

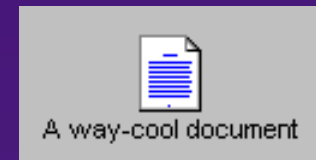
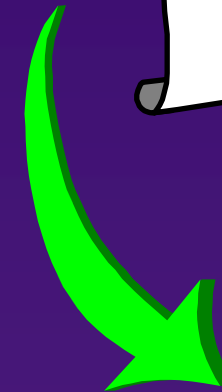


New WindowsTM “Chicago” UI: What It Means For Your Application...

Joe Belfiore
Program Manager
“Chicago” Shell/UI
Microsoft Corporation

A Change In Model...

- ◆ **Document-centric and object-oriented**
 - Users should begin to deal directly with objects (files) that your applications create, using common actions
- ◆ **Greater consistency between applications**
- ◆ **Greater integration among applications and the shell**
- ◆ **Simpler to learn, easier to use**



How Do I Get There From Here?

- ◆ **We're providing new controls you can (and should) use**
- ◆ **We're enhancing common dialogs**
- ◆ **We're making it possible for you to extend the shell**
- ◆ **Guidelines about what you should do first**

New Controls

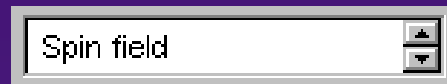
- ◆ **Toolbar**
- ◆ **Status bar**
- ◆ **Column heading**
- ◆ **Slider**
- ◆ **Spin buttons**
- ◆ **Progress indicator (gas gauge)**
- ◆ **Tabs**
- ◆ **Property sheet**
- ◆ **Rich text control**

