *method*
Enumeration	**Enumerator** *object*, **atEnd** *method*, **item** *method*, **moveFirst** *method*, **moveNext** *method*
Function Creation	**Caller** *property*
Operators	Identity (**===**) *operator*, Nonidentity (**!==**) *operator*
Objects	**Enumerator** *object*, **RegExp** *object*, **Regular Expression** *object*, **VBArray** *object*
Regular Expressions and Pattern Matching	**RegExp** *object*, **index** *property*, **input** (**$_**) *property*, **lastIndex** *property*, **lastMatch** (**$&**) *property*, **lastParen** (**$+**) *property*, **leftContext** (**$'**) *property*, **multiline** (**$***) *property*, **rightContext** (**$'**) *property*, **$1...$9** *property*, **global** *property*, **ignoreCase** *property*, **lastIndex** *property*, **source** *property*, **compile** *method*, **exec** *method*, **test** *method*, **Regular Expression Syntax**
Strings	**concat** *method*, **slice** *method*, **match** *method*, **replace** *method*, **search** *method*, **anchor** *method*, **big** *method*, **blink** *method*, **bold** *method*, **fixed** *method*, **fontcolor** *method*, **fontsize** *method*, **italics** *method*, **link** *method*, **small** *method*, **strike** *method*, **sub** *method*, **sup** *method*

JScript 3.0 Availability

JScript 3.0 can be downloaded from the JScript Web site at **http://www.microsoft.com/jscript/**. JScript 3.0 will ship with the following Microsoft products:

- Microsoft Internet Explorer 4
- Microsoft Internet Information Server 4.0
- Microsoft Windows Scripting Host

Software developers can license JScript or Visual Basic Scripting Edition from Microsoft to use as the scripting language in their own applications. JScript and VBScript are available from Microsoft in two forms: as a ready-to-run (compiled) binary for several platforms or as source code. Licensing details can also be found at **http://www.microsoft.com/jscript/**.

Differences between JScript 3.0 and JavaScript 1.2

If you're designing Web pages that may be used by more than one type of browser, and you're using advanced technologies such as JScript and JavaScript, it's important to understand potential compatibility issues that may arise. For example, Netscape's JavaScript 1.1 served as the original basis for ECMA-262, although Netscape has yet to implement the ECMA-262 specification. In the latest releases of Netscape's Navigator software, JavaScript 1.2 does not comply with the ECMA-262 specification, and code written to JavaScript 1.2 works only with the version 1.2 scripting engine in Navigator.

As a result of Microsoft's standards support, code written using JScript 3.0 will work with other industry implementations of the ECMA-262 specification, including a future implementation from Netscape. In addition, Microsoft will ship the same JScript implementation in all versions of Internet Explorer 4, so developers can count on the same scripting behavior, regardless of the browser platform.

The following table details the support for the ECMA-162 Internet scripting standard offered by Microsoft Internet Explorer 4 and Netscape Navigator 4.0 and describes the corresponding behaviors in each browser. Developers can use this table as a tool to help during the migration process if scripting errors or anomalies arise with pages on your intranet.

Standard Feature or Behavior category	Microsoft Internet Explorer 4	Netscape Navigator 4.0
\u1234 used to specify unicode character	Fully supported	Not supported
Hexadecimal constants longer than 8 digits	Fully supported	Not supported
Octal constants that overflow 32 bits	Fully supported	Not supported
Negative Zero	Fully supported	Not supported
Date *object*	Fully supported	7th parameter (**ms**) not supported
Date.toUTCString	Fully supported	Not supported
Date.getFullYear	Fully supported	Not supported
Date.setFullYear	Fully supported	Not supported
Date.getUTCFullYear	Fully supported	Not supported
Date.setUTCFullYear	Fully supported	Not supported
Date.getUTCMonth	Fully supported	Not supported
Date.setUTCMonth	Fully supported	Not supported
Date.getUTCDate	Fully supported	Not supported
Date.setUTCDate	Fully supported	Not supported
Date.getUTCDay	Fully supported	Not supported
Date.setUTCDay	Fully supported	Not supported
Date.getUTCHours	Fully supported	Not supported
Date.setUTCHours	Fully supported	Not supported
Date.getUTCMinutes	Fully supported	Not supported
Date.setUTCMinutes	Fully supported	Not supported
Date.getUTCSeconds	Fully supported	Not supported
Date.setUTCSeconds	Fully supported	Not supported
Date.getMilliseconds	Fully supported	Not supported
Date.setMilliseconds	Fully supported	Not supported
Date.getUTCMilliseconds	Fully supported	Not supported
Date.setUTCMilliseconds	Fully supported	Not supported
NaN Dates	Fully supported	Not supported, set time to 0 (Jan 1, 1970 UTC) instead

