From: Yuval Neeman
Sent: Monday, October 05, 1998 10:10 PM
To: Gregory Leake
Subject: FW: Outline for Vision Doc

Greg, I need help with this, do you have the materials we used in the planning offsite ?

-----Original Message-----
From: Brian Arbogast
Sent: Monday, October 05, 1998 7:28 PM
To: Paul Gross' Direct Reports; Visual Studio PUMs
Subject: FW: Outline for Vision Doc

this is one topic we didn't get to today, so here's a reminder to those of us that are listed below that we have a draft due in two days!

-----Original Message-----
From: Paul Gross
Sent: Thursday, September 17, 1998 7:43 PM
To: Paul Gross' Direct Reports; Visual Studio PUMs
Subject: FW: Outline for Vision Doc

Here is another crack at the outline with owners. See the forwarded message for other hints and the Office 9 vision doc.

Visual Studio
Vision.doc

Vision, Goals/Themes (**pgross**)
Customer Value Proposition (**brian**)
Competition (**yuval**)
Product, Priorities and Timing (**janet**)
Key Scenarios (**aaron**)
Metrics (**tom**)
Tenets (**craig**)

The document is targeted at the tools division but will likely be read by many folks outside the division. The goal is to have it inform many decisions and have interns be able to understand what is important about VS7. I think this should be 5-7 pages in length. I expect that we will publish our schedule by October 1 but I will send this out by October 15th.

Please write your section and send it to pgdir and vspums **by October 7** (when I'm back on email). Don't hesitate to enlist people in getting it written.

Paul
-----Original Message-----
From: Paul Gross
Sent: Tuesday, September 01, 1998 6:08 PM
To: Paul Gross' Direct Reports; Visual Studio PUMs
Subject: Outline for Vision Doc

Here is the vision doc outline

Vision, Goals/Themes -- pgross
Customer Value Proposition (sucks up some of the current vision doc info but describes benefits to each segment).
Competition
Product, Priorities and Timing (milestones and flesh on how the goals/themes will be accomplished)
Key Scenarios
Metrics -- TomB
Tenets (Guiding principles -- process, feature and product)

I need people to sign up for sections by the end of the week.

Paul

Plaintiff's Exhibit
6477_A
Comes v. Microsoft

The Vision for
Microsoft Offic...

Visual Studio Vision

Visual Studio will provide an integrated suite for all levels of professional developers enabling teams to successfully design, develop and deploy highly scalable and manageable corporate applications that work best with NTS, BackOffice servers, and Microsoft clients (including IE, Windows including CE, and Office) but work well with other clients and servers. The center of gravity for these solutions will be a combination of web protocols and distributed components on the server bringing customers the best of the web and the best of client server (Windows DNA). **We must succeed in simplifying how developers design, develop and deploy distributed applications for NT much in the same way that Ruby and MFC simplified the Windows API.** Convergence on a shared programming model and framework for the server is a must – the space is too large and new for us not to put all of our wood behind one arrow. Operations on data on the middle tier become the fundamental activity of the developer building a distributed application. We must give customers a starting point for a working application and not just deliver the pieces leaving it to them to figure out. All other work on Visual Studio directly or indirectly falls out of attempting to address enterprise application requirements.

VS will allow customers to integrate their applications with both business logic and data (structured and unstructured) residing on other platforms including interfacing with 3rd party enterprise applications. By leveraging VS as an extensible platform and aggressively partnering, 3rd party vendors will provide application customization through support of VS and VBA. Their use of our tools platform will be a stepping stone to making NT the fulcrum for enterprise application integration. Furthermore, a single tools platform will ensure that a complete complement of 3rd party tools augments our offering to address customers needs.

Using the web as a foundation for projects, Visual Studio applications will be able to tightly integrate with other MS applications such as FrontPage and Office. For those projects (and their components) to be versioned on a team server, we provide essential services that teams that leverage our platform the most need to be successful in building applications. And if we can store with those projects the application meta data – making it accessible at design time and runtime – VS will enable developers to build, deploy and manage complex applications with a fraction of the knowledge they need today.

