Lecture

23

Review of Last Lecture

- Common dialog: choose colour
- Modeless and Modal dialogs
- Listbox
- Edit, radio, static, **`WM_CTLCOLORSTATIC`**
- Tab stop, Tab order, Groups of controls
- Control Notification Messages

Windows Common Controls *(figures)*

- Date time picker
- List View
- Progress bar
- Status bar
- Toolbar
- Tree View etc.

Windows Common Controls

- Weren't there in Win3.1
- Implemented in Comctl32.dll
- Library versions: IE and before IE shipments
- Effect of library versions
- InitCommonControls(void); registers the common control window classes

InitCommoncontrolsEx()

```
typedef struct tagINITCOMMONCONTROLSEX {
        DWORD dwSize;
        DWORD dwICC;
} INITCOMMONCONTROLSEX;
```

Valid values for the **dwICC** parameter:

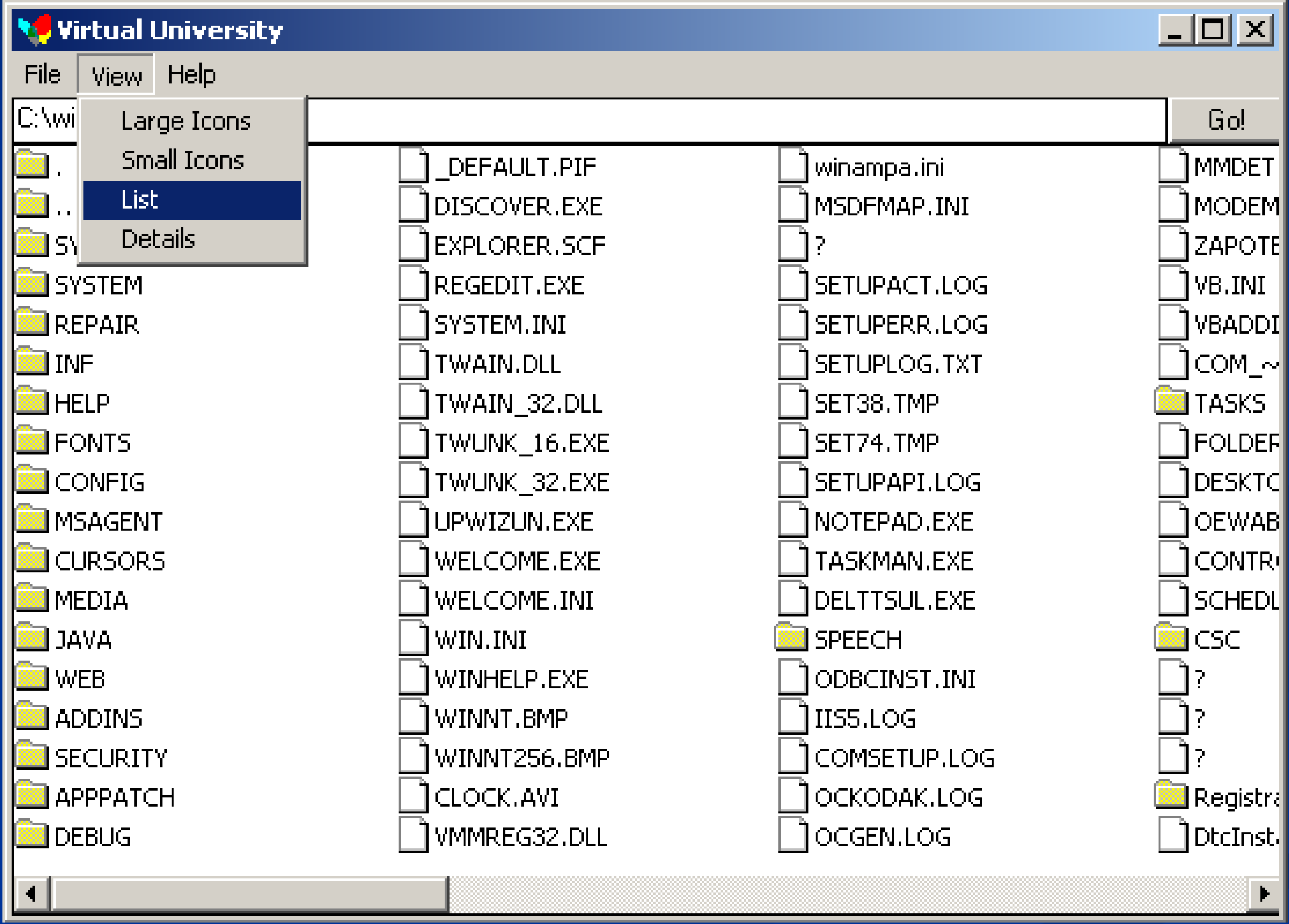
ICC_DATE_CLASSES : Date-Time picker control class

contd…

InitCommoncontrolsEx()

- List View: ICC_LISTVIEW_CLASSES
- Progress Bar: ICC_PROGRESS_CLASS
- Status bar: ICC_BAR_CLASSES
- Toolbar: ICC_BAR_CLASSES
- Tree View: ICC_TREEVIEW_CLASSES etc.