Standard Feature or Behavior category	Microsoft Internet Explorer 4	Netscape Navigator 4.0
Full range of –100,000,000 days before Jan 1, 1970 UTC to 100,000,000 days after Jan 1, 1970 UTC	Fully supported	Not supported
Eval *method*	Fully supported	Supported, but often returns an incorrect value
== operator	Fully supported	Supported, but doesn't match standard specification
+ operator	Fully supported	Supported, but often returns incorrect results
Number formatting	ECMA standard behavior	Supported, but often returns incorrect results
Calling through a local function should pass the sglobal object as the "this" value	ECMA standard behavior	Passes activation object instead
Bind once to left operand on side-effect operators	ECMA standard behavior	Binds twice
Delete *operator*	ECMA standard behavior	Returns undefined
New *operator*	ECMA standard behavior	Grammar parsed incorrectly
Math.atan2 *method*	ECMA standard behavior	Does not implement many of the special cases listed in the standard specification

PART 7

Security and Privacy

Microsoft Internet Explorer 4 provides a full range of security features to help system administrators manage Internet and network security in their organizations. Internet Explorer 4 also provides features to help ensure user privacy. Part 7 describes the security and privacy features of Internet Explorer 4 and explains how these features can be used to help protect the individuals and information in your organization.

Chapter 27: Security Zones

Internet Explorer 4 introduces security zones, an innovative approach to secure browsing that offers greater management control and flexibility. Security zones allow users to divide the Internet and intranet into four groups of trusted (safe) and untrusted (unsafe) areas and to designate that specific Web content belongs to these safe and unsafe areas. This Web content can be anything from an HTML file, a graphic, an ActiveX™ control, a Java™ applet, or an executable program. Once zones of trust have been established, browser security levels can be set for each zone.

Chapter 28: Digital Certificates

To verify the identity of individuals and organizations on the Web and to ensure content integrity, Internet Explorer 4 uses industry-standard digital certificates and Microsoft Authenticode™ 2.0 technology. Together with security zones, these can be used to control user access to online content based on the type, source, and location of the content. For example, system administrators can use security zones in conjunction with certificates to allow users to have full access to Web content on their intranet but limit users' access to Web content from restricted sites located on the Internet.

Chapter 29: Trust-Based Security for Java

Internet Explorer 4 provides trust-based security for Java, which is a cross-platform security model for Java that provides fine-grained management of the permissions granted to Java applets and libraries.

Chapter 30: Content Advisor

System administrators can control the types of content that users can access on the Internet by using the Internet Explorer 4 Content Advisor. You can adjust the content rating settings to reflect what you think is appropriate content in four areas: language, nudity, sex, and violence.

Chapter 31: User Privacy

Internet Explorer 4 supports a wide range of Internet protocols for secure information transfers and financial transactions over the Internet or intranet. Internet Explorer 4 also provides a variety of features to help users assure the privacy of their information.

System administrators can use the Internet Explorer Administration Kit (IEAK) to pre-configure custom packages of Internet Explorer 4, giving them control over the security settings and content rating settings for user groups in their organizations. Once browsers are deployed, you can use the IEAK Profile Manager to manage security settings and content rating settings for user groups through the automatic browser configuration feature of Internet Explorer. You can also lock down security settings, certificates, and content rating settings to prevent users from changing settings.

Overview: Understanding Trust Management

Until recently, most people looked at browsing the Internet as a spectator sport. It didn't occur to them that they weren't just anonymously flipping through an online book; they were actively *exchanging* information and programs. Every visit to a Web site leaves a calling card, every animation fetches a program, and every hacker who becomes temporarily famous for demonstrating a security hole raises awareness that the bad guys are right behind us. Being wired has its price.