<u>Visual Studio 7 Goals</u>

- Simplify the design, development and deployment of distributed applications on Windows NT.
  - Shared, web based multi-tier architecture and framework
  - Architectural guidance in the tools and template applications
  - Convergence on a new forms architectures based on DHTML
  - Effortless installation of applications
- Integrate tools, teams, partners and process.
  - Integrate the tools into a suite to reduce arbitrary differences in using two tools together
  - Support for end to end application development and deployment
  - Enable teams to work productively together
  - Extensible platform integrated with or used by key enterprise partners
- Scale to the enterprise – Built in scalability, reliability, security, interoperability and manageability.
- Provide value for our bread and butter VB and VC Windows developers.

<u>Metrics</u>

- Number of desktops and servers entrenched by US developers
  - Adoption of MS API's (directory, COM+, SQL, Exchange) technologies
- Gartner Group puts us in the top right corner of Enterprise Application Development tools.
- Measure customer success/failure rate of building and deploying scalable apps
- Early data on success of tools before we ship:
  - Review of specs
  - Team of partners/customers who deploy in 30 days of RTM (RDP?)

- Measurable improvement in developer satisfaction

Where is the VS dial setting? A Quartet!

The trouble people have with a dial analogy is that it is a zero sum game. The reality of VS is very different. We aren't reducing our investment in VB and VC to fund VS. We should shift to a musical analogy. When a piano plays, it carries responsibility for the melody and the bass. When the pianist plays in a trio, s/he has more freedom. The bass frees the piano to add greater richness to the sound. Each instrument that is added provides more freedom to the rest and, if well managed, aggregates to a more pleasant, richer sound for the listener. So when thinking of the next release of Visual Studio a quartet is a better "setting". Visual Studio 8 is probably a symphony! Or if you were at the mixing board, you might slide the slider for VS up, but you certainly wouldn't need to slide down the sliders for VB and VC!

Microsoft Office9

Microsoft Confidential

They come, sometimes, without our knowing what brings them. There is always a driving technology, a desirable benefit, or even wishful thinking. Paradigm shifts are as central to our industry as is Moore's Law. Office was a paradigm shift. Beginning with Office 4 and culminating with Office 97, our product profoundly altered the way people use their Windows PCs to create, analyze, present, and store information. As the early successes of Office 97 become apparent, one must take stock of the profound change, a paradigm shift, that is underway. The *Vision for Office9* provides the shared guiding principles behind the next release of Microsoft's most important product.

The Vision for Office9

Vision: Office9 is the best execution of an integrated suite of Internet-centric communication and productivity tools for creating, editing, sharing, synthesizing, and analyzing business information.

Fundamentally, Office9 is a **compelling** and **exciting** product that generates demand when demonstrated to individuals and influential end-users. Office9 is a **must-have** and **no-brainer upgrade** for Office 97 customers.

Office9 demonstrates industry leadership, provides a clear path for long-term customer value, and improves end-user productivity by focusing on the following product areas:

- Migration, Administration, Deployment, and Management
- HTML Document Creation
- Outlook and Outlook+Application Integration
- Web Collaboration and Solutions
- Web-Based Corporate Reporting
- Personal Productivity

How does one use this document? The goal of this document is to provide a play book or battle plan, but not a recipe or feature list for Office9. As you begin the selection of features, design, engineering and ultimately the release of the product it is critical to the success of Office9 that each of you maintain spirit of this vision. The decisions, tradeoffs, and choices you make in building Office9 should relate to the principles outlined below. Like a football game or a battle, there are many situations and circumstances this vision did not foresee, but this vision serves as a guide for what is important and should help each of you to make the *right* decisions. Office9 will be an expression of this vision that builds on the skills, creativity, and foresight of everyone in Desktop Applications.

Customer Value Proposition

With the original Office product Microsoft established the long-term value proposition of an *integrated suite of productivity applications*. The first releases of Office were mere skeletons of this proposition and it was only with Office 97 that Microsoft truly delivered an integrated suite. At the same time, Office has begun to establish a new long-term value proposition based upon communication and the Internet.

