

# DRAFT: Office10 End-User Productivity Team Vision Statement

April 25, 1999

## Program Managers and Areas of Ownership:

GPM: DHach

Core UI: GlennF

MSN Integration / Services on the Web: DavidSwi, JKatzman

Workpane: Zeke

## Vision

Make Office the personal productivity tool of choice for knowledge workers by offering the most exciting, innovative, and easiest experience of getting work done on your computer.

The EUP team targets high-frequency productivity scenarios that are relevant to most users across the suite and work out of the box (no IT support assumed). (Other teams target, in addition, industry specific scenarios (Word and lawyers, Excel and financial sector, etc.) and scenarios that might assume more infrastructure (e.g. Outlook and Exchange).

The EUP team delivers finished features in order to have tremendous, positive end user impact. It also provides some infrastructure for other teams to deliver their EU excitement. For example, enabling the Server and Web Documents teams to deliver great integration through an HTML-pane experience (as we've had prototyped) is a prime candidate. Integration with services on the web (e.g. publishing, property setting) is a unique benefit that we can deliver to users through this mechanism.

We measure this team's success by its ability to deliver

- (1) End user benefit and inspiration for end users to want Office 10
- (2) Infrastructure and standards that enable other teams to maximize suite-wide end user benefit
- (3) A unified end-user productivity story across the suite by driving shared priorities to guide decisions around basic use scenarios and improvements. (Note that every team owns its own destiny within this framework.) The EUP team will lead by example in delivering this story.

EU Excitement has three parts:

- **Innovative.** We don't want "new because we can," but new that also meets the criteria below.
- **Broad appeal / useful.** Think of this in terms of how many users a feature affects or user reach. "High IV" is a good first cut here. Typing, the clipboard, and FileOpen are good examples of this across Office. A feature doesn't have to appear in all the applications in the box to satisfy this. Most applications, in particular the right applications for the feature, is the standard. (This will be a judgment call and the priorities in (3) above should help here.)
- **Finished.** To deliver excitement to end users we need to deliver finished goods that are useful out of the box, not just infrastructure or an existence theorem. One example might be collect and paste. It's good enough for a demo, but in real use is a frustration. FileOpen is finished and useful out of the box.
- **Aesthetics.** These features have emotional as well as intellectual appeal and require a good level of fit and finish.

It's the kind of thing that people should go "Wow!" to without having to hear a long explanation. Getting all of these aspects right is important to success. Example: Solving the alert problem is useful, but not necessarily innovative. Speech is highly innovative but not necessarily broad appeal.

## Quick context for this vision

- **Customers.** We are totally totally totally focused on Influential End Users and End Users. These are the people who actually use Office as an application, in contrast to the people who roll Office out, decide on whether to purchase Office, develop on top of Office, or use solutions developed on top of Office. A theme throughout this vision is that users are upgraders. They know something already and have a base ability to get a set of things done. Key to Office 10's value is allowing them to use what they know and quickly figure out what they need to accomplish tasks outside of their existing skills.
- **Competition.** The real ones are Office 97 and Office 2000 and the workarounds that users have already developed for themselves in using them. Another category is "web productivity services" (<http://officeweb/users/stevesi/voffice/default.htm>). There is a separate issue of mindshare: what is the "cool" thing that people are talking about (e.g. ICQ)?

- **Office internal goals.** The EUP is responsible for Consistency 3.0. Office is "the best execution of an integrated suite" - we need to define what consistency and integration mean in this release and our goals and agree up front how we will execute to these goals, in terms of process and outcome. First draft of this document to come Friday April 30 from DavidSwi.
- **Changes from past release.** This team is picking up responsibility for MSN integration / services on the web. We are decreasing focus on Help and Assistance infrastructure. The payback for doing less work on Help/Assistance should be better execution and "finish" on the features we do take on. We definitely want to do fewer things, much better than last time.
- **Non-goals.** Deep infrastructure change, such as unifying the forms architecture under all our UI, is a non-goal. This is a direct fall out of wanting to deliver excitement and benefit. Telling people "32 bit" or "object oriented" is not enough to excite them.
- **Focused innovation.** We need to ask "what merits doing" very seriously because given the set of resources on the project we cannot do everything or take every single opportunity for innovation. We need to choose them carefully and do a better job of delivering fully on the ones we choose. Lots of half-baked features will not fulfill our goals. Unfocused innovation already has a name - bloatware.

## End-user productivity Feature Standard

Office 10 needs to compete for end-users not so much feature for feature but **feature against workaround**. The bar in displacing old versions of Office is not whether a particular feature (e.g. FileOpen) is better than it was in the previous release but if the new design can beat how the user worked around the previous release's feature limitations.

For example, there are the **features** styles and format painter, and then there are **workarounds** (paste text that's formatted the way you want and type through it; manually reformat). There's the feature of search (find contact edit control, etc.) and then there's the workaround (browse + sort + scroll in contacts; open a message + resolve name + double click, etc.). There are collaboration features like web discussions and revision marks, and there are all the mail messages with "Comments in line below >>>". Delivering more e-mail signature options and Rules wizard settings etc. won't necessarily deliver more end user productivity and excitement. On their own, they won't change people's work. If we pretend to innovate, customers can pretend to upgrade.

Note that a real if extreme form of workaround is people just prioritizing tasks away. Not doing the work, for some tasks, is a reasonable alternative. (An extreme example of this is not caring about formatting or spelling of some email messages.) The more we improve on tasks that users are choosing to punt, the less users will care about our software.

The basic feature standard is (from the section **above** on excitement) broad appeal and finish. The basic test is "For this user problem / scenario, what are users doing today **despite** the commands we're giving them, and how do we address those problems?" Do we expect the proposed solution to change the way people work? The challenge for anyone working on end user productivity and excitement in Office 10 is to address these (among other) questions:

1. What is the fundamental user problem you're trying to solve and user goal you're trying to deliver on? (Ex: "<featurename> is hard" is not a user problem. It's our challenge.)
2. Will the changes you're proposing effectively solve the problem? (Ex: guarantee yourself a new job nextrel. Excel's Data/Sort command hasn't changed in a while.)
3. If users didn't use the command before, why are they going to use it now? Will the changes you're proposing change the shape of the IV data? Of support data? If not, how can we tell if the change will contribute to inspiring users to upgrade?
4. Are the changes you're proposing competitive with the user workaround?
5. How core is the scenario around this problem to this release and to Office usage overall?

## Customers

Main focus is end-user/IEU. We deliver infrastructure for other teams to please the other groups.

<b>If I am a(n)...</b>	<b>what can you do for me? (two sentences, please)</b>
End user	Obviously easier to be productive <ul style="list-style-type: none"> <li>• More productive on the web out of the box because MSN services are tightly integrated into the client.</li> </ul>
Influential End-user	<ul style="list-style-type: none"> <li>• New UI is cool and enables user experience that used to be unique to web sites.</li> </ul>
IT	Can deliver richer, better-integrated user experience of their solutions on top of Office than before.

BDM Tighter integration with servers/services offers compelling business value (e.g. ERP/data integration in the Workpane).  
CIO All of the above yell at him or her to get it.

## Functionality Goals

Scenarios below help motivate this section.

### CoreUI

1. **Auto In control.** Mechanism for controlling Auto actions. Address discoverability of the mechanism (reasonably in the user's face (i.e. not just a hidden checkbox) and also reasonably not too obtrusive (not the assistant)). Reduce user busy work. This applies even if the action is not generated from the event monitor code. In Word's vision this is described as "proactive Word." More detail available at <http://office10/teams/UI/Memo/personalization.htm>. We will also pursue some more auto functionality. Some is the key word here; we will do this selectively. Ex: linking documents and Outlook information as in clicking May 1 brings up the calendar.
2. **Alerts.** Goal is to reduce user frustration. We've added and removed text, help buttons, and most recently added URLs to them. Metric is TBD. Much app specific, feature specific work here. Document with details under construction. PSS/IV data for which alerts or classes of alerts to target is one starting point. Goal will most likely include: reduce the user's need to read, understand / remember the text, and figure out what to do to move on. Some alerts are mostly OK (e.g. do you want to save changes?). Classes of problems in alerts so far are: Not actionable; Not helpful/understandable (too cryptic); Not specific enough; Awkward button mapping; and No local context.
3. **Collect and Paste 2.0.** In 2000 this turned out to be great for the demo, not good enough for real usage. Goal for 10 is same as goal for FileOpen in 9: great for real users. Details available at <http://office10/teams/UI/whatsleft/c&issues.htm>
4. **Speech Command and Control.** Leverage work in command bars for this. This is not a huge standalone work item; this work is focused on being reasonable in the context of the primary speech goal: dictation.
5. **Help.** One goal: performance. No deep changes to architecture, bits, etc.
6. **Visual redesign.** Aesthetics and reduce clutter. Goal: clean look with user appeal. Make Office 2000 look as clunky and dated as Office 4 does today.
7. **eBook.** Integration with eBook is under investigation.

### Pane

Infrastructure: the Workpane is a platform, like command bars, for other teams. (Links to demos and more documents below; short version is to think of this as a browser control hosted in a command bar.)

1. **Tier-1 clients (Must).** These scenarios are key to the Office 10 vision and exploit unique aspects of this mechanism (e.g. server integration).
  - a. Publish/document sharing scenario (Web Documents team)
  - b. Other client integration with web services (e.g. MSN integration, Hosted Office server functionality, perhaps data from the web?). (Server Team, Excel data team?)
  - c. Guidelines for what we put here v not put here. (Functionality? Help? Content?)
2. **Tier-2 Clients (Like).** Functionality we'd like to be able to enable but is more vague and is not guaranteed at this time.
  - a. Area for solutions (e.g. expense template) to present UI. Exploits the ability for dynamic presentation (as compared to dialogs (static) or toolbars (not rich)).

Solutions can be more interactive/actionable, self-descriptive, and modeless, as well as consistent in their browser and application presentation. (Programmability team)

- b. Platform for applications to use to produce better solutions to specific, selectively chosen user problems. Goal is to get basic HTML UI right (usable, useful, desirable), and give users an opportunity to react before we change the whole app under them. Plan is the same as when Wizards became available: apps make the best decision they can locally to take on 0-2 features per app at most here, choosing the features that can get the most out of this mechanism. Initial list from the apps is at <http://office10/teams/ui/workpane/the%20apps.htm>. Right now, there's not much in common to these features.

Finished feature: Personalized UI Part II – Rightsizing the UI after the Anti-Bloat effort. The workpane makes it easier for users to locate (and then use) functionality by supplementing the core menu and toolbar experience with richer browse and search mechanisms. The workpane complements the menu/toolbar space as a personalization mechanism; once the user has used functionality through the workpane, that functionality is bubbled up and into the core UI space. Office 2000 anti-bloat functionality decreased UI by removing what you didn't use. This feature increases UI by adding the items that are relevant to your usage pattern.

Success for the workpane is if the user (1) uses this instead of reading all the menus (in their expanded form) and tool tips and (2) the user succeeds at the task at hand.

1. **Discoverability through Search + Browseable results.** Enable a basic search mechanism (AnswerWizard++) over the UI space that saves the user the effort of browsing the full command space (and sifting through all the tab dialogs, gosubs, etc.). No help, no additional words or explanation: just the command.
2. **Optimal re-presentation of results.** After the user has searched once successfully, he doesn't have to use search as the main UI to the functionality. Personalization implies that once the user has done something it will be easier to get back there.
3. **Poster child "optimally-browsed" UI.** We want to do enough here to try the ideas out (rich presentation and description, modeless, web navigation metaphor, etc) and deliver some benefit. The formatting UI is an initial focus for this effort.

### MSN/Services on the Web

1. **Instant extranet.** Web Folders work out of the box. The entry point is obvious, the registration process is brief and the user can successfully find and share the published document afterward. This is compatible with OSE and the upgrade from these services to MSN services to what the Server team enables makes sense.
2. **Online meetings as easy as URLs.** Setting up and joining a meeting on the Internet is as easy as setting up a meeting in Outlook and clicking a URL. We need more specific goals re: interop with Exchange real time collab as well as integration with Office document review and project management process. We take advantage of additional hardware (microphone, camera) but still work in the base case (keyboard only). The workaround we target here is the speakerphone / conference call.
3. **No excuses.** If the browser works, this functionality works.
4. **Per-user data storage backend and interfaces to this for other teams.** This would allow the Fundamentals team to connect DAD Watson to SOW and the Worldwide team to provide server-side document translation, for instance. It could also allow the e-Book team to provide server-side conversion of Office docs to Open e-Book.
5. **Opportunistic application specific integration.** At the very least this team catalogs and makes sense of all the services on the web integration shipping in the box.

## Won'ts

- We do not completely replace or suppress the existing user interface for the applications.
- We do not clone all the existing services out on the web, but rather focus on understanding what our customers actually want in this space.
- We won't integrate every [MSN] service under the sun into Office, but will happily evangelize to those services how they should integrate themselves.
- We won't Deliver the Ultimate Infrastructure or Initial Implementation of Unifying Help and functionality (the scenario of "I have the tool handy; how do I use it?"). Today, users solve this by experimenting with the UI, reading labels, and going to help. Users continue to do this. All teams solve their most important problems by improving the designs of key features.
- We do not attempt to generate wizards on the fly to address user tasks that span multiple dialogs or multiple steps. The space underlying speech, menus, toolbars, search, and help is only as good as it individual feature designers make it.