New Controls - 2

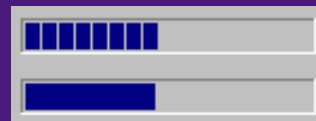
Slider



Spin buttons

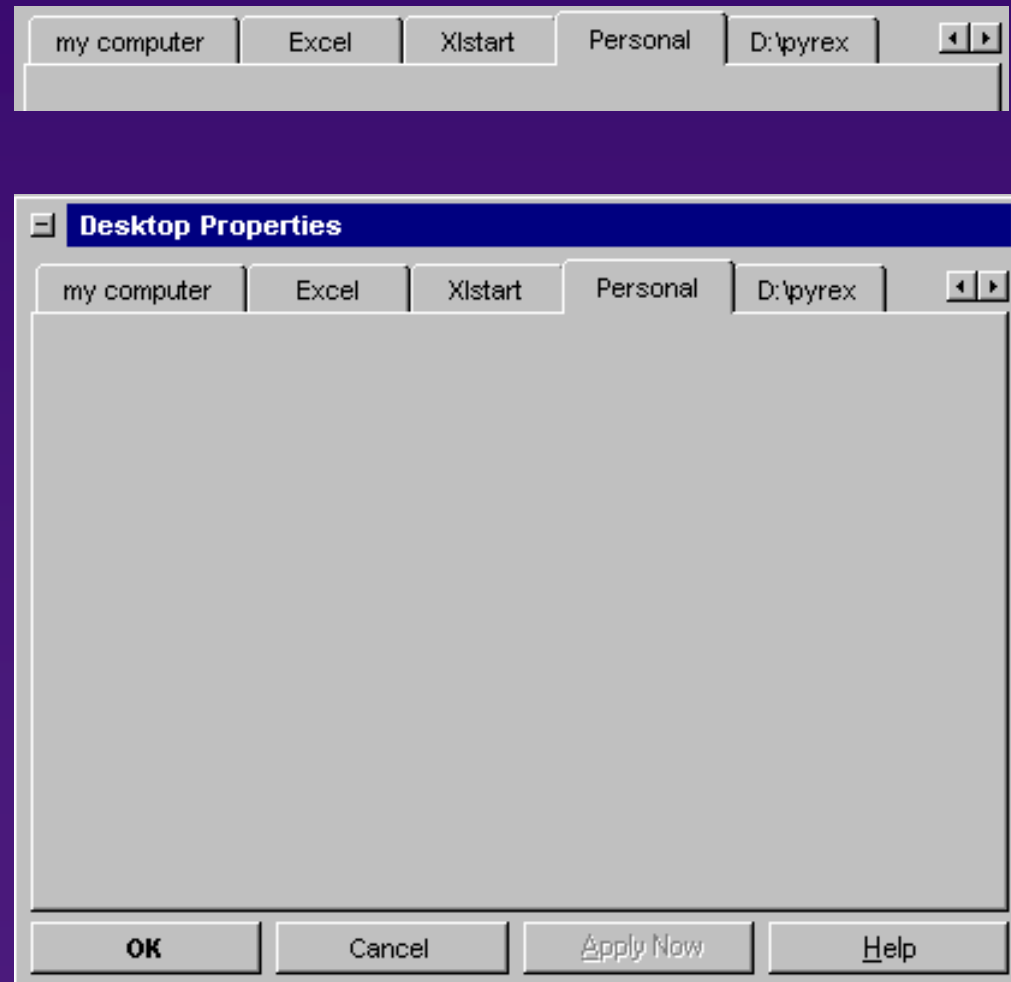


Progress indicator



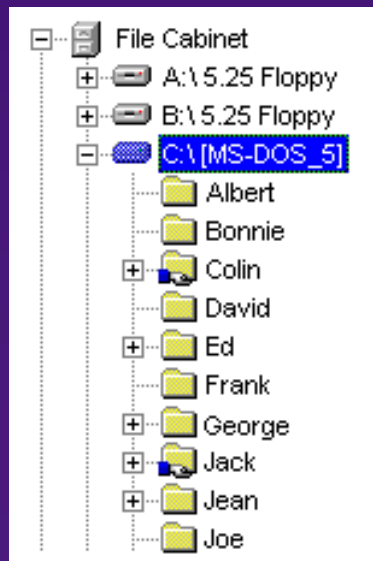
New Controls - 3

- ◆ **Tabs**
- ◆ **Can set the maximum number of rows to be displayed**
- ◆ **Tabs can scroll horizontally**
- ◆ **Property sheet**
- ◆ **You add pages as dialog templates**
- ◆ **Can add/change the buttons displayed across the bottom**



New Controls - 4

- ◆ **ListView**
- ◆ **Supports large icon , small icon (these are positional), list, details views**
- ◆ **Useful for “custom container” implementations**



- ◆ **TreeView**
- ◆ **Use to display hierarchies - containers, outlines, etc.**
- ◆ **Can add your own icons, configure expand/collapse**

New Controls - 5

◆ Rich text edit

- More than 64K of text
- OLE 2.0 client support
- Left, right, center alignment on a per-paragraph basis
- Definable left tabs
- Fonts (bold, italic, single underline, color)
- Simple bulleting
- Find and replace
- Superscript, subscript, strikethrough