It follows that security and privacy in this new generation of the Internet has to be rethought. Traditional network security focuses on protecting valuable server resources from hostile clients. Now the problem also includes protecting valuable client resources from hostile servers. Information is flowing in both directions between publishers and consumers of Internet content, and comprehensive Internet security requires collaboration between both parties. This brings the issue of trust to the foreground.

The very fact that the Internet is open means that using it will remain somewhat dangerous. But not using it at all is not a practical solution. The right approach is to know the risks up front and make educated choices. Ultimately, the idea is not to avoid potentially dangerous operations, but to acknowledge that an action is potentially dangerous and provide the user with tools to make an informed decision about whether to allow it. Translated into Internet security terms, this is trust management.

Trust management is best compared to risk management. Insurance schemes of all kinds depend on managing risk, not avoiding it. Trust is a familiar concept. In its simplest form, it's an intuitive human construct based on three elements:

- Recognition
- Reputation
- Referral

Recognition is the most dependable element. It's one to one—a face, a voice, and by extrapolation and careful positioning, a brand. The problem with recognition is that it's not scalable. Reputation is better. It's indirect and scalable, but it can be unreliable: "great reputation, lousy food." It does, however, have the added property of capital. Reputation capital is important; companies build it based on delivering consistent quality. It's hard to get and easy to lose. Referral is the most complex form of trust because it's always transitive: "the doctor I trust says this is the best doctor for my condition." A combination of reputation and referral can be used to effectively establish trust in a digital world. The raw materials for digital trust come from public key cryptography: digital signatures, certificates, and encryption.

Establishing Trust on the Internet

Any trust proposition has to begin with a simple "Who am I dealing with?" The technical term is *authentication*, and it applies to both entities and to the origin of data. Strong authentication on the Internet relies on digital signatures and certificates.

A *digital signature* provides the cryptographic means to associate a message with its originator. Further, its use guarantees that a recipient of a signed document can verify that the signature is authentic and recover the message in a provably unmodified form. Signature verification is accomplished using an attached certificate.

A *digital certificate* is a computer-based record that provides an electronic means of confirming that individuals and organizations who communicate on the Web are who they claim to be. Essentially, a certificate is evidence of prior authentication, where authentication means providing some traditional form of proof, such as comparing a person's face to a picture on a driver's license or making sure he or she knows some secret that was sent to him or her in paper mail, and so forth. The sole purpose of authentication is to establish transitive trust.

For example, suppose Sam Shareware signs a document making an assertion that "My code contained herein is not intentionally malicious." The only way Sam (who is unknown to the user of his software) can be trusted is by referral. This starts with the fact that he went out and willingly submitted himself to a process that led to his being granted a digital certificate.

Still, the simplicity of Internet Protocol (IP) spoofing makes reading the URL insufficient evidence to authenticate a site. Requesting a site certificate by means of a secure protocol, such as SSL or IPSEC, provides a second level of assurance.

With Authenticode 1.0, a digital signature assures message integrity and can form the basis for a trust decision: Is execution allowed? Authenticode 2.0 allows a software publisher to additionally have its code "stamped" with a date. Certificates are generated using large cryptographic key pairs that have a practical life expectancy based on the amount of time in which their keys could be reasonably compromised. Expiration dates are applied to certificates to ensure that they expire long before then. Time-stamping guarantees that the signature was applied while the keys were good, and therefore the signature itself is valid.

Trusting Java

Authenticode 2.0 also includes finer-grained security for Java. While digital signatures can assure message integrity and authenticate publishers of mobile code, the only authorization decision presented is permission to execute—it's all or nothing. In practice, however, it's often more important to know what resources the mobile code needs than who published it. For example, while you may trust Microsoft, you may not want any code downloaded from the Internet to open any arbitrary file on your system.

Java originally defined a "sandbox" model for security, in which Java classes loaded from the network were granted extremely limited capabilities, and classes loaded from the local disk were given free reign to do virtually anything. Under this binary trust model, many interesting Web applications could not be written to run from the network, while unrestricted local classes could accidentally open arbitrarily bad security holes, either by themselves or when called from a malicious, untrusted applet.