> **Microsoft Office is the most compelling, integrated suite of Internet-centric communication and productivity tools for creating, editing, sharing, synthesizing, and analyzing business information.**

We have a challenge in designing Office9 in that for the vast majority of PC users, Office 97 is still well ahead of the average user's need for Internet functionality. Office9 will be a leading-edge product in terms of focus on Internet scenarios. Office9 is a technology-centric release based on leveraging the standards in infrastructure. It is the case that decisions will be made that take into account where technology trends are taking the industry rather than specifically focusing on today's customer feedback or expectations. Office9 is a leadership product for the industry and customers.

Our objective is to provide a timely release of Office that will migrate existing Office 97 users to new levels of communication and Internet functionality. Office9 will lead the industry in integrating document creation, electronic mail, web servers, and provides functionality that makes even the most Internet-savvy person take note.

In the past our primary Office customer has been end-users (including influentials) and to some degree the solutions developer. Office9 will broaden our focus to include two new customer segments: Administrators and CIO/Influentials. We will necessarily focus less on pure end-user functionality, though this does not change one bit our *to-the-core* dedication to ease of use for all of the work in Office9. We will make decisions that favor the Administrator or Influential over the individual end-user.

Customer	Value Proposition
End-user/Individual	Office9 is the best, easiest to use, and most integrated suite of tools for the creation and analysis of common business information. Office9 is exciting and fun to use at work and people will both want to and need to upgrade. Office9 is as fast or faster than Office 97.
Influential End-User	Office9 is a must-have upgrade that allows the guru/influential user to do new things that impress others in the office. Office9 provides leading edge integration of email and the web that allows the power user to get more impressive work done sooner.
Administrators	Office9 supports customizable and centralized distribution, roaming user-profiles, and self-diagnosing and repairing applications. Office9 is a *no-brainer* upgrade from Office 97 because we do not change the binary file formats and migrate all custom applications and user-settings.
Solutions Builder	Office9 provides the richest set of programmable objects for use in Intranet and Internet custom solutions built on the combination of standard web browsers, servers, and scripting.

Influentials and CIOs	Office9 integrates with and leverages key Microsoft initiatives including Zero Administration Windows, Internet Explorer, Windows NT Server, and BackOffice. At the same time, Office9 embraces open Internet standards and builds unique value on top of those standards. Office9 is not a dead-end or end of an era, but the natural transition from one paradigm to the next. Office9 does not drive hardware purchases and performs as well or better than Office 97 in real-world benchmarking.

Table 1. Office9 Value Propositions by targeted customer segment

The value proposition establishes a long term and multi-release commitment with our customers. Office will continue to lead and to innovate along the vision of producing a communication-centric suite of productivity tools. Office9 is a big first step and with it comes a history of understanding how these transitions take place.

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 competitors (Word Perfect, Lotus, Software Publishing, Borland, and recently Corel) have been left in the dust by Office 97. 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. Thus our competitive focus for Office9 will be on a new breed of communication and collaboration products that do not compete with Office head-on, but offer customers an alternative desktop computing 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 Office and Windows. For Office9 by radically simplifying these aspects of Office and 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 suite, taking advantage of the lessons learned from the Office 4. More broadly, this competitor symbolizes the pre-eminence of electronic mail in the new paradigm. We must deliver on a broad set of features that address the communications scenarios of Communicator by leveraging Outlook as the hub of a communication-centered desktop. We will also integrate seamlessly with Internet Explorer.

HTML editing tools such as NetObjects, FrontPage, Netscape Composer, and a whole range of tools that will continue to be introduced over the year all provide users that have the need or desire to create web documents with alternatives to Office. The richness and *webness* of the documents these tools are able to create sets the absolute minimum bar for Office9 and we must be better across the board at making it easy to produce great looking HTML documents. **Electronic mail editors** will also support HTML and thus pose another threat to Office. The integration of Office documents and Outlook provides unique opportunities to thwart this competition.