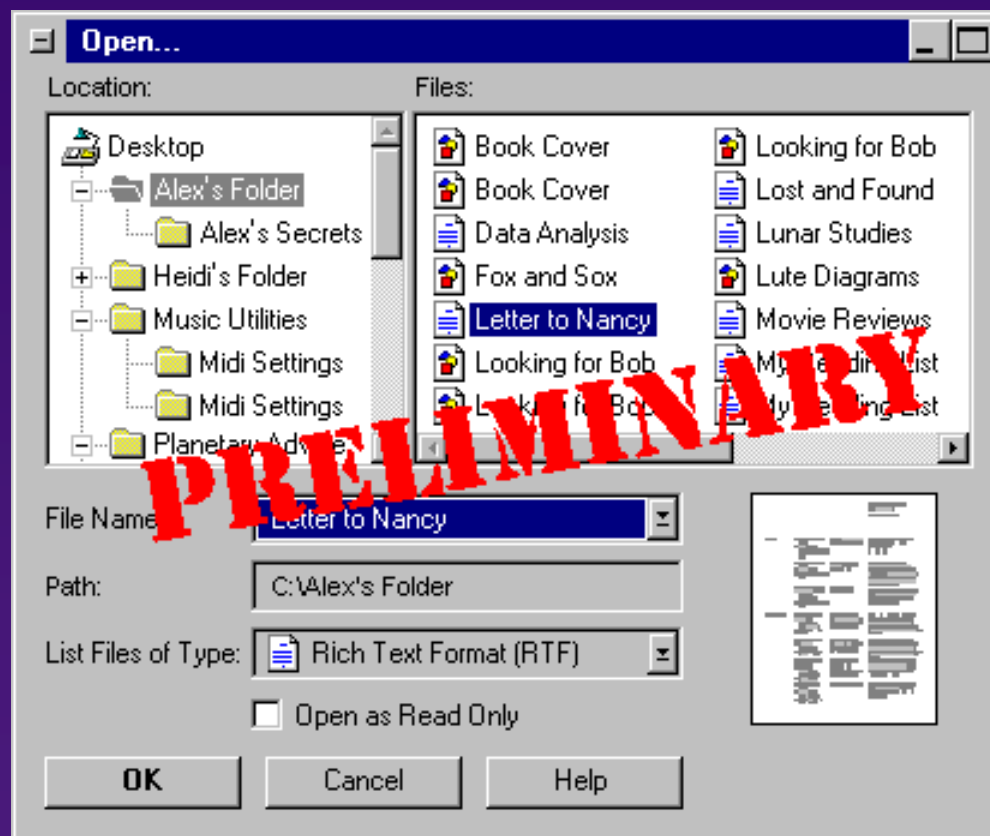
Common Dialogs

- ◆ **File open and save as**
- ◆ **Print**
- ◆ **Print setup (choose printer)**
- ◆ **Page setup**
- ◆ **Find and replace**
- ◆ **Font**
- ◆ **Color**
- ◆ **OLE 2.0 dialogs**

File Open/Save As

◆ Explorer functionality

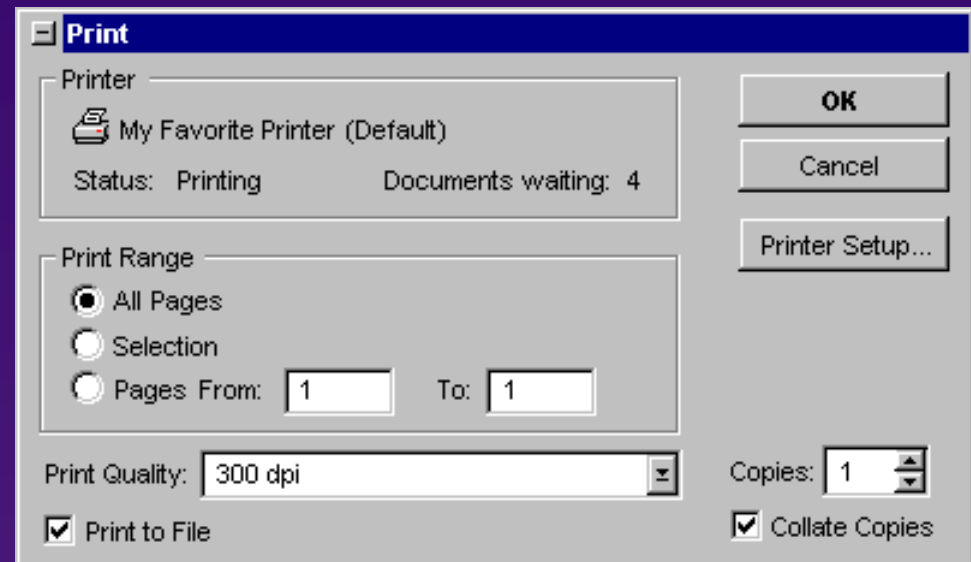
- Direct browsing of the network
- Displays and handles links properly
- Context menus available on files and background, allowing fs operations



◆ Extensibility similar to Windows 3.1

Print

- ◆ **Very similar to Windows 3.1**
- ◆ **Includes printer status information**
- ◆ **Hookable, as in Windows 3.1**