Microsoft's Authenticode signing technology and Sun's JDK 1.1 added the ability to sign applets loaded from the network so that they could enjoy the same privileges as local applications. They did not, however, eliminate the all-or-nothing quality of Java security.

Authenticode 2.0 begins by adding intermediate levels of trust to the existing Java security model. It enhances the administrative options for the virtual machine to include fine-grained control over the capabilities granted to Java code, such as access to scratch space, local files, and network connections. This allows a system administrator to give an application some additional capabilities without offering it unlimited access to every capability in the system. Thus, only system classes with the highest level of trust can extend the model.

More important, the decision whether to trust an applet is presented before the applet is loaded. Two properties of Java (strong typing and the fact that it runs in a virtual machine) give it a security advantage. The developer can assert what resources his code needs and the virtual machine can enforce it. *Capability signing* allows an applet's signature to include a list of the capabilities requested by that applet. (This is in contrast to Netscape's model, which forces applets and class libraries to bury hard-coded capability requests in the Java source code.) Permissions requested for an applet (such as the hosts to which the applet is allowed to connect) are distinct from the code and can be expanded or reduced by an intranet administrator without having to recompile the applet. It also diminishes user interface "noise"—presenting the user with a trust dialog every time a new capability is requested while the code is running.

A new Package Manager allows the installation of local class libraries that are not fully trusted, using capability signing. This is especially important for Java Beans and class libraries. While it's desirable to allow these components to reside locally and to have some expanded capabilities, it's not prudent to give them unlimited power.

But, putting things into perspective, Java is but a component. What about the site itself?

Trusting Everything Else

Historically, the motive for discriminating on the basis of a site's content has been driven by a site's subject matter and the fact that some ideas and images are blatantly offensive to many people. In 1995, the World Wide Web Consortium Platform for Internet Content Selection (PICS) started to define an infrastructure for voluntary self-labeling, and more than a million sites are already labeled. While this is still an admittedly small fraction of the Internet, it's a good start. Internet Explorer 4 provides a Content Advisor with built-in PICS support, which can be used to control the types of content users can access on the Internet, based on the categories of language, nudity, sex, and violence.

Now there are new areas of concern relating to trusting Web content. While a site's subject matter may occasionally be an issue, it's more probable in a business setting that a content *feature*, such as whether the component wants to write to the hard disk, will be the cause of discrimination and will make the difference in whether to allow or disallow execution.

Unfortunately, the explosion of functionality available for the Web—Java with Capabilities, Scripting, ActiveX—has brought with it an explosion of complexity. What was previously a simple idea—"Don't let bad things happen to my machine"—has become a complex web of interrelated choices for the user.

Previous security schemes have depended on an end user or administrator making blanket security decisions for all Web sites and pages encountered. Java was always on or always off. ActiveX controls were always enabled or always disabled. This simplistic scheme is now a victim of the success of the Web.

Previously, all Web content was treated on an approximately level playing field. It really couldn't do much, so all content was given the same level of capability—virtually none. But corporations have recently realized that the combination of a Web browser and HTML can produce powerful internal applications. Systems that otherwise would have to be written in C++ or Visual Basic® programming language and then installed across the corporation can now be created quickly in HTML. With this technology, ongoing maintenance and installation costs are greatly reduced, and user interface is intuitive; it seems an ideal solution.

This solution raises new issues. Many Java applications need to perform application-like functions (such as writing to the disk, or network I/O), or they can use ActiveX controls to do the same thing. Blanket restrictions regarding Java and ActiveX are no longer practical; all content is no longer equal. You may trust active content in some situations, from some sources, but not trust it in other situations, from other sources.

One solution would be to ask users to make case-by-case judgment calls. But most users are not knowledgeable enough to be trusted with these decisions. For example, when browsing through a Web site, a user might be asked if the browser should allow the site to write to the disk, perform network operations, run ActiveX controls, script ActiveX controls, use more than 1 MB of memory, perform cross-frame operations, and so on. It's not likely that most users could adequately answer those questions.

