

Programmability

Software Administration

Urawing

Office Explorer

FL AG 0010974  
CONFIDENTIAL

MX 1189906  
CONFIDENTIAL

## Office Explorer

Lead Office Program Manager:	Richard Wolf (RWolf)
Lead Office Developer:	Navneet Paliwal (NavPal)

[\* Working very closely with the Ren team. Bill Bliss (BillBI) is the Group Program Manager for Ren.]

- Explorer Goals
  - Goals for the Explorer
    - Spec: Draft vision statement
    - Owner: RWolf (Office)
    - File: \wkgroup\shell\expgoals.doc
- Table view on FAT (and OFS) - "File Module Table View"
  - Delta from Ren spec covering table view on MAPI. Deal with issue of view persistence, choice of available properties.
    - Spec: No spec; Draft due by 11/21
    - Owner: WJones (Office)
    - Sharing: all Office (Ren and Doc Mgmt) code
- Chicago superset/replacement
  - How and to what degree the Office Explorer supersedes the Chicago Explorer/Folders and how it replaces them. How the Explorer reuses Chicago code.
    - Spec: No spec; Draft due by 11/14
    - Owner: VinodA (Office)
    - Sharing: Office (Ren/DocMgmt) code
- Folder views
  - How the Office Explorer supports folders. Covers how the Explorer replaces the Chicago folder and how the various pieces of functionality are provided. Whether and how we write this spec depends upon the above spec.
    - Spec: Depends on Chicago approach above
    - Sharing: Office (Ren/DocMgmt) code
- Indexing/Caching document properties
  - Underlying support for rich views on FAT. The functionality and user spec for this comes from the above spec "Table view on FAT." This is a significant piece of work that needs to be designed and planned.
    - Work plan: Nov 21
    - Owner: SteveBr (Office)
    - Sharing: Office code (Doc Mgmt)
- Name space architectural overview
  - How the Explorer supports the back end stores - MAPI, FAT, OFS
    - Spec: Nov 14
    - Owner: DonGa (Ren)
    - Sharing: Office (Ren/DocMgmt) code
- Finding documents in the Office Explorer
  - How querying fits into Ren UI.
    - Spec: Draft due 11/21
    - Owner: WJones (Office)
    - Sharing: Office (Ren/DocMgmt) code
    - File: \wkgroup\docmgmt\find96.doc
- Other views on FAT
  - Other views (calendar, timeline, etc.) not nearly as important for FAT as table.
    - Spec: No spec yet. RWolf plan by 11/21

FL AG 0010975  
CONFIDENTIAL

MX 1189907  
CONFIDENTIAL

- Extensibility  
Namespaces, controls, IDispatch
  - Spec: No spec yet. RWolf plan by 11/21
- Document management
  - The Office librarian  
Versioning documents, document checkin/out, document auditing and how this functionality fits into the Explorer.
    - Spec: Stable
    - Owner: WJones (Office)
    - Sharing: mostly Office (DocMgmt) code, some apps work
    - File: \wkgroup\docmgmt\library.doc
  - Content indexing MAPI  
How doc mgmt group can provide content indexing for MAPI
    - Spec: Plan by 11/21
    - Owner: Wjones (Office)
    - Sharing: Office (DocMgmt) code
  - Content indexing improvements  
Improvements to underlying content indexing functionality
    - Spec: Plan by 11/21
    - Owner: Wjones (Office)
    - Sharing: Mostly Office (DocMgmt) code, some apps work
- File Open  
How File Open works in '96, what code is used. To what extent does File Open unify the name spaces.
  - Spec: Draft by 11/21
  - Owner: Wjones (Office)
  - Sharing: Office (DocMgmt) code
- Document Routing Integration with Ren  
Routing functionality comes from Ren/Exchange, apps must integrate
  - Spec: Plan by 11/21
  - Owner: MikeAng
  - Sharing: Mostly Office (Ren), possibly some apps work
- OLE Properties  
What part of our property work for '96 is needed to support Project X
  - Spec: already exists, needs prioritization, proposal by 11/28
  - Owner: HeikkiK
  - Sharing: Office, possibly some apps

FL AG 0010976  
CONFIDENTIAL

MX 1189908  
CONFIDENTIAL

Microsoft Confidential

## Office Explorer Goals

11/1/94

### Goals

I would like to get some up front discussion and buy in concerning the goals for the Office Explorer. This paper enumerates possible goals that the Explorer could have. I have rank ordered the goals according to my personal preference to provide a stake in the ground, but I want to stimulate feedback on the rank ordering rather than shut it down.

Here is the rank ordering as a summary, ahead of the discussion of each item:

1. Rich views on FAT and OFS
2. Single app for documents and PIM
3. Superset and replace Win 95 Explorer
4. Extensibility
5. New Explorer UI

Naturally, I have provided more details for the goals that are at the head of the list, as they are the ones I understand the most. As we reorder the priorities I will provide detail on the most important goals.

