

```
sprintf("%02s", dechex($parts[2])) .  
sprintf("%02s", dechex($parts[3]));  
return $color;  
}
```

```
function add_line($msg) {  
    $f = fopen("/tmp/wall.html", "a");  
    $dt = date("Y-m-d h:i:s");  
    $msg = strip_tags(strip_tags($msg));  
    $remote = $_SERVER["REMOTE_ADDR"];  
    // generate unique-ish color for IP  
    $color = colorify_ip($remote);  
    fwrite($f, "<span style='color:$color'>");  
    fclose($f);  
}
```

```
function refresh() {  
    $lines = file("/tmp/wall.html");  
    // return the last 25 lines  
    return join("\n", array_slice($lines, -25));  
}
```



# THE BROWSER

# RELOADED



A new development model called **AJAX** breathes **new life** into Web apps

ONE YEAR AGO, THOMAS LACKNER DIDN'T ASK MUCH OF JavaScript. When he sketched out the architecture to a Web application, he knew he could count on the browser language for "set-a-cookie hacks" and for loading images, but he turned to the server side for the heavy lifting. But when Google began launching highly interactive Web sites such as Gmail and Google Suggest, the scales dropped from Lackner's eyes and he saw the opportunity.

"It all clicked in my head the middle of 2004," Lackner says.

"I started trying to add AJAX components to every Web app I worked on."

AJAX is the newly minted acronym encompassing a fresh vision of empowered browsers: Asynchronous JavaScript and XML. Before AJAX, Web pages displayed links, forms, and but-

tons. When a user clicked on a link or a button, the browser sent a message to a distant server asking what to display next. JavaScript would typically be used for nothing more than to check form inputs. Web pages were as static as pages in a book.

The post-AJAX browser can think on its own without saying "Mother may I?" to a distant server. Processing threads running in the background preloads page content. If a user clicks on a link or a button, the browser can update page content using JavaScript. Google's Gmail, for instance, hides or displays parts of an e-mail thread without waiting for a response from its server, eliminating network lag. If information must go to the server, it too is sent by a background process.

Brendan Eich, the creator of JavaScript at Netscape, says the world is just now discovering what he imagined back in 1995. He says, "We [Marc Andreessen and I] always intended JavaScript to enable client-centric apps that did not have to reload pages from a server."

BY PETER WAYNER | ILLUSTRATION BY BEN BARBANTE

## A New Kind of App

AdaptivePath's Jesse James Garrett, who coined the AJAX acronym in an essay written in February, says that the response has been surprising. "Right now, it's all so new that what we've got is a lot of excitement," he says.

The excitement is understandable when you consider the advantages AJAX brings. For one thing, it greatly simplifies software distribution. Browsers load AJAX applications automatically. Customers are often reluctant to install custom applications, but most people can be convinced to visit a Web site.

The AJAX approach has several other advantages, as well. JavaScript running on the client reduces bandwidth and processing demands on the server. Well-designed code that is truly asynchronous also gives the server more time to respond to queries, reducing this load even further by spreading out peak demands. It even increases security by encrypting data on the client side, before it travels across the network.

The Basecamp project-management application from 37signals is a good example of a Web application developed using the new paradigm. Adding a new to-do item pops open a window to accept input without a round-trip to the server. But when the new item is saved, the browser still must wait to ensure that it was stored successfully. JavaScript code updates the server in the background while displaying the note, "Moving ... Just a moment." There is still plenty of interaction with the server, but the JavaScript code speeds it by handling many of a user's clicks.

David Heinemeier Hansson, a programmer at 37signals, says he's concentrating on removing the lag time when a user submits a form. "If you have a Weblog and you add a comment, it will



be updated on the server side in the background. Whenever you need to add or change content, you can do it without reloading."

Some of the buzz surrounding AJAX has been generated by Web designers as well as programmers. AJAX's flexibility is invigorating for Web designers because JavaScript can control any aspect of any images or type on a page. Fonts can grow or shrink. Tables can add or lose lines. Colors can change. Although none of these capabilities are new to programmers accustomed to building client applications — or, for that matter, Java applets — they are novelties to Web designers who would otherwise be forced to rely on Macromedia Flash.

