c# WndProc事件 消息类型

转载 weixin_34272308 于 2019-04-14 17:57:00 发布 阅读量596 收藏 4 点赞数 3
文章标签： 操作系统

转载：https://www.cnblogs.com/idben/p/3783997.html

WM_NULL = 0x0000;

WM_CREATE = 0x0001;
应用程序创建一个窗口

WM_DESTROY = 0x0002;
一个窗口被销毁

WM_MOVE = 0x0003;
移动一个窗口

WM_SIZE = 0x0005;
改变一个窗口的大小

WM_ACTIVATE = 0x0006;
一个窗口被激活或失去激活状态；

WM_SETFOCUS = 0x0007;
获得焦点后

WM_KILLFOCUS = 0x0008;
失去焦点

WM_ENABLE = 0x000A;
改变enable状态

WM_SETREDRAW = 0x000B;
设置窗口是否能重画

WM_SETTEXT = 0x000C;
应用程序发送此消息来设置一个窗口的文本

WM_GETTEXT = 0x000D;
应用程序发送此消息来复制对应窗口的文本到缓冲区

WM_GETTEXTLENGTH = 0x000E;
得到与一个窗口有关的文本的长度（不包含空字符）

WM_PAINT = 0x000F;
要求一个窗口重画自己

WM_CLOSE = 0x0010;
当一个窗口或应用程序要关闭时发送一个信号

WM_QUERYENDSESSION = 0x0011;
当用户选择结束对话框或程序自己调用ExitWindows函数

WM_QUIT = 0x0012;
用来结束程序运行或当程序调用postquitmessage函数

WM_QUERYOPEN = 0x0013;
当用户窗口恢复以前的大小位置时，把此消息发送给某个图标

WM_ERASEBKGND = 0x0014;
当窗口背景必须被擦除时（例在窗口改变大小时）

WM_SYSCOLORCHANGE = 0x0015;
当系统颜色改变时，发送此消息给所有顶级窗口

WM_ENDSESSION = 0x0016;
当系统进程发出WM_QUERYENDSESSION消息后，此消息发送给应用程序，

WM_SYSTEMERROR = 0x0017;

WM_SHOWWINDOW = 0x0018;
当隐藏或显示窗口是发送此消息给这个窗口

WM_ACTIVATEAPP = 0x001C;
发此消息给应用程序哪个窗口是激活的，哪个是非激活的；

WM_FONTCHANGE = 0x001D;
当系统的字体资源库变化时发送此消息给所有顶级窗口

WM_TIMECHANGE = 0x001E;
当系统的时间变化时发送此消息给所有顶级窗口

WM_CANCELMODE = 0x001F;
发送此消息来取消某种正在进行的摸态（操作）

WM_SETCURSOR = 0x0020;
如果鼠标引起光标在某个窗口中移动且鼠标输入没有被捕获时，就发消息给某个窗口

WM_MOUSEACTIVATE = 0x0021;
当光标在某个非激活的窗口中而用户正按着鼠标的某个键发送此消息给当前窗口

WM_CHILDACTIVATE = 0x0022;
发送此消息给MDI子窗口当用户点击此窗口的标题栏，或当窗口被激活，移动，改变大小

WM_QUEUESYNC = 0x0023;
此消息由基于计算机的训练程序发送，通过WH_JOURNALPALYBACK的hook程序

WM_GETMINMAXINFO = 0x0024;
此消息发送给窗口当它将要改变大小或位置；

WM_PAINTICON = 0x0026;
发送给最小化窗口当它图标将要被重画

WM_ICONERASEBKGND = 0x0027;
此消息发送给某个最小化窗口，仅当它在画图标前它的背景必须被重画

WM_NEXTDLGCTL = 0x0028;
发送此消息给一个对话框程序去更改焦点位置

WM_SPOOLERSTATUS = 0x002A;
每当打印管理列队增加或减少一条作业时发出此消息

WM_DRAWITEM = 0x002B;
当button，combobox，listbox，menu的可视外观改变时发送 此消息给这些空件的所有者

WM_MEASUREITEM = 0x002C;
当button, combo box, list box, list view control, or menu item 被创建时 发送此消息给控件的所有者

WM_DELETEITEM = 0x002D;
当the list box 或 combo box 被销毁 或 当 某些项被删除通过LB_DELETESTRING, LB_RESETCONTENT, CB_DELETESTRING, or CB_RESETCONTENT 消息

