

Apps / Chicago Synergy Ideas to get us started

- What's a good Chicago app
- So what's the problem?
- Ideas to get us thinking and moving forward

Plaintiff's Exhibit
7413
Comes V. Microsoft

Microsoft Confidential

What's a Good Chicago App according to today's plan?

- **Plug and play compatible**
 - Supports dynamic environment changes. Display, network, hardware, etc.
 - Easy to install and has uninstall capabilities
 - Makes setting app preferences a no-brainer
- **Supports new UI**
 - Matches 3d visuals including 3d shading colors
 - Uses common dialogs and other system controls where appropriate
 - Supplies small and large icons
 - Utilizes context menus and property sheets internally
 - Does drag/drop everywhere, including to/from the shell via OLE 2.0, to printers, etc.

What's a good Chicago app

Good Chicago app cont.

- **Exploits shell features**
 - Extends the systems' property sheets and context menus for classes that it owns
 - Supplies a great file viewer
 - Uses the recycle bin for deletes and cuts
 - Supports shell links
 - Replaces right pane of cabinet if appropriate. IE mail store or a document library or a database.
- **Supports long file names and UNC names**
- **It's a Win32c app**
 - Preemptively scheduled and async input make it a better app in the system.
 - Utilizes enhanced metafiles

What's a good Chicago app

Good Chicago app cont.

- **OLE 2.0 enabled**
 - supports drag/drop with the shell
 - registers any special verbs like “play” if need be
 - does all other cool OLE interoperability things
- **Network app, has workgroup features**
 - Works great setting up and running off the net
 - Follows file sharing rules
 - Uses MAPI to be mail enabled
 - Uses MAPI PAB to get at name and address info
 - Uses RPC for client/server interactions
- **Is a great mobile app**
 - Slow link awareness
 - Dock/Undock support (plug and play)
 - Power management smart

MS 5046400
CONFIDENTIAL

Microsoft Confidential

What's a good Chicago app

Good Chicago app cont.

- **Supports other platform improvements**
 - Comm apps use TAPI
 - International support. Uses NLS api to get system supplied locale information, sort tables, etc. Utilizes multilingual API to get/set keyboard layout and fonts for i/o in multiple languages.
 - leverages printing improvements
- **It's fast and robust**
 - app is tuned for size and performance
 - efficient when multiple instances are used
 - uses structured exception handling
 - tested under debug Windows and on NT

So what's the problem?

- OLE and Chicago are not leveraging each other
- Shell/app integration isn't as good as it could be
- Agreement on UI improvements are slow
- Workgroup synergy is weak
- Programability synergy is weak
- Groups are not moving forward together and sharing technology as well as they should.
- Will the apps group do everything listed?
- Is the list complete? Is everything there that the apps group wants?
- Lack of feedback that leverages the apps groups' experience to improve the system

Joint OLE 2.0 / Chicago goals

- Chicago should be the preferred platform for OLE apps. (IE An OLE app needs to be a whole bunch better on Chicago than on Win 3.1)
 - tune OLE 2.0 size, perf, features for Chicago
 - add value to OLE apps via the shell
- The best Chicago apps are OLE enabled.
 - apps take advantage of the above to deliver unique and useful function

Chicago's planned usage

- **Use OLE code for:**
 - shell <-> app drag/drop; scraps, files
 - GetclassID for opening documents, not for enumerate due to performance problems.
 - doc file summary page in file property sheet
 - iDispatch support in the shell for some limited set of commands tbd.
 - applets
- **Don't use OLE code for:**
 - right pane replacement in cabinet.
 - context menus extensions
 - property sheets extensions
 - Chicago has an "OLE like" but more efficient mechanism for these
 - Links

Ideas for Making Chicago a better platform for OLE apps

- Size and speed work in OLE
- OLE client enabling of folders.
- Make OLE links and Chicago links the "same" for users.
- Find files based on OLE properties
- Use OLE as the shell extensibility mechanism
- include useful OLE servers.
- Universal MDI container app for projects (would be a new OLE feature)
- more....

OLE size and speed work

- doc file integration into the file system for size and speed improvements.
- Drag/Drop code
- GetClassFile
- Getting doc file summary page
- doc files (work in progress)
- insitu editing
- iDispatch

Shell integration ideas for improved multi-app scenarios

- system wide MRU document list
- system wide window list (document centric)
- utilize “the tray” for app commands
- Windows button in every Chicago app which calls up common, useful function supplied by the system. (alternative to the tray)
- we need to look at what MS Office is doing and leverage

Getting UI agreement

- **MDI/SDI, Workspaces/Workbooks, projects, etc.**
- **Transfer model**
- **UI details: property sheets, transfer model, popups, 3d look, toolbar design, etc. Progress here is pretty good actually.**
- **Common dialogs, Toolbar control and the rest of the system controls need to made good enough for major apps**

Programmability

- **Plan of record**
 - the shell will expose some commands via iDispatch, but exact list and format completely TBD
 - vbasic applet is plan of record, but no decision on if it's VB3 or VB4 based.
- **How to improve**
 - exposed commands should be consistent with Excel and Word
 - should we put object basic in the system?
 - some language is needed for logon scripts
 - where do component forms and VBA/VB 4 fit in?

Areas We Could Improve

- **Workgroup**
 - **Library-type storage in the system**
 - **Routing, Delegation Scenarios**
 - **Links used on the Network. Creating a link locally and moving it around the net does not work currently**
 - **Clipbook/Templates. There ought to be a very simple way of getting to shared libraries of clipart, templates, etc**
 - **Alerts**
 - **Standard shared ISAM implementation. (or local ISAM implementation for that matter)**

Areas We Could Improve

- **Security and Users**
 - **Single notion of a "User". All our apps and system components should use one API for user-based data**
 - **Storage of User prefs**
 - **User-based operations in Apps -- Send, Dial, Chat, Alert, Share, etc.**
 - **Authentication of users**
 - **Signatures, ACLs, or other security mechanisms on individual files**

Areas We Could Improve

- **Management**
- **Network setup. Currently every WFW user installs apps locally and then tries to share them, this rarely works**
- **Software Distribution**
- **Configuration Management**
-

Other areas to improve synergy

- **Spell checker; OLE API and dictionary**
- **File sync merge handlers and apps doc management**
- **Chicago File viewers / app filters**
- **Text Editing: adopt the new selection model that word has, assuming it's cool**
- **Install/Uninstall: can Chicago help with uninstalling the shared components?**
- **Size. Getting Chicago and more than one app to work on a 4 meg system**
- **Default 256 color pallet (done)**

Still more...

- Shoud we include ODBC in the box?
- Multimedia. is this being leveraged?
- Licensing API?