## Scenarios

### Core UI

We assume that users are upgraders of some kind and already have a basic way to be productive in the applications. It might not be optimal or even the way we intended the applications to be used, but users have a way. We do not attempt to re-educate them.

1. **Users feel in control of the applications.**
  - **Users can control unwanted Auto easily.** Try the following in Word 2000 (you may need to reset Tools/AutoCorrect/Exceptions/Other Corrections). Type "comm" and notice the period is added. Backspace over the period and hit space. Repeat until you want to use another word processor. This is not just event monitor features; Excel date autofmt is another interesting candidate for this (see <http://office10/teams/UI/Memo/personalization.htm>).
  - **Users are less ticked at Alerts.** We know users see them. We need to define the scenario and metric we address here. We know it's not the raw count of alerts in the products (leads to vague alert text) or more text in more alerts (users don't enjoy reading this stuff). We know there's a lot of app specific work to be done here.
2. **Users doing common, high-frequency tasks are better off**
  - **More good Auto happens.** Connecting documents to Outlook information is a good example here. We can recognize dates and, potentially, contacts, and make them linked. Example: user types May 1 and it becomes May 1 with Outlook://date/information under it. Note "unwanted is easy to control" makes this item a more reasonable scenario.
  - **Moving information between documents becomes easier.** Last release fake SDI helped here. Collect and paste doesn't quite nail the scenario. We need to define the scenario better here.
3. **Users who are dictating speech have basic control of their applications through speech as well.** We are still defining scenarios here.
4. **Users react positively to Office visuals.** OK, this is a stretch as a user scenario. Key here is that a visual redesign to make Office 10 distinctive from previous releases does not make it too distinct from the success of previous releases.

### Workpane

This is a particular sub-focus of Core UI with the same assumptions. The basic scenario is the baseline usability test. Do users have a better success rate at tasks aren't well represented on the default toolbars?

1. **Can users locate the tools they need for a task?** There are at least two parts to this: "I know what task I want to accomplish; what tools do I need?" and "I know what tool I need but I don't know exactly where it is." Today users solve these problems by reading the menu items, trying them out, and trying help. We usually demo Help, but believe that users just manually search the space. We think that command search is an interesting solution here, especially if the results of the search are easily browsed.
2. **Do users find this a cool, exciting UI change?** Have we successfully addressed the coexistence and interop with current UI, performance, etc.? Specifically, do we balance well between existing entry points for what users already know (menus/toolbars) and being a well integrated entry point to functionality that users don't know well? (Hard to beat menus/toolbars for stuff the user already knows. In fact, if we mess with the top-level entry points, it's probably a net negative.) Will users notice and react positively?

### MSN/Services on the Web

We assume only that the browser can get to the Internet after setup.

1. **Web folders work out of the box.** The user has a place on the net to save files to that just works out of the box. What minimal setup that is required (e.g. Passport identity?) can happen during the Save As Web page process. The user can exercise the basic functionality (save, open, rename, permissions / notifications) required for basic collaboration against her shared space on the net. Office functionality (e.g. Web components) works great against this default place.
2. **Share more than documents.** Outlook just works against this mail store. Contacts and calendar are sharable.
3. **Real time collaboration.** The user can easily find other users on line and successfully start real time communication with them. This process integrates well with the document editing process in any MSO client. (Make the experience of Office users trying to do real time collab against Internet services simple and consistent.)
4. **Supportability.** The user can resolve a blocking problem (error message, task that Help on the client does not address) without making a phone call by getting to additional services on the web. Key: this assumes that PSS wants to do this and that the apps are working / booting and can connect (cf. Fundamentals team). Appeal here is that the client software helps you find ways to get stuff done on the server (connects to a URL that can redirect appropriately to Help on the web, PSS offerings, ORK, Office Update, etc.) rather than making you seek it out separately. Key for us is to identify the integration points and who is signing up to be on the other side, and what this means in an admin'ed environment.)
5. **Information integration.** While working on a document, the user needs additional information that is online. The process of getting that additional information, as long as that information is available through the browser, into the user's document is easier and better integrated into the document creation/editing cycle. (Ex sample information: contact names, project dates and deadlines (as stored on a team web page), directions to a place, etc.). Note: in addition to overlapping with collect and paste, this is a danger point WRT unified search. There is no clear owner for the value add here; as it stands this would just be re-presenting browser+ search in the context of the editing application.

Notes:

- The default place is just a starting point. Its primary goals are to just work out of the box and to scale. FP and Outlook/Exchange are platforms that exceed the functionality offered here very quickly (but do not offer out of the box benefit in the same way).
- While we have NetMeeting integration in 2000, out of the box it doesn't just work.

## Demo Script

Driven from the scenarios and functionality goals. End goal for demo: a broadly useful set of applications that are attractive, innovative, and consistent (read: across the applications, things are better the same ways). We know we've succeeded when end-users cheer at the demo but admins/IT feel a little scared.

This area is currently under construction.

## Customer Input

- Baseline study (success rate of users just trying to use Office) – <http://office10/teams/ldu/usability/baselinesummary.htm> is a good starting point (mailto:chrisker)
- Reviews re: IntelliSense and being difficult to control are cited in the personalization document (link above, in line)
- Instrumented version
- PSS (esp for alerts)

## Technical Investments

- Workpane: command bar with Trident that allows a rich presentation of UI but doesn't create security, performance, or "new component" problems. Also, mechanism so that this UI and classic Win32 UI look "right" next to each other.
- Server-side code at MSN-scale for services to Office users.

## Project Tenets

- We don't write the same solution three times against each server architecture.
- Before we take on a dependency or feature, we think through the defining scenario and offer contingency plans.
- When we actually cut or scale features we go back to the defining scenario to see if what's left makes sense to the user.
- Users don't like reading (alerts, UI, etc.).
- Users learn by doing, and don't mind and playing with software (they just don't do it for a living).

## Related Documents

- Initial goals when the team was defined – <http://office10/teams/ui/overview.htm>
- What we should investigate for NRO in the UI space document – <http://office10/teams/UI/visions/user%20interface.htm>
- Initial prototypes of the workpane  
Shockwave (old, but relatively complete)  
<http://office10/teams/ldu/design/PrototypeGallery/WebUIOffice10/ShockwaveDemo2.htm>  
The Browser only version of the working prototype  
<http://office10/bin/workpane/FakeWorkpaneDemo.htm>

## Dependencies

See <http://office10/teams/gpm/dependencies.htm#UI>; contingency plan TBD.

Other teams should assume that we are building the workpane and maintaining (until otherwise noted) the shared code we delivered in Office 9. FileOpen ownership will most likely move to WebDocuments (to help them deliver on a great publishing story). Command bars, Help and Assistant, SDM, and SDI seem like no-brainers to stay in this team.

Alert feature area assumes some app specific work to be done by the applications (in the name of fundamentals).

Other teams will make their own investments in application specific end-user excitement and personal productivity features. This team will be responsible for making the collective set of features sensible when taken together as a whole.

#### **MMO possibilities**

Working with dev on this list. Main items right now are bug/perf clean up from last rel and some fit and finish items (e.g. end user customization of places bar in FileOpen).

#### **Competitors**

Competitors start with Office 97/2000 + user workarounds. The usual suspects follow Corel WordPerfect suite, Smart Suite (Lotus/IBM)). The list of others (e.g. AOL?) and drilldown will follow.

#### **Assumptions**

- We can scope and prioritize Workpane to be valuable without being unpredictable (e.g. random in where it's used).
- We can get the perf and safety we need when driving Office through a Trident front end (e.g. the workpane work to deliver server integration, as in Publish)
- We can write server code (a la Hotmail) that is robust and valuable to end-users.
- We are not affected by any severe changes that might happen to MSN.

#### **Issues**

- Command bars as redist for vstudio. No promises; we just need to figure this out better in this release.
- Updatable content on the client: not clear how doable this is now, if we care about enabling even better or more. Main issue now is help. If we think we'll author features in HTML + script in the workpane, then this is for more than Help. Do we need to worry about rollback? Versioning? Interop?
- Not sure if we've thought through the annuity / Office update integration v these services on the web v MSN enough.

#### **End-user productivity Feature Standard (long version)**

Toward the end of Word 2 and as we started Word 6 planning, Chris Peters, the Word GM at the time, made clear that the product had two clear paths moving forward. One of them involved going "high end" in word processing, toward DTP and multi-author composition packages like Frame. The other was a return to basic use, fundamentals, and real users. The mantra of the second was Power Made Easy (the Excel 3 tagline). The poster child for the evils of the first was color separation. We believed, with Chris, that while this was a great feature for Publisher's category, it should be a source of ridicule for us in the word processing category. I don't think anyone really believed that color separation would set our users free; rather, we were at a turning point when "what to do next" wasn't clear at all despite many alternatives. For example, Word and Excel had teams working on "workgroup features" that, at the time, meant things like master documents and document routing. We focused on what was important to users at that time: typing, formatting, and consistency across Office. At the end of Office 97, we were at a similar turning point, and successfully (we believe at this time) made our focus TCD (LORG customers) and HTML (new web scenarios).

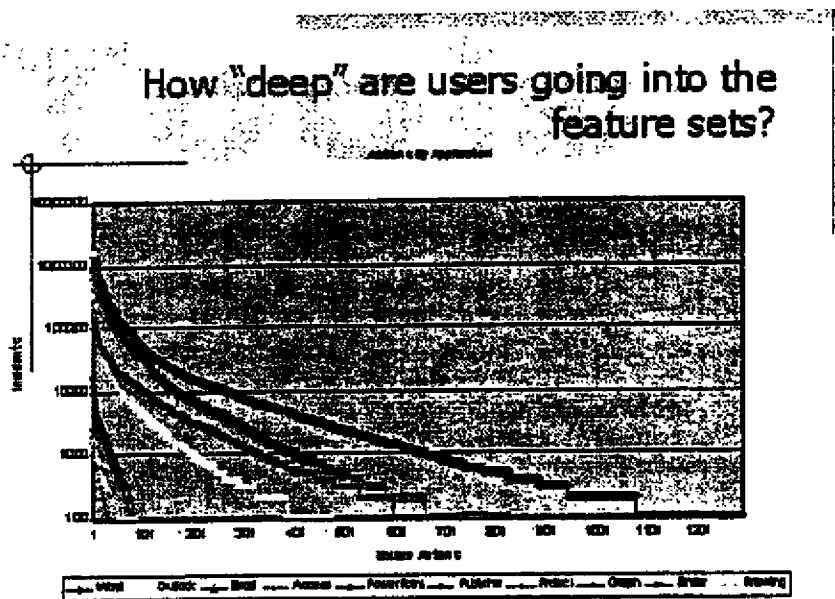


As we start Office 10, we're in a similar position to decide our own future: technology through a bunch of features or product delivering clear customer benefit through technology and features. Don't get me wrong — we'll do features and we'll change some part of the visible UI (and some small set of menu items and toolbar buttons). We just need to decide if the goal of these changes is "more of the same, and even more competitive than before" or "deliver a coherent, MUST-have Ultimate Personal Productivity Tool."

If we want to succeed at the latter and deliver an inspiring upgrade, we need to compete effectively not only with Office 97 and Office 2000, not only with "the promise of Notes" and the Web, but with productivity tool alternatives. In addition to fax, phone, and paper, there's a productivity tool that we should be very scared of: people prioritizing tasks away. Not doing the work, for some tasks, is a reasonable alternative. (An extreme example of this is not caring about formatting or spelling of some email messages.) The more we improve on tasks that users are choosing to punt, the less users will care about our software.

This means we need to compete, not so much feature for feature, but feature against workaround. For example, there's the feature of search (find file, find contact, etc.) and then there's the workaround (sending mail asking someone for a file, browse + sort + scroll; open a message + resolve name + double click, etc.). There are the features styles and format painter, and then there are workarounds (paste text that's formatted the way you want and type through it; manually reformat). There are collaboration features like web discussions and revision marks, and there are all the mail messages with "Comments in line below >>>". Delivering more Rules wizard and Views, more e-mail signature options and clipart, more Pivot table and Chart wizard etc. won't deliver more user productivity and excitement. On their own, they won't change people's work. If we pretend to innovate, customers can pretend to upgrade.

Based on preliminary Office 97 Instrumented Version data, here's one way to visualize the problem. (Kudos to <mailto:churner>. This graph is based on partial data and will be updated when more is available. There's much more that we can learn from the IV; an initial presentation is available at <http://office10/planning/iv/office97iv.htm>.)



The curves describe command usage drop off in the applications, or how much the most used commands of the product are used compared to the rest. For example, the yellow curve (Outlook) falling more steeply than the red curve (Word) means that for all the commands recorded ("Incidents"), the number of distinct commands ("unique actions") that made up those incidents were fewer in Outlook than in Word. Basically, the drop from Delete to Check for new mail (#1 command to #11 command for Outlook) is sharper than the drop from Save to Underline (#1 to #11 command for Word).

Our goal is a better product — not getting users to choose more of our applications' commands. A better question than "Why aren't people using more of our commands and how do we drive command usage up," I think, is "What are users doing despite the commands we're giving them, and how do we address those problems?" If we address those problems, we will offer a valuable and exciting upgrade and win. (Note also that there are tasks not captured by the IV (e.g. data

entry tasks and scrolling/navigation). If we can make those significantly better (possible example: Excel's AutoComplete; counter example: Word's navigation widget by the scroll bar), we can deliver significant user benefit.)