WM_VKEYTOITEM = 0x002E;
此消息有一个LBS_WANTKEYBOARDINPUT风格的发出给它的所有者来响应WM_KEYDOWN消息

WM_CHARTOITEM = 0x002F;
此消息由一个LBS_WANTKEYBOARDINPUT风格的列表框发送给他的所有者来响应WM_CHAR消息

WM_SETFONT = 0x0030;
当绘制文本时程序发送此消息得到控件要用的颜色

WM_GETFONT = 0x0031;
应用程序发送此消息得到当前控件绘制文本的字体

WM_SETHOTKEY = 0x0032;
应用程序发送此消息让一个窗口与一个热键相关连

WM_GETHOTKEY = 0x0033;
应用程序发送此消息来判断热键与某个窗口是否有关联

WM_QUERYDRAGICON = 0x0037;
此消息发送给最小化窗口，当此窗口将要被拖放而它的类中没有定义图标，应用程序能返回一个图标或光标的句柄，当用户拖放图标时系统显示这个

WM_COMPAREITEM = 0x0039;
发送此消息来判定combobox或listbox新增加的项的相对位置

WM_GETOBJECT = 0x003D;
WM_COMPACTING = 0x0041;
显示内存已经很少了

WM_WINDOWPOSCHANGING = 0x0046;
发送此消息给那个窗口的大小和位置将要被改变时，来调用setwindowpos函数或其它窗口管理函数

WM_WINDOWPOSCHANGED = 0x0047;
发送此消息给那个窗口的大小和位置已经被改变时，来调用setwindowpos函数或其它窗口管理函数

WM_POWER = 0x0048;（适用于16位的windows）
当系统将要进入暂停状态时发送此消息

WM_COPYDATA = 0x004A;
当一个应用程序传递数据给另一个应用程序时发送此消息

WM_CANCELJOURNAL = 0x004B;
当某个用户取消程序日志激活状态，提交此消息给程序

WM_NOTIFY = 0x004E;
当某个控件的某个事件已经发生或这个控件需要得到一些信息时，发送此消息给它的父窗口

WM_INPUTLANGCHANGEREQUEST = 0x0050;
当用户选择某种输入语言，或输入语言的热键改变

WM_INPUTLANGCHANGE = 0x0051;
当平台现场已经被改变后发送此消息给受影响的最顶级窗口

WM_TCARD = 0x0052;
当程序已经初始化windows帮助例程时发送此消息给应用程序

WM_HELP = 0x0053;
此消息显示用户按下了F1，如果某个菜单是激活的，就发送此消息个此窗口关联的菜单，否则就
发送给有焦点的窗口，如果当前都没有焦点，就把此消息发送给当前激活的窗口

WM_USERCHANGED = 0x0054;
当用户已经登入或退出后发送此消息给所有的窗口，当用户登入或退出时系统更新用户的具体
设置信息，在用户更新设置时系统马上发送此消息；

WM_NOTIFYFORMAT = 0x0055;
公用控件，自定义控件和他们的父窗口通过此消息来判断控件是使用ANSI还是UNICODE结构
在WM_NOTIFY消息，使用此控件能使某个控件与它的父控件之间进行相互通信