The excitement may signal a newfound synergy between designers and programmers, who can now speak the same language about the page. Notations from CSS, a language spoken by Web designers, are used directly by the JavaScript coders. The result is a Web page that behaves with some of the snappiness of a dedicated client application — without any of the headaches of distributing and installing software on a client machine.

"For me, it's not about building something big, it's about making sure that the experience is right for the right people," says Jason Fried, president of 37signals.

## Browser Problems Persist

Enthusiasm for these new techniques,

however, is usually tempered by reality. There are few good tools for AJAX development, the platform can be unstable, and adherence to standards is inconsistent. Even the biggest proponents complain about differences between Web browsers and concede that they don't understand the best way to add many interactive features. What's more, these new capabilities can confuse users who don't expect the features and, in some cases, can even open up new security gaps.

Many AJAX applications require features that are only included in the version of JavaScript bundled with newer browsers such as Firefox or IE 5.5. Others use functions that work correctly in one version of a browser but not another. The XMLHttpRequest object, for example, parses XML from Web services directly, a feature that makes it much easier for a programmer to interact with any source of XML. Before this feature debuted in IE 5.0, developers could still download information from distant Web sites, but they needed to use different frames.

Some of the newest browsers offer stable platforms for using XSLT (XSL Transformation), but the details seem to be fluid. There are big differences in the way IE 5.0, 5.5, and 6.0 handled namespaces. Mozilla 1.8 now shares many of the same capabilities, but earlier versions don't.

Programmers are forced to smooth over some of these ambiguities by building custom loaders that match the code to the browser version. Rob Brown, an early AJAX developer and creator of the Firefox plug-in Aardvark, is optimistic. "Luckily, most browser differences can be fairly easily encapsulated into a few utility functions, and the 'meat' of your AJAX application can be free from ugly conditional code," he says.

# Development tools for building JavaScript are still few and they often have a rough feel to them.

But these contortions can be painful, and some are simply abandoning earlier browsers. “We’re working on a project now called Backpack and it’s going to be one of the most advanced AJAX apps outside of Gmail,” 37signals’ Fried says. “We made the decision to just say no to IE 5. It was a conscious decision we made. It’s about time.”

This same challenge faces many enterprise designers. If a Web applica-

tion will only be used internally, an enterprise architect can reasonably assume that every browser in the company meets a certain specification. Such a uniform infrastructure is much easier for programmers. But a company that distributes Web applications to customers can’t make the same assumptions. There will always be some customers running old versions of browsers.

## Code Headaches

Dealing with browser ambiguities is just the first challenge. Development tools for building JavaScript are still few and they often have a rough feel to them. Features that programmers of other languages take for granted are just now appearing in the JavaScript world — for example, debuggers such as Mozilla’s Venkman aren’t widely used by JavaScript coders accustomed

## Pushing AJAX Forward

THE RAPID SPREAD OF THE TERM AJAX (ASYNCHRONOUS JavaScript and XML) — from Weblog to Wall Street Journal within weeks! — might lead developers to assume it’s a breakthrough that heralds the death of desktop applications. There’s certainly a kernel of truth in that: The recent spate of new Web applications under the AJAX banner have redefined end-users’ expectation of what’s even possible within a Web browser by offering smooth scrolling, incremental updates, and more responsive input forms.

Nevertheless, so-called fat-client UIs still retain one fundamental advantage over Web UIs: real-time event notification. AJAX alone does not address IM, stock tickers, and other collaborative applications that require “push” data streaming.

The key goal of AJAX-style applications is to decompose jarring transitions that download an entire new Web page into a series of smaller, more frequent transactions. Developers consider AJAX to be “asynchronous” because data can be updated without interrupting the user. For example, Google Maps dramatically reduces the perceived latency of scrolling a map by only downloading the newly visible tiles and moving the rest.