The challenge for anyone working on end user productivity and excitement in Office 10 is to address these (among other) questions:

1. What is the fundamental user problem you're trying to solve and user goal you're trying to deliver on? (Ex: "<featurename> is hard" is not a user problem. It's our challenge.)
2. Will the changes you're proposing effectively solve the problem? (Ex: guarantee yourself a new job nextrel. Excel's Data/Sort command hasn't changed in a while.)
3. How core is the scenario around this problem to this release and to Office usage overall?
4. Especially if there's an external technology, will we be able to make a feature work out of the box, as is? (Ex: Excel, Data/Get External Data/Run Saved Query basically works; Tools/Online Collaboration/Meet now is a harder problem and takes some jiggling of the handle.)
5. Do the changes you're proposing make Microsoft's offerings make more sense collectively? (Ex: this could be the first release in which Mail Merge commands in Word, Outlook, Access, and Publisher really consider each other.)
6. If users didn't use the command before, why are they going to use it now? Will the changes you're proposing change the shape of the IV data? Of support data? If not, how can we tell if the change will contribute to inspiring users to upgrade?
7. Are the changes you're proposing competitive with the user workaround?

**DRAFT**

## **Office10 Web Documents Vision Statement**

4/26/99

### **Key focus areas are:**

- Improve the Save/Publish/Send experience. Make saving to a web for collaboration and saving a document into a workflow process great.
- Make Office the best tool for contributing to a FP/OSE/Platinum/Tahoe web site through tighter integration with web based tools and services.
- Continue to invest in targeted HTML infrastructure improvements (CF\_HTML, smaller/cleaner HTML files, XML support for new scenarios in the apps), and provide customer focused changes to our HTML support

### **Program Managers and Areas of Ownership:**

GPM: Marc Olson

Team (initial focus areas):

Mandira Virmani: Web Server integration

Alan Ramaley: Server based collaboration, HTML file handling

Teresa Fung: Save/Publish/Open improvements

Noah Egorin: CF\_HTML improvements

Noah Edelstein: Metadata Integration

Valerie Serdy: SendTo10 and Email collaboration

### **Charter**

The charter of the Web Documents team is to enable better sharing of documents by knowledge workers through the use of team webs and collaborative email scenarios. This team will continue to invest in our suite-wide use of HTML and remove barriers to adoption of HTML as a file format. We own defining key user scenarios around working with documents intended for sharing with others, either in email or on a web. Web sites should be able to be treated the same as file servers everywhere in the product.

### **Vision**

The vision of the Web Documents team is to continue to deliver on the promise of Web Servers as the place for documents that are to be shared with others. Part of the work is the HTML file format enabling that was done in Office 2000,

**MS/CR 0005751  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**

but there remains more work to be done around exposing the context of the web as part of the document creation process. This includes knowing what other documents are in the same space, understanding the structure of the site and how that relates to where I save things, as well as enabling a smart server to simplify the decisions around how I put a document on a server and make it available to other users. We've given users a UNC view of their web site, but as sites get smarter and have more structure, we should look at exposing that structure differently through the namespace extension so that the user's document go to the right place in the web. These key enhancements build on features made available from the FP/Office Web Server web as well as our new storage in Platinum.

HTML as a file format brings issues and complexity with it that we need to continue to manage for the user to give them as seamless an experience as possible. There remain issues around sharing content between Office documents, enabling other tools to create and edit Office generated HTML, and some unsolved challenges around the complexity of what 'publish' means as it relates to Saving a file. Some of these challenges include the integration of the "peeler" to strip out round-trip information for archival purposes, what it means to save a copy, what options are applicable to which target browser, how does the user get the end URL of the document they just saved, and managing the multiple individual files created by saving HTML documents. In Office9, various HTML options were feature-oriented (e.g. use CSS or not) rather than browser-oriented. We've heard from early feedback that this is difficult to understand and that end users do not have a good grasp of differences in browser functionality.

Sharing documents and information isn't just about web servers. SendTo9 enabled a new way for people to send documents via email, as well as providing the framework for a better HTML mail experience using WordMail. The solution isn't perfect, and there is additional work to make the experience better for Wordmail and for the compound document scenarios that exist for Excel and PowerPoint. In addition, there is the possibility of extending the Send functionality to provide context for scenarios enabled within the Outlook/Platinum collaboration space.

As our corporate email store moves to a web server model in Platinum, we need to understand the key integration points from a document centric perspective and make sure that users can gain the benefits of Platinum when their documents are stored there, enabling exposing metadata for the server as part of the document creation and saving experience. As Outlook works to make documents a key part of the information they understand and manage, provide the right integration model to make customers want to put their documents in the same place that they put their mail and calendaring information. Getting

**MS/CR 0005752**  
**CONFIDENTIAL**  
**HIGHLY**  
**CONFIDENTIAL**

more content into this space enables better scenarios for workgroup solutions, team projects, and an easier way to find information using the underlying search infrastructure and rich views on information available in Outlook.

## Customers

<b>If I am a(n):</b>	<b>what can you do for me? (two sentences, please)</b>
End user	Easier to share documents with others through the use of web sites and email Easy to find related/relevant information to my documents
Influential End-user	Tight integration of document creation with the web sites I care about I can easily create web sites that are the focus for shared work
IT	No more webmasters to manage content Get value from Platinum for document storage without changing mail infrastructure
Influentials	Can successfully deploy OSE+Office to enable departmental or workgroup level hosted webs
BDM	
CIO	

## Functionality Goals (list of possibilities)

### HTML:

- Enable better sharing of document content between applications via CF\_HTML
- Provide Web Archive support (MHTML) to simplify file management
- Investigate improvements to handling unknown HTML to enable scenarios with VizAct
- Targeted improvements based on customer and review feedback

### XML (tentative):

- Create an XML compliant version of our HTML output
- Allow for export of specific portions of a document as XML
- Build an export to XML wizard

### Open/save:

- Enable web sites to be accessible everywhere that I load/save files within Office

- Expose project spaces and team pages as target locations
- Understand better the Publish vs. Save model that customers have for documents stored on a web
- Provide a consistent user experience when saving documents to a Platinum store or to a OSE/FP Web
- Provide metadata promotion and synchronization from the client into the server
- Review server based metadata from within the document
- Enable templates to contain default metadata and project information
- File, New works against a web site so that I can create documents that are part of the web context

### **Send to 9/Email:**

- Figure out possible improvements to current model (comments section/cover sheet separate from body of document)
- Make sense of the right way to present send single sheet/slide and send entire presentation/workbook.
- Look at improvements to sending attachments for collaboration
- Investigate ways to unify the Save/Send/Publish experience

### **Collaboration:**

- Enable initiation of a review process that works against Platinum or OWS from within the application
- Track review/comment state from within the document
- Incorporate server context for document (discussions, document state, tasks)
- Show related documents on a server
- Make web discussion features work better across Office, just like in Word

### **Working better with documents in webs:**

- FP and Publisher focused enhancements to Hyperlink dialog
- Enhancements to File Open to make working in Frontpage better
- Better integration of FP client features into the Office applications (bots, views, list components)

### **Won'ts**

- Re-invent our HTML file format to be a pure XML implementation (XML+XSL). We need more feedback from customers before we go down a path that further reduces the reach of our 'view everywhere' file format.

## Scenarios

Contributing content to a web site is easy, easy, easy. I can find places to save to, create documents that reflect the context of the site, put the documents in the right place, and easily enable others (and myself) to find and collaborate on those documents again.

Sharing documents with others for collaboration/review/shared authoring is enabled in a consistent way for web sites and platinum/outlook team folder scenarios. I can start a review process from within the application context.

Other scenarios documents

Document collaboration scenario

Web authoring Scenario

Save to Web

Server Context

Server search

## Demo

//under construction

## Customer Inputs

- Site visits to understand how customers are deploying Office 2000 and OWS
- Customer and review feedback about our HTML implementation

## Technical Investments

Updating File/Open to provide an extensibility model to enable hosting of other namespace extensions

Implement a client side object model to File open to improve customization for ISVs

Integration with Web Server defined abstraction layer for talking to

OSE/Platinum/Tahoe/Generic Webs

Web Archives (MHTML) as a file format

## Project Tenets

All new features need to be persisted in HTML in a way that doesn't break backwards compatibility

New features can't rely on a new version of IE

MS/CR 0005755  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

## **Related Documents**

Other versions of this document along the way:

Original vision in December:

<http://office10/teams/WebClient/visions/Web%20Client%20ideas%20for%20NRO.htm>

More thinking once the feature teams were defined:

<http://office10/teams/WebClient/visions/web%20documents.htm>

## **Dependencies**

In the Office Box:

Office Web Server  
Tahoe/Rosebud  
Frontpage

Things your team is building that other folks rely on:

Enhancements to File Open/Save focused on Web server/Platinum scenario  
integration  
HTML plumbing and infrastructure, particularly CF\_HTML  
SendTo10 integration in the apps

### ***MMO possibilities***

Our list is kept on the server, but it's a bit out of date. Here are the prime candidates for work:

Architecture work to fix a limit in our long pathname/hyperlink problem  
(problems with MaxPath being too short).  
Investigating some FrontPage feature integration  
Leveraging work on Web Archive add-in to see how it maps to making MHTML a  
core piece of functionality.  
Clean up parser infrastructure.

### ***Competitors***

**MS/CR 0005756  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**



Office 2000—can we improve web publishing and server integration experience enough to compel upgrades  
Traditional suite products that provide web integration (WordPerfect, SmartSuite). Wordperfect 2000 will be especially interesting to evaluate due to its claim of tight integration of Trellix as well as native support for XML document authoring (at least in WP, not clear about the other applications in the suite).  
Trellix

### ***Assumptions***

### ***Issues***

The biggest open issue now is the evolution of our HTML file format. When we started Office2000, it was clear that sharing document via webs was going to be critical. Between then and the time we finished, there have been an number of advances in the core technology available, particularly in the area of XML. Our focus in Office2000 was on universal viewing of Office documents across browsers, and our format was optimized for that and the ability to make the save process lossless. It would be conceivable to want to produce a pure XML version of our documents that's based on XML plus some XSL or CSS transform, but that would only work on a subset of browsers. This type of feature would be more targeted at developers and vertical document management solutions rather than end users wanting to publish their documents. Doing this work means re-defining our web file format, perhaps in advance of any tangible user benefit. Another alternative is to look at implementing our HTML file format as XHTML, a format currently in a working group at the W3C. See <http://www.w3.org/TR/WD-html-in-xml/>.

**MS/CR 0005757**  
**CONFIDENTIAL**  
**HIGHLY**  
**CONFIDENTIAL**

**DRAFT**

## **Office 10 Release and Deployment**

March 23, 1999

### **Program Managers and Areas of Ownership:**

Greg Harrelson  
John Jendrezak  
Darrin Hatekeda  
Rob Howe  
Asa Whalstrom  
Teresa Fernandez  
Harry Sinn  
Gordon Hardy  
Javier Delgado

### **Charter**

The charter of the Release and Deployment team will continue to assure that the current and future Microsoft Office customers will find the new releases of Office compelling and economically feasible to acquire and upgrade to. We will continue to develop and implement features and process that allow all of our customers (End Users, IT Departments, and OEM's) to continue to lower the cost of deploying maintaining, and updating (Service Releases) installations of Office.

### **Vision**

"To provide solutions that reduce the complexity and expense of installing, customizing, deploying, maintaining, and updating Microsoft Office"

### **Customers**

There are three fundamental customer groups that we will be addressing with this release of Office. The approximate amount of focus for each area is listed as a percentage of the total.

#### **Administrators (70% of focus)**

The cost of ownership is addressed in this customer segment by providing applications that are easy to deploy (Initially and on an on-going basis) and require little to no maintenance (resilient). This was the customer

**MS/CR 0005758  
CONFIDENTIAL**

**HIGHLY  
CONFIDENTIAL**

segment that was addressed by Office 2000 and in Office 10 many of the same success factors should be focused on and improved.

- Easy to Customize
- Easy to Deploy
- Easy to Maintain
- Resilient
- Upgrade Fidelity
- Easy SR Rollout

### **End Users (20% of focus)**

We lower the cost of ownership here by providing a killer OOBE (Out of Box Experience), and by making the software self-maintaining. We need to reduce the number of choices that a user needs to make to get our software installed. If they selected 'Custom' last in Office 2000 we should be able to migrate their choices forward for the Office 10 install. We also gain end-user satisfaction by simply making the overall setup experience faster.

- Simpler Install (Drag Drop, Install same as last, etc.)
- Faster Install
- Tolerant, Forgiving, Resilient Applications

### **OEM's (10% of focus)**

OEM's are all about speed, how fast can they deploy software using our tools and processes. To lower costs to the OEM's we need to think about how to deploy software in a factory floor environment. We should modify tools that make OEM type assumptions about target machines. (Disk space available, OS on the machine, other software being bundled, etc.)

- Speed Focused
- Enhanced the set of deployment tools to make OEM deployment quick and easy

## **Functionality Goals**

### **IntelliSetup**

- Support Drag Drop Setup
- Support "Same as current Install"

MS/CR 0005759  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

- Migrate users settings from previous versions of Office
- Re-Claim space wizard
- Make Setup recognize and use web servers
- Support "Push Updates" Create a single unified install of Office
- Allow Setup to span multiple CDs during unified install
- Support DVD as a distribution media

### **Annuity**

- Investigate how we could adapt a DIVX model to Office
- Investigate Office Tone
- Have user setting stored for them on the Web
- Make updates automatic

### **Automated Deployment**

- Have a good SMS story
- Investigate other solutions such as Tivoli, etc.
- Smooth deployment of all of the applications shipped in the Office box
- Easier Multilanguage installs
- Better integration with IME and other multi-language feature
- Push updates
- Easy deployment for all the bits in the Office box, not just the core applications.