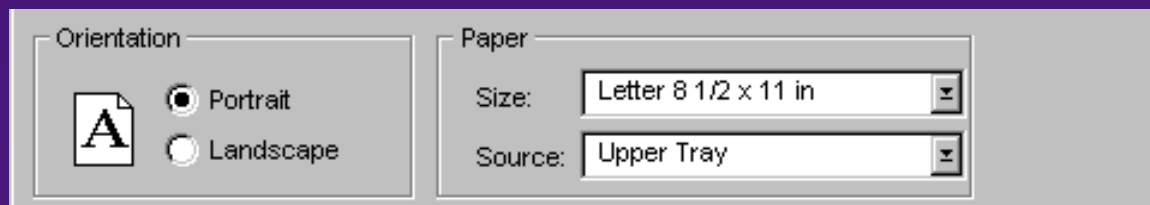
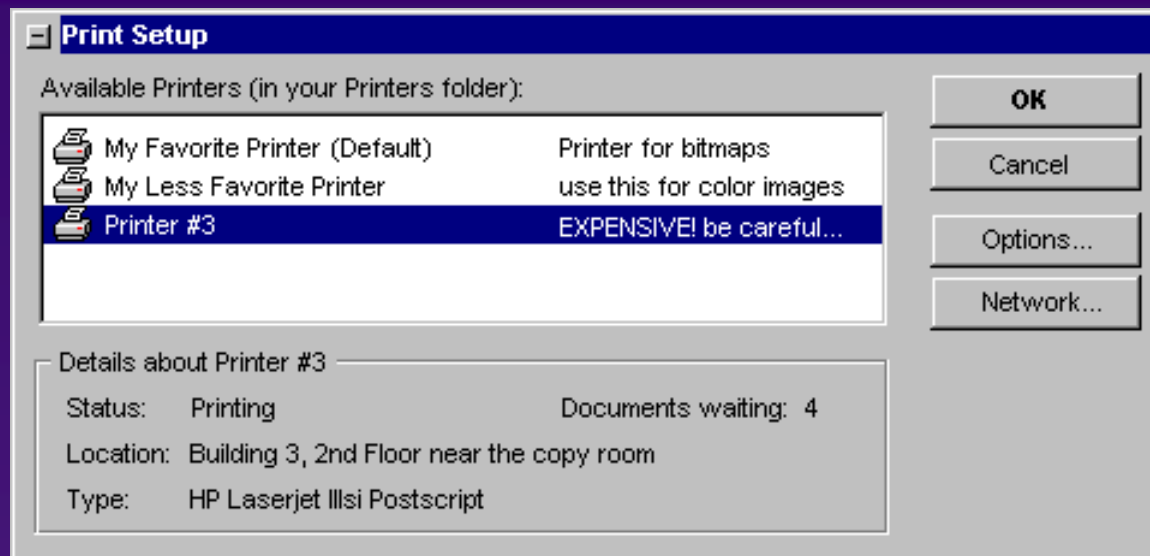
Choose Printer (Print Setup)

◆ Use for printer selection UI

- Choices are all the printers in your printers folder
- Status info for each printer is shown

◆ Compatible with today's applications

- Landscape/portrait and paper tray choices are displayed for old applications (bottom section shown above)
- New applications can set these using the page setup dialog