Figure –List View control



```
C:\wi
.           _DEFAULT.PIF     winampa.ini      MMDET
..          DISCOVER.EXE     MSDFMAP.INI      MODEM
SY          EXPLORER.SCF     ?                ZAPOTE
SYSTEM      REGEDIT.EXE      SETUPACT.LOG     VB.INI
REPAIR      SYSTEM.INI       SETUPERR.LOG     VBADDI
INF         TWAIN.DLL        SETUPLOG.TXT     COM_~
HELP        TWAIN_32.DLL     SET38.TMP        TASKS
FONTS       TWUNK_16.EXE     SET74.TMP        FOLDEF
CONFIG      TWUNK_32.EXE     SETUPAPI.LOG     DESKTC
MSAGENT     UPWIZUN.EXE      NOTEPAD.EXE      OEWAB
CURSORS     WELCOME.EXE      TASKMAN.EXE      CONTR
MEDIA       WELCOME.INI      DELTTSUL.EXE     SCHEDL
JAVA        WIN.INI          SPEECH           CSC
WEB         WINHELP.EXE      ODBCINST.INI     ?
ADDINS      WINNT.BMP        IIS5.LOG         ?
SECURITY    WINNT256.BMP     COMSETUP.LOG     ?
APPPATCH    CLOCK.AVI        OCKODAK.LOG      Registr
DEBUG       VMMREG32.DLL     OCGEN.LOG        DtcInst
```

Today's Goals

- ListView common control
- Description of the control's appearance and the application
- Image List

Today's Goals

- ListView control parts
- Views: Large, small, list, report
- Columns: header
- Items and subitems
- Images associated with LView: big, small and state image
- Image List

ImageList

- An *image list* is a collection of images of the same size, each of which can be referred to by its index.
- InitCommonControls needs to be called before image lists can be used

ImageList_Create

```
HIMAGELIST ImageList_Create(
    int cx, // width
    int cy, // height

    UINT flags, ILC_COLOR4 , ILC_MASK

    int cInitial, Number of images that the image
  list initially contains


     int cGrow This parameter represents the number of
  new images that the resized image list can contain

 );
```

```
int ImageList_AddIcon(
                    HIMAGELIST himl, HICON hicon);
```

Multiple device icons (16x16, 32x32 etc.) can be present in an single .ICO file

An icon contains a bitmap that is displayed when the icon is drawn, and a **mask** that specifies whether a certain pixel in the icon background will be overdrawn by the icon

```
int ImageList_AddIcon(
    HIMAGELIST himl,    // handle to image list
    HICON hicon         // bitmap and mask
  );
```

