

# Integrated Office Proposal

5 August, 1993

## Key Characteristics of Integrated Office

### - Combines world-class applications

- Product-specific features continue to evolve
- Sim-ship with stand alone products

### - Feels like one application

- Strong unifying metaphors
- Cross-app activities are as easy as in-app activities
- Unifies activities that are common across document types
- High performance and efficient working set

Note: As we move to an object centric user model, where applications are less visible, we might say Integrated Office "feels like no applications". i.e. It feels like you are working directly on your documents or other objects.

### - Extensible and configurable

- Components can be added, removed or replaced
- Easy to use tools for building and customizing UI
- Programmable

- access to information  
- on lib  
- ACW  
- P. J. ...  
- RT  
- WAF

MSA  
Product  
Controls

asked to leave  
cost staff

become  
miss

## Market Trends

- Dramatic increase in demand for "Office Suite" type products
- Consistency and integration are becoming important software purchase criteria

### Leverage our control of platform technology

- Allow our apps to anticipate the coming "left turn" of platform
- Repeat advantage we gained from bet on Windows

### High development cost increase entry barrier for competitors

### Integrated Office vs Previous Integrated Applications

- Components applications will be full featured (Word, Excel, etc.)
- Integrated Office will be extensible
- Components applications already have partially unified UI: fewer artificial compromises

## Key User Activities in Integrated Office

### Working with documents

- Finding, creating, using, modifying, printing, sending, saving documents
- Working with compound documents
- Working with groups of documents

### Working with data

- Activities: Finding, viewing, managing, collecting, transforming, analyzing, reporting, specialized apps
- Examples: Project, Access, Fox, RPM, most MIS apps, EIS systems, accounting systems, etc.
- These are typically non-document oriented applications

### Working with other users

- Mail, BBS, routing, doc library, collaboration
- Most of these are not product-specific

### Customizing the system

- Adjusting UI
- Adding/removing capabilities
- Automating tasks
- Building new systems with Office components
- May be done by: end user, MIS, ISVs or Microsoft at appropriate levels, of sophistication

### Learning

- Consistency, help, cue cards, help channel, tip wizard

## Benefits to End Users

- Synergy between applications

MS 0150226  
CONFIDENTIAL

MICROSOFT CONFIDENTIAL

08/05/93 01:42 PM

Plaintiff's Exhibit

5591

Comes V. Microsoft

Easier to work with mixed document types  
 Products more closely tailored to their needs  
 by themselves  
 by their MIS departments  
 by ISVs  
 by Microsoft

Freedom to choose appropriate apps  
 high vs low end  
 specialized apps

Reduced working sets.  
 Higher performance for mixed app activities  
 Functional synergy between components

Components of Office can be used together to accomplish things not other otherwise possible.  
 Hypothetical example: Using Excel recalc engine to compute values in a Word table, or to compute constraints between drawing objects.

Easier installation and upgrading

**Benefits to MIS & System Integrators**

Platform for vertical & MIS applications  
 Easier to create solutions targeted at exact requirements  
 Office as run-time for custom solutions  
 Easier to maintain  
 More robust  
 Building blocks for down sizing  
 Opportunity for 3rd parties to offer add-ins, vertical products or services  
 Security  
 Real time processing  
 Business Opportunity  
 Reduced training costs

**Benefits to Microsoft**

More customers will choose buy the office suite  
 Lock in users, lock out competition  
 Competitive advantages  
 Lotus: does not have a development tools business  
 Borland: incomplete family of apps  
 Word Perfect: incomplete family of apps, no development tools  
 More features possible through extensibility  
 Development efficiency  
 Opportunity for industry-specific solutions  
 Steady revenue flow through annual licenses  
 Keep prices up and increase market share  
 Changes in user model, high level of integration and new shared functionality could allow Integrated Office apps to be positioned as a new class of software, and thus transcend the feature and price wars.

**Integrated Office Contents**

Standard Edition  
 Word processor (Word)  
 Spreadsheet (Excel)  
 Presentation (Powerpoint)  
 Drawing  
 Charting  
 List manager (low end database)  
 PIM (Ren)  
 Mail (Included in Ren)  
 ISAM? Query Tool? Document library?  
 "Professional Edition" adds

**MS 0150227  
 CONFIDENTIAL**

- Access
- OB development environment
- Library of custom controls
- Other add-ins
- OFS (as server edition?)

Possible Server Edition  
Adds EMS or OFS

**Office is also synergistic with**

- Shell
- Document routing
- Project management
- Smart office equipment
- Hand-held computers
- Notes?

**User Interface**

UI is document-centric or object-centric

- Create new documents by replicating templates or "stationery"
- Apps less visible. User focus is on working directly with documents or other objects.
- Commands and tools come and go as needed based on selection
- Also need to incorporate non-document oriented tools

The Shell is the overall container in which Office runs

- This assumes that we can design the shell for excellent support of application requirements
- Would make extensive use of shell extensibility to support Office scenarios.
- Explorer extensions to work with
  - object oriented stores (Ren)
  - document stores (Word's Doc Lib)
  - structured stores (DDT's navigator)
- Possibly tray extensions for support of cross app scenarios
- In SDI, some app functionality would be shared across windows. This may shared Office palettes or toolbars which we may want to incorporate into the system.

Workbooks will be a key type of document

- UI metaphor for accessing components of documents
- Act as a container of other documents

Great support of browsing and querying for documents, objects, and information.

- Chicago Explorer, Cairo Explorer, Ren, Navigator, Word doc library, Excel information finders
- Need to rationalize and unify these designs where possible.

Simplify UI and reduce number of concepts. Candidates for unification/simplification:

- Charting
- Drawing
- Tool bar/tool bar customization
- Options/Settings
- Proofing tools
- General paper model
- Print Preview
- Mail
- Document Library
- Property sheets e.g. Font, borders
- Styles
- Print layout, headers, footers, etc.
- Workbooks
- Outlining
- Tables/spreadsheet grid
- Annotation
- Naming
- UI customization
- Add-in management
- Programmability

MS 0150228  
CONFIDENTIAL

(Note: UI unification could also imply code sharing)

## Development Strategy

### Key APIs to Standardize

#### OLE 3

- Layout negotiation
- Transparent object support
- Irregular object support
- Search and spell
- Ambient properties

App documents as component controls

App documents as forms (containers for controls)

Standard interfaces to storage and stored objects

- Access to storage
- Reconciliation
- Context indexing

Standard app object models, programmability interfaces and events

Standard interfaces between addins and host apps

Note: These interfaces should be compatible with the component object model.

### SDI

### Shared Components

### Development Process

## Platforms

### Chicago

May include development of a unified shell for Chicago. See separate paper on Unified Shell Proposal.

### Cairo

As above, plus leveraging of OFS, smart folders, security, summary catalogs.

### Mac

Supported with enhanced portability layer. Some features may not be available.

### Win 3.1?

Assume not supported, or more limited feature set.

Most users who would upgrade Office would also upgrade to Chicago.

## Integrated Office Process

Confirm vision and fill in details

User visits and ABP

Write spec and develop prototype

Interop, Desktop product groups, Res group, Cairo, Chicago, DDT

Provide input to Chicago, Cairo, OB and forms effort

Any feature should have exactly one owner

Design & development tradeoffs need to be able to be made at a level below senior VP

Need to maintain the "hallway effect", at least for feature teams.

Should consider using better workgroup software

MS 0150229  
CONFIDENTIAL