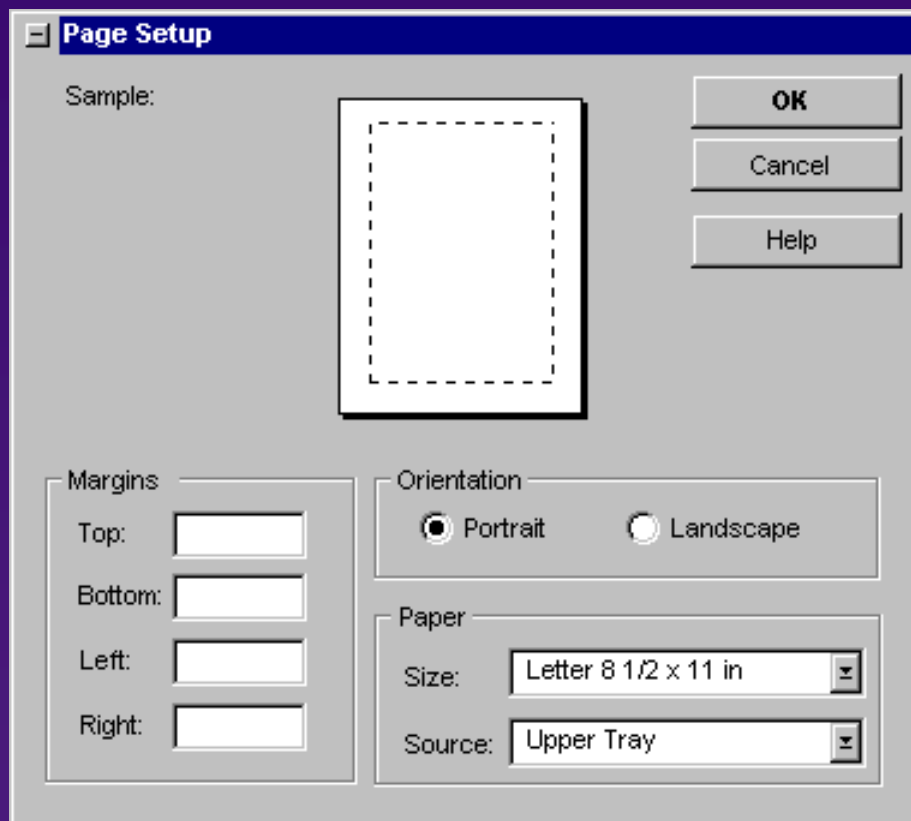




Page Setup

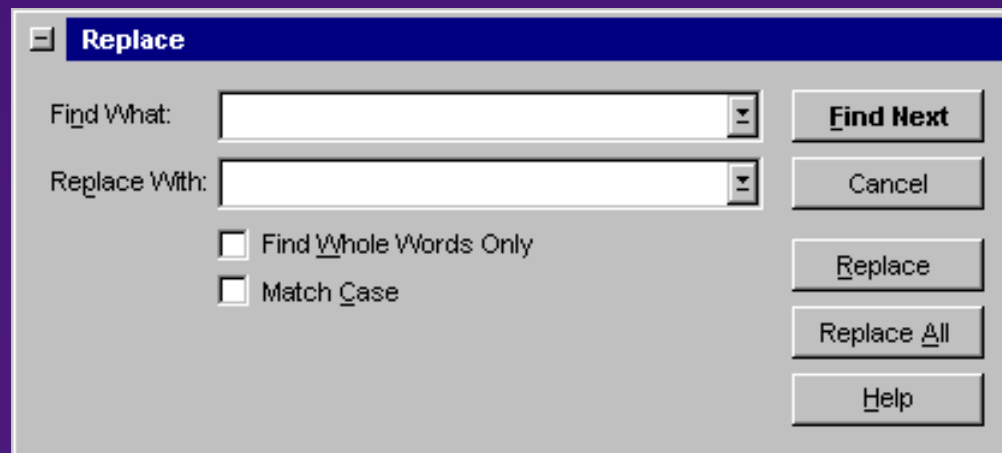
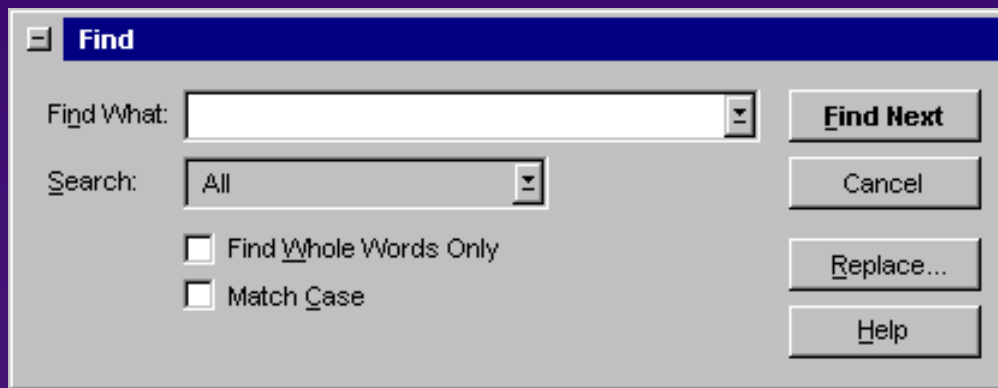
◆ General UI to set page layout

