From: Chris Jones
Sent: Saturday, May 31, 1997 7:28 PM
To: David Cole
Subject: RE: ie5 planning

here's a rough start. basically took sinofsky's office9 vision and started from that, then included ie 5 stuff. will work on it more tomorrow. pls don't fwd yet, note that i want hadip to drive this so i need to spend some time with him as well. you should read stevesi's stuff (http://officeweb/Visions/office9.htm), i think it is pretty good (though wordy).

mission statement.doc

-- chris

-----Original Message-----
From: David Cole
Sent: Saturday, May 31, 1997 5:32 PM
To: Chris Jones
Subject: RE: ie5 planning

great. down one level from the top level mission, might be worth while to think about the mission in terms of customer sets. what will we do for end users, corps, icps, isps, developers, etc. The stuff I listed below is busy work we want done, but certainly doesn't touch on the mission.

-----Original Message-----
From: Chris Jones
Sent: Saturday, May 31, 1997 1:38 PM
To: David Cole
Subject: RE: ie5 planning

i am sort of trying to think about this today. i will try to send you something by end of this weekend on exactly this subject. your list is a good start.

-----Original Message-----
From: David Cole
Sent: Saturday, May 31, 1997 8:56 AM
To: Chris Jones
Subject: ie5 planning

maybe this must wait til we get beta 2 out, but we need some formal planning in each of the product areas for this. the standard practice is the developers just go do random work and put us in a hole wrt our ability to start on the real product we want. we must make sure each team has a clear agenda from day we rtm ie4.

how do you want to go about making this work? the ie5 mission and focus must be clear from the start.

Trident: there is some incremental work for supporting VI98, and they have to embark on the weblications support, which includes supporting authoring apps.

Shell: split the browser and shell code to allow us to better release separately. fix bugs for memphis, add shell features for NT5.

Info delivery: I haven't thought about things much here.

The above is just needed busy work, in addition to fixing bugs, doing QFEs, etc.

GOVERNMENT EXHIBIT 494

IE 5 Vision

The Vision for IE 5

Vision: IE 5 is the most popular client platform for developing and deploying Internet-centric communication and publishing solutions.

At the start, IE 5 is a compelling and exciting product that generates demand when demonstrated to individuals and influential end-users. IE 5 is a must-have and no-brainer upgrade for both IE 4 and Navigator 4 customers.

IE 5 is the foundation of the HTML applications platform, and enables the next level of richness in Internet-centric applications as well as HTML as a presentation language.

IE 5 is optimized for Windows 95 and NT operating systems. IE 5 runs best on Win32, and delivers a complementary UI and platform message with existing and future Win32 systems. In order to ensure adoption of the HTML applications platform and deliver a great communications solution, a sub-set of IE 5 runs on Windows 3.1, Mac PPC, and UNIX® operating systems.

While IE 5 works great against all Web servers, IE 5 runs best when connected to BackOffice/IIS servers.

IE 5 is designed to be easy for authoring tools and solutions to target. In particular, IE 5 will deliver the best platform for tools such as Microsoft Office, Microsoft Visual Studio, and other third party tools.

IE 5 demonstrates industry leadership, provides a clear path for long-term customer value, and improves end-user and developer productivity by focusing on the following product areas:

- Migration, Administration, Deployment, and Management
- HTML as an Application Platform
- HTTP as a Collaborative Protocol
- Offline and Remote Usage

Customer Value Proposition

Part of the challenge for the Internet Explorer team is defining the value we deliver to customers. For Internet Explorer 3.0, our value proposition was as good or better than Navigator and free. This proposition, while interesting, is a "me too" strategy that does not focus our innovation on solving real customer problems or delivering solutions. We need a single message that allows our team to focus and take leadership by delivering solutions for customers that are leading edge and showcase our innovation.

Internet Explorer is the most compelling and cost effective client platform for developing and deploying Internet-centric communication and publishing solutions (or "weblications").