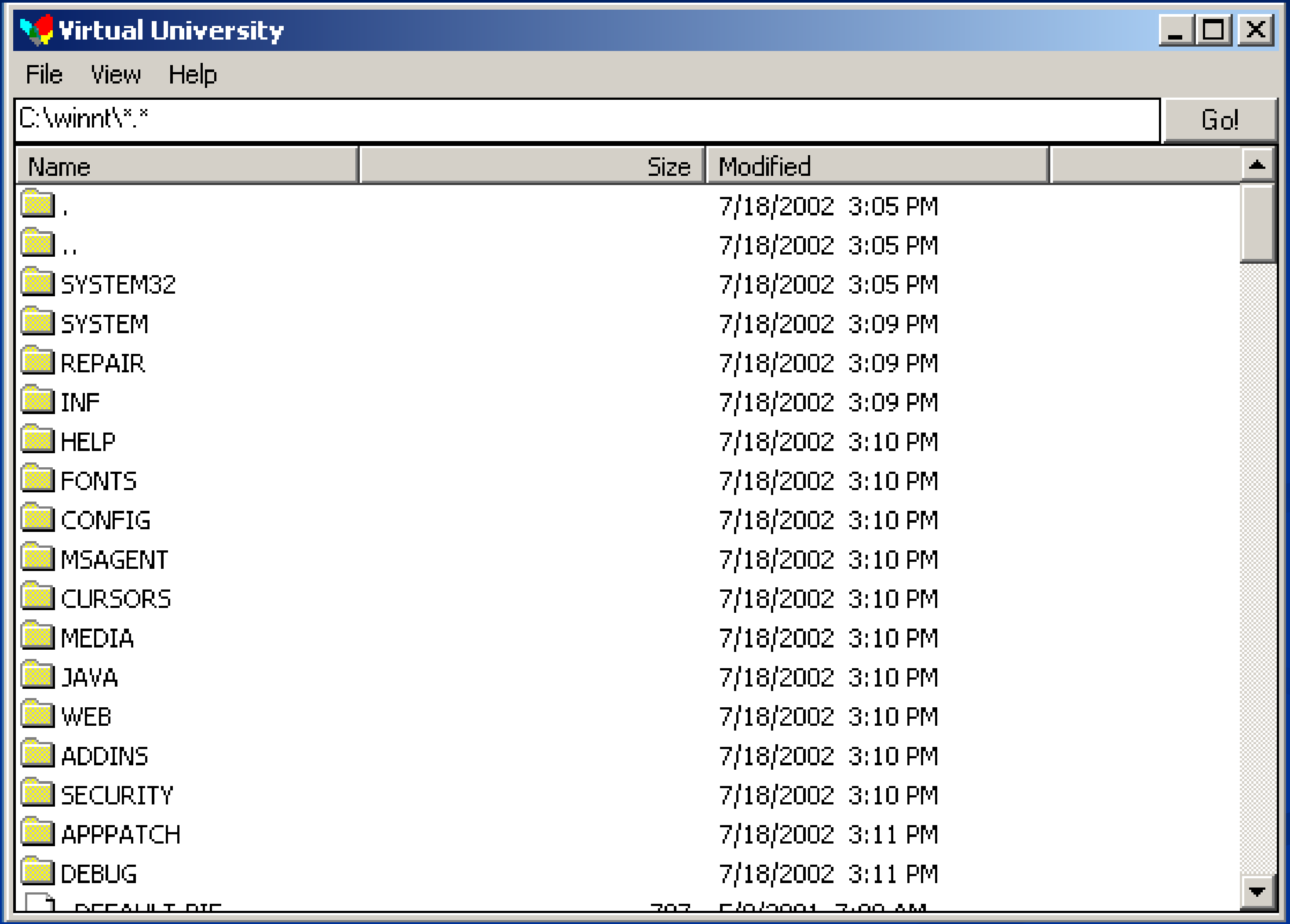
ImageList_ReplaceIcon( )

```
int ImageList_ReplaceIcon(
    HIMAGELIST himl,
    int i,
```

Index of the image to replace. If *i* is -1, the function appends the image to the end of the list.

```
    HICON hicon      bitmap+mask
  );
```

Screen-shot of Example Application



C:\winnt\*.*

Name	Size	Modified
.		7/18/2002 3:05 PM
..		7/18/2002 3:05 PM
SYSTEM32		7/18/2002 3:05 PM
SYSTEM		7/18/2002 3:09 PM
REPAIR		7/18/2002 3:09 PM
INF		7/18/2002 3:09 PM
HELP		7/18/2002 3:10 PM
FONTS		7/18/2002 3:10 PM
CONFIG		7/18/2002 3:10 PM
MSAGENT		7/18/2002 3:10 PM
CURSORS		7/18/2002 3:10 PM
MEDIA		7/18/2002 3:10 PM
JAVA		7/18/2002 3:10 PM
WEB		7/18/2002 3:10 PM
ADDINS		7/18/2002 3:10 PM
SECURITY		7/18/2002 3:10 PM
APPPATCH		7/18/2002 3:11 PM
DEBUG		7/18/2002 3:11 PM
_DEFAULT.PIF	707	5/9/2001 7:00 AM

Creating the interface (very brief)

- `wc.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_ICON_VU));`
- `MainWindow`
- A child edit control `IDC_EDIT_DIRECTORY`
- A "Go" button `IDC_BUTTON_GO`
- Font setting

```
SendMessage(hwndGo, WM_SETFONT, (LPARAM)hFont, MAKELPARAM(TRUE, 0));
```

- `icex.dwSize = sizeof(INITCOMMONCONTROLSEX);`

```
icex.dwICC = ICC_LISTVIEW_CLASSES;
InitCommonControlsEx(&icex);
```