To help simplify the decision-making process, Internet Explorer 4 has introduced *security zones,* a model of common security settings for related sites. Security zones do two things:

1. Group sets of sites together.
2. Assign security settings to that zone.

The security zone model breaks online content into four areas:

- The Internet
- The intranet
- Trusted Web sites
- Untrusted Web sites

Settings for ActiveX controls, downloading and installation, scripting, cookie management, password authentication, cross-frame security, and of course, Java capabilities can all be controlled based on the trust zone to which a site belongs. Users may accept the default zone settings or fully customize the zones and security levels to meet their needs.

The familiar High, Medium, and Low security levels are carried over from Internet Explorer 3, but with some significant changes. Users often questioned exactly what High, Medium, and Low security meant, and corporations requested additional control beyond these three settings. You can now see exactly what security options are disabled and enabled, and can have precise control of more than 50 options by choosing the new Custom setting.

If all of these choices are too much for an end user, an administrator using the IEAK can pre-configure the security options before the user installs Internet Explorer 4. Settings can also be locked down to prevent them from being altered by end users.

Thus, security zones reduce the number of intrusive and unanswerable dialog boxes that the end user encounters within a browsing session, while giving administrators fine-grained control over security options. Zones provide a means to express and enforce an organization's security rules or policies. And updates to zone membership and security settings can be automatically pushed to each user's desktop machine, so that administrators can manage security policy dynamically across all machines on their networks.

Summary

Any serious approach to Internet security must be comprehensive. And, ideally, it should bring order to the complex web of dangers, problems, and questions. Microsoft's Security Management Architecture acknowledges that it's not enough to provide assurances of component security. The problem is bigger than that, and it starts with trusting sites themselves. Trust is the unavoidable common denominator, and the decision whether to trust something or not must be based on fine-grained, but easy-to-understand choices.

Security zones and Authenticode 2.0 are two new tools that enable users to make good trust decisions. Zones give users and administrators a convenient means to specify Internet security policy based on origin. Authenticode 2.0 extends the digital shrink-wrapped software model to include time-stamping and signed Java capabilities. Together, they form the basis of a workable, extensible security scheme for the Internet.

CHAPTER 27

Security Zones

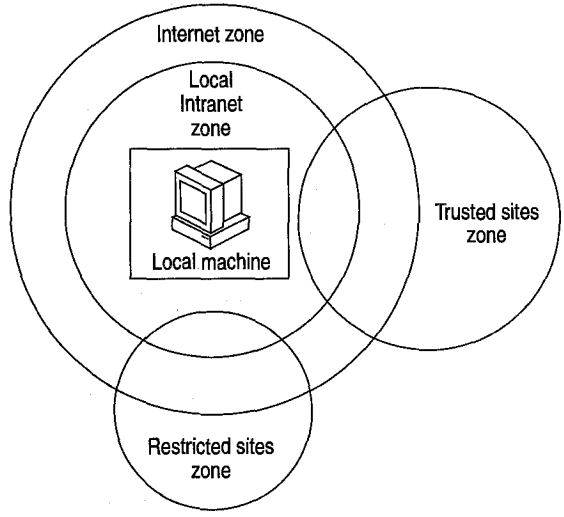
In this chapter

Security Zones Overview 448
Zone Architecture 450
Setting Up Security Zones 451
Designating the Sites Included in Zones 452
Local Intranet Zone 452
Restricted and Trusted Sites Zones 454
Changing Security Levels 456
Customizing Security Zones 456
Configuring Custom Settings 456
ActiveX Controls and Plug-ins 457
Java 459
Scripting 460
Downloads 460
User Authentication 461
Miscellaneous 461
Deploying Security Zones in User Groups 463
Locking Down Security Options 464
Managing Security Settings for Deployed Browsers 465

Security Zones Overview

Zone security is a system that allows the user or system administrator to divide into groups the Web content a browser can visit. This Web content can be anything from an HTML file, a graphic, an ActiveX control, a Java applet, or an executable file. Each group, or *zone,* then has a security level assigned to it. In this way, security zones control users' access to Web sites depending on the zone in which the site is located and the level of trust assigned to each zone. Internet Explorer 4 provides the following predefined security zones:

- **Local Intranet zone.** This zone includes all sites inside an organization's firewall (for computers connected to a local network). The Local Intranet zone is also home to Web applications that need access to the user's hard disk. The default security level is Medium.
- **Trusted Sites zone.** This is the zone where you specify Internet sites you know you can trust. These sites might include corporate subsidiaries or perhaps the Web site of a trusted business partner. The default security level is Low.
- **Internet zone.** All sites on the Internet that are not in the trusted or restricted sites zones are included in this zone. The default security level is Medium.
- **Restricted Sites zone.** This is the zone where you specify sites that you know you cannot trust. The default security level is High.



A fifth zone, Local Machine, exists but it is not included in the user interface and is not configurable through the security options. Local Machine can only be configured in the registry. Local Machine includes most content on the local disk, excluding cached classes in the Temporary Internet Files folder and classes that are signed with restricted privileges. This area forms a completely trusted zone, to which few or no security restrictions apply. System administrators can also exclude network and other drives by explicitly mapping them into other zones, if necessary.

Using security zones, it's easy to provide the appropriate level of security for the various types of Web content that users are likely to encounter. You can accept the default setting for each zone; change the setting to another of the preset High, Medium, and Low choices; or use the Custom setting for more precise control.

For example, it's likely that you fully trust sites on your company's intranet, so you probably want to allow all types of active content to run there. Simply set the Local Intranet zone to a low level of security. You probably don't feel as confident about sites on the Internet, so you can assign the entire Internet zone a higher level of security to prevent active content from being run and code from being downloaded to your computer. However, if there are specific sites you trust, you can place individual URLs or entire domains in the Trusted Sites zone. Other sites on the Internet are known to be sources of potentially harmful Web content, so you can choose to place the highest restrictions on those sites.

The default Low, Medium, and High security levels are detailed in the following table. You can accept the defaults or configure custom zone settings to meet your needs.

Security Option	Low Level	Medium Level	High Level
Script ActiveX controls marked safe for scripting	Enable	Enable	Enable
Run ActiveX controls and plugins	Enable	Enable	Disable
Download signed ActiveX controls	Enable	Prompt	Disable
Download unsigned ActiveX controls	Prompt	Disable	Disable
Initialize and script ActiveX controls not marked as safe	Prompt	Prompt	Disable
Java permissions	Low safety	High safety	High safety

Security Option	Low Level	Medium Level	High Level
Active scripting	Enable	Enable	Enable
Scripting of Java applets	Enable	Enable	Disable
File download	Enable	Enable	Disable
Font download	Enable	Enable	Prompt
User Authentication - Logon	Automatic logon with current name and password	Automatic logon only in intranet zone	Prompt for user name and password
Submit nonencrypted form data	Enable	Prompt	Prompt
Launching applications and files from an IFrame	Enable	Prompt	Disable
Installation of desktop items	Enable	Prompt	Disable
Drag and drop or copy and paste files	Enable	Enable	Prompt
Software channel permissions	Low safety	Medium safety	High safety

Administrators can also use the Internet Explorer Administration Kit (IEAK) Configuration Wizard and the IEAK Profile Manager to configure and manage security zones for custom packages to be deployed in user groups.

For more background information about security zones, see the "Trusting Everything Else" section of "Overview: Understanding Trust Management" at the beginning of Part 7.

Zone Architecture

When an HTML page is opened, a dynamic-link library called Urlmon.dll is used to determine from which zone it was loaded. To do this, two things happen. First, Urlmon.dll checks to see if a proxy server was used to get the HTML page; if so, Urlmon.dll automatically knows that the page is from the Internet. Next, it checks the registry to see if the page is from a designated Trusted or Untrusted location, and the security zone is set appropriately. If no proxy server is involved, the URL is then parsed to determine where the page comes from.

