

Написание полноценных **Windows** приложений на **Delphi** требует от программиста навыков применения ресурсов самой

Windows

. Под ресурсами

Windows

подразумеваем ресурсы (

Bitmap, Icon, Avi, Dialog

) содержащиеся в системных файлах . Применение данных ресурсов позволяет значительно ускорить написания приложений и придания им профессионального вида.

При написании

Windows

приложений львиную часть времени отнимает проектирование интерфейса, (нахождение нужной иконки, или Avi файла).

С переходом на winxp простое выдёргивание тех же иконок из системных файлов не даёт желаемого результата, к тому же, такой путь значительно утяжеляет приложение.

Из всего обилия ресурсов находящихся в системной директории **Windows** возьмём:

- **shell32.dll** - ресурсы общего назначения
- **netshell.dll** - ресурсы иконки сети, Bitmap, диалоговые окна, Avi
- **mmsystem.dll** - медиа ресурсы
- **comdlg32.dll** - диалоговые окна
- **compstui.dll** - ресурсы принтера

Для поиска ресурсов можно использовать программу **ResourceExplorer**, которая находится в папке Program filesborlanddelphi6demosresexplor, которую необходимо предварительно скомпилировать.

Запустив программу и открыв, к примеру, файл **netshell.dll** мы увидим ресурсы которые хранятся в данной DLL.

Рассмотрим пример размещения иконки из ресурса в трее.

```
uses shellapi; //Подключаем необходимые библиотеки const //определяем
сообщение при клике по трею wm_notifytrayicon = wm_user + 151; var //определяем
переменные buffer: array [0..255] of char; path: string; //путь hwndhicon : hicon;
//хендл иконки tray: tnotifyicondata; //экземпляр трея //***** определяем путь к файлу
с ресурсом **** function GetPath(stres: string): string; begin GetSystemDirectory(buffer,
SizeOf(buffer)); //системная директория path:= buffer; result:= path + " + 'stres'; end;
//***** получение хендла иконки из ресурса **** function GetHIcon(path:string; idicon:
DWORD): HICON; begin result:= ExtractIcon(HInstance, PChar(path), idicon); end; //*****
создаём иконку в трее **** procedure CreateTray; begin with tray do begin cbsize :=
SizeOf(TNotifylconData); wnd := Form1.Handle; uid := 0; uflags := nif_icon or
nif_message or nif_tip; ucallbackmessage := wm_notifytrayicon; //обрабатываемое
сообщение hicon := GetHIcon('netshell.dll', 112); sztip := ('test'); //хинт end;
shell_notifyicon(nim_add, @tray); //добавить в трей end; //*****удаляем иконку в трее ****
procedure DeleteTray; begin Shell_NotifyIcon(nim_delete, @tray); //удалить из трея end;
```

Для обработки клика по трею используется стандартная обработка созданного нами сообщения.

С иконками разобрались, вывод анимации ещё проще. Переносим на форму компонент вывода анимации *Animate1*. Открываем DLL, в которой содержатся ресурсы:

```
var hlib: integer; hlib:= LoadLibrary(pchar(getpath('shell32.dll')); SendDlgItemMessage(
Handle, Animate1.Handle, wm_user + 100, hlib, 166); //открываем анимацию ресурса
id:= 166; SendDlgItemMessage( Handle, Animate1.Handle, wm_user + 101, 2,
makelong(1, 100)); //проигрываем два раза с 1 по 100 кадр // так как не знаем точного
количества кадров
```

Следующий вопрос, вывод диалоговых окон, делится на два этапа:

Создание из ресурса диалогового окна

```
var hlib: integer; mainwnd1: hwnd; begin lib:=
LoadLibrary(pchar(GetPath('shell32.dll')); mainwnd1:= CreateDialog(hLib,
MakeIntResource(28672), 0, nil); //создаём окно форматирования диска
ShowWindow(mainwnd1, sw_shownormal); //отображаем окно while GetMessage(msg,
0, 0, 0) do begin WindowDlg(mainwnd1, msg); //обработчик оконной процедуры
DispatchMessage(msg); TranslateMessage(msg); end; end; Обработка
```

сообщений компонентов диалогового окна

```
const pbm_setpos = wm_user+2; function WindowDlg(wnd: hwnd; msg: tmsg): IResult;
//обработчик оконной процедуры var inf: integer; begin {далее происходит цикл
обработки сообщений} case msg.message of wm_command: begin //удаление окна
PostQuitMessage(0); DestroyWindow(wnd); exit; end; wm_lbuttondown: case
GetWindowLong(msg.hwnd, gwl_id) of //нажатие левой клавиши определяем id
```

```
компонента 1: begin //нажатие клавиши "Начать"  SendDlgItemMessage(wnd, 28678 ,  
rbm_setpos, 50, 0); //сдвигает процесс бар на 50  end; end;
```

Для определения **ID** компонентов диалогового окна необходимо в обработчик оконной процедуры добавить функцию

```
GetWindowLong (msg.hwnd, gwl_id);
```

По материалам <http://www.interface.ru>