Creating a ListView control

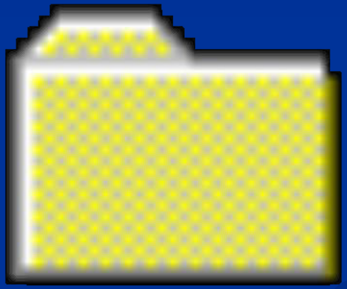
We Created a window with **WNDCLASS wc**

```
#define ID_LISTVIEW           5
hWndListView = CreateWindow(WC_LISTVIEW,
      "Window Name",
      WS_TABSTOP | WS_CHILD | WS_BORDER |
      WS_VISIBLE | LVS_AUTOARRANGE | LVS_REPORT,
      10, 10, 350, 280, hWndMain,
      (HMENU)ID_LISTVIEW, hInstance, NULL);

if(!hWndListView)
      return 1;
```

Creating imglist

```
hLarge =
   ImageList_Create(GetSystemMetrics(SM_CXICO
   N), GetSystemMetrics(SM_CYICON), ILC_MASK,
   1, 1);

hSmall =
   ImageList_Create(GetSystemMetrics(SM_CXSMI
   CON), GetSystemMetrics(SM_CYSMICON),
   ILC_MASK, 1, 1);

hIcon = LoadIcon(hInstance,
   MAKEINTRESOURCE(IDI_ICON_FOLDER));

ImageList_AddIcon(hLarge, hIcon);

ImageList_AddIcon(hSmall, hIcon);

hIcon = LoadIcon(…,
   MAKEINTRESOURCE(IDI_ICON_FILE));
```

Window default folder Icon



Folder.ico

Add image list

```
ListView_SetImageList(hWndListView,
  hLarge, LVSIL_NORMAL);

ListView_SetImageList(hWndListView,
  hSmall, LVSIL_SMALL);

HIMAGELIST ListView_SetImageList(

     HWND hwnd,

     HIMAGELIST himl,

     int iImageList
  type of Image List: LVSIL_NORMAL | LVSIL_SMALL |
  LVSIL_STATE

);
```

Add coloumns to listview

```
lvc.mask = LVCF_FMT | LVCF_WIDTH |
           LVCF_TEXT | LVCF_SUBITEM;

lvc.cx = COL_WIDTH;

for(i=0; i<3; ++i)
{
  lvc.iSubItem = i;
  lvc.fmt = alignments[i];
  lvc.pszText = columnHeadings[i];

  if(ListView_InsertColumn(hWndListView,i,&lvc)==-1)
           return 1;
}
```

Add an Item

```
// add an item with 3 subitems = 4 coloumns
lvi.state = 0;             // no state specified
lvi.stateMask = 0;         // no state specified
lvi.lParam = (LPARAM)1234; // item specific data
do
{
  lvi.mask = LVIF_TEXT | LVIF_IMAGE |
             LVIF_PARAM | LVIF_STATE;
  lvi.iItem = itemNo++;    // which item it refers to
  lvi.iSubItem = 0;        // refers to an ITEM
  lvi.iImage = (findFileData.dwFileAttributes &
      FILE_ATTRIBUTE_DIRECTORY) ? 0 : 1; // proper image
  lvi.pszText = findFileData.cFileName;
                                         // add the item
  if(ListView_InsertItem(hWndListView, &lvi) == -1)
      return 1;
```

add subitems for this item

```
lvi.mask = LVIF_TEXT;
lvi.iSubItem = 1;
// (findFileData.nFileSizeHigh * (MAXDWORD+1)) +
  findFileData.nFileSizeLow;
if(findFileData.dwFileAttributes &
                           FILE_ATTRIBUTE_DIRECTORY)
  wsprintf(buf, "");
else
  wsprintf(buf, "%10lu", findFileData.nFileSizeLow);

lvi.pszText = buf;
if(ListView_SetItem(hWndListView, &lvi) == -1)
  return 1;
```

Last modified date of file

```
FileTimeToLocalFileTime(
                &findFileData.ftLastWriteTime, &fileTime);
FileTimeToSystemTime(&fileTime, &systemTime);

strcpy(strAMPM, systemTime.wHour>=12 ? "PM" : "AM");
if(systemTime.wHour>=12)
    systemTime.wHour -= 12;
if(!systemTime.wHour)
    systemTime.wHour = 12;

wsprintf(buf, "%d/%d/%d %2d:%02d %s", systemTime.wMonth,
    systemTime.wDay, systemTime.wYear, systemTime.wHour,
    systemTime.wMinute, strAMPM);

lvi.iSubItem = 2;
lvi.pszText = buf;

if(ListView_SetItem(hWndListView, &lvi) == -1)
    return 1;
```