### **Cluster Server Support**

- Support in a 'Run from Server' install
- Support for running saving Office documents to the server

### **Deployment and Management of SR's and QFE's**

- Team dedicated to the planning and execution of SR's and QFE's
- Distribute SR's and QFE's with Darwin patches
- Migrate user settings from release to release

### **Limit what we Ship to Office**

- Don't ship any System files (other than those from Office 2000)
- Get rid of all non-darwin installed components (MDAC, IE5 etc).

### **Deliver Pre-Install Kits to OEM's**

- Provide 'Quick Install' (No costing, etc.)

**HIGHLY  
CONFIDENTIAL**

**MS/CR 0005760  
CONFIDENTIAL**

- Provide a OEM targeted enhancements to CIW

### **Tools**

- Improve Authoring Tools/Processes
- Take over Custom Maintenance Wizard
- Continue to improve CIW
- Integrate OPW into Office for machine replacement

### **Internal Enhancements**

- Expose internal ORAPI defaults
- Improve Windows Terminal Server functionality and deployment
- Make policy and defaults easy to deploy and manage
- Keep roaming user support
- Install All Office Premium bits without administrator rights
- No feature degradation running in a High Security environment

### **User Assistance**

- Tutorial on demand for hard to use features. Get a 10 min Pivot table tutorial when you need it.

### **Supportability**

- We need to investigate what changes we need to make to prevent support calls or shorten their length.

### **Release**

- Multi-Lingual Sim Ship
- Super-tight Dev/PM/Test processes
- Eliminate Self Reg
- Increased International ownership of SKU/Feature definition and overall project management.
- SKU Issues (Layout, Offerings, Schedule, etc.)
- Drive External Dependency contracts/processes
- Evaluate/Redevelop stand-alone build process
- Eliminate system file (noted above in "Limit what we Ship to Office")

### **Won'ts**

- We will not create a new set of release tools

- We will not migrate CIW transforms to next version
- We will not take new Darwin beyond 1.1 and maybe not even 1.1

## **Scenarios**

### **Make deployment and updates the same AND Update the server = updates to the clients**

An administrator uses the CIW and other tools to create their customizations for deployment. Later, they use the same tools to make changes, set policy, set new defaults, or other changes, and these changes are automatically deployed to their end users.

### **Install and Running on hydra, NT High security, non-administrator.**

#### **Drag & Drop setup**

A novice user is excited about buying a copy of Office 2000 and wants to start using it as quickly and as easily as possible. They insert the CD and the autorun application displays an Icon that tells them to drag this icon to their local hard drive. After a period of copying bits a 'Welcome to Office' dialog appears and asks them how they prefer to work with Office.

#### **OEM Pre-Install**

A system engineer at an OEM is using an enhanced version of CIW to prepare a pre-install image of Office 2000. This image is to be copied to the hard drive of each machine that is being built that day. The resulting setup is quiet and very fast, but still is very much our own setup. Once the setup finishes in the factory it is removed from the hard drive and made ready for the customers first run experience.

#### **Tutorials when you need it**

**Go to pivot tables, table of contents, mail merge and get tutorials when you need it.**

#### **Demo**

- Office is customized for deployment and placed on a share.
- Users drag and drop or an administrator does a file copy to their end users machines
- User launches an application and all of the shortcuts. OLE goop, etc. everything shows up and is installed.

- User needs help and gets a tutorial just when they need it.
- Administrator realizes that they need make a change and uses the same tools to update the image
- User gets new updates

### **Technical Investments**

- Drag Drop Setup
- Web Deployment
- Annuity Model
- Pulling out system files
- Change the way authoring is done (developer vs Author)

### **Project Tenets**

- All features function in a High Security environment
- Run from source works in all modes including High Security environment
- Install and run without administration rights on Windows 2000
- No change in file formats
- All applications ORAPIfied
- Mixed environment between the new version of Office and previous versions fully
- No new hardware
- No new operating system.
- No new system software as part of Office
- Old applications will be able to load files created by new Office All old macros and custom solutions run 100%
- No change unless at least two times better
- All settings support Roaming user

### **Related Documents**

- Darwin
- Windows Update
- Cluster Server
- Windows 2000
- Terminal Server

### **MMO possibilities**

- Exploratory: Table-driven custom action consolidation?

If we are going to have to live with Darwin 1.0/1.1 in 10, are there any wins from consolidating our custom actions so that we have fewer and

they're more declarative/table-driven? Example that may or may not be practical: Many of our CA's do one of two things, provide sequenced self-reg or express feature dependencies (turn one feature on or off to follow another or a condition). If we added a DynFeature and SequencedCA table to our MSIs and implemented *one* custom action per table to read the table and process it, would that make our authoring easier/more manageable?

- Exploratory: External UI for Gimme layer using SDM

I believe AndrewH is already investigating writing an SDM UI handler for Gimme to deal with our need for more control over source resolution dialogs. Implementing this would also give us a lot more control over the look and feel of our runtime install, might give us easier integration with Component Manager, etc. Is it worth prototyping and seeing if there's a cool feature lurking here?

- Getting rid of the Terminal Server transform

We should have integrated this into mainstream setup for 9 (it would have been less work), and should definitely get rid of it for 10. It's not 100% clear to me if wiping this out is MM0-type work or if it's something we should look at as later feature work. (May be better to put this off given that we're already planning to do a lot of work on CAs that shift around default feature states for IntelliSetup. May be able to leverage that code here. Conversely, this or the DynFeature task might be useful scout work for IntelliSetup.)

- Unit test for Gimme layer/Darwin integration

In 9, every single quicktest was testing our API and Darwin integration. Given the gap we're deliberately introducing between dev and customer machines for 10, should we look at implementing unit test infrastructure for Gimme so that we can keep a high quality gate on tcutil checkins?

- Terminal Server compatibility and optimizations.

Investigate what we need to do to make Office 10 a great Terminal Server application suite. Look at graphics optimizations and integration with Terminal Server, e.g. potential opportunities using Virtual Channel extensions to RDP, "Program Neighborhood" functionality, Terminal Server per-user licensing integration, and possible RDP-DCOM integration.

- Developer tools for release authoring



- Setupless work ("Backdoor" or "Alternative path")
- Release authoring guidelines and standards
- Investigate cluster server app requirements. Set up a small cluster server and test against it.
- Review entire Office 9 setup for efficiencies. Effectively a setup logic code review.
- Review all use of Gimme API in Office code for usage of correct flags. We've got a lot of postponed bugs where we don't repair correctly due to incorrect API usage in the apps.
- Develop pipecleaner plan for 10. How will we deal with removing 9?
- Identify application deployment work items. What do we need to do to fully leverage Win 2000 deployment and SMS/Tivoli?
- Take a fresh look at license verification. What will we do for 10 and what do we need to do to enable annuity licenses, i.e. expirations?
- Examine using Darwin 2.0 with a transform against a Darwin 1.1. database -- i.e. if we detect Darwin 2.0 on the machine at install time, we automatically deploy with a transform that adds Darwin 2.0-specific functions. We should test with this now to ensure that adding columns to all database tables works in Darwin 1.1, i.e. it just ignores the additional columns.
- Identify setup bootstrapper work for web deployment.

### ***Competitors***

<http://www.installshield.com/>

**OilChange and FirstAid** ([www.cybermedia.com](http://www.cybermedia.com)) – a company brokering patches etc. The value they add is tool that analyses what is installed on the machine and what

### ***Assumptions***

### ***Issues***

MS/CR 0005765  
 CONFIDENTIAL  
 HIGHLY  
 CONFIDENTIAL

**DRAFT**

## **Office10 Programmability Vision Statement**

4/9/99

**Program Managers and Areas of Ownership:**

GPM: PJ Hough

### **Charter**

The charter of the Programmability team is to advance Office as a solutions and development platform. We are part of the competitive response to Notes R5 and the growing demand for collaborative solutions.

### **Vision**

***"Real Solutions, Real Fast"***

Our goal is to enable Office developers and end users to build and deploy workgroup collaborative apps. The team will focus on building a great solutions model, building a series of usable solutions that work out of the box, and delivering the customization environment and tools to enable Office users to enhance the solutions. We will also maintain the existing developer community and expand the relevance of Office to a broader set of professional developers.

We do not intend to compete head-to-head with Notes, but to use the combined assets of Microsoft (including Visual Studio) to provide a more compelling roadmap for solution developers. The key pivot points for the Office programmability team will be

1. Bringing Office ease-of-use to solutions development
2. Providing a seamless transition from Office to Visual Studio for the professional solution developer.

### **Customers**

<b><u>If I am a(n)...</u></b>	<b><u>what can you do for me? (two sentences, please)</u></b>
End user	End users will be able to easily instance applications from the template solutions that ship in the Office

**MS/CR 0005766  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**

box. End users will also be easily able to create solutions that "surround" Office document templates. IEU's will be easily able to customize the solutions that ship in Office 10 using the Office designer. IEU's will be able to make relevant and significant modifications without writing code. IEU's will also be able to use the solutions we ship as internal sales tools to demonstrate real business benefits of Office.

**Influential End-user**

IT will be able to deploy custom collaboration solutions that have been derived from the template solutions which ship in Office, or which they have built without leveraging templates. IT will also be able to smoothly transition these solutions to Visual Studio for extensive enhancements.

**IT**

The BDM will see real end-user value in the solutions we deliver, and will see this work as delivering on the promise of Office as a development platform for the masses.

**BDM**

The CIO will see significant Office/BackOffice synergy, and will have demonstrated examples of solutions that leverage all of the technology deployed in his/her organization.

**CIO**

Cheat sheet is at <http://office10/teams/gpm/customer.htm>

## **Functionality Goals**

Compete with Notes designer on ease-of-use and end-user-benefit with great out-of-the-box solutions that can be easily customized.

Enable powerful solutions to be built simply. Enable complex apps to be started, with a great transition to Visual Studio, Data Access Pages etc.

Solve the "end-to-end" scenario for application design to deployment and update that is a general weakness of our tool set today.

Great project model interoperability with VS

Will work best with Platinum – we may be able to design some simple apps against SQL server.

**HIGHLY  
CONFIDENTIAL**

**MS/CR 0005767  
CONFIDENTIAL**

Will be a great designer for Outlook 10 applications, but many of our templates will work well in HTML 3.2 (read: no DHTML, no Applets, no CSS) browsers. We will provide a great cross-platform story for our simple apps.

**Issue: What does it mean to be a great Outlook 10 application?**

We will also ship some applications that require Office on the client desktop (document library, for example).

We will build some template solutions on the Programmability team and we will also coordinate with other teams that are building templates to ensure consistency.

### **Won'ts**

We won't solve the Exchange/SQL problem. Our designer will build solutions that leverage the power of Exchange. However, we will support the notion of "foreign pages" in our solutions, so that a DAP or other page could be included as part of a solution application.

We are not targeting vanilla Web servers

We are not targeting hosted scenarios

We are not picking up any components that have a COM+ dependency on the client or server.

### **Scenarios**

Outlines here: details to follow

#### **Scenario 1: End user builds a solution**

1. Boot Outlook
2. File:New Database:Survey
3. Answer some questions (who should the survey be mailed to?)
4. Done!

#### **Scenario 2:End user/IEU customizes a solution**

1. Steps 1-3 above
2. Tools:Design
3. Add a graphic to the survey form
4. Add another question
5. User uses "Script Wizard" to process incoming survey responses

**MS/CR 0005768  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**

6. Publish Application template to central location
7. Save&Exit
8. Done !

#### **Scenario 3:End user re-uses a custom template**

1. User browses central location looking for a custom application
2. User finds custom template
3. File:New Database:Custom
4. User answers some questions
5. Save&Exit !

#### **Scenario 4:Developer builds new application without re-using a template**

1. Start:Office:Application Designer
2. User constructs schema/forms using a form or table view
3. User/developer constructs frames, navigation, agents, pages etc
4. Developer writes some simple script
5. Publish Application template to central location
6. Save&Exit
7. Done!

#### **Scenario 5:Office user builds solution from within Office application**

1. User Boots Word
2. User creates document template (business plan template)
3. User saves template to Platinum Server
4. User launches Designer (automatically scoped to the server location where the doc was stored)
5. User chooses "Document Library" template
6. User identifies Word document as the "new document" template for the solution
7. User saves solution
8. Done!

#### **Demo**

<5 demos based on the scenarios above>

#### **Customer input**

MS/CR 0005769  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

We are classifying our user base into three broad groups:

1. End users that consume solutions and solution templates. This group of users will not use the Designer directly, but will heavily use the solutions and template solutions built in the designer. The target end user is a knowledge worker that is trying to build an application to share information with others.
2. Influential End Users (IEU's) / Power Users that customize solutions and build application instances or solutions that are shared with others. This group of users will use the designer exclusively as their authoring tool for solutions. They will primarily rely on non-coding techniques to get their job done, and will only use the designer when the template solutions don't meet their need.
3. Power User / Developer that heavily customize templates, but frequently build entire applications. This group of users will use the Designer as another tool (probably their initial tool) to create collaborative solutions, but they will also use Visual Studio to supplement their needs. The developer will code in the Designer, and will also expect to see such features as debugging integrated into the tool.