Due to the nature of communication and publishing, it is essential that we have a story

Our top objective is to build a platform; this means that we live and die based on the number of people using our platform, not by applications we build. While we will still build and deliver applications, their focus is to promote the platform, and they are products only as a by-product of the need to start adoption. Our applications will all be build fully on top of our platform, and their focus and purpose is to drive platform adoption and hence share.

This means that we will optimize our decisions around features that favor the developer or administrator, and focus less on end users and influentials.

Table 1. IE 5 Value Propositions by targeted customer segment

Customer	Value Proposition
End-user/Individual	IE 5 is the best, easiest to use, and most integrated suite of applications for reading, browsing, and communicating. IE 5 is exciting and fun to use and people will both want to and need to upgrade. IE 5 is as fast or faster than IE 4 or Navigator 4.
Influential End-User	IE 5 is a must-have upgrade that allows the guru/influential user to do new things that impress others in the office. IE 5 provides leading edge content and collaboration solutions that allows the power user to view the best content sooner.
Administrators	IE 5 supports customizable and centralized distribution, roaming user-profiles, and self-

	diagnosing and repairing applications. IE 5 is a no-brainer upgrade from both IE 4 and Navigator 4 because we are fully compatible with existing content and solutions and migrate all custom applications and user-settings.
Developers	IE 5 provides the most compelling platform for developing the next generation of applications. IE and IIS, plus Visual Studio, make it easy to deliver full featured Internet-centric applications which degrade for access anywhere. Dynamic HTML provide a new way to build user interface for applications which enables more richness and interactivity than ever before.
Influentials and CIOs	IE 5 integrates with and leverages key Microsoft initiatives including Zero Administration Windows, Office, Windows NT Server, and BackOffice. At the same time, IE 5 embraces open Internet standards and builds unique value on top of those standards. IE 5 is a natural complement to the next generation of the Windows platform and makes it easy to migrate machines to the next Windows. IE 5 does not drive hardware purchases and performs as well or better than IE 4 in real-world benchmarking.

The value proposition establishes a long term and multi-release commitment with our customers. IE 3 and IE 4 began this commitment, IE 5 is the next step, and IE will continue to deliver on this proposition.

Competition

The watchwords for dealing with the current competitive situation can be taken from Andy Grove's recent best seller-Only the Paranoid Survive. It is a fact that our traditional operating system competitors (Apple, OS/2, and UNIX) have been left in the dust by Windows 95 and Windows NT on the client. There is simply no chance that we will lose sales because of lack of feature parity with those traditional products. Of course these companies, and all the new competitors, know that and are rapidly devising strategies that might lead to victory through an end-run or changing the rules of the game, mostly through an effort to build "middle-ware" solutions that render Windows and Windows services irrelevant. Thus our competitive focus for IE 5 will be on a new breed of communication and collaboration products that do not compete with Windows head-on, but offer customers an alternative platform environment that might accomplish the more relevant tasks (communication and collaboration) either more efficiently or more cost-effectively.

Network Computer (NC) and Java both represent our customers' best hope for a correction of the sins of the Windows PC. The NC platform with Java applications represents a world where new technologies and paradigms solve the management, deployment, and training issues negatively associated with Windows. For IE 5 by radically simplifying these aspects of Windows, and at the same time delivering a much richer user-experience, we will bridge the existing users to this new paradigm.

Netscape Communicator defines a new platform, taking advantage of the lessons learned from Visual Basic, Visual C++, Java, and Web content. They are completely focused on turning their applications framework (HTML, object model, scripting, and JFC) into the primary way developers deliver Internet-centric applications. As part of this strategy, they are leading the way in integrating browsing with electronic mail and discussions. They are also working to deliver a new user model and paradigm for navigating the local computer in an effort ot make the Windows shell obsolete. We must deliver on a broad set of features that address the platform threat of HTML and scripting, plus integrate communication seamlessly with the browser. In addition, we have to work together with the Windows team to promote the Windows shell as the premiere desktop environment.

HTML authoring tools such as Visual JavaScript, Netscape Composer, and others that uniquely target the Netscape/Sun platform are what enables developers to deliver solutions. While these are not directly features of our product, enabling the best tools and applications is the most critical part of our success. IT is critical that we make it easy for Visual Studio, Office, FrontPage, and 3rd party tools to target the Internet Explorer 5 platform.