WM_CONTEXTMENU = 0x007B;
当用户某个窗口中点击了一下右键就发送此消息给这个窗口

WM_STYLECHANGING = 0x007C;
当调用SETWINDOWLONG函数将要改变一个或多个 窗口的风格时发送此消息给那个窗口

WM_STYLECHANGED = 0x007D;
当调用SETWINDOWLONG函数一个或多个 窗口的风格后发送此消息给那个窗口

WM_DISPLAYCHANGE = 0x007E;
当显示器的分辨率改变后发送此消息给所有的窗口

WM_GETICON = 0x007F;
此消息发送给某个窗口来返回与某个窗口有关连的大图标或小图标的句柄；

WM_SETICON = 0x0080;
程序发送此消息让一个新的大图标或小图标与某个窗口关联；

WM_NCCREATE = 0x0081;
当某个窗口第一次被创建时，此消息在WM_CREATE消息发送前发送；

WM_NCDESTROY = 0x0082;
此消息通知某个窗口，非客户区正在销毁

WM_NCCALCSIZE = 0x0083;
当某个窗口的客户区域必须被核算时发送此消息

WM_NCHITTEST = 0x0084;
移动鼠标，按住或释放鼠标时发生

WM_NCPAINT = 0x0085;
程序发送此消息给某个窗口当它（窗口）的框架必须被绘制时；

WM_NCACTIVATE = 0x0086;
此消息发送给某个窗口 仅当它的非客户区需要被改变来显示是激活还是非激活状态；

WM_GETDLGCODE = 0x0087;
发送此消息给某个与对话框程序关联的控件，widdows控制方位键和TAB键使输入进入此控件
通过响应WM_GETDLGCODE消息，应用程序可以把他当成一个特殊的输入控件并能处理它

WM_NCMOUSEMOVE = 0x00A0;
当光标在一个窗口的非客户区内移动时发送此消息给这个窗口 ，非客户区为,窗体的标题栏及窗的边框体

WM_NCLBUTTONDOWN = 0x00A1;
当光标在一个窗口的非客户区同时按下鼠标左键时提交此消息

WM_NCLBUTTONUP = 0x00A2;
当用户释放鼠标左键同时光标某个窗口在非客户区十发送此消息；

WM_NCLBUTTONDBLCLK = 0x00A3;
当用户双击鼠标左键同时光标某个窗口在非客户区十发送此消息

WM_NCRBUTTONDOWN = 0x00A4;
当用户按下鼠标右键同时光标又在窗口的非客户区时发送此消息

WM_NCRBUTTONUP = 0x00A5;
当用户释放鼠标右键同时光标又在窗口的非客户区时发送此消息

WM_NCRBUTTONDBLCLK = 0x00A6;
当用户双击鼠标右键同时光标某个窗口在非客户区十发送此消息

WM_NCMBUTTONDOWN = 0x00A7;
当用户按下鼠标中键同时光标又在窗口的非客户区时发送此消息

WM_NCMBUTTONUP = 0x00A8;
当用户释放鼠标中键同时光标又在窗口的非客户区时发送此消息

WM_NCMBUTTONDBLCLK = 0x00A9;
当用户双击鼠标中键同时光标又在窗口的非客户区时发送此消息

WM_KEYFIRST = 0x0100;
WM_KEYDOWN = 0x0100;
//按下一个键

WM_KEYUP = 0x0101;
//释放一个键

WM_CHAR = 0x0102;
//按下某键，并已发出WM_KEYDOWN， WM_KEYUP消息

WM_DEADCHAR = 0x0103;
当用translatemessage函数翻译WM_KEYUP消息时发送此消息给拥有焦点的窗口

WM_SYSKEYDOWN = 0x0104;
当用户按住ALT键同时按下其它键时提交此消息给拥有焦点的窗口；

WM_SYSKEYUP = 0x0105;
当用户释放一个键同时ALT 键还按着时提交此消息给拥有焦点的窗口

WM_SYSCHAR = 0x0106;
当WM_SYSKEYDOWN消息被TRANSLATEMESSAGE函数翻译后提交此消息给拥有焦点的窗口

WM_SYSDEADCHAR = 0x0107;
当WM_SYSKEYDOWN消息被TRANSLATEMESSAGE函数翻译后发送此消息给拥有焦点的窗口

WM_KEYLAST = 0x0108;
WM_INITDIALOG = 0x0110;
在一个对话框程序被显示前发送此消息给它，通常用此消息初始化控件和执行其它任务

WM_COMMAND = 0x0111;
当用户选择一条菜单命令项或当某个控件发送一条消息给它的父窗口，一个快捷键被翻译

WM_SYSCOMMAND = 0x0112;
当用户选择窗口菜单的一条命令或当用户选择最大化或最小化时那个窗口会收到此消息

WM_TIMER = 0x0113; //发生了定时器事件
WM_HSCROLL = 0x0114;
当一个窗口标准水平滚动条产生一个滚动事件时发送此消息给那个窗口，也发送给拥有它的控件

