

# Описание IDL32.DLL

---

1. Введение
2. Описание сообщений
3. Описание `configure.ini`
4. Минимальные требования
5. Авторское право

## Введение

В данном материале приведена информация, по средствам которой вы сможете использовать компонент LIVE\_PANEL. Когда я начинал писать библиотеку, я не представлял, насколько сильно нужно обдумывать логику, каждое звено в коде было связано с 3, а то и 10 блоками оконных обработчиков сообщений. Цель была в том, что бы создать универсальную для всех языков программирования простую графическую систему позволяющую видеть прогресс, скорость и другую информацию в компоненте о закачиваемом файле из интернет сети. Так же большая часть усилий пошла на оптимизацию, дабы получить размер DLL не больше 20 кб. И вот в результате вы получили долгожданный результат.

## Описание сообщений

Компонент использует простую для понимания систему сообщений. В программе обязательно нужно произвести инициализацию компонента и деструкцию.

Пример для Delphi:

```
var
  Form1      : TForm1;
  hpct       : HWND;
  hpan       : HWND;

procedure TForm1.FormCreate(Sender: TObject);
begin
  hpct:= Loadlibrary('IDL32.dll');
  hpan:= CreateWindowEx(WS_EX_OVERLAPPEDWINDOW , 'LIVE_PANEL', nil,
    WS_CHILD+WS_VISIBLE+WS_VSCROLL, 0, 0, width, height,
    form1.handle, 0, Getmodulehandle(nil), nil);
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
  DestroyWindow(hpan);
  FreeLibrary(hpct);
end;
```

Для того что бы добавить задание вы можете использовать сообщение FM\_DOWNLOAD в **lparam** передаете адрес на имя файла для сохранения, а в **wparam** URL страницы или файла на сервере.

```
procedure TForm1.N2Click(Sender: TObject);
const
  FM_DOWNLOAD = WM_USER + 1;
begin
  SendMessage(hpan, FM_DOWNLOAD, Longint(url), Longint(filename));
end;
```

Так же для работы с очередью предусмотрено еще три сообщения:

**FM\_WORKS** = **WM\_USER + 11**

данное сообщение заставляет компонент пересмотреть очередь.

**FM\_SET\_QUEUE** = **WM\_USER + 14**

устанавливает максимальное значение одновременно работающих закачек, при этом число должно передаваться в lparam. После FM\_SET\_QUEUE желательно послать FM\_WORKS.

**FM\_GET\_QUEUE** = **WM\_USER + 15**

возвращается установленное на данный момент число допустимых задач в очереди.

**FM\_GET SOCK\_MODE** = **WM\_USER + 18**

Возвращает нуль при выключенном SOCKS или один при включенном.

**FM\_SET SOCK\_MODE** = **WM\_USER + 19**

В lparam передается режим 1 или 0, включить или выключить SOCKS.

### *Описание configure.ini*

Если файл конфигурации будет не найден, подфункция выведет сообщение на экран с описанием ошибки и примет настройки по умолчанию. Максимально хотелось сократить файл и сделать его удобочитаемым:

<pre>[ОЧЕРЕДЬ] РАЗРЕШАТЬ ЗАКАЧЕК=1 [ЯДРО] РАЗРЕШАТЬ SOCKS=0</pre>
---

Как видно из листинга сохраняются только два параметра, рассмотрим подробнее:

В ключе «РАЗРЕШАТЬ ЗАКАЧЕК» может быть любое число от 1 до 1024. В ключе «РАЗРЕШАТЬ SOCKS» вы можете установить только два значения это 0 «без SOCKS» и 1 «Включить SOCKS».

### *Минимальные требования*

Для правильной работы в LINUX требуется WINE 1.1.6.

Для семейства операционных систем от Microsoft требуется: ядро не меньше NT 4.0, библиотеки wsock32.dll версии не меньше 4.x.x.x. и wininet.dll желательно версии 5.x.x.x

### *Авторское право*

Все авторские права принадлежат Сеницыну Антону Алексеевичу.

За подробной информацией обращайтесь на форум по адресу <http://fasm.moy.su/> или на электронную почту [Everhest1@yandex.ru](mailto:Everhest1@yandex.ru)