Lotus/IBM Notes will continue to attempt to deliver middleware on Windows and provide an alternate platform, pushing a server-centric (and often user-hostile) strategy that advocates reducing client side processing. We can expect them to continue to portray Windows as a negative in favor of pushing a strategy of components and glue or even no client code at all. We will address this threat by a combination of better administration on the client and an architecture that supports specific IE scenarios by leveraging the support of web servers running IE/IIS code as well as tight integration with Windows NT and BackOffice.

Product, Priorities, and Timing

IE 5 will be delivered as a single integrated product that represents the needs of the majority of our customers. We will actively move lesser-used components, especially content, out of this core product and make them available via the web where end-users can obtain them and corporations can choose to incorporate them in custom deployments. This core product includes the browser, Outlook Express, and Java.

Schedule. TBD.

Table 2. Major milestones for the IE 5 schedule.

Date	Milestone
TBD	Visions Completed
TBD	Coding Starts
TBD	Code Complete!
TBD	Beta I
TBD	Beta II
TBD	RTM USA
TBD	RTM Japan

IE 5 is a focused release that will ship in 1998. Unlike IE 4, IE 5 will not be tied to a shell upgrade and instead focus solely on delivering the application platform. In order to make this realistic, IE 5 will be focused on Total Administration, HTML Applications Platform, HTTP as a Collaborative Protocol, and Offline Usage. If you need to bend over backwards to make something fit into one of these buckets, then there is a good chance the feature should be reconsidered for Office9.

Total Administration

IE 5 (with ZAW) will deliver on the promises of the NC in terms of manageability and deployment. Although we tend to focus on the setup process and technology, the primary objective for this effort is to remove setup from the equation and deliver our applications so that setup is really only a glorified copy process. Every single decision we make in the product has an impact on the cost of ownership for our most important corporate customers.

The functionality goals for the Office9's efforts on manageability and administration include:

- Install and deploy: IE 5 makes it easy to deploy from a central location to a set of desktops in a networked environment. An important option will include controlling how much gets installed locally and how much will be run from the central location. All features of IE 5 will be advertised from the running applications, even if they have not been installed, and accessed via just-in-time installation.
- Customize: IE 5 provides the ability to choose only the components that are needed on a reasonably granular level and to set each preference prior to the deployment.
- Upgrade: IE 5 will smoothly upgrade from previous IE and Netscape releases, which includes migrating the old settings and user's work to the new installation and management metaphor. Key to this work is recognizing that IE 5 will be run in a mixed environment with other versions of IE and Netscape.

- Unify preferences and settings: IE 5 will consistently store, maintain, and customize preferences and settings. We will aggressively seek to provide administrators the option to lock down important settings and provide a means for customization of key settings at deployment.
- Support roaming user and laptops: IE 5 will implement features to enable documents and customizations to roam with the user between multiple machines. IE 5 will address the needs of laptop users by providing an easy way to control what travels with the machine and what roams via the server.
- Deliver resilient applications: IE 5 applications will be run even when not all components are installed, when the registry is corrupted, when files have been mistakenly deleted, or when the default source for the just-in-time installable components disappears or changes.
- Administer and maintain: IE 5 supports central administration via NT 5 or the IEAK. Updates and patches can be pushed from a central location and administrators will be able to monitor and report the installation problems and application status from a central place.

HTML as an Applications Platform

HTML will play a central role in Office9, as it is the common substrate upon which communication scenarios are enabled. This includes electronic mail in Outlook (Outlook will have an HTML-based mail note released this year). We face many challenges in moving our proprietary document formats to HTML as there is a deep relationship between the features in Office and the storage format used to persist them. Nevertheless, the importance of HTML is such that we will give up any single feature, or even a large number of features, to enable the bulk of user-created documents to be stored, all the time, in HTML. By leveraging standard HTML for representing our documents we gain the full benefits of universal viewing in the user's choice of browser and the ability of our documents to be processed (indexed, annotated, source controlled, etc.) by independent code on the server or client.

