

Office 96 Spec Competitive: Chicago Explorer Superset and Replacement

Chicago Explorer Superset and Replacement

Program Mgr VinodA ¹

Summary

Office Explorer will leverage the property promotion work done in Office 95, providing users rich views on documents based on the properties that they expose. Office Explorer will superset and replace the Chicago Explorer to become the single place where users can find and manipulate all their information irrespective of its type, including all documents and files, in addition to personal information such as appointments, task lists and mail. By allowing Office users to browse rich views on documents without requiring them to be connected to a groupware store, Office 96 undercuts Lotus Notes, giving away a large part of the Notes functionality for free.

¹ Portions of this spec derived from *Office Explorer Goals*, by Richard Wolf

CONFIDENTIAL

Strategic-1

EXHIBIT KR
EA 15
Notes 3/4/09

MS-PCA 1566798
CONFIDENTIAL

Office '96 Spec. Competitive: Chicago Explorer Superset and Replacement

Justification

Office '95 documents already publish their standard and custom properties in docfile streams. Users can create custom document properties, and even bind custom properties to native document content, such as bookmarks in Word or range names in Excel, enabling many important user scenarios.

The drawback of property promotion today is that it is exposed only when the user is connected to a MAPI store. On the standard Chicago Explorer, however, these properties are visible through property sheets on a per-document basis only, rather than in more useful aggregated views. By providing rich views on FAT (OFS as well if this is available on NT in the Office 96 time frame), Office Explorer makes rich views available to the average user without requiring them to be connected to a groupware store.

By implementing a feature superset of Chicago's Explorer, Office Explorer can replace the Chicago Explorer across the board and be the single place where users can find, view and manipulate, in a consistent way, all their documents, appointments, tasks and mail. This makes Office Explorer inherently more information-centric, it presents users with the all the information they need without them having to worry about its type.

Description

Some terminology:

Term	Definition
Folder Object	An object that appears in the Explorer that is also a container of other objects, e.g., file system folders (such as a FAT directory), the Control Panel folder, the Fonts folder.
Item Object	An object that appears in the Explorer that is not a container for other objects, e.g., a file in a FAT folder, the Keyboard item in the Control Panel folder, the Arial font item in the Fonts folder.
Special Folders	Folder objects that are not file system folders, and which contain very special kinds of item objects. For the purposes of this discussion, the term Special Folders will refer to the following set of Chicago folders: Control Panel, Fonts, Printers, Recycle Bin, Dial-Up Networking and Remote Access. Note that My Computer and Network Neighborhood are not considered Special Folders for the purposes of this discussion, Office Explorer will handle these two folders along the same lines as the FAT folders.
Scope Pane	The pane that displays the hierarchy of folder objects contained in the Explorer.
Contents Pane	The pane that displays the contents of a folder object.
View Control	The control that displays a folder object's contents in the contents pane.
View Modes	The display modes supported by a view control. Chicago's Explorer has 4 view modes: Small Icons, Large Icons, List and Details.
Explorer in Explorer Mode	This refers to an Explorer window with a scope pane.
Explorer in Folder Mode	This refers to an Explorer window without a scope pane.

CONFIDENTIAL

Strategic-2

MS-PCA 1566799

CONFIDENTIAL

Office 96 Spec Competitive: Chicago Explorer Superset and Replacement

There are three main parts to making Office Explorer the browser of choice for rich views on documents:

- a) Implementing rich views on FAT and OFS.
- b) Implementing a "document module" in Ren, which allows the Ren app window to be the single locus for locating, viewing and manipulating all documents, in addition to personal information.
- c) Implementing a feature superset of Chicago's Explorer, to be able to replace the latter across the board when Office 96 installs on Chicago.

This spec is mostly concerned with (c) above, i.e., superset and replace the Chicago Explorer. There are 3 sub parts to this:

1. Implement a feature superset of Chicago's Explorer

By doing rich views and providing a single place for documents and personal information, Office Explorer adds valuable new functionality to that provided by the Chicago Explorer. However, Office Explorer has to provide all of the existing functionality of the Chicago Explorer, otherwise it forces users to choose between the two Explorers.