### **Technical Investments & vision for other teams**

We are betting on Platinum, and will rely on the default features installed on a Platinum server.

Platinum implies a Windows 2000 server bet also.

The following list represents current thinking on the other big technology bets we will be making

- Trident - HTML editing surface/Forms designer
- Visual Studio - Designer shell
- VS script engines - Version 6.0 -- client and server scripting
- MDAC - Data stack
- Data Access Pages - Shared data binding, and possibly some shared components

Here is a list of technologies that we do not plan on changing dramatically in Office 10

- VBA

MS/CR 0005770  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

- MSE
- Digital Signatures
- Com Addins
- Forms^3
- Office Object Models

## **Project Tenets**

Solutions that we ship work great in Outlook 10 and in all browsers that support HTML 3.2

End-to-end wins over everything else. The features that we add to the designer, to Outlook and to the solutions that we ship must converge on great end-to-end experiences. We will limit work in some areas to expend effort evenly on the life cycle of a solution.

No additional server install required by our solutions. Our solutions work against Platinum as it ships. We will not require an additional server-side install to enable our out-of-the-box applications.

Building and deploying a solution is an End User Feature.

## **Related Documents**

Dependencies both in and out of the Office box. Try to include contingencies (if this dependency doesn't come through, what's our fallback plan)?

Things your team is building that other folks rely on

*MM0 possibilities*

*Competitors*

Notes <http://www.notes.net>

*Assumptions*

Platform is Office 10+Platinum. All other desktops (including Office 2000) will be treated as "reach" desktops.

SQL is not a target store for the Designer

MS/CR 0005771  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

Offline is a lower priority than Reach (HTML 3.2) and is largely a function of the Outlook work to integrate the new local store.

While we will not directly support building forms that connect to non-Platinum stores, we will allow arbitrary pages to be added to our solution.

### ***Issues***

This is a placeholder for additional sections that have yet to be written.

- Security
- Roles
- Replication
- Data Model
- Programming Model (what's the right api?)
- Web components and the designer
- Application/template publication and discovery
- Offline (runtime and development)
- Inheritance
- Persistence
- Data Binding
- CDO / Outlook object model
- Extranet solutions
- HTTP-save (Rosebud) enabling of the designer?

### **Code that we may write**

We own the UI and developer experience in the Office designer

We may also deliver UI components, such as an enhanced Rules Wizard that run in Outlook

We may also deliver Admin components that help manage a departmental Exchange server, focused on the scenarios and features that we enable.

### **Architecture:**

TBD

**MS/CR 0005772  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**



↑	<b>Office Art 10</b>	7/2
	Author <u>tuann</u>	
	Area Office Graphics	
	Created 2/11/99	

---

## “Compelling Graphics Made Easy”

### Mission

For the last 2 Office releases, Office Art has delivered the shared graphics support in the main Office applications: Word, Excel, and PowerPoint. In Office97, we offered new graphics features that showcased the ability to easily create professional looking graphics. A recent user study ([\\Gpu\prodplan\documents\research\GUS3\\_files\frame.htm](\\Gpu\prodplan\documents\research\GUS3_files\frame.htm)) collected by our product planning team has confirmed the popularity of Office Art features among Office97 users. In Office2000, we mainly lined up behind the HTML push and focused on delivering the powerful feature of round-tripping Office Art data via HTML. VML, a W3C submission to the Vector Scalable Graphics working group, came out of such effort.

In this release, our mission should mainly focus on expanding the shared graphics foundation by integrating with more clients and by leveraging new Windows graphics platforms. It is important that we made a genuine attempt to line up our feature ideas behind the main Office10-wide themes:

- Increase Office upgrades by providing better performance, robustness, and simplicity
- Leverage or develop innovative technology (graphics) that blows people away.
- Solve business problems out of the Office box.

One interesting finding in the Office97 graphics usage survey is that a substantial amount of Office users still resort to using a dedicated illustration application (i.e. Photoshop, Visio...) to create graphics artworks. We need to understand and drill down on this data point better and find the appropriate fixes to ensure that such tasks can be simply and professionally performed within the Office applications using Office Art.

### Feature Ideas

Earlier planning meetings in Office2000 and conclusions drawn from the Office97 graphics user study have helped us to tabulate the following set of feature ideas. They can be used as a baseline for the upcoming Office Art10 planning. A careful examination of these features shows that they can line up pretty well with the Office10 overall mission themes that were discussed in the previous section. This feature list is not yet sorted based on the priority order.

**MS/CR 0005773  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**

**Better HTML/CSS support.** With the Office2000 release, we have made great strides in reading/writing Office Art data in HTML. However, due to schedule constraints, we have deliberately opted to offer only the minimum support for CSS-P, especially in the context of reading generic HTML Web pages, i.e. Web contents created by non-Office applications (Word, Excel, PowerPoint). Such minimum support has triggered several bug reports and concerns about our ability to offer Office applications as the HTML editors of choice. In this release, we have to offer a more seamless support for reading generic web pages that contain CSS positioned graphics and CSS style sheets. We also have to address the important HTML layout bugs that were postponed late in the Office2000 product cycle.

**GDI+ Integration.** GDI+ is the next generation of GDI on the Windows platform. It is a total rewrite and is planned to ship in Q2 2000 coinciding with the Win98 OSR2 release. It promises several advantages: bug fixes, performance improvement, modularization, and new graphics features. During the Office2000 project cycle, we have been very proactive in communicating the Office Art graphics requirements to the GDI+ team. Our common goal is to establish a thin mapping between the Office art graphics properties/effects and the GDI+ public interfaces. For example, GDI+ gradients should be a strict superset of the Office Art gradients.

Even though this undertaking is definitively risky and implies a substantial regression testing effort, we believe that the simple promises of more robustness in the graphics rendering pipeline and a drastic performance improvement in printing are compelling enough to seriously consider our investment in GDI+. Leveraging new Windows graphics platforms like GDI+ also enables us to easily add new graphics features to Office users by hooking in the necessary user interface. The possibilities are many: Web-savvy raster effects on Office Art shapes, full alpha (opacity support) both on screen and on print...

**VML.** VML has turned into a Microsoft strategic initiative. As it is defined now, VML contains 2 levels: Modeling and Rendering. The Modeling level includes features such as the Office Art Autoshapes and the constraint transformation. The Rendering level encompasses stroke/fill styles, NT-based paths, and color transformations. In synchronization with the GDI+ team, we will evolve the VML rendering level to be the XML-based declarative language for the GDI+ interchange format, i.e. next generation of Windows metafiles. Based on future Web developer feedback on the IE5/VML feature, Office Art will continue to improve the VML modeling level to add more high-level semantics such as our rule engine. The goal is to reinforce the positioning of VML as the optimum vector graphics interchange format for HTML documents originated from Office.

**Publisher Integration.** In this release, the Publisher team has already started to investigate the replacement of its graphics engine with Office Art. The benefits of using Office Art are immediate: more graphics features, better integration/interchange with Office applications (Word, Excel, and PowerPoint). It is within our mission goals to work closely with the Publisher team to ensure that Office Art10 will meet the Publisher

requirements in all areas: features, quality, and performance. We will have to provide additional support for the Publisher legacy graphics features (i.e. metafile/bitmap rotation, CMYK color model, "size-to-text" shapes...) that can also benefit the other Office Art-based applications.

**Ease-of-use.** The Office Art UI needs better discoverability. The Office Art "fancy" text-on-a-path feature, WordArt, is very popular. But Office users have to struggle to use this feature. They should not have to make the distinction between regular text and WordArt text. Unfortunately, the current WordArt text entry model (via a modal dialog) implicitly reinforces this unneeded separation between the 2 "text" models. We should improve the WordArt UI by offering a modeless mechanism to create WordArt text. A tighter integration of the WordArt feature into the application text handling (e.g. Drop Caps in Word) is also highly desirable.

Offering the look and feel of a familiar drawing application will help to attract Office users to choose Office Art as the tool to create their graphics artworks. Simple and cheap additions of "drawing" UI elements such as a Zoom tool, visible and adjustable grids, precise positioning of Office Art lines/shapes, may be the right ingredients to position Office Art as the tool of choice for the creation of professional-looking graphical contents.

Another area of improvement is the desired unification of line/fill styles between Word tables and Office Art shapes. In Office2000, the new PowerPoint "table" feature keenly illustrates this divergence. Even though its UI is based to the Word table feature, a PowerPoint table is based on Office Art structures and thus inherits a much richer set of line/fill styles relative to its sibling, a Word table. We can easily extend the same rationale to address other areas such as Excel cell borders.

**Diagramming.** One of the most frequent activities is the creation of charts and diagrams in Office documents. Office Art already has a solid foundation to support basic diagramming. As illustrated by the connector feature, the Office Art rule engine is already adequate and is designed to adapt to the diagramming requirements. In Office97, we already prototyped the automatic creation of popular diagrams (cyclic, pyramid, radial) via VB scripts. What is needed is the UI model on how to bind data to the diagramming rendering backend. Applications like Visio are too rich in complexity and features to solve the simple needs to create basic diagrams. We can solve it within Office Art in an elegant and compact way. But on the pragmatic side, adding connectors to Word is not a simple task and it needs more thoughts on how to simplify/solve the problem of routing connectors across page boundaries. Support for diagramming also implies the need to have text attached to connectors.

**Org-Chart.** Previous Office releases have shipped the Org-Chart OLE server that has not been updated for the last several releases. We know that its usage is still not negligible and for contractual reasons, we have now to decide if we should extend the current license or to buy off the distribution rights. The Office Art foundation is rich enough for us to emulate the Org-Chart application and to enrich it with all the cool Office Art graphical features. For backward compatibility reasons, the main challenge is

to reverse engineer the Org-Chart file format. We should keep an open mind in this area and be opportunistic on providing an Office Art-based replacement for the now obsolete Org-Chart application. Such effort will give us more control on improving the feature set of this component and will remove an important dependency on a non-strategic third party vendor.

**Seamless coexistence with PhotoDraw.** In Office2000, Office Art has incorporated some basic steps to offer a tighter integration with PhotoDraw: promoting the PNG format as the preferred raster interchange format on the clipboard is one way to propagate the fundamental alpha channel between the applications. In this release, we should explore further tighter integration scenarios that go beyond the current publishing model. We can draw the parallel with what PowerPoint has achieved with Graph via the OLE data exchange mechanism. For example, Graph understands the PowerPoint color schemes and can handshake with PowerPoint to support building/animating a Graph bar chart in the PowerPoint slideshow. As in that case, we can use the OLE protocol to negotiate privately with a PhotoDraw OLE server to offer the optimum format for exchanging shared data between the applications.

**Client/Server model.** From the Office Art perspective, this is a new area that we can put more thoughts into. For Office Art, it might simply mean a tighter integration with persistent storage on the server end, and a better link management within our HTML processing. We will probably leverage new operating systems services to provide such support. Another area of interest is the idea of server-based VML generators. An interesting application of that technique can already be found in the geological mapping domain as illustrated by a small firm in Spain (<http://www.lmapper.com/index.htm>). We can envision a server-based application that can generate VML data describing a specific Office Art drawing to be integrated into the Office applications on the client side: photo library, clipart repository, train station maps (a very popular feature in Japan)...

## Next Steps

The ideas discussed above are only a small portion of what we have in mind for an ideal Office Art code base. Actual constraints on the current resources (assume no change in team size) and the potential short product cycle of Office10 will force us to prioritize this feature list and to choose the most important items to be implemented in this release.

This document is intended to outline our early thoughts on how to shape the Office Art10 release. As food for thoughts, it is expected to trigger valuable constructive feedback. We really appreciate your comments. Be sure to send them to TuanN.

## Change History

Date	Changes Made	Program Manager
2/10/99	First pass	TuanN

MS/CR 0005776  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

## **DRAFT Office10 Word Vision Statement**

April 26, 1999

### **Program Managers and Areas of Ownership:**

GPM: Reed Koch

Lead PMs: Chris Pratley

PMs: Dixon Miller, Ashok Kuppusamy, Jeff Reynar, Jose Luis Montero, Marcin Sawicki, Mike Rigler, Rahul Sonnad, Roberto Taboada, Young Koh

### **Charter**

The charter of the Word team is to produce a world-class word processing application that makes the process of writing fundamentally easier and more productive.

### **Vision**

Word 10 is the easiest editor around the world for common web, email, and print documents produced individually or in a group.

These are the key focus areas for Word 10:

- Collaboration
- Authoring for Webs
- Empowering & Exciting Users
- Fundamentals

### **Collaboration**

Collaborative authoring features in Word integrate tightly with document management, knowledge management, and Notes-competitive features provided by groups both within and external to Office. However, Word's suite of collaborative features will work well out of the box, assuming nothing more than Word documents mailed around as email attachments. Servers aren't required. This is a common collaboration scenario for numerous customers who haven't invested in or are underutilizing a server infrastructure.

For Word 10, we envision a world where users can collaborate on the creation and editing of a document without having to know anything about collaboration features. Simplicity and ad-hoc collaboration are guiding principles. Everything they try to do simply works as a user would expect. If multiple people open the document for editing at once, it just works. If someone copies the document locally to edit, then copies it back, it just works. If a user wants to send the document via email for review it's as easy as sending a DOC attachment is today. When he gets the document back, merging is automatic, and accepting/rejecting changes is simple. Commenting a co-worker's presentation is as easy as clicking

**MS/CR 0005777  
CONFIDENTIAL**

**HIGHLY  
CONFIDENTIAL**

and typing. If multiple copies of a document are generated in the course of sending it for review, Office tracks them for you so you don't have to.