In the middleware community, however, the formal definition of asynchrony refers to the capability of sending a message at any time, in either direction. AJAX provides the upstream direction, but HTTP would appear to make server-initiated transmission impossible.

Fortunately, clever developers have exploited a loophole in HTML to address this. Browsers are designed to display pages incrementally while downloading pages from slow Web sites. Using hidden frames and JavaScript tags, HTTP

can be used to hold open a long-lived response connection, allowing an application to stream data into the browser.

The simplest way to exploit this is to turn the browser into a 21st-century “green screen” dumb terminal. Manuel Kiessling’s open-source ARSC (A Really Simple Chat) uses AJAX techniques to send input lines upstream, whereas a modified HTTP server that holds open thousands of simultaneous connections rebroadcasts the chat stream to other users. Another example is KnowNow’s SpeedReader product, which is useful for alerting employees to relevant RSS news items.

The subtler and broader implication of combining AJAX with asynchronous event notification is to extend publish-and-subscribe application integration across the Internet.

Several open-source platforms provide powerful abstractions for connecting fully interactive Web UIs to enterprise applications and Web services. Nevow (née Woven LivePage) and Pushlets extend the event loop familiar from model-view-controller GUIs for Python and Java, respectively. Mod\_PubSub is designed as an event bus that uses URL path names as topics to implicitly invoke programs written in a wide range of languages. Commercially, KnowNow’s LiveServer provides enterprise-class scalability (and even connects to and from Excel spreadsheets).

The clear benefits of migrating desktop applications to the Web in terms of maintenance, security, and scalability must be weighed against the costs of slower response times, limited interactivity, and less-than-beautiful graphical interfaces. With AJAX, push technology, and the ubiquitous plug-ins for PDF and Flash, the Web is closer than ever to becoming a viable default platform for application development.

— Rohit Khare

## Accelerating the Web

The AJAX model frees Web applications from the traditional request/response cycle, allowing network communications to take place asynchronously.

to reloading pages to look for errors.

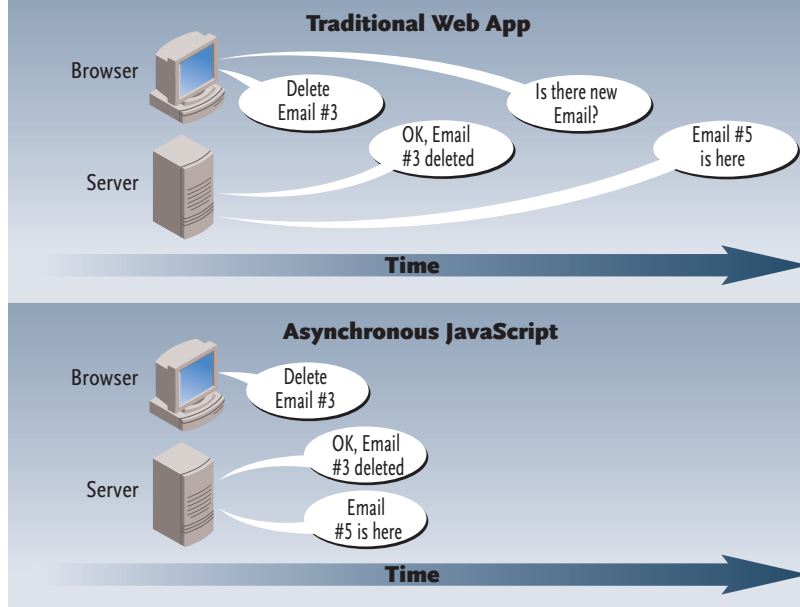
Ed Felten, a professor of computer science at Princeton University, cautions that programmers may inadvertently insert security holes when translating server-side features into JavaScript code. Although JavaScript runs incoming code in a “sandbox” with no API for accessing local files, opportunities for mayhem remain because the code can still access external Web sites. DDoS attacks, for instance, can be easy to code.