The existing Chicago Explorer functionality that Office must support include:

1. Network browsing.
2. Browsing special folders.
3. Supporting the Small icons, Large icons and List view modes available in Chicago Explorer's view control.
4. Correct UI elements for file system and special folders, as well as their item objects. These UI elements include context menus, icons, and property sheets, as well as any registered shell extensions for these.
5. Shell compatible cut, copy, paste, delete and drag and drop operations, all of which need to work in the framework of the shell's Undo mechanism. Deleted files need to go to the special Recycle Bin folder.
6. Supporting the base menu and toolbar items in the Chicago Explorer, which are created by the Explorer frame.
7. Supporting the additional menu and toolbar items added by the Chicago Explorer's content pane, and those added by the selected item in that pane. Providing keyboard accelerators for these.
8. Updating views on file system change notifications.

The Office Explorer implementation strategy is to leverage the Chicago shell team's work as much as possible. Chicago provides some of the crucial interfaces that will simplify our work, these include:

IShellFolder This interface, derived from IOleContainer, is the key to doing rich document views in the Office Explorer. It provides an enumeration of the folder and item objects in the Chicago Explorer, including special folders and their items (the Fonts special folder is an exception, its item objects are not enumerated this way). IShellFolder also provides name parsing and all the UI elements (e.g., icons, property pages, context menus) for these objects, handles any registered shell extensions, shell-compatible cut, copy, paste, delete and drag/drop functionality. It provides support for handling the menu items added to

M CONFIDENTIAL

Strategic-3

MS-PCA 1566800

CONFIDENTIAL

~~Office '98 Spec Competitive Chicago Explorer Superset and Replacement~~

Explorer by the selected item object in the content pane, through the IContextMenu interface.

IShellView This interface handles the display of the IShellFolder enumerated objects in contents pane of Chicago's Explorer. Office Explorer will attempt to use IShellView to display the contents of special folders since there is code specific to special folders in Chicago's IShellView - Office Explorer can ensure that it will not lose such functionality by using IShellView. Office Explorer views on non-special folders will not use this interface, since IShellView does not support rich views.

Summary of the Office Explorer Implementation Strategy:

1. Leverage IShellFolder to enumerate all the Chicago Explorer folder, and for their basic functionality, as discussed above. Use this to build the Office Explorer scope pane.
2. Use IShellFolder to enumerate item objects in the above folders. Use this to construct a specialized Office view that adds the all-important rich view mode (in addition to providing Chicago Explorer's Small Icons, Large Icons and List view modes) in the contents pane for all FAT folders.
3. For the contents panes of Chicago's special folders, use IShellView, in addition to IShellFolder. Special folders have special properties and attributes that are implemented in IShellView code, by using IShellView we ensure we do not lose any of this functionality.
4. Implement the base menu and toolbar items created by Chicago Explorer's frame. These include Find Files or Folders, Find Computer, View Options, Map Network Drive and Disconnect Network Drive menu items, and some simple toolbar buttons.
5. Have Chicago export and publish certain APIs that will make our life easier. These APIs include shell Undo, SHChangeNotifyRegister, SHChangeNotifyDeregister, and SHFindComputer. All of these are already implemented in Chicago today, only these are internal and need to be published. Some, like the notification ones, are untested and therefore not robust. Chicago needs to test these APIs more extensively, if published.

NOTE: The table in Appendix A has a more detailed list of all the work items, it includes an implementation approach for each item, as well as any support required of the Chicago shell team.

2. Replace Explorer

The Explorer is invoked from the Chicago shell in a couple of ways. It is an item on the Start menu's Programs item. Also, the right mouse button context menu on a folder includes the "Explore" verb that runs Explorer with that folder opened.

If the Office Explorer simply supersedes the Chicago Explorer but does not replace it, then the user will invoke the standard Chicago Explorer from the Start menu or the context menu, making it extremely easy and likely for the user to invoke the wrong Explorer.

To replace the Chicago Explorer, the following entries under Chicago's registry HKEY_CLASSES_ROOT\Folder\shell\explore branch need to be modified - the "command" entry should be replaced with the path of the Office Explorer executable, and the "ddeexec" sub branch entries need to be suitably modified to ensure that once

CONFIDENTIAL

Strategic-4

MS-PCA 1566801

CONFIDENTIAL

Office '96 Spec Competitive: Chicago Explorer Superset and Replacement

the Office Explorer is already running, it can respond to a DDE message asking for a folder to be explored by simply creating a new thread to handle this request, rather than run another whole new instance of itself.