Here is what Word is doing.

### **Document Clean-up and Format Stripping**

Combining content from multiple document types is a manual and laborious task. Text from email messages and web pages may be in a different base font than the target document. Text from plain text applications may have carriage returns at the end of each paragraph. Empty paragraph marks, commonly used to create whitespace between paragraphs, combine with Body Text style space before/after to create "too much" whitespace.

While attaining consistent formatting is time-consuming for even the most advanced user, an extra difficulty blocks beginning users. In many cases, the same "appearance" can be accomplished using one of several formatting features (consider borders under text in a table cell). Users who do not have widespread familiarity with Word's features, often find themselves unable to remove unwanted features. WordPerfect migrators have long requested a Reveal Codes feature to address this user problem.

Cleaning-up formatting of content added to a document in Word 10 is at most a discoverable, one button operation. Where possible, it is automatic.

Format stripping and clean-up is a prevalent task in Word and Word will take a front seat in driving innovations (office-wide) in this space.

### **Templates**

Templates are a highly desirable feature that works poorly today. They are hard to customize (a top support issue), difficult to distribute and get to work with Office UI (a top beta newsgroup issue), and easy to "break". Entire word-processing departments at major legal, consulting, accounting and other service-oriented firms are created because professionals easily and unintentionally step around features and formatting provided in templates. These companies find it easier to centralize document editing than to clean-up documents that are formatted incorrectly up-stream.

Templates are easy to create and distribute. Template users intuitively discover and use customizations provided by the template creator.

Template creation and distribution are Office-wide needs, and Word will take a back seat role in these two areas. Word moves to the front seat for making templates less "breakable" (or easier to "lock down").

### **Styles (Make this document look like "that")**

Closely related to templates are styles and structure. Styles are difficult across the board (creation, application, modification). Styles in Word 10 are easy to apply and modify. Word supports the integration of XML constructs with documents.

MS/CR 0005778  
CONFIDENTIAL

HIGHLY  
CONFIDENTIAL

Styles Innovations will be driven by Word and Publisher. Support for arbitrary XML will be driven by Word with support from Office shared teams.

**Review (Revision Tracking and Commenting)**

Office and Word include several features (Change Tracking, Comments, Discussions) in the "review" space. None of these is frequently used. Reasons include lack of ease (how can you beat paper, pencil and post-its?), discoverability, lack of robustness, fear (when and how are revisions turned off? Will they be seen by someone who should not see them?), and problems with down-stream applications (document comparison and conversion systems). Track changes and comments really belong in the tier of extremely simple, no forethought required, no process needed collaboration features. The UI needs some unifying between all the collaboration features, but especially track changes & commenting need to be unified. This is really the Word vs. paper battle, except that really Word coexists with paper. The "rightness" paper will not disappear for many reviewing tasks, but here's where we get to add value that paper can't.

Document Review and Commenting in Office 10 is intuitive, puts the author in control, and works well with existing processes.

Revision tracking and commenting are Office-wide needs. Word will partner with the Office Shared teams to provide breakthroughs in this space. Currently, revision tracking and commenting within the document will be done by Word. Commenting in the browser is probably done by an Office Shared team. Sending a document for review is probably done by the Outlook and Web Documents team.

**Editing as a Group (Check-in/Check-out, Multi-user editing, Versioning and Version Comparisons, Workflow and Approval)**

Multiple users can make changes to the same document (some offline, some online), and easily combine them into one master document that merges all the changes in an easily readable way.

Workflow and approval can require a bit of forethought by users to setup and deploy. Almost by definition these require some sort of process. Our emphasis should be how much can Office do without requiring explicit user action. Users should get the benefits of these features without requiring the cognitive load to think about them ahead of time.

Word will be doing simultaneous multi-user editing for documents, but the rest of the scenario probably lives with the shared Office teams and Outlook.

**Word as Default Mail Editor for Outlook**

MS/CR 0005779  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

In Office 10, Word will be the default mail editor for Outlook. Here is a list of things we need to do to get there:

<http://outlook/outlook10/specs/vserdv/Word%20as%20default%20editor%20reqs.htm>.

---

### **Authoring for Webs**

Word is the best editor for creating documents that reside in web sites for knowledge workers and IEUs.

As in Word 2000, we will prioritize our focus areas to keep up with evolving web standards and common authoring practices. However, it is not a goal of Word 10 to support every HTML and browser feature.

### **More Complete Support of HTML & Browser Constructs**

- Integrate HTML peeler into Word 10 (huge corporate customer interest)
- Linked CSS management
- Misc common HTML tags and attributes (body margins, nobr, div)

### **Full Support of Browser Layout**

- Autospacing
- CSS Positioning
- Support the IE5 box model for margins & padding
- Nested boxes (div)

### **Help Drive Browser Implementation and Standardization**

- Drive W3C proposal efforts in the area of text wrapping, floating objects, and other areas that Word has support for and expertise in
- Work with the IE teams on prioritizing new browser features that aid Office HTML files look good on the web

### **XML**

Customers often want to tag or type certain content in Word documents to be post-processed by another tool. We observed this in site visit to the WSJ where the articles were created in Word, but some text was tagged with meta-information. Word will better handle XML constructs within HTML files.

Since XML is not a defined standard yet, it is important to keep track of the interesting developments and respond to them as appropriate. Currently, we plan to do the following:

- Do a better job of preserving XML and unknown HTML.
- Respect the syntax making content within XML tags visible or invisible.
- Provide some mechanism to associate additional information such as metadata with actual text in the document.
- Provide the ability to associate stylesheets to XML tags.

**MS/CR 0005780  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**



Word will not:

- Word 10 will not be a full blown XML editor.
- Word 10 won't provide the ability to create any arbitrary markup or a pure XML file.

### **Easing the transition from print documents to the Web - MiniWebs**

Getting word documents onto a web and look good in a web is still an interesting problem. We made some definite progress in Word 2000 with Save as Web Page and Themes. We've seen many customers struggle with converting long Word documents into a format (separate web pages, navigation, etc.) that is more appropriate for the web. Word 10 enables a document to look great for both print and the web without maintaining separate versions of the document. Word 10 will do the following:

- Automate the process of getting a word document onto a web
- Automatically provide navigation to the webified document
- Show the document WYSIWEB (separate web pages) in Web Layout View
- Allow the entire document to still be editable in a streaming Print Layout View
- Allow the ability to print the entire miniweb
- Allow managing of the miniweb (adding & deleting document, provide web page breaks, etc.)

---

### **Empowering & Exciting Users**

Word 10 empowers users by providing new, helpful features and giving them more control over existing features. Word 10 empowers users in five crucial ways using technology from within Office and from NLG and DTAS.

**Speech.** Japanese users are at least partially liberated from the IME and poor typists in the U.S. will see their document creation times reduced. Spoken command and control gives great typists the opportunity to keep their hands on the home row and control Word by voice. Knowledge works on the go dictate to handheld devices and synchronize them with Word. Word owns dictation. Command and control is likely to be owned by the End User Productivity team. Other speech features (e.g. sync with handhelds, voice as a datatype) are possible, but may be owned by other teams.

**IntelliSense rules.** The dangers of automatic actions are well known: users feel out of control, undoing mistaken actions is difficult and time consuming and only power users ever learn how to turn off particular rules. Word 10 solves these problems and, though there are more automatic actions in Word 10, users never feel that Word's desires for their document are more important than theirs.

MS/CR 0005781  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

Also, IntelliSense rules in Word 10 can either be on or in semi-automatic mode. In semiautomatic mode, Word will offer to take the action on the user's behalf, if they so choose. For example, want the sequence (c) to turn into © this time but not normally? Select the copyright symbol from the right-click menu.

**Learning.** Word 10 watches what you type and how you interact with the IntelliSense rules. For example, if a user always hyperlinks Microsoft to [www.microsoft.com](http://www.microsoft.com), Word 10 will do that for them. If they only do it occasionally, Word 10 lets them choose to add the hyperlink, but will still save them from typing it from scratch.

**Leveraging Factoids.** Strings in documents that NLG labels with type information, such as people, places and dates; and providing strong ties to Outlook. Because of factoids, Word 10 users can access contact information directly from names in their documents. They can also find maps to places, change date formats without retyping and find company home pages. Accessing all the e-mail you've exchanged with the recipient of the letter you're writing is easy, too. AutoComplete is beefed up for Factoids, as well, and even lets users complete multiword names.

---

## **Fundamentals**

We need to ensure that Word does essential things very well, and improves with each release. We will focus on addressing key areas of weakness in our product, with an emphasis on reducing customer cost and pain even when the main theme of the release doesn't address these areas.

### **Address The Top Customer Complaints**

Often things that many customers complain about such as basic use glitches or a useful feature that got dropped from the product go unaddressed for multiple releases, or worse, never get addressed. Working with PSS and MS Wish Data, we identify the most common of these complaints and eliminate the problems.

We already know of the following areas that we are doing:

- mail merge (one of our top PSS issues for the last couple of releases)
- document comparison improvements for the legal industry
- reveal codes
- selectable lists

### **World-Wide**

We have several worldwide improvements to make. Some of the most important are:

- Add Indic, Thai, and Vietnamese to the single worldwide exe
- Add support for Unicode 3.0
- Support all language locales in Windows 2000

MS/CR 0005782  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

- Improve BiDi integration
- Language autodetect improvements
- A more complete list is here:  
<http://office10/teams/Word/whatsleft/International/Intl%20Work%20Items.htm>

### **Significant Punted Bugs/Capability are Fixed/Added**

"We shipped it in Office9 and No One Complained" is not good enough. In the push to ship we punt bugs or feature capability we really wish we didn't punt because we know that customers will run into the problem area and it looks bad. Word 10 allocates PM/development time to identify and address the worst of these punted bugs and punted capabilities.

### **Increased Reliability**

Word doesn't crash. (Note: the meaning of doesn't TBD) Customers rarely – if ever – experience unexpected data loss. Word protects data integrity under more circumstances than it does in Office9. It is fully resilient to crashes in the components it uses. We unify our efforts with the Office fundamental team's "Safe Mode" work.

### **Increased Perceived Performance**

We put effort into making the most frequently executed functions faster. Things like enabling background load & save of HTML documents. What are the areas we can improve here?

### **Customers**

<b>If I am a(n)...</b>	<b>What can you do for me? (two sentences, please)</b>
End user	Word can work on documents with other people easily. Formatting is smooth, easy to do, and works well with formatting others have applied.
Influential End-user	Word is a richer and more powerful web authoring tool. Graphics, picture bullets, and frame authoring all work better.
IT	The file format doesn't change.
BDM	Word will help your knowledge workers work more quickly and efficiently on documents together.
CIO	Word helps my employees work together more quickly and efficiently and I don't incur a lot of organization costs during rollout.

### **Functionality Goals**

List them.

### **Won'ts**

MS/CR 0005783  
 CONFIDENTIAL  
 HIGHLY  
 CONFIDENTIAL

### **Word as FP Editor**

Word does not become the default editor in FrontPage.

There are 3 scenarios of usage in Frontpage.

1. High-end site authoring for a Webmaster
2. Intranet site managed in FrontPage and running OSE or FrontPage server extensions (ex: <http://office10>)
3. Simple, basic webs authored entirely in Frontpage

Scenario 1 isn't interesting to Word because Word is unlikely to implement FP features of interest to these users such as preservation of exact HTML tags. Scenario 2 is interesting, but Word is already doing well here because opening & saving to the web is enabled in Office 2000. Most of the documents on <http://office10> are authored using Word. The site is managed with Visual Studio or Frontpage, but most of the document authors never need to use the FP editor.

Scenario 3 is where making Word the default editor would make a difference. In talking with the FrontPage team, the problems in this scenario for users are more structural than the FP editor. Most of the problems are in getting started with a hosted web site, getting a domain, republishing to the hosted web site, etc. Being in-place with the FP explorer is a big advantage here since having a separate editing app was a serious usability problem in previous versions of FrontPage. Word won't be able to go in-place in the FP explorer.

Scenario 2 is by far the largest set of users. I doubt we care enough about scenario 3 to do all the work to make Word the default editor.

### **Scenarios**

For the collaborative authoring scenario see the Word customer profile up on <http://wordweb/customerprofile>.

### **Dependencies**

#### **Web Documents**

Here's what we're expecting from the Web Documents team:

- **Collaboration scenarios** – Signifying and sending a document for review are probably in Web Documents. Revisions, commenting, and merging revisions is within Word, but coordinated cross Office.
- **FP Functionality in Word** – knowing location & structure of the current web, Shared Borders viewable in Word, hosting FP bots, Exposing FP views (navigation, hyperlinks) in a pane
- **HTML Forms** - We need a cross-Office story for HTML forms. Perhaps we host something in Trident.

MS/CR 0005784  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL

### **Trident**

Closely cooperate with Trident on new feature implementation to avoid inconsistencies in layout.