Lotus/IBM Notes and **Oracle** (and to some degree **Sun**) will continue to attempt to marginalize Office in favor of pushing a server-centric (and often user-hostile) strategy. We can expect them to continue to portray Office 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 Office scenarios by leveraging the support of web servers running Office9 code as well as tight integration with Windows NT and BackOffice.

Corel Perfect Office is a traditional competitor in every sense and presents us mostly with a marketing challenge rather than a product challenge. On the other hand **Corel Office for Java** represents a significant challenge of Microsoft's leadership as well as a legitimate product threat if the NC gains market share.

Finally, our own **Microsoft Office 4.x** and **Office 95**, while certainly a great asset, are also our strongest competitors. We will likely upgrade a significant portion of our installed base to Office 97, but at the same time we must raise the benefits and lower the costs associated with any upgrade to these products.

Product, Priorities, and Timing

Office9 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 Outlook, Word, Excel, PowerPoint, Access, with major contributions from User-Assistance, Data Access, and Programmability.

The Office9 product development process will be more centralized and will reflect the recent organizational changes in terms of an emphasis on smaller self-deterministic teams. There are some specific areas that are clear areas for improvement based on the Office 97 post-mortem including a shared specification process, more unified scheduling, a central build process, and a stronger central role for release and configuration management. Fundamentally, the process of building Office9 will be improved over Office 97 by maintaining a more stable product throughout the process (including self-hosting), managing the overall bug counts, and improving our test planning and execution. Office9 will adhere to a much stricter and more realistic definition of Code Complete by locking down the product before the end of the development milestone, and not after.

Over the next weeks, the Directors of Development and Testing will deliver the guidelines for the specific changes we are making to the product development process.

It goes without saying that given the competitive situation the ideal would be to release Office9 sooner rather than later. There is a high risk of having the right product at the wrong time should we fail to deliver at the promised time. As it stands we have an aggressive schedule that will give us the time to do the right level of communications support; while at the same time the schedule provides as timely a release as we can make. There is little margin for error and this schedule is dependent on improving the process we used in Office 97.

Date	Milestone
3/3/97	Visions Completed
5/1/97	MM1 Coding Starts
8/1/97	MM2 Coding Starts
11/1/97	MM3 Coding Starts

2/15/98	Code Complete!
3/1/98	Beta I
4/1/98	ZBR I
5/1/98	Beta II
6/1/98	ZBR II, Marketing Beta
7/1/98	RTM USA
8/15/98	RTM Japan

Table 2. Major milestones for the Office9 schedule.

Office9 will be between Office 95 and Office 97 in terms of schedule, investment, and focus. Office 95 was a small effort almost entirely focused on Win32 while Office 97 was a huge effort with, as described in the post-mortem, a lack of shared vision and product focus. Where Office 97 was about sharing and application innovation, Office9 is about efficiency/simplicity and consistent communication-centered innovation. To this end, Office9 will focus on Total Administration, HTML, Web Collaboration, Personal Productivity, and Data Access and Reporting. 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.

TAO of Office9

The *Totally Administerable Office* 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. Much like in years past where development became increasingly aware of the impact their decisions could have on the ability to localize Office, Office9 calls for us to make that same dedication to understanding the needs of our corporate customers.

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

- **Install and deploy**: Office9 will make 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 Office9 will be *advertised* from the running applications, even if they have not been installed, and accessed via *just-in-time* installation.
- **Customize**: Office9 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**: Office9 will smoothly upgrade from previous Office 95/97 release, which includes migrating the old settings and user's work to the new installation and management metaphor. Key to this work is recognizing that Office9 will be run in a mixed environment with other versions of Office.
- **Unify preferences and settings**: Office9 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**: Office9 will implement features to enable documents and customizations to roam with the user between multiple machines. Office9 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**: Office9 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**: Office9 supports central administration. 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 in Office9

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 and Outlook+Application Integration

Office9 fully recognizes that electronic mail will become the center of the user's focus in the next few years. We are already seeing erosion of Word's *facetime* in front of users as email becomes paramount in importance. 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:

- **WordMail**: It is imperative to the long-term success of Word, not to mention a significant user benefit, that Office9 deliver on the promises of using Word to compose email. For Office9, Word will remove the barriers to usage of WordMail by improving performance and stability, while at the same time adding compelling features that will enhance mail creation. Word will also seek to establish WordMail as the premium editor for use in other popular email clients.
- **Office Documents as Mail**: Each Office9 document creation tool (Word, Excel, PowerPoint) will be able to create native HTML documents and incorporate page scripting. A compelling use of Office would be as a forms creation tool for corporate scenarios, such as expense reports. By enabling a shared mail header available from any Office9 document, Office9 users will be able to create a mail message from inside any document. This reverses the norm of creating a brief mail message with an attached document and allows the more natural metaphor of creating a document with a brief attached note. This builds on the popularity of the Office File Send command, something that we see significant usage of in corporate settings.
- **Web Collaboration Integration**: The Office9 Server functionality (see below) will create server-based data that will be viewable by Outlook (or any threaded conversation/data viewer). The threaded information will be relevant to a particular document/set of documents and include comments and annotations, both structured and unstructured. There will be times when it is best to view this data in a structured viewer capable of defining grid or table views, and Outlook integration will address this need.
- **Mail/Office Document User Model**: For corporate users that are entirely mail-centric, Office9 will allow users to create, save, and manage their Office documents entirely as though they were mail messages. Office9 will incorporate the user model for managing mail documents with the user-model for traditional documents. This will remove the separate notions of File Save from the applications and allow the simple metaphor of mail folders to be used for document storage. For example, instead of saving a document to a folder a user will be able to just hit the Escape key and acknowledge saving changes, which will place the rich Office9 document in what looks just like another mail folder to the user.
- **Enhanced Personal Integration**: The Journal feature in Outlook has proven quite popular with the press and influentials (though is currently underused in real world experience). There are numerous places where Office9 can build on this infrastructure in terms of providing a personal history of the work being done by an individual.

Web Collaboration and Solutions in Office9

Office9 will make dramatic improvements in the use of servers and networking infrastructure. In this new paradigm, servers become an integral part of the everyday user's experience with Office. This new server functionality goes well beyond today's treatment of servers as merely giant hard drives. Office9 will provide corporations and workgroups with a rich collaboration and information-sharing environment based on web technologies. The collaboration support will be enriched by full support for HTML-based scripting in all of the Office9 applications, which will allow solutions developers to build solutions starting with Office9's easy-to-use creation tools.

Office9 will introduce a new user model for document storage and publication, build a new collaboration infrastructure, and provide easy-to-use document management and support for basic workflow. A key aspect of the server support in Office9 is the leverage of other Microsoft products and strategies. Office9 will leverage the FrontPage server extensions to provide a nearly universal level of server support for the base scenarios of getting documents to and from the web server. Extending this functionality, the Office9 server support will leverage the Windows NT Server and Internet Information Server products to provide a rich end-to-end experience that is also very easy to install and administer. The server functionality will be a key focal point for Office9. We believe the Office9 Server has the potential to be received as a *new product* in the Office9 box, much as Outlook was received.

The functionality goals for the Office9's efforts in web collaboration include:

- **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.

Web-Based Corporate Reporting in Office9

Increasingly corporations are using Office as the preferred front end for their corporate reporting systems - when the data gets to the desktop, it is delivered within Excel, often within pivot tables, and within Access, often as banded reports. With the data in Excel, users can customize reports, produce charts, do analyses, etc, all without requiring effort on the part of MIS. With the data in Access the users can create banded reports, queries, filter, sort, etc., again without requiring MIS work. This allows IT/MIS to focus on making the backend data available, rather than the specifics of all the reports desired by users. Users like this because of the flexibility it gives them and IT/MIS likes this because it frees them up to do what's most important for them.