3. Replace Folders

Many users of Windows 95 will rarely, if ever, encounter the Explorer *per se*, rather they will only use it in the guise of a Folder, a simplified Explorer that lacks a scope pane. These are typically invoked from the Chicago shell by double clicking a folder, which is essentially the same as running the default "Open" verb from the right mouse button context menu on that folder.

If the Office Explorer replaces only the full Explorer but not Folders, several users will never encounter it to realize its benefits.

To replace these Folders on Chicago, the following entries, under Chicago's registry HKEY_CLASSES_ROOT\Folder\shell\open branch, need modification - the "command" entry should be replaced with the path of the Office Explorer executable. And the "ddeexec" sub branch entries need to be modified as appropriate, again, this is to ensure that once the Office Explorer is already running, it can respond to a DDE message asking for a folder to be opened by creating a new thread to handles the request, rather than running another new instance.

Special Considerations for Views on Chicago Special Folders

As described in the implementation strategy list (on item 3), Office Explorer will attempt to leverage IShellView for the contents panes of special folders. However, integrating IShellView (for special folders) with the Office Explorer custom view (for FAT folders) may present some tricky problems, e.g., with toolbar and menu negotiation. This calls for an alternative, fallback strategy for the situation where Office Explorer is unable to leverage IShellView in this manner. The goal is for Office Explorer to provide workarounds so the user can continue to access special folder information as easily as possible.

In this alternative strategy, Office Explorer will not implement native views for special folders. Rather, when it installs, Office Explorer will associate special folders with a new *SpecialFolder* class that it will create in the Chicago registry, and, by customizing the registry entries for the Open and Explore verbs for the *SpecialFolder* class, these folders will be made to invoke the Chicago Explorer instead of the Office Explorer. This means:

1. All *SpecialFolder* folders in the Chicago shell (e.g., Recycle Bin on the desktop, and any other special folder shortcuts created by the user on the desktop) will always invoke the Chicago Explorer when the Open verb is invoked on these folders (i.e., either by double clicking, or through the default "Open" menu item on their context menu), or when the Explore verb is invoked on these folders (through the Explore menu item on their context menu). If the Open verb was used, Chicago Explorer is launched in folder mode. If the Explore verb was used, then Chicago Explorer is launched in explorer mode.

M CONFIDENTIAL

Strategic-5

MS-PCA 1566802

CONFIDENTIAL

Office 98 Spec Competitive: Chicago Explorer Superset and Replacement

Issue: Once Office launches the Chicago Explorer in explorer mode as described above, users are free to browse other folders using the Chicago Explorer, where they will not find rich views on FAT. Will this lead to confusion?

Should we consider making the Explore verb on special folders do the same thing as the Open verb, i.e., bring up Chicago Explorer in folder mode? This way we can altogether hide the Chicago Explorer's explorer mode from the user. Even in folder mode, we must still try and root the Chicago Explorer to the specific special folder being opened.

2. All *SpecialFolder* folders that appear in the Office Explorer will also respond in a similar way when opened or explored, i.e., by bringing up the Chicago Explorer.

Issue: This will launch the Chicago Explorer in its own window, on top of the Office Explorer. This makes it apparent to the user that these folders are in fact, "special". Is this behavior acceptable?

Note that this alternative strategy depends on the ability of the Chicago registry to support a user-defined *SpecialFolder* class of folders whose behavior overrides that of regular folders. If it turns out that Chicago will not support this in their registry, Office Explorer by itself will take over the responsibility of dispatching to the Chicago Explorer using DDE whenever it recognizes a special folder. Here is how this will work - Office will associate the Open and Explore verbs for all folders with the Office Explorer executable. Now, whenever the user runs either the Open or Explore verb on a folder in the Chicago shell, Office Explorer gets a DDE notification, with a value called the *pidl* (pointer to *ItemIDLList*) for that folder as one of the parameters. This *pidl* uniquely identifies a folder, Office Explorer can use the *pidl* and an API called *SHGetSpecialFolder* to find out if the folder in question is a special folder, and if so, dispatch this DDE message to Chicago's Explorer, making the latter activate.