### **Search**

Knowledge workers would greatly benefit from search tool that covers multiple stores (e.g. exchange, your hard drive, the corporate intranet) with only one UI (a la AltaVista discovery) within Word since finding documents is become more important when writing. This should be a cross-office feature and we should work with teams that may invest in this (e.g. Outlook combined with PKM). Implicit searches, like those the research group has proposed in which documents relevant to what the user is writing are identified without an explicit search request, would be fantastic as well. Again, Word is unlikely to own this, but this too would be really useful for knowledge workers:-

### **Customer input**

- Site visits in the US and Japan to understand how customers are deploying Word 2000
- Instrumented version of Word 97 and Word 2000 in the US and Japan
- Customer usage and satisfaction survey for Word 97 and Word 2000 in the US and Japan
- Legal Advisory board
- PF Web from PSS
- Voice recognition usability study
- Web authoring survey
- Consulting site visits and summary report
- Usability testing
- Competitive product reviews for Lotus Millennium Edition, Corel Wordperfect 8, Ichitaro, ARA, HMJE 95, and StarOffice.

### **Technical Investments**

- TBD

### **Project Tenets**

- The file format will not change in this release.
- All new features need to be persisted in HTML in a way that doesn't break backwards compatibility
- New features can't rely on a new version of IE

### **Related Documents**

MS/CR 0005785  
CONFIDENTIAL

HIGHLY  
CONFIDENTIAL

- For Word mm0 work see <http://officeweb/specs/word/team/MM0.htm>
- For what's next documents for Word (and there are many) see <http://office10/bin/tabies.asp?docType=whatsleft&sortBy=Team>
- For customer information see <http://wordweb/customerprofile>

### **MM0 possibilities**

Due to how early we shipped relative to Office these have become too numerous to list here. However, you can see a list of them up on <http://officeweb/specs/word/team/MM0.htm>

### **Competitors**

Our traditional competitors continue have been left in the dust by Office 9. However, the threats to our business are from shifts to email and web authoring.

**Office 4.x, 95, 97, and 2000.** Our best competitor. The most recent version has more features than all of the other competitors and a high rate of customer satisfaction. We will need to work extra hard to make sure migration to Word 10 has a very low cost.

**WordPerfect.** They continue to be in financial difficulties with a churning development organization. The threat from them is primarily marketing.

**StarOffice.** They continue to be a threat in Europe. They too are having cash difficulties since the \$20M infusion from IBM from the aborted StarOffice purchase has run out. However, they have a product which is integrated from the ground up and has decent HTML capabilities. Office 9 covers their current advantages however there is a continued nationalism threat that could keep them alive and give them additional money to match us on the development front.

**Ichitaro.** They continue their decline in revenue, lack of product focus, and development organization churn. There are no immediate threats from Ichitaro.

**WordPro.** They have not been able to leverage their success in Notes into WordPro share gains. They will keep trying though.

**ARA Hangu.** They came within a week of going broke this year however due to dillgent efforts on the part of Steve Ballmer and MS we've kept them afloat. Yes, that is dry humor. However, the nationalistic drive that raised the cash did not raise sufficient cash and their development organization has churned. We will need to make sure we produce a very Korean word processor in 10 to maintain this market.

MS/CR 0005786  
 CONFIDENTIAL  
 HIGHLY  
 CONFIDENTIAL

Native Chinese word processors. This is a wild card. No one is making much money here but there are native competitors and it is a big country. We will continue to monitor this country closely.

**MS/CR 0005787**  
**CONFIDENTIAL**  
**HIGHLY**  
**CONFIDENTIAL**

## **DRAFT**

### **Office10 Small Business Tools Vision Statement**

Last Updated: 4/9/99

#### **Charter**

*To increase the value of Office to small businesses by making it easier to improve their customer interactions.*

#### **Vision**

*To make it easy for small businesses to:*

- *collect in a single place all information about current and potential customers*
- *use that information to enhance their communication methods*
- *better track those customer communications and interactions*
- *more effectively run their business*

We will accomplish this by:

- Building tools in and on top of Office that enhance small businesses' customer management capabilities
- Enabling data integration across current applications, including Office, key accounting and LOB applications, and potentially key outside appliances like the phone and fax machine
- Providing the natural connections between Office and small business web sites

#### **Customers**

##### **SORG Characteristics -- (SORGs are NOT just Little LORGs!!)**

- **Small.** (Duh!) The majority of SORGs have less than ten employees. Therefore, scalability and large-scale administrative TCO features aren't highly relevant - at least not in the same ways as they are in LORGs. Efficiency is much more important than growth. This is especially true outside the US. SORGs are much more likely to want to stay the same size in terms of number of employees, but be more profitable in less time.
- **Multiple roles is the norm.** SORG workers, and especially white collar workers and BDM/owners "wear multiple hats". Doing many different jobs every day means they have less time to develop expertise in tools for those jobs and have a higher need for information about how best to do particular roles (e.g. financial, legal, or HR functions). Although few SORGs have dedicated "knowledge workers", because SORG workers wear multiple hats, one of their part-time roles is as knowledge worker.
- **No IT (internal or external).** Vast majority of SORGs have no formal, trained IT employees or good VAR support. This means that there is no training, support or customization help within the business. Instead they rely

**MS/CR 0005788  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**



on the most knowledgeable employee, the owner, or some cheap help like a high schooler working after-hours to muddle through. The only exception is for installing networks, but less than half of SORGs have networks.

- **Want Out-Of-Box-Solutions (OOPS).** Because workers are not technically savvy, are not well trained, do not have access to good customization support, and do not have the time to learn or create their own solutions, they typically live with whatever solution they get out of the box.
- **Business applications are key.** SORGs depend on accounting applications or vertical applications, often built around an accounting core. Critical business data resides in these applications, not in Office.
- **Externally focused.** The rule of thumb is that SORG communications are the inverse of LORGs: 80% is external and only 20% is internal. The graphic representation of this is the commonly heard response to questions about usefulness of PCs for internal communications: "You don't understand the scale of my business. When I want to talk to [co-worker] I just shout!" By contrast, keeping in touch with customers, partners, and suppliers is a much more challenging task for SORGs.
- **High usage of externally-focused, non-traditional computers.** Fax machines, cash registers, phones, etc are very common in SORGs. These machines are critical to running the business, yet with very few exceptions are never connected to an Office pc. Just as LORGs live and die by email, SORGs today live and die by Fax, typically using physical, dedicated Fax machines.
- **Business critical information difficult to get at.** Paper as a storage mechanism is very common. SORGs typically have no centralized or specialized computer system to track documents and critical business information. They are faced with keeping a handle on the "paper beast".
- **Internet, Not PC Net:** SORGs are mostly (soon universally) connected to the Internet, even if only through simple unshared dial-up access to AOL (ISP for 47% of SORGs in US). However, only about a third of SORGs have LANs installed and, even on these, the most common server NOS is still Windows 95. Increasingly, business-business communication is as important as business-customer communication. Usage currently mainly browsing. SORGs dream of e-commerce, but no concrete plans, except where driven by larger customer or supplier needs.

**IT I am a(n) \_\_\_\_\_ what can you do for me? (two sentences, please)**

End user	Make is simple to share information about customers with other employees in your business, and simple to access information about your customers and their projects.
Influential End-user	In addition to what is provided for end users above, provide tools to customize for your specific needs and the needs of your business, while also providing some cool new productivity features supporting customer management.
IT	Nothing – you don't exist in small businesses. Actually, this is an overstatement. While most small businesses, don't have anyone functioning in an IT role, some do and others use VAPs as their IT department. Therefore, we will ensure that the solutions we build are customizable to meet individual business' needs.
BDM	Provide a set of tools that allow you to distinguish your small business from your competition by providing superior customer

MS/CR 0005789  
CONFIDENTIAL

HIGHLY  
CONFIDENTIAL

service, and increase your business health by making your employees more efficient and effective.

CIO

See IT above.

## Functionality Goals

The functionality goals for the SBT team fall into four, prioritized buckets:

- Nailing customer management
- Leveraging Office as value add for small businesses using MSN and other hosting services
- Ensuring Office tools meet the needs of small businesses
- Extending Business Management offerings

Because many of these goals rely on work being done in the NRO timeframe, both from Office and other teams, we are considering options for delivering some of the functionality via the web on NRO street-date, which implies staggering our dev milestones from those of Office.

### Nail Customer Management

Customer management is critical to survival for small businesses. Interacting and communicating with customers and managing those communications is how small business employees spend most of their time. Good customer management is one way that a small business can differentiate itself from its competition. Increasingly, small businesses are facing fierce competition, whether from large companies (e.g., local book stores vs Barnes & Noble, Borders, etc) or from competitors offering competing products or services via the internet (e.g., the Lake Union boat part/repair shop losing business to a small business in Florida promoting itself via the internet).

In Office 2000, SBT provided Small Business Customer Manager. It provides a shared, integrated datastore combining contact information from Outlook with customer related information from small business accounting apps. SBCM then provides different views of that customer information (Top Customers, Customers by Time Since Last Order, etc), and provides "one-click" solutions for automating common customer-related actions - sending a Thank You letter to customers (automates Word mail merge), sending a "We've Moved" postcard to all customers (automates Publisher mail merge), etc.

With NRO, we need to complete the work we started with SBCM to enabling small businesses to more efficiently manage their customers, provide better customer service and increase the number of customers. We need to automatically or easily capture all customer communication and customer related information and expose that information in appropriate ways.

Obviously, this means rearchitecting SBCM to leverage the new features of NRO. Doing this right means being tightly integrated with Outlook, including using Outlook's schema, storage and forms designer.

Specifically, the work we are planning on includes:

- Extend Outlook Contacts to support basic tracking in a shared context (meeting notes, phone calls, etc.)
- Extend Outlook Contacts to support customer management functions (last order placed, prospective customer status, etc.)

MS/CR 0005790  
CONFIDENTIAL

HIGHLY  
CONFIDENTIAL

- Host SBCM views inside Outlook. No more SBCM standalone shell?
- Associate all documents created with Office with the relevant customer(s). This is done today if you initiate the document creation from within SBCM or Outlook Contacts folder. We need to generalize to that regardless of where the user starts, we track.
- Improve Data Exchange with Accounting Applications. We have filters for all key small business accounting apps that let us import data from their stores. We need to improve the exchange to use a standardized format (XML) and to enable real-time syncing of data. Pushing data into accounting data stores is a possibility, but a lower priority.
- Possibly provide tools that integrate Office with the phone and fax – two key non-pc devices in use in small businesses. Ideas include displaying customer record in Outlook when receive a phone call from that customer; integrate better fax services, possibly services provided via the web.
- Ensure customer data from the web can be integrated with Outlook. This includes visitor information, mail from email links in the businesses web site, request for information/brochure, even orders – any customer related communication.
- Provide solutions for better customer communication and interaction:
  - Prospecting rules wizard
  - Support job ticket tracking
  - Customer satisfaction follow-up wizard
- Provide customer views on Business Today

#### **Leveraging Office as value add for small businesses using MSN and other hosting services**

Many companies are wooing small businesses today. Most are doing it with services offered through the web. Small businesses are no longer have to buy packaged monolithic productivity apps to install on various machines in order to get their job done. Increasingly, services on the web are providing the key tools they need. And, in many cases, those services are free. Whether it's email, an outsourced network, a tool for writing a business plan or finding a loan, even accounting, small businesses are looking to these task specific services to accomplish their work.

We need to ensure that there is still a reason to have Office on the client when using these services. Examples include working with Hydrogen to connect the e-commerce data with Outlook and small business accounting apps, MerchantPoint to ensure that the minimal customer management services they provide integrate well with Outlook and the rest of Office.

The SBT team will work with MSN and the Exchange hosting group to ensure that there is value add for the Office customer when using these services. Exactly which team provides which tools and templates is still tbd.

#### **Ensuring Office tools meet the needs of small businesses**

Two years ago, small businesses didn't really know what to do about the internet, many certainly didn't have the time to create their own sites, and most didn't even see the value it could bring to their business. This is certainly no longer true. Whether to create a simple web presence, provide an e-commerce site, use the web to interact with customers or as an outsourced network for internal communications, "getting on the web" is becoming a major focus of small businesses today.

The SBT team will work with other teams including Office Web Server, MSN and the Exchange hosting group to ensure that there are small business specific tools to:

**MS/CR 0005791  
CONFIDENTIAL**

**HIGHLY  
CONFIDENTIAL**

- Create a web presence site. Extend the general web site wizards to hook to specific ISP's offering small business services, customize the web site creation wizard to include information we already know about the business (including name, address, type of business, products/services, etc).
- Extend the site to include e-commerce. Easily create an e-commerce site based on the product/service information we know about.
- Allow small businesses to interact with customers for simple communication, product and service inquiries and project tracking.
- Enable the information gathered via a web site to be integrated with the Outlook datastore where customer information is stored.
- Allow small businesses to communicate internally and collaborate on internal documents.

Exactly which team provides these tools and templates is still tbd.

#### **Extend the Business Management offerings**

Although customer related activities account for more than 50% of small businesses attention, business management is also key. Office 2000 provides two Small Business Tools – Small Business Financial Manager and Business Planner – that address this need. In NRO, SBT will rationalize these offerings with the services provide via the web and the Customer Management tools.

#### **Won'ts**

- Provide a hosting infrastructure. Although hosting is a hot button for small business focus, the Office SBT team will not invest in hosting infracture. We will, however, work with the teams whose charter it is to do this.
- Provide a small business web portal. Like hosting, the charter for this falls to a different group, in this case MSN. We will, however, work with MSN to ensure that the web portal services integrate with Office, and that there is added value for the Office customer when using these services.
- Provide an accounting application. Our expertise is in productivity applications, not accounting applications. While we will continue to work with the small business accounting isv's, we will not provide an accounting application.