WM_VSCROLL = 0x0115;
当一个窗口标准垂直滚动条产生一个滚动事件时发送此消息给那个窗口也，发送给拥有它的控件

WM_INITMENU = 0x0116;
当一个菜单将要被激活时发送此消息，它发生在用户菜单条中的某项或按下某个菜单键，它允许程序在显示前更改菜单

WM_INITMENUPOPUP = 0x0117;
当一个下拉菜单或子菜单将要被激活时发送此消息，它允许程序在它显示前更改菜单，而不要改变全部

WM_MENUSELECT = 0x011F;
当用户选择一条菜单项时发送此消息给菜单的所有者（一般是窗口）

WM_MENUCHAR = 0x0120;
当菜单已被激活用户按下了某个键（不同于加速键），发送此消息给菜单的所有者；

WM_ENTERIDLE = 0x0121;
当一个模态对话框或菜单进入空载状态时发送此消息给它的所有者，一个模态对话框或菜单进入空载状态就是在处理完一条或几条先前的消息后没有队中等待

WM_MENURBUTTONUP = 0x0122;
WM_MENUDRAG = 0x0123;
WM_MENUGETOBJECT = 0x0124;
WM_UNINITMENUPOPUP = 0x0125;
WM_MENUCOMMAND = 0x0126;
WM_CHANGEUISTATE = 0x0127;
WM_UPDATEUISTATE = 0x0128;
WM_QUERYUISTATE = 0x0129;
WM_CTLCOLORMSGBOX = 0x0132;
在windows绘制消息框前发送此消息给消息框的所有者窗口，通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置消息框的颜色

WM_CTLCOLOREDIT = 0x0133;
当一个编辑型控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置编辑框的颜色

WM_CTLCOLORLISTBOX = 0x0134;
当一个列表框控件将要被绘制前发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置列表框的颜色

WM_CTLCOLORBTN = 0x0135;
当一个按钮控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置按纽的文本

WM_CTLCOLORDLG = 0x0136;
当一个对话框控件将要被绘制前发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置对话框的颜色

WM_CTLCOLORSCROLLBAR= 0x0137;
当一个滚动条控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置滚动条的

WM_CTLCOLORSTATIC = 0x0138;
当一个静态控件将要被绘制时发送此消息给它的父窗口；通过响应这条消息，所有者窗口可以通过使用给定的相关显示设备的句柄来设置静态控件的颜色

WM_MOUSEFIRST = 0x0200;

WM_MOUSEMOVE = 0x0200;
// 移动鼠标
WM_LBUTTONDOWN = 0x0201;
//按下鼠标左键

WM_LBUTTONUP = 0x0202;
//释放鼠标左键

WM_LBUTTONDBLCLK = 0x0203;
//双击鼠标左键

WM_RBUTTONDOWN = 0x0204;
//按下鼠标右键

WM_RBUTTONUP = 0x0205;
//释放鼠标右键

WM_RBUTTONDBLCLK = 0x0206;
//双击鼠标右键

WM_MBUTTONDOWN = 0x0207;
//按下鼠标中键

WM_MBUTTONUP = 0x0208;
//释放鼠标中键

WM_MBUTTONDBLCLK = 0x0209;
//双击鼠标中键

WM_MOUSEWHEEL = 0x020A;
当鼠标轮子转动时发送此消息个当前有焦点的控件

WM_MOUSELAST = 0x020A;
WM_PARENTNOTIFY = 0x0210;
当MDI子窗口被创建或被销毁，或用户按了一下鼠标键而光标在子窗口上时发送此消息给它的父窗口

WM_ENTERMENULOOP = 0x0211;
发送此消息通知应用程序的主窗口that已经进入了菜单循环模式

WM_EXITMENULOOP = 0x0212;
发送此消息通知应用程序的主窗口that已退出了菜单循环模式

WM_NEXTMENU = 0x0213;
WM_SIZING = 532;
当用户正在调整窗口大小时发送此消息给窗口；通过此消息应用程序可以监视窗口大小和位置也可以修改他们

WM_CAPTURECHANGED = 533;
发送此消息 给窗口当它失去捕获的鼠标时；

WM_MOVING = 534;
当用户在移动窗口时发送此消息，通过此消息应用程序可以监视窗口大小和位置也可以修改他们；

WM_POWERBROADCAST = 536;
此消息发送给应用程序来通知它有关电源管理事件；

WM_DEVICECHANGE = 537;
当设备的硬件配置改变时发送此消息给应用程序或设备驱动程序

WM_IME_STARTCOMPOSITION = 0x010D;
WM_IME_ENDCOMPOSITION = 0x010E;

