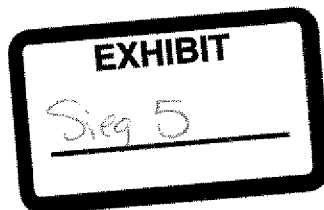

From: Sean Nolan
To: russs
Subject: IShellBrowserJunk
Date: Wednesday, October 05, 1994 1:47PM

The Explorer EXE is currently 190K retail.
--- S

<<File Attachment: nobrowse.doc>>



M 1010506
CONFIDENTIAL
MX 5103233
CONFIDENTIAL

Marvel Explorer & IShellBrowser

◆ IMPLICATIONS OF LOSING CHICAGO IMPLEMENTATION

The Explorer provides us with two main services:

1. **Window & Control UI**, including a toolbar, statusbar, tree view, split pane for explore view, default menu and toolbar commands/tooltips/help text, view options, real estate negotiation, left pane drag & drop, saved view states, keyboard shortcuts, etc. All basic stuff that just takes time to write and debug, much like an OLE2 insitu container.
2. **Hierarchy Management**. Chicago caches our IShellFolders, remembers our parent, keeps a global tree structure to preserve resources, manipulates IDLists, and notices when parts of the hierarchy change via FSNotify.

Since we are a Chicago-only product and did not expect this bombshell, we do not have the luxury of an NT-based IShellBrowser host to fall back on, as does Capone. Redesigning our shell from the ground up is not a realistic solution due to time constraints. The only viable alternative is to find or write a host we can plug our existing extension into, as detailed below.

◆ POSSIBLE SOLUTIONS

1. Continue to use Chicago Implementation
 - + No New Work
 - "Undocumented"
 - Dependency on Chicago
2. Port Ren/Office Source to Marvel Shell (*BrianMac, MarkGo, ErikGav, DougC*)
 - + Interfaces likely to be supported long-term
 - TIME
 - New interfaces mean non-trivial changes to our extension
 - New Codebase not intended to ship in our timeframe
3. Port Explorer Source to Marvel Shell
 - + Same Code means Marvel mimics Chicago behavior
 - + Code we've been testing Marvel against already
 - TIME
 - cabinet + shelldll = 270,000 LOC to wade through (DDE, net, tray, fail-safe, cpanel, etc. - all the other stuff we don't care about but need to weed out)
 - BradSi against sharing source
4. Port Capone Source to Marvel Shell (*JohnKal*)
 - + Smaller/Cleaner than Chi (about 5K LOC)
 - TIME
 - Major assumptions about MailView/MAPI
 - Assumes 1 thread per window
 - JohnKal believes straight port not practical
5. Write our own Marvel Shell (*SeanNo, DavidSan*)
 - + Our Code = No Dependencies
 - + Can use Chi TreeView
 - + Have Capone/Chi for Reference
 - TIME

M 1010507
CONFIDENTIAL

MX 5103234
CONFIDENTIAL

- New Code = New Bugs

◆ ANALYSIS OF POSSIBLE SOLUTIONS

From our perspective, option #1 is clearly superior. We can proceed as expected, squash our bugs and ship. If this is not possible, my recommendation is #5, i.e., we should write it ourselves.

The Ren/Office alternative is not workable. The killer is the fact that, beyond having to port their shell code, they do not provide IShellBrowser, and we would have to rewrite our extension to fit with their APIs. Beyond this, the codebase is new and relatively untested - another new risk we definitely do not need.

The other two options are ports of existing codebases: Explorer and Capone. Explorer is enormous and all-encompassing. The work to understand it and pare it down to the bare bones that we need seems as likely to take as long as to complete and produce bugs as writing it ourselves. Capone is much more attractive, and definitely has pieces we can steal. However, the code assumes much about MAPI and MailView; so much that a port seems futile.

Writing the code ourselves has a number of disadvantages. Our rough current estimate is that it will take one developer 2.5 months to complete and work out the major bugs. There is no way we will have this code in place for the M7 release. But owning the code offers a number of advantages as well, and it appears that the time required to port Explorer/Capone is almost identical. Internally, the immediate loss would be Sysop Tools, which would have to be postponed until after 1.0. This is a workable fallback, but not one I'm very happy about facing.

M 1010508
CONFIDENTIAL

MX 5103235
CONFIDENTIAL