<<more to come>>

#### **Scenarios**

<<under construction>>

#### **Demo**

<<under construction>>

#### **Customer input**

<<tons of site visits and several studies; will update with consolidated info >>

**MS/CR 0005792**  
**CONFIDENTIAL**  
**HIGHLY**  
**CONFIDENTIAL**

## **Technical Investments**

## **Project Tenets**

## **Related Documents**

## **Dependencies**

- Magma
  1. Datastore
  2. Query joins with SQL data
- SQL Server
  1. Datastore
  2. Query joins with Magma data
- Outlook
  1. Object model enhancements.
  2. Contact improvements.
  3. Ability to sync the Magma store in a non-Exchange server world.
  4. Ability to host datapage/page from Office designer connected to Magma/SQL store.
- Office Designer
  1. Brain-dead simple form's designer that supports both Magma contact's schema and SQL schemas.
  2. Ability to create pages that integrate data from both Magma and SQL stores.
  3. Logical schema representation for Magma and SQL stores (so I can add "Address" and we know it's a formatted combination of fields)
- Word/shared teams.
  1. Consistent, improved merge across all apps; contact/data everywhere support.
  2. Name/Address/Data Everywhere.
- Publisher.
  1. Exposed automation api's.
  2. Web Site wizard that integrates data we have about company, and can feed data gathered about customers back into Outlook.
- OFX/Latinum.
  1. Shared schema
  2. Standardized data interchange format
- Office
  1. Darwin/Setup
  2. Help
- Accounting ISVs
  1. Filters based on new data interchange format

## ***MMO possibilities***

We started M0 on 4/5/99. M0 has three focus areas: Backend, Frontend, Automation

## ***Competitors***

## ***Assumptions***

**MS/CR 0005793**  
**CONFIDENTIAL**  
**HIGHLY**  
**CONFIDENTIAL**

***Issues***

**MS/CR 0005794  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**



## Worldwide Office 10

Author [chrisor](#)

Created 4/6/99

Updated 4/13/99

### Summary

Office2000 is clearly a quantum leap over past releases in terms of international TCO reduction and general multilingual functionality. Customers love this aspect of Office and can immediately perceive its value to them. However, Office2000 is really only a partial solution and has too many limitations for us to claim that Office *Premium* is worldwide enabled. We also have some work to do to complete our multilingual functionality and we fall short in supporting the complete range of languages that Windows2000 supports, something that some of our largest customers request. The areas for investment in Office10 for worldwide are therefore as follows:

### Finish the job

- **Unicode support for Outlook, FrontPage, and PhotoDraw** - These applications don't use Unicode to store text, creating problems in multi-lingual usage scenarios within the apps, and in exchanging information between apps.
- **Pluggable UI for Publisher and PhotoDraw** - Publisher and PhotoDraw are part of the "Premium" SKU we're targeting at large organizations, but are missing the features we designed to help LORGS deploy Office. They should adopt a single exe with a Pluggable UI to ease corporate deployment.
- **BIDI/Complex Script support in FrontPage, Publisher, and PhotoDraw**. In Office2000 we shipped products that could not support the local language, even though we label the suite as supporting them.
- **One exe for all apps** - Integrate Thai/Vietnamese/Indic code for all apps so there are no longer two builds.
- **True worldwide single exe** - Remove "exemode" for Excel and Access so that users don't have to choose between features and compatibility. These apps can be rewritten as was done for other apps to allow full support in all language versions, but full compatibility only with a particular legacy version. Essentially, reduce code dependency on exemode where possible, and use it only as a compatibility flag.
- **Darwin MSIs use Unicode** - Currently the MSIs are authored in a particular code pages, which makes intl deployment difficult (some of our content does not run unless localized names are used). For Office10, all MSIs should use UTF-8 as the encoding. This almost works today, but the authoring tools/processes need to be modified.
- **Help refers to the active UI terms, not UI terms in the language of Help**. In Office2000 we had to cut this to the dismay of several large accounts. In Office10, the Help will reference the UI terms in the language of the current UI, not the language of Help. A few lines of script allow each Help file to modify the UI terms referenced to match what is being displayed.
- **More complete globalization/pluggability of components**. Clean up remaining items that do not plug UI, or are not globalized. Reduce the list of exceptions to our worldwide story. Excel add-ins, PPT org chart, reg entries, etc.
- **English Mini-LPK supported on all localized builds**. Currently accounts that deploy localized versions have no way to get plug UI to English.

### Consistent

MS/CR 0005795  
CONFIDENTIAL

HIGHLY  
CONFIDENTIAL

- **Global IME support in all apps** – all applications should support the installable "Global IME" in all edit controls to allow Asian typing. This API will be used to provide speech support for non-crucial areas, so it is necessary.
- **Language auto-detection in Publisher, PowerPoint, FrontPage, Outlook** - Leverage Word2000's feature to help users of these other language-aware apps.
- **IME2000-level IMM support in all apps** - Word2000 is ahead of the other apps in this area, and we should bring them all up to a consistent level of IME integration.
- **A4/Letter paper resizing for Excel/Access** – common request from LOGs – want this feature in all apps.
- **Asian Typography** – extend support in Word to other applications, including HTML I/O (Ruby/horizontal in vertical)

## Complete

- **Coverage for all Win 2000 Languages** - We should use Uniscribe to add support for the rest of the scripts covered in Win2000.
- **Unicode 3.0 and surrogate pair support** – All Asian governments (esp. Hong Kong, Taiwan) are now pushing for increased coverage of Chinese characters, beyond what Unicode 2.1 handles. New characters are being added as double-word (4-byte) values called surrogate pairs. Win2000 has received approval/orders to add support for these characters post Beta3, and Office10 should do this across the board. Approximately 40,000 new Asian characters will be defined in Unicode by Office 10 code complete, and IMEs/fonts will also be available.
- **Old Hangul support** – via Uniscribe. Handle the 1mil+ old Korean characters via combining Jamos.

## Appropriate

- **Collaboration in Asia (Ringl)** – Asian companies have particular ways of collaborating, and many US collaboration scenarios do not directly apply due to the nature of office layout. We can significantly impact the Asian market since they are just now making investments to change business practices over to electronic methods
- **Seamless IME usage (Word)** – as part of the speech effort, Word will adopt the new input interface from IIT/Cicero which allows modeless typing with IMEs. This will be a test bed for eventual adoption of the new interface in the system and other applications.
- **Drawing canvas (Word, other apps as necessary)** – Asian users, and indeed all users, feel that drawing tools in the apps are powerful but often don't work the way they expect – they're too hard to control. The drawing canvas will re-introduce the idea of an optional blank rectangle to draw on – a container that holds all the elements of a drawing, making it easier to develop diagrams. Having the drawing elements in this controllable rectangle will make drawing easier, as well as allow things like connectors in Word – something that today forces us to ship Draw2000 in the JPN Word box.
- **Improved localization** – continue to research improved UI terms to help with international usability in localized versions. Currently terms are decided by vendors who may not pick a term that best describes the feature. Product planning will drive the collection of new UI terms in Office10 and work with subsidiaries to get appropriate terms selected.

## Compelling

- **Speech recognition** – Japanese, English, and (possibly) Tier 1 European languages and Mandarin Chinese. The English design will be worldwide so that it handles all languages well. Only Japanese and English are guaranteed at this time.
- **Handwriting Recognition/Ink Integration with WinCE and tablets.** Leverage MS Handwriting recognition for English, Japanese, Chinese and other languages to enable several

MS/CR 0005796  
CONFIDENTIAL

HIGHLY  
CONFIDENTIAL



scenarios. A) Note taking can use electronic Ink. Import notes into Word and other apps as Ink. Recognize text later, convert diagrams to drawings (if desired). B) Direct handwriting support in Word via tablet. C) Annotation/Collaboration scenarios with Ink on Word documents, possibly with Pocket Word.

- **Machine Translation Integration** – Current technology now has business value. Although far from perfect, it works well enough to tell the reader what a document is about, and empowers them to take action on it. Many large customers and OAC members tell us they want us to provide the ability to easily integrate machine translation into Word, Outlook, IE and other apps to help in international collaboration and information sharing. They are fully aware of the current level of technology, and that is sufficient for them. For a preview of what works today, try <http://babelfish.altavista.com>. Lotus has also announced they intend to integrate MT into Notes for the same reasons. See the announcement. Scenarios include browsing the web in "your" language (auto-translate while browsing, with optional "View Original"), an aid to authoring in your second language (frequently English), a way to read foreign language mail/docs sent to you (perhaps via an alias you are a member of, or because your company operates in English – a foreign language for you). MT engines will not be developed in MS, but licensed as we do with proofing tools.
- **International web server** - HTTP provides information about the language of web pages and the browsers have information on the language of clients. One compelling web scenario is to have a web server that can serve up pages in the language of the user. This can be combined with optional machine translation to allow surfing of international web pages. In addition, there is some work needed to make web use worldwide for Office users. This is an area where FrontPage/Server extensions/IE/MS can work to improve key international scenarios.
- **Fit on X pages (Word)** – Asian users often start a document with the goal of having it fit on one page. Because this is a constant challenge in an app like Word, customers tell us they use PowerPoint or other applications for documents. We should make it easier to force documents to fit on a page. We can leverage the existing Shrink to Fit feature and add value for all users at the same time (students, etc.). This has always been a top WP feature.

## Process/Internals

- **Simultaneous RTM of US, JPN, GER, and ARA** – And all Redmond languages to RTM within 4 weeks of US. We managed to make a single exe in Office2000. In Office10, we will be able to release localized setup as well as core code at the same time. After the simultaneous RTM, the only work remaining is creating and verifying media. Other localized versions will have compressed deltas from initial RTM, since the remaining Redmond languages should be about ready to RTM anyway. This provides process benefits to improve our quality, and ability to turnaround for the next project. It provides market benefits to create synergy with English office and get localized version revenue earlier. Finally, it is what our customers want – many LOGs wait until all the languages they need are RTM'd before rolling out Office anyway.
- **Beta 2 in Redmond and Ireland languages** – Our marketing betas need to include additional European languages – English, German and Greek are not enough. As with Win2000 betas, Office10 betas (at least beta 2) should include Spanish, French, and Italian as additional beta languages (adding to English, German, Greek, Japanese, Korean, Traditional Chinese, Simplified Chinese, Arabic, Hebrew).
- **Auto-resizing dialog layout** – PowerPoint has the ability to layout its dialogs on the fly. This helps reduce cost in localization, as well as reduce bugs dealing with cosmetics. These kinds of bugs are prevalent at the end of the project, and contribute to the difficulty in delivering final localized versions. For Office10, this code will live in MSO, and can work with SDM dialogs. We should do the work to integrate this across Office to make simultaneous shipping easier.

MS/CR 0005797  
CONFIDENTIAL

HIGHLY  
CONFIDENTIAL

## Customers

### LORGS/IT

Office-wide support for Plug UI, single exe, same set of languages in all apps makes rollout and deployment a no-brainer for all of Office. No gotchas or exceptions.  
Simultaneous RTM means localized versions available within weeks of US, not months, and roll out is sooner, smoother.

### BDM

Intl web server allows easy data gathering across regions, publishing info  
Machine Translation aids productivity in multinational LORGS  
Asia: Office Collab works well in Asian scenarios.

### Influential End User

Speech in my language  
Machine Translation aids working in English (or other), browsing web in English (or other)

### End User

Asia: better IME experience  
All: better drawing (Word)  
All: fit docs to one page (Word)  
Apps easier to use due to improved localization

## Scenarios:

Multinational LORG wants to roll out Office with Multilanguage Pack. With Office2000, they need to remove Publisher and PhotoDraw since these apps are either not Unicode or don't do plug UI. With Office10, all concerns about per-app limitations are removed, and they can deploy the full Office Premium to their desktops.

US Gov't or other multilingual account needs to work in multiple languages in Office apps. Currently they cannot use the full set of languages in all apps. Some apps don't handle Complex Scripts; others force an either/or situation. With Office10, these accounts (US Gov't, UN, World Bank, Mormon Church, Credit Suisse, Volvo, etc) are able to use all languages in each app, particularly Excel, Access, and Outlook. They are also able to use many uncommon languages, which they need for their work in developing countries (e.g., Armenian, Hindi, other Unicode-only langs). Taiwan government able to discard Unix proprietary systems now needed for 50,000+ character support. Hong Kong users can support all their names and addresses. Japan/Korea/China get support for uncommon name characters.

LORG uses Outlook for international communication, and maintaining contact info on customers. With Office10, data is not trashed moving from machine to machine, user to user.

French user tries to collect information from WWW or corpnet on corporate marketing plans in various regions. Search auto-translates her French query terms into 14 different languages, and hits are returned on pages in many different langs. User views the pages, and they are auto-converted to French from the ubiquitous English and some other langs. User sifts out the unrelated pages by reading the French translations, and for pages with directly applicable material, she uses the French

translation as an aid while reviewing the English source page for specific info. User also receives English-language mail on several corporate mail aliases she is a member of. She auto-translates these to figure out what they are about, and deletes those that are of no interest. For others, she uses the translation as a tool to read the English original. Same for documents and presentations.

## **App-specific Intl Plans:**

Access:

Excel

FrontPage:

Outlook

PhotoDraw:

PowerPoint: <http://gou/personal/DAVEKES/pp10intlfeatures.htm>

Publisher

Word

**MS/CR 0005799  
CONFIDENTIAL  
HIGHLY  
CONFIDENTIAL**