WM_IME_COMPOSITION = 0x010F;
WM_IME_KEYLAST = 0x010F;
WM_IME_SETCONTEXT = 0x0281;
WM_IME_NOTIFY = 0x0282;
WM_IME_CONTROL = 0x0283;
WM_IME_COMPOSITIONFULL = 0x0284;
WM_IME_SELECT = 0x0285;
WM_IME_CHAR = 0x0286;
WM_IME_REQUEST = 0x0288;
WM_IME_KEYDOWN = 0x0290;
WM_IME_KEYUP = 0x0291;
WM_MDICREATE = 0x0220;
应用程序发送此消息给多文档的客户窗口来创建一个MDI 子窗口

WM_MDIDESTROY = 0x0221;
应用程序发送此消息给多文档的客户窗口来关闭一个MDI 子窗口

WM_MDIACTIVATE = 0x0222;
应用程序发送此消息给多文档的客户窗口通知客户窗口激活另一个MDI子窗口，当客户窗口收到此消息后，它发出WM_MDIACTIVE消息给MDI子窗口（未它；

WM_MDIRESTORE = 0x0223;
程序 发送此消息给MDI客户窗口让子窗口从最大最小化恢复到原来大小

WM_MDINEXT = 0x0224;
程序 发送此消息给MDI客户窗口激活下一个或前一个窗口

WM_MDIMAXIMIZE = 0x0225;
程序发送此消息给MDI客户窗口来最大化一个MDI子窗口；

WM_MDITILE = 0x0226;
程序 发送此消息给MDI客户窗口以平铺方式重新排列所有MDI子窗口

WM_MDICASCADE = 0x0227;
程序 发送此消息给MDI客户窗口以层叠方式重新排列所有MDI子窗口

WM_MDIICONARRANGE = 0x0228;
程序 发送此消息给MDI客户窗口重新排列所有最小化的MDI子窗口

WM_MDIGETACTIVE = 0x0229;
程序 发送此消息给MDI客户窗口来找到激活的子窗口的句柄

WM_MDISETMENU = 0x0230;
程序 发送此消息给MDI客户窗口用MDI菜单代替子窗口的菜单

WM_ENTERSIZEMOVE = 0x0231;
WM_EXITSIZEMOVE = 0x0232;
WM_DROPFILES = 0x0233;
WM_MDIREFRESHMENU = 0x0234;
WM_MOUSEHOVER = 0x02A1;
WM_MOUSELEAVE = 0x02A3;
WM_CUT = 0x0300;
程序发送此消息给一个编辑框或combobox来删除当前选择的文本

WM_COPY = 0x0301;
程序发送此消息给一个编辑框或combobox来复制当前选择的文本到剪贴板

WM_PASTE = 0x0302;
程序发送此消息给editcontrol或combobox从剪贴板中得到数据

WM_CLEAR = 0x0303;
程序发送此消息给editcontrol或combobox清除当前选择的内容

WM_UNDO = 0x0304;
程序发送此消息给editcontrol或combobox撤消最后一次操作
WM_RENDERFORMAT = 0x0305;

WM_RENDERALLFORMATS = 0x0306;
WM_DESTROYCLIPBOARD = 0x0307;
当调用ENPTYCLIPBOARD函数时 发送此消息给剪贴板的所有者

WM_DRAWCLIPBOARD = 0x0308;
当剪贴板的内容变化时发送此消息给剪贴板观察链的第一个窗口；它允许用剪贴板观察窗口来
显示剪贴板的新内容；

WM_PAINTCLIPBOARD = 0x0309;
当剪贴板包含CF_OWNERDIPLAY格式的数据并且剪贴板观察窗口的客户区需要重画；

WM_VSCROLLCLIPBOARD = 0x030A;
WM_SIZECLIPBOARD = 0x030B;
当剪贴板包含CF_OWNERDIPLAY格式的数据并且剪贴板观察窗口的客户区域的大小已经改变是此消息通过剪贴板观察窗口发送给剪贴板的所有者；