### *Rich views on FAT and OFS*

Office has done lots of good work in '95 so that documents publish their standard and custom properties in docfile streams. We have a standard functionality and user interface across Word, Excel, and PowerPoint that allows the user to define custom properties and assign values to those properties. Additionally, Word, Excel, and PowerPoint allow the user to bind custom properties to native document content, bookmarks in the Word case and range names in the Excel case, enabling many important user scenarios.

The only problem with this property promotion work is that it is exposed exclusively when the user is connected to a MAPI store, either the EMS or the PST (the local MAPI store). The user can employ the Capone viewer to browse rich views that are enabled by the combination of property promotion, the MAPI store, and the Capone viewer. The sole advantage that a user gleans from property promotion in the standard Chicago Explorer looking on a FAT file system is that the user can invoke a property sheet that views the properties. This feature lets the user look at the properties of each document one by one, rather than aggregated in a view, so its appeal is extremely limited.

FAT will still be the most popular store in the Office '96 time frame. Providing rich views on FAT makes this feature available to the average user without requiring that they use an esoteric "groupware" store. Of course, FAT does not provide additional important features like replication and security, which the user obtains with an EMS server.

This feature undercuts Notes by providing a large part of the value of Notes for free with Office and will be a potent threat against Notes.

OFS will probably be available on NT in the Office 96 time frame and we should support it as well.

### **Both FAT and OFS**

Here is the additional functionality beyond the standard Explorer functionality, such as large and small icon views, that we should support on both FAT and OFS. This functionality is currently available using the Capone or Ren viewer against an EMS or PST store.

FL AG 0010977  
CONFIDENTIAL

MX 1189909  
CONFIDENTIAL

***Additional column(s) bound to a property***

Add columns to the table view and bind each column to a property, so the property is displayed in the column.

***Sort on arbitrary column***

Sort on a column in both ascending and descending order.

***Group by arbitrary property***

Group by a property.

***Maintain list of properties available in the current folder and select property for column, sort, group by from list***

Adding columns, sorting, and group by require that the user name the property they want to add, sort, group by.

The Explorer should provide a list of the properties that are available for the current folder, so the user can pick the property from a list rather than remembering and typing it. The list of properties available for the current folder is composed of the union of the properties that are available on documents stored within the folder.

***Property promotion from docfile property stream for both standard and custom properties on drag/drop into Explorer***

When the user drags/drops a document into the Explorer, promote the properties from the docfile into native indexed structures maintained by the Explorer.

***View persistence***

Maintain a set of saved views associated with each folder, on both a shared and private basis.

First, this means we have the ability to save views. Second, we should associate a set of saved views with each folder, so users who go that folder can see the convenient views that have already been defined. Third, these views should be shared, so one user can create useful views for many other users, and private, so a user can create their own views that are unique to their needs and are not exposed to others.

***Ship with standard set of views***

Office apps support a standard set of document properties, so we can ship a set of views that are preconfigured to support the standard properties. For example, we can ship a view that groups documents by author.

**FAT**

FAT does not natively provide the requisite support for rich views. Here is the list of additional support required. The Office Document Management group will provide this support.

***Index maintained so performance for sort, group by is fast***

Without specialized support the viewer would have to open every document in a folder, extract the properties from the docfile stream, and then perform the required sort(s). Maintaining an index allows us

FL AG 0010978  
CONFIDENTIAL

MX 1189910  
CONFIDENTIAL

to achieve acceptable performance. This is the technique that "object stores" such as OFS, Notes, and EMS employ.

### ***Maintain list of properties on folder basis***

Support this feature by maintaining the list of properties for each folder.

### ***View persistence on folder basis***

Support this feature by providing saved views, associated with a folder, on a shared and private basis.

### **OFS**

OFS will probably be available on NT around the time Office 96 ships. It provides many of the benefits of EMS built into NT. We should support the same functionality on OFS as on FAT.

Not sure about OFS vs. EMS trade offs. Systems will have to sort this out.

### ***Index***

OFS maintains an index, so we do not have to provide an index. However, it would be desirable to make the OFS index and the document management supplied indexing for FAT appear the same to the viewer.

### ***List of properties***

Not sure what OFS does here. I suspect nothing. There probably was some work on this as part of the CDE, but I don't know whether it will be included in the OFS that ships with NT.

### ***View persistence***

Not sure what OFS does here. I suspect nothing. There probably was some work on this as part of the CDE, but I don't know whether it will be included in the OFS that ships with NT.

### ***Single app for viewing documents and personal information***

The Office Explorer provides a single consistent locus for the user to find and manipulate their information regardless of its type. The Explorer deals with documents, appointments, tasks, and mail in a single place in a consistent manner.

The single locus is a single app window - Ren. The Explorer provides this single locus via a new module within Ren - the "Document module" (it might actually just be a favorite place).

Providing a single locus does not mean that the Explorer provides a single hierarchy for all types of information. In fact, the document and MAPI based hierarchies are disjoint. A single locus simply means that the views on the various types of data - documents, appointments, tasks, and mail - multiplex the same app window.

In connection with this goal it would be desirable to use the Ren view code for documents (FAT/OFS). There are two reasons for this. First, the code already handles rich views. Second, since this code will be loaded and in the working set for MAPI data, it would be more economical in terms of resource consumption to use the same code.