For Office9, our applications will all target HTML as a first class file format. Given that many of our customers, especially the important Influential End Users, will be targeting web servers as their primary document repositories it is essential that we provide enough HTML support such that these users can reasonably set their default file format to HTML. The benefits of HTML combined with the new HTML document creation features in Office9 will be so compelling that the loss of some features will be tolerated. For all users, we will continue to support fully the Office 97 binary file format so for those corporations that need to run in a mixed environment where many users of different releases of Office must collaborate on documents we have a TCO-friendly answer. For users that prefer the HTML file format, we have a new benefit of universal viewing which allows Office9 documents to be viewed by all users on all platforms for which there is an HTML level 3 browser.

When moving our documents from binary formats to HTML it is easy to get caught up in trying to map each of our application features to some new HTML tag, a control, or a binary blob interpreted only by our applications. We will do a lot of this work and it will be painful. But we will be innovative and find new features in our applications that would not previously have been done in as universal a manner. The evangelist Billy Sunday once said, "[M]any men...have got just enough religion to make them miserable. If there is no joy in religion, you have got a leak in your religion." With less fire and brimstone, the measure of Office9's success with HTML will be measured by more than just pain. The upside for providing innovative editing functionality is huge and an opportunity Office is uniquely positioned to leverage. For existing features that do not move to HTML, the user will still have the option of 100% compatibility with Office 97 by using the binary file format. Innovative features for Office9 must first be designed for HTML and only if absolutely necessary should the work be done to provide this support for the binary format as well. As we transition to this new paradigm there will be some discontinuities.

We will target the common browser for our HTML work in Office9. This means that Office9 documents must look great in Internet Explorer 3.0 and Navigator 3.0. Although the browser space is fast moving and updates are made available frequently, corporate adoption and the general inertia of an installed base will provide a majority of those browsers, or most certainly a highly fragmented installed base, for the foreseeable future. Additionally, a primary advantage Office9 gains from leveraging HTML is the availability of ubiquitous viewing on any platform, which does force a certain lowest common denominator approach. We must succeed in delivering a standard HTML representation of Office documents that looks great in the current generation of browsers.

There are a number of key scenarios where the static representation of a page is not adequate. In the Office 97 environment the user would be running Office to view the document and see this enhanced behavior. For Office9 we will enhance the online presentation of Office documents by leveraging a combination of Java applications/applets and Internet Explorer 4.0's object model. Office9 documents must look amazing in Internet Explorer 4.0.

A key aspect of HTML in Office9 is the use of the user's preferred browser as the primary means of locating, reading, and even interacting with Office documents. Office9 users will see documents in the browser before they choose to edit them, just as today most people see documents on paper rather than online. Office9 applications will continue to be the premier mechanism for high fidelity printing, and interactive viewing. We will make the transition between browser and editor an easy one for users that use HTML.

The functionality goals for the Office9's efforts in HTML editing include:

- Embrace: Office9 will enhance each application to support HTML as a round-trip file format for common documents (Access will continue to store data in MDB). It will be entirely possible for a corporation to standardize on HTML as their Office9 file format at deployment time. The key elements of embracing HTML functionality include: round tripping edit state, maintaining the raw HTML fidelity necessary for third party processing (do not strip comments, adjust white-space, etc.), supporting embeddable objects with a standard mechanism (do not lose the idea of a file containing words and pictures), transfer (i.e. import and export) of HTML-based data between our applications and other HTML applications (browsers, editors, news/mail clients). There will be a base of shared infrastructure to support a consistent and efficient implementation of HTML.
- Extend: HTML 3.2 by itself is not always compelling enough. Office9 will build on the foundation created by embracing HTML to deliver richer documents and online reading/analysis capabilities. The key elements of extending HTML functionality include: creating (and proposing to the W3C) new tag attributes that only our applications understand yet render in today's browsers, working with Internet Explorer on rendering tags offering a unique advantage to Office9+Internet Explorer users, providing a richer viewing and analysis experience with the addition of various runtime components (such as vector graphics and a grid) as a companion to Office9 applications. Office9 will not extend HTML support until the full embrace level of support has been completed.
- Mail: Office9 will leverage the work in HTML to provide rich mail support through the use of WordMail as the standard email editor companion to popular mail clients such as Internet Explorer, Netscape Communicator, Eudora, with emphasis on extremely rich integration with Outlook.
- Graphics: Online documents necessitate great support in graphics and we have a considerable start in Office 97. To date the use of graphics on web pages has been focused on reuse of clip art or on the very high end. Office9's improved