All of the zone security settings can be found in two registry keys:

- **HKEY_CURRENT_USER\Software\Microsoft\Windows\Currentversion\Internet Settings\Zonemap**

  and

- **HKEY_CURRENT_USER\Software\Microsoft\Windows\Currentversion\Internet Settings\Zones**

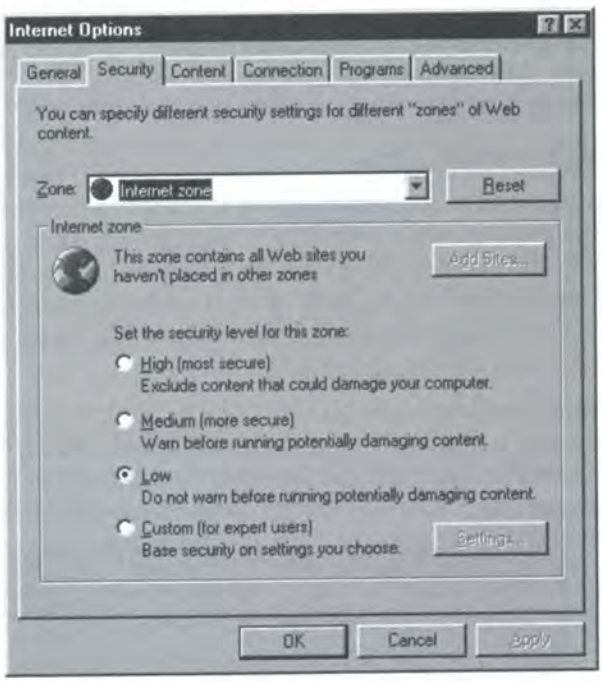
The **ZONEMAP** keys contain the list of Trusted and Untrusted (Restricted) Web sites, domains, and protocols. The **ZONES** key contains the actual zones and the security settings that apply to each zone.

Setting Up Security Zones

The options for setting and configuring security zones are the same whether you access them from Internet Explorer 4, the IEAK Configuration Wizard, or the IEAK Profile Manager.

▶ **To open the Security tab from Internet Explorer**

1. On the **View** menu, click **Internet Options**.
2. Click the **Security** tab.



Note Because security works differently in Internet Explorer 4, any existing Internet Explorer 3 settings are not preserved.

Designating the Sites Included in Zones

Users can specify the sites included in the Local Intranet, Restricted Sites, and Trusted Sites security zones. The Internet zone consists of all Web sites not included in any of the other zones.

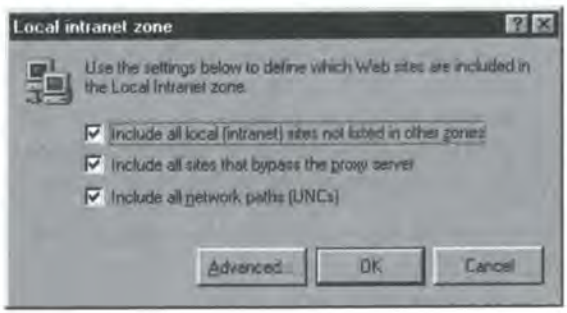
Note Web content can be addressed either by Domain Name System (DNS) name or by IP address. For sites that use both, it is important to configure both references to the same zone. In the common cases, the Local Intranet sites are identifiable by either local names or by IP addresses in the proxy bypass list. All other names and IP addresses would be mapped to the Internet zone. However, if a site name is entered into the Trusted Sites or Restricted Sites zone site list, but its IP address range isn't, then the site may be treated as part of the Internet zone if it is accessed by the IP address.

Local Intranet Zone

To be secure, it is imperative that the Local Intranet zone be set up in conjunction with the proxy server and firewall configurations. All sites in the zone should be inside the firewall, and proxy servers should be configured so that they do not allow an external DNS name to be resolved to this zone. Configuring the client zone security requires a detailed knowledge of the existing network configuration, proxy servers, and secure firewalls.

By default, the Local Intranet zone consists of local domain names and the addresses of any proxy exceptions specified in the **Proxy Settings** dialog box. (To access this dialog box, on the **View** menu, click **Options**, and then click the **Connection** tab.) You should confirm that these settings are indeed secure for the installation, or adjust the settings to be secure.

When setting up the zone, you can specify which categories of URLs should be considered. You can also add specific sites to the zone.

▶ **To set up sites for the Local Intranet zone**

1. On the **Security** tab, click **Add Sites**.
2. Select the sites to be included in this zone.