WM_ASKCBFORMATNAME = 0x030C;
通过剪贴板观察窗口发送此消息给剪贴板的所有者来请求一个CF_OWNERDISPLAY格式的剪贴板的名字

WM_CHANGECBCHAIN = 0x030D;
当一个窗口从剪贴板观察链中移去时发送此消息给剪贴板观察链的第一个窗口；

WM_HSCROLLCLIPBOARD = 0x030E;
此消息通过一个剪贴板观察窗口发送给剪贴板的所有者 ；它发生在当剪贴板包含CFOWNERDISPALY格式的数据并且有个事件在剪贴板观察窗的水平滚动
者应滚动剪贴板图象并更新滚动条的值；

WM_QUERYNEWPALETTE = 0x030F;
此消息发送给将要收到焦点的窗口，此消息能使窗口在收到焦点时同时有机会实现他的逻辑调色板

WM_PALETTEISCHANGING= 0x0310;
当一个应用程序正要实现它的逻辑调色板时发此消息通知所有的应用程序

WM_PALETTECHANGED = 0x0311;
此消息在一个拥有焦点的窗口实现它的逻辑调色板后发送此消息给所有顶级并重叠的窗口，以此来改变系统调色板

WM_HOTKEY = 0x0312;
当用户按下由REGISTERHOTKEY函数注册的热键时提交此消息

WM_PRINT = 791;
应用程序发送此消息仅当WINDOWS或其它应用程序发出一个请求要求绘制一个应用程序的一部分；

WM_PRINTCLIENT = 792;
WM_HANDHELDFIRST = 856;
WM_HANDHELDLAST = 863;
WM_PENWINFIRST = 0x0380;
WM_PENWINLAST = 0x038F;
WM_COALESCE_FIRST = 0x0390;
WM_COALESCE_LAST = 0x039F;
WM_DDE_FIRST = 0x03E0;
WM_DDE_INITIATE = WM_DDE_FIRST + 0;
一个DDE客户程序提交此消息开始一个与服务器程序的会话来响应那个指定的程序和主题名；

WM_DDE_TERMINATE = WM_DDE_FIRST + 1;
一个DDE应用程序（无论是客户还是服务器）提交此消息来终止一个会话；

WM_DDE_ADVISE = WM_DDE_FIRST + 2;
一个DDE客户程序提交此消息给一个DDE服务程序来请求服务器每当数据项改变时更新它

WM_DDE_UNADVISE = WM_DDE_FIRST + 3;
一个DDE客户程序通过此消息通知一个DDE服务程序不更新指定的项或一个特殊的剪贴板格式的项

WM_DDE_ACK = WM_DDE_FIRST + 4;
此消息通知一个DDE（动态数据交换）程序已收到并正在处理WM_DDE_POKE, WM_DDE_EXECUTE, WM_DDE_DATA, WM_DDE_ADVISE, WM_DDE_UNADVISE, or
WM_DDE_INITIAT消息

WM_DDE_DATA = WM_DDE_FIRST + 5;
一个DDE服务程序提交此消息给DDE客户程序来传递个一数据项给客户或通知客户的一条可用数据项

WM_DDE_REQUEST = WM_DDE_FIRST + 6;
一个DDE客户程序提交此消息给一个DDE服务程序来请求一个数据项的值；

WM_DDE_POKE = WM_DDE_FIRST + 7;
一个DDE客户程序提交此消息给一个DDE服务程序，客户使用此消息来请求服务器接收一个未经同意的数据项；服务器通过答复WM_DDE_ACK消息提示是个数据项；

WM_DDE_EXECUTE = WM_DDE_FIRST + 8;
一个DDE客户程序提交此消息给一个DDE服务程序来发送一个字符串给服务器让它象串行命令一样被处理，服务器通过提交WM_DDE_ACK消息来作回应；

WM_DDE_LAST = WM_DDE_FIRST + 8;
WM_APP = 0x8000;
WM_USER = 0x0400;
此消息能帮助应用程序自定义私有消息；