Modifying list view control

```
VOID SetView(HWND hwndListView, DWORD dwStyle)
{
  DWORD dwCurrentStyle;

  dwCurrentStyle = GetWindowLong(hwndListView,
                                           GWL_STYLE);

  SetWindowLong(hwndListView, GWL_STYLE,
        (dwCurrentStyle & ~LVS_TYPEMASK) | dwStyle);
}
```

Getting the list of files in a directory

```
hFind = FindFirstFile(DEFAULT_PATH, &findFileData);

if(hFind == INVALID_HANDLE_VALUE)
{
  MessageBox(NULL, "Error calling FindFirstFile",
                                    "Error", MB_OK);

  return 1;
}
```

Creating a ListView control

```
hLarge = ImageList_Create(
        GetSystemMetrics(SM_CXICON),
        GetSystemMetrics(SM_CYICON),
        ILC_MASK, 1, 1);

hSmall = ImageList_Create(
        GetSystemMetrics(SM_CXSMICON),
        GetSystemMetrics(SM_CYSMICON),
        ILC_MASK, 1, 1);

hIcon = wc.hIcon;

ImageList_AddIcon(hLarge, hIcon);

ImageList_AddIcon(hSmall, hIcon);
```

Adding columns: LVCOLUMN structure

- **mask**
  - Variable specifying which members contain valid information. This member can be zero, or some valid value:

LVCF_FMT	The **fmt** member is valid.
LVCF_SUBITEM	The **iSubItem** member is valid.

- **fmt**
  - Alignment of the column header and the subitem text in the column. A few of the valid values are:

LVCFMT_LEFT	Text is left-aligned.
LVCFMT_RIGHT	Text is right-aligned.

- **cx**
  - Width of the column, in pixels.
- **pszText**
  - If column information is being set, this member is the address of a null-terminated string that contains the column header text. If the structure is receiving information about a column, this member specifies the address of the buffer that receives the column header text.
- **cchTextMax**
  - Size of the buffer pointed to by the **pszText** member. If the structure is not receiving information about a column, this member is ignored.
- **iSubItem**
  - Index of subitem associated with the column.

ListView_InsertColumn

- Inserts a new column in a list view control. You can use this macro or send the LVM_INSERTCOLUMN message explicitly.

-
  ```
  int ListView_InsertColumn(HWND hwnd, int iCol,
                            const LPLVCOLUMN pcol);
  ```

Parameters

- hwnd
  - Handle to the list view control.
- iCol
  - Index of the new column.
- pcol
  - Address of an LVCOLUMN structure that contains the attributes of the new column

Subitems and their associated text

-
  ```
  int ListView_InsertItem( HWND hwnd,
                           const LPLVITEM pitem);
  ```
- Insert an item into a ListView.
- The address of an **LVITEM** structure is passed, whose **iSubItem** must be 0, and **iItem** must contain zero based index

-
  ```
  BOOL ListView_SetItem( HWND hwnd,
                         const LPLVITEM pitem );
  ```
- Set the text of a sub-item. **iSubitem** is 1-based index of the sub-item. **iItem** is 0-based item index as normal

About the accompanying code

Refer to actual code file where we

- Create Image Lists and assign them to the ListView.
- Use **ListView_InsertColumn** to add columns.
- Use **FindFirstFile()** and **FindNextFile()** to retrieve directory listing
- Use **ListView_InsertItem** to insert items.
- Use **ListView_SetItem** to add subitems and set their text.

Setting view of a ListView control

- Switching view at menu commands
- Use subclassing, i.e. `SetWindowLong()`

```
VOID SetView(HWND hwndL.View, DWORD dwStyle)
{
   DWORD dwCurrentStyle;
   dwCurrentStyle = GetWindowLong(
                         hwndListView,GWL_STYLE);
   SetWindowLong(hwndListView, GWL_STYLE,
            (dwCurrentStyle & ~LVS_TYPEMASK) |
             dwStyle);
}
```

Need to know more ?

Consult your documentation to know about the following:

- Callbacks
- Virtual list view control
- Sorting of columns
- Working areas
- Label editing

Something To Do! ☺ → ☹

- Impelment command line arguments to get the directory name in command line whose contents are to be listed
- Implement the Go button
- Implement a context menu to switch between views

Challenging Tasks

- Add a Tree View control to make it look more like the standard Windows Explorer
- When you right-click a file in the standard Windows Explorer of today, a Properties dialog is displayed, that shows, among other information, the file type and a specific icon. Implement this Properties dialog.
- Add functionality so that type-specific icons are displayed with files just like as they are in Windows.