OfficeArt delivers built-in raster and vector graphics that make it easy for you to author cool web documents. Great support for saving OfficeArt as HTML including high quality GIF and JPEG output. By extending the base HTML with client-side code or leveraging new 2-D layout in the browser users will be able to support vector-based graphics and better layout.

Outlook Integration

Even though the browser was the first killer Internet applications, it is clear that electronic mail will become the center of the user's focus in the next few years. We must ensure that Outlook and IE are integrated. Outlook will represent the hub of an Office9 user's workspace and Office9 document creation tools will become an integral part of this workspace.

Given the current planning cycles the major scenarios that will be critical to the success of this mission have not yet been determined. Outlook will release an Internet-standards based upgrade in 1997. During this time Office9 will begin co-designing several marquee application-integration features that will be completed during the second half of development of Office9. There are several important scenarios that are being considered and the list below includes some preliminary ideas. The Office Web Client shared feature team will be driving this integration.

An initial cut at functionality goals for the Office9's efforts on Outlook integration include:

- HTML Forms: IE will deliver great HTML forms solutions designed to make sure that all e-forms can be based on Trident.
- Exchange/IIS Integration: Work great with BackOffice servers.
- Subscriptions and Offline Management: Make news, mail, and web work like one namespace.

Web Collaboration

DAV, etc. Support Office9 scenarios.

- User Model: Office9 will make it easy for users to publish documents to web servers. But more than providing a simple mechanism for POST, the server support will free users from managing the process and implementation of sharing information electronically. Office9 will allow a user to seamlessly replicate documents from the server to a local machine or laptop. The goal of this effort is to bring the ease of use and simple conceptual model of email to the process of creating web pages for use by one, a few, or many others.
- Information Push: Office9 will introduce specific server-side features that will support a push model for distributing and locating documents. Office9 users will be able to subscribe to documents (or the author can create a readership list). There will be an event model for documents (or sets of documents) stored on the server that will allow rich customized behavior for both push and workflow routing.
- Collaboration: Office9 will introduce a new level of collaboration into the document creation process. Essentially documents will contain anchored conversation threads, which can be thought of as live annotations, but with the added structure and security provided by a rich back end data store. These threads will be viewable from within the document, a browser, or a thread viewer (Outlook). Although annotation is immediately interesting, the true value of this functionality comes when documents themselves are created dynamically on the server as the result of merging inputs from various users.
- Annotation: Office9 will support the annotation of documents even when users do not have write access to the document. This will allow a group to annotate any page on a corporate intranet or even the Internet itself. These annotations will be stored and managed on the server.
- Document management and reconciliation: Office9 will provide end-users with traditional document management features such as check-in, check-out,

versioning, and rollback. These will not be required and can be used only when needed: thus this functionality will be unobtrusive.

- FrontPage Integration: The basic functionality of saving documents and link verification and repair will be carried out by the FrontPage server architecture. In this way, Office9 and FrontPage become even more tightly integrated.
- Presentation conferencing: Building on the presentation tools in PowerPoint, Office9 will integrate with the server based NetShow technology to provide streaming multi-cast presentations.

Providing scripting tools within the Office9 applications will enrich the server-side functionality. Just as VBE led the way in providing in-document solutions, the browser-hosted model will be used to allow end-users to create live documents and forms using Office9 applications as editors.