- Includes: orientation, paper settings, and margins
- Separates the old print setup dialog into printer selection and page setup components
- Available in property sheet mode



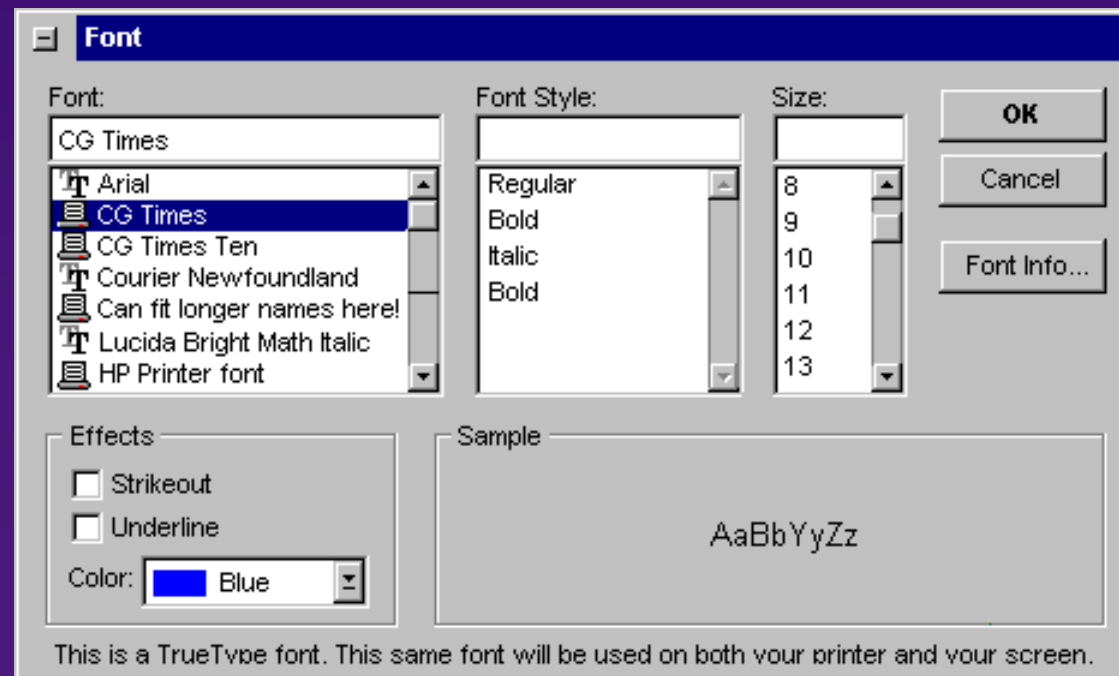
Find/Replace