What’s more, JavaScript code is not compiled the way languages such as C and Java are. And end-users can view or even edit the JavaScript code before it runs. For example, an attacker could search for strings of text containing SQL instructions and replace that code with different queries that retrieve different information from the server. “Every time a string goes anywhere, you have to think about it,” Felten says.

The appearance of new glitches such as these requires Web designers to think twice about new features. Browser users have come to understand the way traditional Web applications work. They have been warned many times, for example, that clicking a button twice could create a double charge on their credit card. AJAX applications break many of those assumptions.

“People are used to clicking a check box and confirming things. Now you don’t have to. That introduces some confusion,” 37signals’ Fried explains. “Now in some places there’s a button and in some places there isn’t. There are a lot of pitfalls, but they’re not insurmountable by any means.”

What’s more, the JavaScript language itself is still evolving. Whereas languages such as Java and C++ are highly standardized and governed by relatively rigorous committees, JavaScript still



feels like an experiment at times. There are huge differences between the way that IE, Mozilla, and Safari execute some JavaScript code and often substantial differences between different editions of the browser. The core of the language is a standard known officially as ECMAScript, but each browser often handles other features differently.

### Will AJAX Prevail?

Despite the cross-platform challenges — and some developers point out that there is little reason for Microsoft to make its browser completely compatible with Firefox or Opera — this new model shows promise. AJAX applications can have much if not all of the functionality of client-side applications. Many programmers react with surprise when they see editors such as Bitflux, FckEditor, or Kupu written entirely in JavaScript, but applications such as these are likely to become commonplace, and even more features are bound to emerge (see “Pushing AJAX

Forward,” page 41).

Some companies see an opportunity here. Thomas Lackner’s development group at ModernMethod is rolling some of its development efforts into an open-source library Lackner calls SAJAX, short for Simple AJAX. JackBe, on the other hand, sells a collection of small libraries that abstract away many browser incompatibilities. A programmer can add Excel-like spreadsheets or forms to a Web page and the JackBe libraries will handle much of the interaction with the client.

Which brings us back to the future as the Web finally discovers what Eich and the original team at Netscape imagined long ago. When asked what held it back, Eich speculates, “It was hampered by incompleteness, a rush to standardize, the browser wars killing Netscape, and a dependence on Java to do ‘real graphics’ and ‘real network input/output.’” Now that JavaScript has matured beyond those problems, Eich is enthusiastic to see what comes next. ☛

NEW SUBSCRIBERS ONLY!

Apply online at: <http://subscribe.infoworld.com>

Priority Code: **MT4PDF**

Form: 18

I WISH TO RECEIVE A FREE SUBSCRIPTION TO **InfoWorld**

1. Yes  0. No

SIGNATURE \_\_\_\_\_ DATE \_\_\_\_\_

## A MAILING ADDRESS

Publisher reserves the right to limit the number of complimentary subscriptions. Free subscriptions available in the U.S. (including APO and FPO) and Canada.

NAME \_\_\_\_\_

TITLE \_\_\_\_\_

COMPANY NAME \_\_\_\_\_

DIVISION / DEPT / MAIL STOP \_\_\_\_\_

MAILING ADDRESS \_\_\_\_\_

CITY / STATE / ZIP / POSTAL CODE \_\_\_\_\_

Is the above address a home address?  1. Yes  0. No

BUSINESS PHONE (INCLUDING AREA CODE) \_\_\_\_\_ FAX NO. (INCLUDING AREA CODE) \_\_\_\_\_

E-MAIL ADDRESS \_\_\_\_\_

## B SPECIAL REQUEST:

You may receive a renewal reminder via e-mail. May we send other information about **InfoWorld** products or services via e-mail?  1. Yes  0. No

Reader feedback is important to us. May a member of our editorial team contact you?  1. Yes  0. No