Note: In the current implementation of Chicago, all special folders that are in the Start menu (e.g., Settings Control Panel, or Printers) will always launch the Chicago Explorer, irrespective of what executable is associated with these special folders in the registry. Something to keep in mind.

Plan "B"

This is a backup plan for supersetting and replacing the Chicago Explorer altogether.

Chicago has the namespace extension mechanism of having a directory with a special .INI file provide its own *IShellView*, this allows any application to put its own view over the file system in Chicago, on all folders that have been marked by this .INI file. A small change to Chicago would allow this addition to be made on a per-class basis rather than instance basis, so that views on all FAT folders can be replaced this way (the mechanism would be changed to be based on a registry entry rather than INI files).

So, under the backup Plan "B", Office will still implement its specialized view that provides rich views on FAT. However, when Office installs, this Office view will plug into the contents pane of the Chicago Explorer, for all FAT folders. So there will be no special "document module" in Ren, users will be able to browse rich views on FAT through the Chicago Explorer itself.

CONFIDENTIAL

Strategic-6

MS-PCA 1566803

CONFIDENTIAL

Office '96 Spec Competitive: Chicago Explorer Superseed and Replacement

Platform Mac: Ren does not target the Mac.

NT: This depends on whether the NT version that we target has the Chicago shell ported on it. If it does, we will provide identical functionality as on Chicago; if we target just Daytona, there is no Explorer to replace.

International -

Visual Basic Office Explorer will use the same object model for the Office Explorer as whatever is being planned for the Chicago Explorer. The current Chicago plan seems to be to support IDispatch and publish their object model so the Explorer can be automated using VB or VBA.

User Ed If Office Explorer replaces the Chicago Explorer, we need to worry about shipping with the Chicago Explorer documentation - this will increase our COGS in a big way.

Q & A -

Changes

14-Nov-94	The table in Appendix A	vinoda
17-Nov-94	Description, updated table	vinoda
06- Dec-94	IShellView and special folder stuff	vinoda

M CONFIDENTIAL

Strategic-7

MS-PCA 1566804

CONFIDENTIAL

Office '96 Spec Competitive: Chicago Explorer Superset and Replacement

APPENDIX A

Features Office Explorer must implement to superset and replace the Chicago Explorer.

	<u>WORK ITEM</u>	<u>OFFICE EXPLORER</u>	<u>CHICAGO</u>
1	Network browsing	Use IShellFolder.	-
2	Special folders - Control Panel Fonts Printers Recycle Bin Dial-Up Networking Remote Access	Use IShellFolder & IShellView. There is code in the IShellView implementation that is specific to special folders (e.g., their special properties, seen in the Details view, the "Similarity" view mode for Fonts, etc.). Office Explorer must try and use IShellView for special folders to get all this functionality. Also, the Fonts special folder does not expose its item objects through IShellFolder. Also see the discussion on Special Considerations for Views on Chicago Special Folders , in the Description section of this spec.	-
3	Shell-compatible property sheets for file system and special objects. This includes property sheet page extensions created by ISVs on a per class basis.	Use IShellFolder.	-
4	Drag and drop. Again, this is for FAT as well as the network, and other special folders. Has to work in the framework of the shell's Undo mechanism.	Using IShellFolder allows us to add to the shell undo stack.. Office needs API support from Chicago to invoke the Undo.	Provide shell Undo APIs. This is currently in the Chicago shell, but is internal and not exported.
5	Shell-compatible cut, copy, paste and delete operations. Deleted files should go to the Recycle Bin. Should work with the shell Undo mechanism.	Works via IContextMenu.	Expose Undo APIs
6	Generate correct icons for each object.	Use IShellFolder. Chicago's has a system imagelist (i.e., the icon cache) but this cannot be exposed by APIs because it cannot function the same way on porting to NT. Office explorer has to implement its own icon caching mechanism, has to use the public Imagelist APIs in Chicago and duplicate Chicago's icon caching.	Chicago cannot expose this through APIs, it uses a global imagelist accessed by multiple processes, which needs to be per-process on NT.