◆ Very similar to Windows 3.1



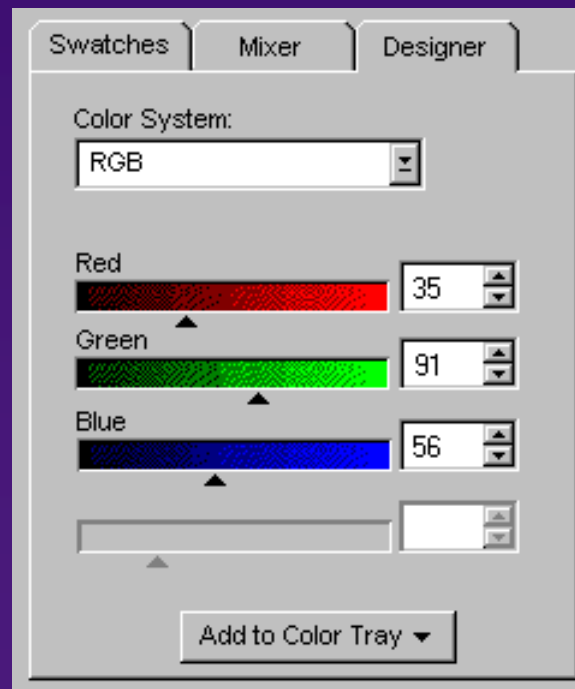
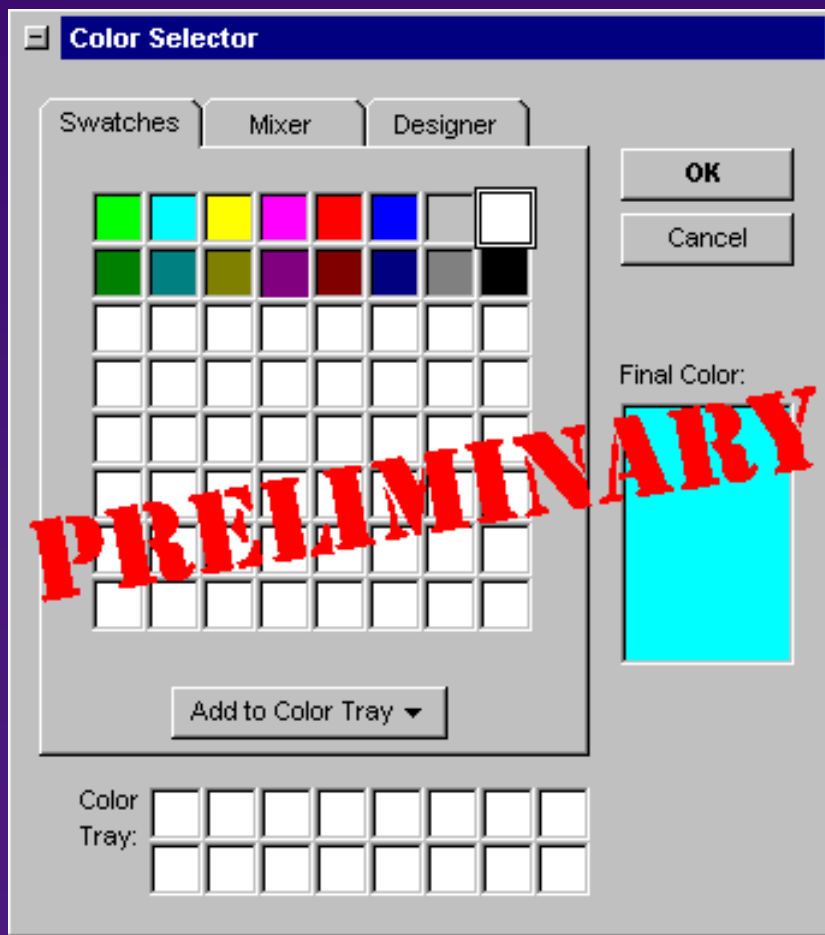
Fonts

- ◆ **Very similar to Windows 3.1**
 - **Adds “font info”**
 - **Also available in “property-sheet mode”**