## 1 WHAT IS YOUR ORGANIZATION'S PRIMARY BUSINESS ACTIVITY AT THIS LOCATION? (PLEASE CHECK ONE ONLY):

### General Business Industries

- 01. Defense Contractor / Aerospace
- 02. Retail
- 03. Wholesale / Distribution (non-computer)
- 04. Pharmaceutical / Medical / Dental / Healthcare
- 05. Financial Services / Banking
- 06. Insurance / Real Estate / Legal
- 07. Transportation / Utilities
- 08. Media (print / electronic)
- 09. Communication Carriers (telecomm, data comm., TV / cable)
- 10. Construction / Architecture / Engineering
- 11. Manufacturing & Process Industries (other than computer-related)
- 12. Research / Development

### Technology Providers

- 13. Managed Service Provider / Business Service Provider
- 14. Technology Service Provider (ISP / ASP / MSP, etc.)
- 15. Computer / Network Consultant
- 16. Systems or Network Integrator
- 17. VAR / VAD
- 18. Technology Manufacturer (hardware, software, peripherals, etc.)
- 19. Technology - Related Retailer / Wholesaler / Distributor
- 20. Government: federal (including military)
- 21. Government: state or local
- 22. Education
- 98. Other \_\_\_\_\_ (Please specify)

### Government / Education

## 2 WHAT IS YOUR PRIMARY JOB TITLE? (PLEASE CHECK ONLY ONE):

### IT / Technology Professionals

- 01. Chief Technology Officer (CTO)
- 02. Chief Information Officer (CIO)
- 03. Chief Security Officer (CSO)
- 04. Vice President (including SVP, EVP, etc.)
- 05. Director
- 06. Manager / Supervisor
- 07. Engineer
- 08. Systems Analyst / Programmer / Architect
- 09. Consultant / Integrator
- 10. Developer
- 11. IT Staff
- 12. Other IT Professional \_\_\_\_\_ (Please specify)

### Corporate / Business Management

- 13. CEO, COO, President, Owner
- 14. CFO, Controller, Treasurer
- 15. Vice President (including SVP, EVP, etc.)
- 16. Director
- 17. Manager / Supervisor
- 18. Other Business Management Title \_\_\_\_\_ (Please specify)

98. Other Title \_\_\_\_\_ (Please specify)

## 3 PLEASE INDICATE YOUR JOB FUNCTION(S)? (PLEASE CHECK ALL THAT APPLY):

### IT / Technology Functions

- 01. Executive
- 02. Department Management - IT
- 03. Research and Development Management
- 04. Systems / Network Management
- 05. Management of Enterprise Applications (CRM, ERP, SCM, etc.)
- 06. Applications Development
- 07. Consultant / Integrator
- 08. Other IT Department Management \_\_\_\_\_ (Please describe)
- 09. Other IT - Staff \_\_\_\_\_ (Please describe)

### Corporate / Business Functions

- 10. Executive
- 11. Department Management - Business
- 12. Financial / Accounting Management
- 13. Research and Development Management
- 14. Sales / Marketing Management
- 15. Other Department Management
- 16. Other Department Staff \_\_\_\_\_ (Please describe)
- 98. Other \_\_\_\_\_ (Please describe)

## 4 HOW MANY PEOPLE ARE EMPLOYED AT THIS ORGANIZATION, INCLUDING ALL OF ITS BRANCHES, DIVISIONS AND SUBSIDIARIES? (PLEASE CHECK ONE ONLY):

- 01. 20,000 or more
- 02. 10,000 - 19,999
- 03. 5,000 - 9,999
- 04. 1,000 - 4,999
- 05. 500 - 999
- 06. 100 - 499
- 07. 50 - 99
- 08. Less than 49

## 5 OVER THE COURSE OF ONE YEAR, DO YOU BUY, SPECIFY, RECOMMEND, OR APPROVE THE PURCHASE OF THE FOLLOWING PRODUCTS OR SERVICES WORTH:

\* CONSULTANTS: PLEASE INCLUDE WHAT YOU RECOMMEND FOR YOUR CLIENTS AS WELL AS WHAT YOU BUY FOR YOUR OWN BUSINESS, IF APPLICABLE. IF YOU CANNOT DISTINGUISH BETWEEN THIS AND OTHER LOCATIONS, PUT RESPONSE IN THE FIRST COLUMN.