Note that this is a secondary goal. We could achieve the goal of having a single app by using non-Ren view code but having it appear in the Ren app window as the "document module". Conversely, if desired we could also use the Ren view code but have it appear in two separate apps - a PIM and a document Explorer.

FL AG 0010979  
CONFIDENTIAL

MX 1189911  
CONFIDENTIAL

## **Superset and Replace Win 95 Explorer**

This goal has three parts: superset the Win 95 Explorer, replace the Explorer, and replace Folders.

### **Superset the Explorer**

The functionality I have described so far, rich views and a single app for documents and PIM, adds valuable new functionality to that provided by the Chicago Explorer. However, unless the Office Explorer provides all of the functionality of the native Chicago Explorer, it will force users to choose between the two Explorers. This functionality includes: views (large and small icon), utility functions (backup, compression), specialized folders (print, font), and perhaps other functions. Requiring users to go to one Explorer to view their documents in the desired fashion and another Explorer to back up their files would be highly undesirable and confusing.

### **Replace the Explorer**

The Explorer is invoked from the Chicago shell in several ways. It is an item on the Start menu located on the tray. Second, a right mouse button context menu is provided that allows any folder to be opened in an Explorer view rather than a folder view. If the Office Explorer simply supersedes the Win 95 Explorer but does not replace it, then the user will invoke the standard Win 95 Explorer from the Start menu or the context menu, making it extremely easy and likely for the user to invoke the wrong Explorer.

So, we should replace the Win 95 Explorer so that if the user invokes it they get the Office Explorer.

### **Replace Folders**

Many of the user of Win 95 will rarely or never encounter the Explorer *per se*, but will only use it in the guise of a Folder, a simplified Explorer that lacks a scope pane. If the Office Explorer replaces only the full Explorer but not Folders, many (perhaps most) users will never encounter it or realize its benefits.

Much of the functionality of the Office Explorer, such as its combination with personal information, does not make sense for Folders. That is, allowing a user to double click on any folder and get not only the contents of that folder but also the full blown Explorer interface with access to other types of data via a left most "manager" or "favorite places" pane would excessively complicate the Folder user interface.

Some of the Office Explorer functionality, such as rich views on FAT and OFS, would be valuable for Folders as well as the full blown Explorer. On the other hand, one could make the case that this extended browsing functionality need only be provided by the full blown Explorer. However, if we do that many users would never realize the benefits.

### **Use Chicago code base**

Chicago has written lots of optimized and heavily tested code. Additionally, it has lots of code to handle print folders, font folders, control panels, various views like small icon and large icon, and necessary functionality like backup, compression, and so on. It would be a huge effort for the Office Explorer to write this functionality without using the Chicago code base.

Note that this goal is highly dependent upon the outcome of replacing the Explorer. If the Office Explorer replaces the Chicago Explorer, then this goal is extremely important. If the Office Explorer does not replace the Chicago Explorer, then this goal probably is not important.

FL AG 0010980  
CONFIDENTIAL

### **Extensibility**

An extensible Explorer would solve a variety of problems. DDT wants an Explorer that can view into databases, enumerating tables and other database objects. Document Management would like the Explorer to view and operate on high end document management stores, such as Saros, or on stores that support

emerging document management standards, such as Shamrock. Even the functionality that we are proposing for the Office Explorer could have been done as an extension to a standard Explorer rather than a replacement if suitable extension interfaces had existed.

Several types of extensibility have been proposed.

### **Name space extensibility**

Name space extensibility provides a strategy to allow various storage back ends to plug in to the Explorer results pane. At a minimum this extensibility should span FAT, and OFS.

Additional stores include:

- MAPI - should MAPI be supported by the name space extensibility API or supported by special case.
- Document management functionality - should the name space extensibility mechanism handle advanced document management functionality.
- Database - should the name space extensibility mechanism handle database management functionality.

### **Container for controls**

Another means to provide extensibility is to provide an architecture where the Explorer is a container for controls and any control provider can plug in a suitable control. To some extent an Explorer is naturally constructed in this fashion, but the interfaces do not necessarily have to support the installation of externally supplied controls.

This is a very general purpose extensibility mechanism and can be used to provide name space extensibility by allowing other name space providers to plug in their controls. However, name space extensibility achieved in this manner does not reuse the viewing code.

### **Views as controls**

If Explorer views are implemented as controls then it is possible to use these views in other contexts. This could provide a degree of reusability, for example, use of the Explorer controls in the File Open dialog. It could also enable new scenarios, such as a view as a ply in a binder or a view as an embedding in a document.

### ***New Explorer UI***

Improve the Explorer UI. This is probably a non-goal, as it will make the issue of replacing the standard Chicago Explorer with the Office Explorer more problematic for our customers. Diverge of the Office Explorer from the Chicago Explorer will probably make working with Systems more difficult. For example, providing future versions of the Explorer to Systems or even having Systems track the Office Explorer will be more difficult.

FL AG 0010981  
CONFIDENTIAL

MX 1189913  
CONFIDENTIAL