As corporations move to intranets for delivery of their corporate reports, we need to make sure that Excel 9 and Access 9 provide the necessary support. We will not only be a good web citizen, but will be particularly strong in data access capabilities used in the web environment. Further, we will beef up general reporting capabilities in Excel to make it more attractive as a reporting tool.

The plan is to allow people to *design to Office*, but run without modification on plain browsers. So corporate developers can set up reports in Office and have the reports viewable both in Office and in plain browsers.

Succeeding here will help keep Office entrenched in strategic corporate applications. Office's reporting and analytic power will provide strong incentive for IT/MIS to continue to design around Office, rather than smaller, better-behaved components. The plan is to keep Office ahead of this curve.

The functionality goals for the Office9's efforts in Data Access and Reporting include:

- **Access Data Pages**: Data Pages are Access forms and reports built specifically for the web. Users can create forms and reports that leverage web technologies and are rendered with WYSIWYG fidelity in the browser.
- **Excel Components ("WebXL")**: WebXL is a set of reusable spreadsheet components designed for use on web pages. These include display, recalc, cell table, data binding, HTML load/save, and an architecture for extensions such as pivot tables. These

components provide services in a variety of contexts. In Web-based corporate reports WebXL provides interactive spreadsheets and PivotTables. In Access DataPages WebXL provides recalc and a grid for displaying relational data sources. In Internet Explorer, WebXL can provide recalc behind HTML tables and controls. On the server, WebXL will provide high-speed recalc and the ability to generate HTML from server-calculated data.

- **Next Generation Data Binding**: Excel, Access and Microsoft Query will consume and expose OLEDB to deliver database power with better performance against these stores. Access' new Data Pages and improved Lists will be designed from the ground up with SQL server as a primary data store. Excel will leverage *Tensor* for online analytical processing. Excel will also include new formulas that can directly access stored data.
- **Excel Pivot Table Enhancements**: Client-Server Pivot Tables will greatly increase the capacity of pivot tables for analyzing large corporate data sources available from both data warehouses and traditional corporate databases. Hierarchical dimension support, using Tensor, will provide richer display and manipulation of the data being analyzed. Pivot Charts will provide interactive analysis through charts.
- **Access Internet Replication (Improved)**: Replication from a local machine to a server via Data Pages enables a whole range of new scenarios. Access will also support direct replication between SQL Server and Access.

Personal Productivity in Office9

The most important aspect of the efforts in productivity for Office9 is in providing cool suite-wide features that are relevant to communication scenarios and compelling to IEUs, while at the same time making these accessible to individual end-users. Office9's overall attractiveness to end-users depends on suite-wide advances in personal productivity.

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:

- **File New, File Open, File Save, Universal Insert**: The most used and most criticized dialogs in Office will be made relevant, improved, and unified with the web and with the collaboration and server-based user model. This poses a particularly difficult migration challenge as users have ver high expectations of both functionality and performance for the existing scenarios, while we want to move them forward to the new scenarios without adding yet another namespace browsing and management tool. There is ample opportunity to improve this area with personalization and intelligence.