- |                                  |                                |                            |
|----------------------------------|--------------------------------|----------------------------|
| 01. \$100 million or more        | 06. \$5,000,000 to \$9,999,999 | 11. \$100,000 to \$399,999 |
| 02. \$50,000,000 to \$99,999,999 | 07. \$2,500,000 to \$4,999,999 | 12. \$50,000 to \$99,999   |
| 03. \$30,000,000 to \$49,999,999 | 08. \$1,000,000 to \$2,499,999 | 13. Less than \$49,999     |
| 04. \$20,000,000 to \$29,999,999 | 09. \$600,000 to \$999,999     | 14. None                   |
| 05. \$10,000,000 to \$19,999,999 | 10. \$400,000 to \$599,999     |                            |

### Product category

For this location:  
(write code in box)

For other locations:  
(write code in box)

Large systems	<input type="text"/>	<input type="text"/>
Client computers	<input type="text"/>	<input type="text"/>
Networking / Telecom (including servers)	<input type="text"/>	<input type="text"/>
Internet / Intranet / Extranet	<input type="text"/>	<input type="text"/>
Security	<input type="text"/>	<input type="text"/>
Storage	<input type="text"/>	<input type="text"/>
Peripheral equipment	<input type="text"/>	<input type="text"/>
Software	<input type="text"/>	<input type="text"/>
Service / Support	<input type="text"/>	<input type="text"/>

Please answer the questions on the following page. 

**6 PLEASE TELL US YOUR INVOLVEMENT WITH YOUR COMPANY'S STRATEGIC TECHNOLOGY INITIATIVES (PLEASE CHECK ALL THAT APPLY):**

- 01. Integrate Technology with company goals
- 02. Define Architecture
- 03. Choose Technology Platforms
- 04. Develop Technology Integration Strategy
- 05. Test, pilot, implement emerging technologies
- 06. Scalability Planning
- 07. Build, Run Web Services
- 08. Internet / Network Infrastructure
- 09. Customer Relationship Management
- 10. External Partnership Management
- 11. Budgeting
- 12. Recruitment & Retention
- 13. Other \_\_\_\_\_ (Please describe)
- 99. None of the above

**9 ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING TECHNOLOGY SERVICES? (PLEASE CHECK ALL THAT APPLY):**

- 01. Technology Services
- 02. Systems / Application Integration
- 03. E-Business / Internet / Intranet / Extranet
- 04. Application Development
- 05. Application Hosting (ASP)
- 06. Web Hosting
- 07. Web Development
- 08. Security
- 09. Storage
- 10. Content Delivery Networks
- 11. Disaster Recovery / Business Continuity
- 12. Outsourcing
- 13. Utility Computing Services
- 14. Telecommunications
- 15. Call Center / IT Services
- 16. Consulting
- 17. Other Technology Services

**7 ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING SOFTWARE? (PLEASE CHECK ALL THAT APPLY):**

- 01. Enterprise / E-Business Applications
  - 02. Customer Relationship Management (CRM / eCRM)
  - 03. Enterprise Resource Planning (ERP)
  - 04. Supply Chain / Procurement
  - 05. Business Process Management
  - 06. Business Intelligence / Data Mining
  - 07. Knowledge Management
  - 08. Portals
  - 09. Collaborative Applications / Groupware
  - 10. Project Management
  - 11. Financial / Payroll / Billing
  - 12. E-business / E-commerce
  - 13. Database Management Systems (DBMS)
  - 14. Data Warehouse
  - 15. Manufacturing
  - 16. Asset Management / Software Distribution
  - 17. Performance / Application Management
  - 18. Streaming Media
  - 19. Other Enterprise / E-Business Applications
- 20. Integration Software
  - 21. Web Services
  - 22. Web Services Orchestration
  - 23. Application Servers
  - 24. Enterprise Application Integration (EAI) / Middleware
  - 25. Business Process Management
  - 26. Legacy Application Integration Tools
  - 27. Other Integration Software
- 28. Application Development
  - 29. Application Development Tools
  - 30. Application Servers
  - 31. Web services
  - 32. Java / J2EE
  - 33. XML
  - 34. .NET
  - 35. Testing Tools
  - 36. Other Application Development Software