The functionality goals for the Office9's efforts in programmable scripting include:

- Language-independent scripting: Office9 will support scripting (JavaScript and VBScript) for browser-based HTML documents. The VBE development environment and integration will be extended to include the necessary tools to create, edit, and save scripts. Office9 will provide a number of pre-built scripts (analogous to today's ActiveX component well) for many of the common operations
- Object model: Office9 will support mapping today's object model to the object model of the browser. This will allow easy transition between application-hosted solutions and browse-hosted solutions. This will also include improved support for events within the applications to allow for better scripting in browser scenarios.
- WebBots: Office9 will support inserting FrontPage WebBots that add interesting functionality without the overhead of learning and understanding server programming.
- Java components and controls: Office9 will introduce several elements of functionality as browser components. Excel will introduce a component grid that supplies a subset of Excel's functionality as a control. Access will introduce DataPages as a programmable data access page. OfficeArt will introduce advanced rendering controls for vector graphics.

Personal Productivity in IE 5

Easy to find and manage information.

The hallmark of Applications has always been attention to the end-user and making it easy for the typical person to be productive with Office applications. Office9 will continue to invest in the area of improving end-user productivity. Unlike previous releases of Office, traditional ease-of-use will be at the broad suite level rather than incremental improvements in specific applications. In particular, Office9 will make significant investments in hard-to-clone initiatives that will permit an easier to use platform of applications. Office9 will look for opportunities to uniformly improve the ease-of-use of all the applications, improving the overall level of consistency and user-experience.

Productivity is also critical for the influential end-users (IEUs) who will drive upgrades to Office9. Thus there will be a significant investment in making the IEU usage scenarios much more productive. These new scenarios will focus on integrating communication functionality that is increasingly important to these communication savvy users. The challenge of the Office9 productivity efforts will be to focus on making these new communication scenarios (information sharing, browser integration, information management and dissemination) powerful without further bloating Office.

The priority for productivity features across all of Office9 is to enable web editing and Internet scenarios, while investing in infrastructure for the long term.

Office9 prioritize extending existing metaphors and concepts. over introducing new concepts that force training.

The functionality goals for the Office9's efforts in personal productivity include:

- Personal Profile: Office9 will learn from users and keep track of this information. As Office9 applications are used, the assistance mechanisms become smarter and more tuned to the individual. These automatic customizations will roam with the user and it will be easy to customize the amount of learning Office9 does.
- Core Interface: Leverage Win32.
- Worldwide Support: TBD.
- Consistency: As with each release it will critical that we maintain (and improve) the level of consistency in our user interface and there will be efforts to insure that we do not go backwards in this area as we introduce new concepts. In particular with Windows and Office.

Key Scenarios

Each application and shared feature team as a small set of targeted scenarios, which are documented on http://ie40/specs. Program management and Product Design are working on sketches that show these scenarios in action.

Tenets

IE 5 will begin with a base set of assumptions that cross all applications and functional areas. These, in some way, represent the commandments in that they should be taken seriously as each decision is made that impacts the product. These might sound somewhat preachy, but the intent is to set out a framework where hundreds of contributors to IE 5 can each be certain of a small set of global issues.

There are three categories of these tenets. The first group represents the set of basic operating principles or process standards to be used in IE 5. The feature tenets cover the design of the product. The overall product tenets represent the global product issues.

Process Tenets

- All members of the IE 5 team, regardless of the reporting structure, are responsible for the innovations in the IE 5 product. By corollary, the shared feature teams are responsible for the integration of their work in each application.
- Development and process efficiency is critical to the success of the IE 5 schedule, and therefore it is better to do things the same way once rather than doing things in multiple places. This refers both to features and process. In other words, it is better to be the same rather than different.

Feature Tenets

- All features must fit easily within the product vision described. For IE 5 to be successful and focused it is important to follow the spirit of the vision. IE 5 will not advance a feature
- When there is any doubt regarding backward compatibility when making a change to existing functionality, favor the upgrade scenario over improving things for the new user. By implication, IE 5 will not make gratuitous changes to the user-interface.
- IE 5 will embrace before it extends. Fully embracing the Internet standards and experience is core to the IE 5 product. Extending this experience is the long-term direction. Office9 will support the Internet Explorer efforts at leveraging the W3C.
- Features will clearly target specific customer segments and will offer clear advantages for the administration and cost of ownership mission. All features changing existing functionality will solve known user-problems.