Color

◆ Design still underway



OLE 2.0 Dialogs

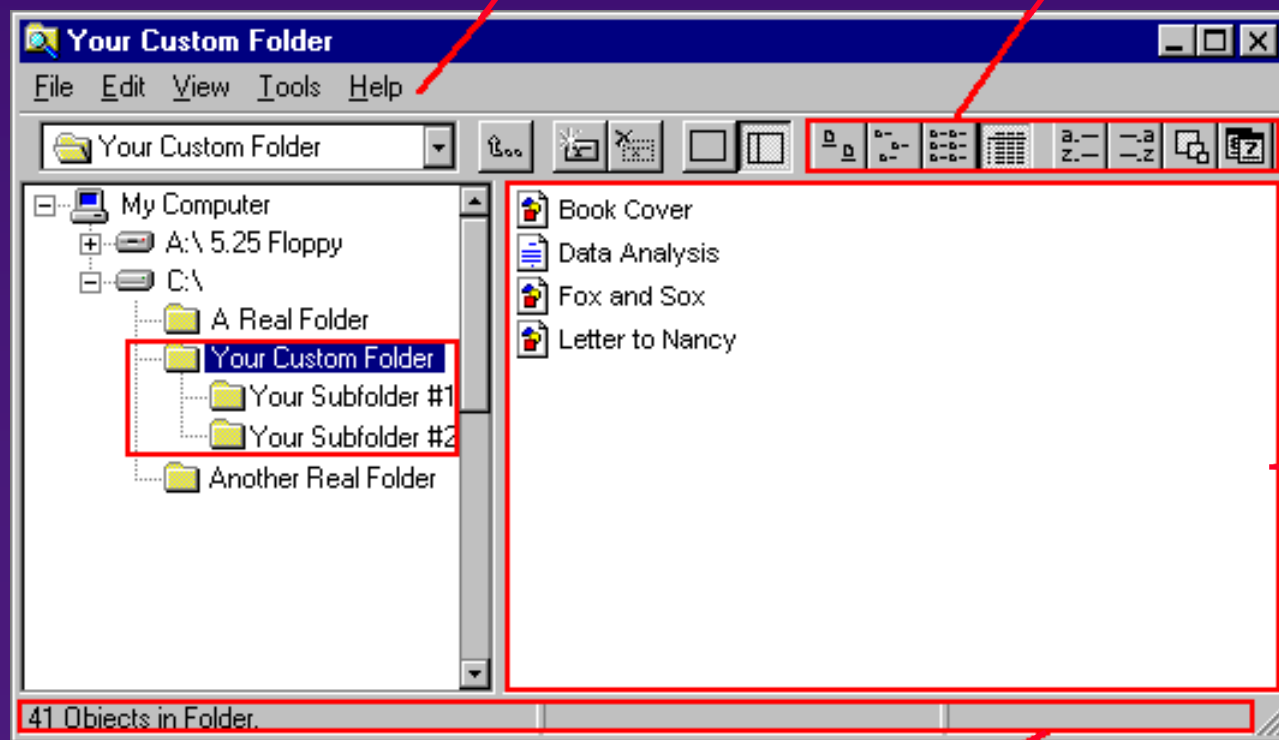
- ◆ We'll include all the OLE 2.0 dialogs, revised for look and feel of "Chicago":
 - Insert object
 - Change icon
 - Put here as/paste special
 - Convert
 - Links (link properties)
 - Object properties 
- ◆ Attend Randy Kerr's OLE 2.0 UI talk for more details!!

Shell Extensibility

- ◆ **OLE 2.0 integration**
 - Drag-and-Drop feature
 - Summary properties shown by default
 - Verbs you add to the registry will be displayed on object menus
- ◆ **Property sheet extensibility**
 - Add your own property sheet pages to objects on a per-class basis
- ◆ **Viewers**
 - Hook your own file viewers directly in to the shell
- ◆ **Explorer UI integration**
 - *If you have an application that displays a collection of file-like objects, you can create your own “custom container” displayed in the folder/explorer hierarchy*

Explorer Integration Details

- ◆ **Not for most applications!**
 - Only should be used if your application displays a pseudo-folder: electronic mail, document management, etc.
 - Users should **NOT** edit documents with an explorer extension!
- ◆ **How it works:**



Customize status bar

Microsoft Confidential

What About MDI?

- ◆ **MDI is ok**
 - **Make MDI behavior as consistent and non-mysterious as possible in the shell**
 - **Think about evolving away from MDI - read the style guide!**

Summary: How To Build A Great Windows “Chicago” Application UI

- ◆ Win32
- ◆ OLE 2.0
- ◆ “Chicago” UI style
- ◆ Plug and Play
- ◆ Shell integration

10 Keys For Making A Great Windows “Chicago” Application UI

1. Support long filenames
2. Support UNC path
3. Make sure your documents/data files are accurately displayed and used in the shell, with multisize icons, verbs, and no visible “.3” extensions
4. Use the commdlg “File Open”, or recreate its namespace accurately, including network browsing and links
5. Support OLE 2.0 Drag-and-Drop feature and the system menu-based transfer model consistently and extensively

10 Keys For Making A Great Windows “Chicago” Application UI

6. Be careful about multiple instances of your application being started *too easily* at the same time
7. Maintain a consistent user interface and object paradigm between your application and the shell. (Use right-click menus, prop sheets, our controls, etc.)
8. Support pen input for pen notebooks and desktop tablets
9. Support “Chicago”-style Help. (WinHelp authoring style, context-sensitive Help)
10. Make sure your visuals don’t break in “Chicago”! (3-D color, window metric changes, maximize to the desktop toolbar, not behind it)