**10 ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING PRODUCTS OR TECHNOLOGIES? (PLEASE CHECK ALL THAT APPLY):**

- 01. Networking
  - 02. LANs (Local Area Networks)
  - 03. WANs (Wide Area Networks)
  - 04. Switches / Routers / Hubs
  - 05. Caching / Load Balancing
  - 06. Grid / Utility Computing
  - 07. E-mail
  - 08. Instant Messaging / Peer-to-Peer
  - 09. Content Delivery Networks
  - 10. Network and Systems Management
  - 11. Traffic Monitoring and Analysis
  - 12. QoS (Quality of Service)
  - 13. VoIP (Voice over IP)
  - 14. Telecommunications
  - 15. IP Telephony
  - 16. Wireless
  - 17. Remote Access
  - 18. Web / Video Conferencing
  - 19. Other Networking
- 20. Storage
  - 21. High-end / Enterprise Class Storage
  - 22. Network Attached Storage (NAS)
  - 23. Storage Area Networks (SANs)
  - 24. Storage Management Software
  - 25. IP Storage
  - 26. Direct Attached Storage (DAS)
  - 27. Storage Blades
  - 28. Storage Backup (Tape, Disk, Optical, RAID)
  - 29. Removable / Portable Storage
  - 30. Disaster Recovery
  - 31. Other Storage
- 32. Security
  - 33. Anti-Virus / Content Filtering
  - 34. Firewall
  - 35. VPN (Virtual Private Network)
  - 36. Identity Management / Authentication
  - 37. Intrusion Detection
  - 38. Encryption
  - 39. Other Security
- 40. Internet / Intranet / Extranet
  - 41. Web Servers
  - 42. Web Development / Authoring Tools
  - 43. Web Performance Management / Monitoring Software
  - 44. Content Management / Document Management
  - 45. Content Delivery Networks
  - 46. Internet Software
  - 47. Other Internet / Intranet / Extranet

**8 ARE YOU INVOLVED IN BUYING, SPECIFYING, RECOMMENDING OR APPROVING THE FOLLOWING HARDWARE? (PLEASE CHECK ALL THAT APPLY):**

- 01. Hardware
  - 02. Mainframes
  - 03. NT / Windows 2000 / .NET Servers
  - 04. Unix Servers
  - 05. Linux Servers
  - 06. Blade Servers
  - 07. PCs / Workstations
  - 08. Notebooks / Laptops
  - 09. PDAs / Handhelds / Pocket PC / Wireless Devices
  - 10. Other Hardware
- 11. Peripherals
  - 12. Laser Printers
  - 13. Inkjet Printers
  - 14. Monitors
  - 15. Flat Panel Displays
  - 16. UPS (Uninterruptible Power Supply)
  - 17. Network Copiers
  - 18. Other Peripherals

**11 WHICH OF THE FOLLOWING OPERATING SYSTEMS ARE IN USE OR PLANNED FOR USE AT THIS LOCATION? (PLEASE CHECK ALL THAT APPLY):**

- 01. Windows XP
- 02. Windows 2000
- 03. Windows NT
- 04. Windows 95/98
- 05. Windows CE
- 06. Mac OS (Macintosh)
- 07. Solaris
- 08. UNIX
- 09. Linux
- 10. MVS, VMS, ESA
- 11. VM
- 12. OS 400
- 13. Netware
- 14. Palm OS
- 15. Other OS

**Return this form to InfoWorld,  
P.O. Box 3511,  
Northbrook, IL 60065-3511 or  
FAX to (847) 291-4816.  
You can also apply ONLINE at <http://subscribe.infoworld.com>**