通知消息(Notification message)是指这样一种消息，一个窗口内的子控件发生了一些事情，
需要通知父窗口。通知消息只适用于标准的窗口控件如按钮、列表框、组合框、编辑框，
以及Windows 95公共控件如树状视图、列表视图等。
例如，单击或双击一个控件、在控件中选择部分文本、操作控件的滚动条都会产生通知消息。
按扭
B N _ C L I C K E D //用户单击了按钮
B N _ D I S A B L E //按钮被禁止
B N _ D O U B L E C L I C K E D //用户双击了按钮
B N _ H I L I T E //用户加亮了按钮
B N _ PA I N T按钮应当重画
B N _ U N H I L I T E加亮应当去掉
组合框
C B N _ C L O S E U P组合框的列表框被关闭
C B N _ D B L C L K用户双击了一个字符串
C B N _ D R O P D O W N组合框的列表框被拉出
C B N _ E D I T C H A N G E用户修改了编辑框中的文本
C B N _ E D I T U P D AT E编辑框内的文本即将更新
C B N _ E R R S PA C E组合框内存不足
C B N _ K I L L F O C U S组合框失去输入焦点
C B N _ S E L C H A N G E在组合框中选择了一项
C B N _ S E L E N D C A N C E L用户的选择应当被取消
C B N _ S E L E N D O K用户的选择是合法的
C B N _ S E T F O C U S组合框获得输入焦点
编辑框
E N _ C H A N G E编辑框中的文本已更新
E N _ E R R S PA C E编辑框内存不足
E N _ H S C R O L L用户点击了水平滚动条
E N _ K I L L F O C U S编辑框正在失去输入焦点
E N _ M A X T E X T插入的内容被截断
E N _ S E T F O C U S编辑框获得输入焦点
E N _ U P D AT E编辑框中的文本将要更新
E N _ V S C R O L L用户点击了垂直滚动条消息含义
列表框
L B N _ D B L C L K用户双击了一项
L B N _ E R R S PA C E列表框内存不够
L B N _ K I L L F O C U S列表框正在失去输入焦点
L B N _ S E L C A N C E L选择被取消
L B N _ S E L C H A N G E选择了另一项
L B N _ S E T F O C U S列表框获得输入焦点

转载于:https://www.cnblogs.com/gaara-zhang/p/10706224.html

Windows消息大全

最新整理Windows各类函数消息大全， Windows是一消息（Message）驱动式系统，Windows消息提供了应用程序与应用程序之间、应用程序与Windows系统之间进行

消息映射的实现_vc# 2022 wndproc函数 message相关种类与映射表-CSDN...

Windows维护一个系统消息队列(System message queue),每个GUI线程有一个线程消息队列(Thread message queue)。 鼠标、键盘事件由鼠标或键盘驱动程序转换成输入

C# 监测 Windows 设备变动事件_设备已添加到系统或从系统中删除。-CSDN...

本程序通过WPF窗口的 WindowProc 函数处理Windows的硬件或配置改变的事件。开发环境为VS 2022。 基础信息 硬件或配置改变的基础有以下内容: 消息: WM_DEVIC

windows消息大全

windows消息大全。里面含有一些windows经常处理的一些消息

Windows窗口消息大全,全不全自己看 zmq5411的

转载地址：http://www.cnblogs.com/Sunwayking/articles/2817580.html 1 Windows窗口消息大全,全不全自己看 2 3 // 4 #incl

c#中的消息处理函数和vc中的消息处理函数区别_vc中,pretranslatemessage...

对于WndProc(跟vc中的windowproc名字稍有改变)函数来说: protected override void WndProc(ref Message msg) { if(msg.Msg==0x101) { MessageBox.Show(msg.HWnd

介绍Windows的窗口、消息、子类化和超类化_c# afxwnd100

static LRESULT CALLBACK WindowProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) { switch (uMsg) { case WM_DESTROY: PostQuitMessage(0

C# 重写WndProc 拦截 发送 系统消息 + windows消息常量值

通过重载虚函数WndProc在C#下处理Windows系统消息。

C#获取USB事件API实例分析

本文实例讲述了C#获取USB事件API。分享给大家供大家参考。具体如下： const int WM_DEVICECHANGE = 0x2190; const int DBT_DEVICEARRIVAL = 0x8000; const

wParam与lParam的区别

lParam 和 wParam 是宏定义,一般在消息函数中带这两个类型的参数,通常用来存储窗口消息的参数。 LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPA

创建Windows窗体 : WinMain() 与 WndProc()