- **Collect and Edit**: Using the web for reading information is very easy. Using our applications to create documents is also easy. However, when you try to combine the two users are left wanting more. Office9 will support a new paradigm of individual document creation called *collect and edit*, which will allow one to seamlessly browse web content and incorporate that into documents as they are created. Formatting, and source location will be maintained and the new document will look as though the user created it from scratch, with appropriate attribution.
- **Structural Editing/Event Monitor**: Office applications can be made much easier to use by leveraging the structure inherent in documents. Structure can be determined by analyzing the user's document and suggesting changes (such as formatting or chart types). Also by tracking the user actions as they take place and suggesting functionality or automating the most likely task (such as creating a hyperlink automatically) we can improve the richness of the document without forcing the user to *learn* more of the application. Office9 will provide a shared event monitor, based on the Word 97 implementation, to be used by all the applications. We will invest in a large rule base for each application to provide a consistent and rich set of automated tasks in Office9. Both Excel and Access will build significantly in the area of structured editing in the area of automated list maintenance. In addition, Excel and Access will develop the notion of structured *parts* that provide pre-built pieces for incorporation into documents. Broad areas for improvement include AutoCorrect and general detection of composite operations (AutoAuto). The features in this area will enable Office9 to deliver on more personalized editing.
- **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**: The key guidelines for the core interface improvements in Office9 will be to unify disparate elements of the products and to connect the interface elements together and to Internet resources. Core interface work will focus on finding ways to reduce the perception of bloatware that comes from an overwhelming number of interface elements seen by typical users. Office9 will build on the investments in consistency and shared code (Command Bars) to provide a new level of improved interface. The primary innovation will be in the area of connecting user-interface elements together via a *WebUI*. Office9 will also connect elements of the interface to Internet resources, for example alerts will be actionable and allow administrators (by default these URLs will point to Microsoft) to point alerts at specific URLs for content or assistance. A key aspect of core interface will be the unification with the new metaphors introduced by the shell update portion of Internet Explorer 4.0. The Assistant (Social Interface) is also included in the core interface work. Office9 will improve the Office Assistant itself (better screen management, more exciting customization for solutions, etc.) Additionally, the Assistant's role in providing a front-end for the user to assistance and discovery of new functionality will be enhanced as it is integrated with the work in event monitoring and personalization. The social interface will be made accessible to users that require Accessibility services from Windows.
- **Virus Protection**: For the existing Office 97 file formats Office 9 will add support for enhanced virus protection and digital signatures. In addition, the general issues of document security will be scrutinized and Office9 will proactively provide better security for documents and solutions.
- **Worldwide Support**: Recognizing the fact that Office applications are used increasingly to create multi-lingual documents in a multi-lingual environment, Office9 will improve international proofing tools and provide better overall support for international writing. Office9 will have single worldwide executables and allow installable language support and switchable language user-interface. Office9 will also add specific Far East editing features for Word.

- **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.

Key Scenarios

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

Tenets

Office9 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 Office9 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 Office9. The feature tenets cover the design of the product. The overall product tenets represent the global product issues.

Process Tenets

- All members of the Office9 team, regardless of the reporting structure, are responsible for the innovations in the Office9 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 Office9 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 Office9 to be successful and focused it is important to follow the spirit of the vision. Office9 will not unilaterally advance a single application when it is clear that the advance is applicable across applications.
- 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, Office9 will not make gratuitous changes to the user-interface and the Office 97 binary file format (Access MDB excluded) is the binary file format for Office9 (or no additional code is required on Office 97 machines for 100% save/load interoperability with Office 97).
- Optimizing for the web is critical to Office9, while at the same time it is necessary to upgrade Office 97 users. Changes to the existing functionality and interface of Office 97 to enable new web scenarios need to be made with attention to the upgrade scenario--these types of changes need to be twice as good, not just 10% better.
- Office9 applications will ship with the same default mode of HTML or binary and this will be decided as development of Office9 progresses. The goal is for HTML to be a reasonable and useful default for whole corporations. When saving as HTML, Office9 has only a single

way to do this without any wizards or options to complicate the use of this format.

- Office9 will embrace before it extends. Fully embracing the Internet standards and experience is core to the Office9 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

- Office9 will target Windows 95 and Windows NT Workstation 4.0 (with the requirement of particular service packs to be determined).
- Office9 will support the user's preference for browsing. Office9 documents will look great in Navigator 3.0, and even greater in Internet Explorer 4.0. Office9 will support using any web server (supported by FrontPage) for basic collaboration support, with Windows NT Server/IIS 3.0 representing the target platform for innovation on collaboration and workgroup.
- Office9 recognizes it is not the only suite and will co-exist in a mixed environment with Office 4, Office 95, and Office 97.
- Office9 will perform better or the same as Office 97 on the 16MB/Pentium scenario on Windows 95 (NT 4.0 is 32MB). Office9 will be benchmarked in real-world scenarios that include Outlook and/or a browser. Office9 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 Office 97 will be migrated 100% without modification. Office 95 and Office 4 solutions will be migrated the same or better as they did Office 97.