Product Tenets

- Internet Explorer will target Windows 95, Windows NT 4.0, Windows 3.1, Macintosh, and Unix.

- Internet Explorer recognizes it is not the only browser and messaging client and will co-exist in a mixed environment with IE 3.0 and Navigator 3 0 or higher.
- IE 5 will perform better or the same as IE4 on the 16MB/Pentium scenario on Windows 95 (NT 4.0 is 32MB). IE 5 will be benchmarked in real-world scenarios that include Outlook and/or an Office 97 application. IE 4 will continue to benchmark on 8MB/486 machines in a controlled environment to maintain the absolute performance levels of the core applications. The 16MB scenario will be used to define the ship/noship criteria. The goals of performance measurements are to benchmark our own engineering as well as to ensure a great customer experience.
- Existing solutions for IE 4 and Navigator 4.0 will be migrated 100% without modification. IE 3 and Nav 3 solutions will be migrated the same or better as they did IE 4.

IE 5 Feature Teams

Based on vision above, need to come up with focused missions for each feature team on IE.

Browser Team

Own the most popular client platform for developing and deploying HTML applications.

Shell Team

To deliver the system services that make Windows simpler and easier to use and manage.

Outlook Team

To deliver the most popular client platform for messaging and communication.

Microsoft's Goal

Microsoft's top-level goal is to continue to make Windows the most popular operating system for computing devices.

The Threat

While we have a great business in Windows today, the threat to our platform is clear. Windows market share is but an indicator of the great work we've done. The critical issue is one of replacement or cloning – the threat to Windows is that some company can introduce an alternative operating system or device that can replace a Windows PC for customers.

Observations

- Top Goal is to Gain Share. While our focus is on platform, in order to succeed we must achieve critical market share for In order to deliver the platform we must build a world class browser that beats Netscape across the board. This is cost of entry in this marketplace.
- Reach is Critical. Even if we gain 70% market share, customers who target our platform will demand that their customers can fully reach other
- Drive Windows Share. We must remember that our focus is to drive and enhance Windows share. Note that we cannot fabricate a Windows advantage, this path is disaster. If it is possible and reasonable to deliver a feature cross platform we should do so. However, we should always look for ways to make the Windows experience better, by driving features into the operating system or by taking advantage of OS specific enhancements or by conditionally coding for the OS.
- Cross Platform. We need to optimize for Win32, and at the same time delivery information viewing clients on other platforms. Our cross platform work should focus on driving adoption of our standards for Weblications. A good metric here is that we should deliver the right cross platform feature set combined with IIS so it is easy to target the full IE feature set. This means a full port of the Trident engine, scripting, as well as any offline support.
- Windows Support. We will spend 25% of our development time optimizing for the next version of Windows. We are building a Windows component, we should act like one.

Critical Points

HTML/Scripting/Components and Java is the platform for the future. We must be completely focused on extending this platform so it can be used for everything on the system.

However, this does not mean we should clone Netscape and Sun. By cloning Netscape and Sun (via IE and our Java implementation), we are really helping them in their efforts to create an alternate platform. There is definitely a minimum bar we need to hit in order to drive our own share. However, we must strive to make this bar as low as possible. Every time we clone Netscape's HTML we raise the lowest common denominator across browsers and ensure continued evolution. Similarly, every time we clone Sun's Java, we again raise the lowest common denominator across our product and Netscape's/Sun's.

We now have 30% share of the browser market. After IE 4, we will hopefully achieve 50% share. We need to get on a clear extend strategy, where we make the Microsoft platform the superior platform for running all classes of applications, and in particular deliver a superior and feature rich solution on Windows.

We need to stop following and start leading. We need to become passionate about turning HTML/Scripting/Java into Windows features. The next version of Windows must use HTML as the presentation vehicle for dialogs, Wizards, and other page based interface elements. We need to drive the object model for

At the same time, we cannot try to create advantage where none exists. We must focus on understanding what customers are doing, and solve the specific customer concerns and problems.