ShowWindow (hwnd,iCmdShow);//显示窗口UpdateWindow (hwnd) ;//更新窗体while(GetMessage(&msg,NULL,0,0)) { TranslateMessage (&msg);//翻译消息并发送到wind

c# 重载WndProc，实现重写“最小化”的实现方法

在做“亦歌桌面版”的时候，发现当打开歌词状态下，用最小化隐藏窗体到托盘的话（如下code #1），在调出发现歌词缩小了（虽然显现的窗体大小跟刚才一样），从这点

c# winform 鼠标、键盘消息拦截

vs2013 c# winform ， PreFilterMessage拦截本程序内的鼠标消息，获取鼠标与键盘钩子获取鼠标与键盘消息，或者取消键盘消息

转载 Windows消息 16进制编码(1)C#API anivdqdt84098的

Windows消息 16进制编码 WM_NULL = 0x0000, WM_CREATE = 0x0001, WM_DESTROY = 0x0002, WM_MOVE = 0x0003, WM_SIZE = 0x0005, WM_ACTIVATE = 0x0

API查询工具,API函数查询

可以查询一些知道功能但不会定义的api函数（需要安装.net1.0）

关于WindowsAPI常数的值 Yentre的

键盘扫描码(十六进制------十进制) VK_LBUTTON----------01----------1----------鼠标的左键 VK_RBUTTON----------02----------2----------鼠标的右键 VK_CANCEL----------03------

VC中的消息 水上漂的

VC中的消息 消息，就是Windows向应用程序发出的通知。事件，就是你对硬件设备的操作。例如，当你单击鼠标、敲击键盘等，这本身就是一个事件的发生，Windo

Windows中的消息详细列表(附常量值) lxxlql的

消息，就是指Windows发出的一个通知，告诉应用程序某个事情发生了。例如，单击鼠标、改变窗口尺寸、按下键盘上的一个键都会使Windows发送一个消息给应用程序，

C# winform启动外部exe后，如何完全阻断父界面接收事件，扩展waitforexit weixin_30607659的

公司的系统搭载了好多奇奇怪怪的exe，以前启动exe后，系统还能接着操作。但是后面又提出额外的需求，说是打开外部exe之后，启动exe的父界面要完全不能进行任何

Windows系统消息大全使用详细介绍 weixin_34384681

【IT168 编程开发】 消息，就是指Windows发出的一个通知，告诉应用程序某个事情发生了。例如，单击鼠标、改变窗口尺寸、按下键盘上的一个键都会使Windows发送

Windows消息 16进制编码(1)C#API wangxiaona356的

Windows消息 16进制编码 WM_NULL = 0x0000, WM_CREATE = 0x0001, WM_DESTROY = 0x0002, WM_MOVE = 0x0003, W

[转] C#，WinForm中的KeyDown,KeyPress 和KeyUp事件相关知识点 Sky的

Windows窗体通过引发键盘事件来处理键盘输入以响应Windows消息，大多数Windows窗体应用程序都通过处理键盘事件来以独占方式处理键盘输入。 1．按键的类型

C# WndProc是什么事件 **最新发布**

C是一种强大的编程语言，被广泛用于操作系统、数据库、...C语言中常用的数据类型包括整型、浮点型和字符型等，同时也支持指针和数组等数据结构。C语言由Dennis

weixin_34272308
码龄8年 暂无认证

157	-	178万+	123万+	
原创	周排名	总排名	访问	等级
7315	4808	206	16	1268
积分	粉丝	获赞	评论	收藏

私信 关注

热门文章

最新评论

UIActivityViewController系统原生分享

Nu_: 大佬你好，UIActivityViewController如何分享和显示摘要部分的内容

初次使用pycharm 的interpreter option为…

quanquancutest: 您好 求问我的package全是空的 该怎么添加interpreter呀 谢谢

linux testparm(test parameter) 命令详解

我想尽情享受你: testparm未找到命令，怎么回事 楼主

基于Vue+Vuex+iView的电子商城网站

路新天: 后台怎么运行

您愿意向朋友推荐“博客详情页”吗?

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

在Linux下编译APUE的例子

[Microsoft.BizTalk.Deployment.DeploymentException] Assembly "AutoProcess, Version=1.0.0.0, Culture=n...

Calling an Application Engine from PeopleCode

2019年 370篇	2018年 686篇
2017年 907篇	2016年 491篇
2015年 450篇	2014年 306篇
2013年 327篇	2012年 236篇
2011年 219篇	2010年 142篇
2009年 108篇	2008年 94篇
2007年 83篇	2006年 26篇
2005年 23篇	2004年 1篇