CONFIDENTIAL

Strategic - 8

MS-PCA 1566805

CONFIDENTIAL

Office '96 Spec Competitive: Chicago Explorer, Superstar, and Replacement

	<u>WORK ITEM</u>	<u>OFFICE EXPLORER</u>	<u>CHICAGO</u>
7	Support all registered shell extensions: Context Menu Handler Property Sheet Handler Icon Handler Drag/Drop Handler Copy Hook Handler	Use IShellFolder::GetUIObject to get the menu and property sheet (via IContextMenu), icon (IExtractIcon), and drag/drop (IDropTarget). Copy hook handler is taken care of through the shell copy mechanism.	-
8	Display correct names for all objects shown in the explorer. File names may or may not include extensions, show labels for disks, prettified names for network objects.	Taken care of by IShellFolder	-
9	Views on FAT have to update on change notifications from the underlying container.	The preferred solution is for Chicago to publish their shell notification APIs so we can use them. The alternative is to use standard Win32 APIs, possibly borrow Chicago's scheme that translates FS notifications into internal messages.	Chicago should preferably publish the existing SHChangeNotifyRegister & SHChangeNotifyDeregister APIs in Chicago. However, these APIs are currently untested and therefore not robust. If we resort to doing this from scratch (with the Win32 APIs), we must try to leverage Chicago's implementation.
10	Implement the Explorer frame toolbar buttons, there are just a couple of these - the dropdown with the folder hierarchy, and the "One Level Up" button.	To be added in Office Explorer.	
11	Provide Large Icons, Small Icons and List views on FAT, including their toolbar buttons.	Office explorer will implement these. Office can leverage Chicago's ListView control to implement these views.	-
12	File menu - Create Shortcut Properties	Handled by IContextMenu	-
13	Edit Menu - Select All Invert Selection	Code for these exists in Ren today.	-
14	View Menu - Arrange Icons (by Name, Type, Size, and Date, also AutoArrange for Small and Large modes) Line Up Icons	Office Explorer needs to implement these.	-
15	View Options dialog Office Explorer needs to honor the options the user has set here, automatically filtering file classes that the user doesn't wish to see.	In the absence of a Chicago API to call this, Office Explorer has to create this dialog, update and keep in sync with the registry settings for the filters and other options the user has set.	View Options is in explorer.exe. Chicago does not plan to move this to the shell DLL to expose this as an API.

M CONFIDENTIAL

Strategic-9

MS-PCA 1566806

CONFIDENTIAL

Office 96 Spec Comparison: Chicago Explorer, Superior and Replacement

	<u>WORK ITEM</u>	<u>OFFICE EXPLORER</u>	<u>CHICAGO</u>
16	Tools menu: Find (Files or Folders/Computer) Map Network Drive & Disconnect Network Drive (including the toolbar buttons added for these) Goto	We will use our own Find File, need to get Chicago to publish the FindComputer API. Issue: Will we need to change Find Computer to look like our Find File? The two network commands are relatively easy to implement in Office Explorer using Win32 network APIs. Goto is easy to do in Ren.	Publish the SHFindComputer API that exists in Chicago today.
17	Menu items that get added to the Explorer from verbs from the selected item object in the contents pane, e.g.: New (Folder or Shortcut) Open Open With: when there's no registry association for a file; brings up a dialog Send To Print QuickView Explore, Sharing & Find (for folders only)	These are handled via IContextMenu.	
18	Keyboard accelerators for merged menu items	This needs to be implemented in Ren's views. Chicago does this in their IShellView implementation and IContextMenu does not expose accelerators.	
19	Open in place - Explorer in folder mode has the option to open other folders in place, i.e., instead of bringing them up in a new window.	New functionality to be implemented in Ren.	-
20	Load time needs to be at least as good as Chicago's explorer	Performance implication for Ren.	-
21	Support formatting disks	Format is just another verb, appears for all drives. This again gets handled via IContextMenu.	-
22	Views onto floppies compressed using Diamond compression	-	This is likely handled the same way as views into special folders. This feature may not ship on Chicago at all.
23	Replacing the default explorer through registry settings	In the registry's HKEY_CLASSES_ROOT\Folder\shell\<exploreopen> branches, change the "command" and the "ddeexec" entries to point to the Office Explorer executable.	Start Menu's Settings folders (Control Panel, Printers, etc.) will run explorer.exe even after this registry change, this might be a bug. This behavior will need to be tied to these registry settings as well to allow Ren to replace Explorer across the board.

X CONFIDENTIAL

Strategic-10

MS-PCA 1566807

CONFIDENTIAL