

[Q]: Как получить window handle для os/2 window session?

[A]: Rinat Sadretdinow (2:5020/620)

SWBLOCK содержит в себе массив SWENTRY, каждый из которых содержит в себе SWCNTRL, который в свою очередь содержит поле hwnd. Ага?

[A]: Sergey Eremin (2:451/1)

Q> вот тут возник вопрос: каким образом получить список активных Q> задач ? собственно, это вопрос не мой, а автора OS/2 Commander'a Q> :) он сказал, что сделает task switcher (как в Capitan Nemo), Q> если ему расскажут как получить этот самый список.

Вот сейчас выдрал кусочки из dv2... Пускай делает. Copyright он может не вставлять, но thank в доке может написать :)

```
----- >% swl_exam.cpp %< ----- #define INCL_DOSPROCESS #define
INCL_DOSINFOSEG #define INCL_WINSWITCHLIST #define INCL_NOCOMMON #define INCL_SUB
#include <os2.h> #include <string.h> #include <stdio.h>
```

/ Written by Sergey Eremin. © 1994 2:451/1@fidonet se@prior.belpak.grodno.by /

```
static PSWBLOCK pswb; static LONG current;
```

```
enum { ATTR_BORDER=0x07, ATTR_NORMAL=0x17, ATTR_SELECT=0x71 };
```

```
int count( void ) {
```

```
    return (pswb) ? pswb->cswentry : -1;
```

```
}
```

```
int fill_session_list( void ) {
```

```
    ULONG cbItems,
        all_cbItems,
        cbBuf;
    PSWBLOCK tmp_pswb;
```

```
    if (pswb)
        delete pswb;
```

```
    //
    // get all switch list entries
    all_cbItems = WinQuerySwitchList( 0, 0, 0);
    cbBuf = (all_cbItems * sizeof(SWENTRY)) + sizeof(HSWITCH);
    tmp_pswb = (_SWBLOCK*) new char[cbBuf];
    WinQuerySwitchList( 0, tmp_pswb, cbBuf);
    //
    // count VISIBLE entries
    cbItems = 0;
    for ( int i=0; i <= tmp_pswb->cswentry; i++ )
```

```

    if ( tmp_pswb->aswentry[i].swctl.uchVisibility == SWL_VISIBLE )
        cbItems++;
// allocate memory for visible entries
cbBuf = (cbItems * sizeof(SWENTRY)) + sizeof(HSWITCH);
pswb = (_SWBLOCK*) new char[cbBuf];
//
// copy only visible entries to our array
int k = i = 0;
do {
    if ( tmp_pswb->aswentry[i].swctl.uchVisibility == SWL_VISIBLE )
        memcpy( (void*)&pswb->aswentry[k++],
                (void*)&tmp_pswb->aswentry[i],
                sizeof(SWENTRY) );

    i++;
}
while ( all_cbItems-- );
pswb->cswentry = cbItems;
current = 0;
delete tmp_pswb;
return 1;

```

```

}

```

```

void draw_list( void ) {

```

```

    BYTE abCell[2];
    char temp[120];
    int  usLeft,
        usTop,
        len,
        menuWidth = 0,
        menuHeight = 0;
    BYTE bAttr;

```

```

    int entries = count();
    int i = 0;
    do
    {
        if ((len=strlen(pswb->aswentry[i++].swctl.szSwttitle)) > menuWidth)
            menuWidth=len;

```

```

        menuHeight++;
    }
    while ( entries-- );

```

```

/* Leave space for the border */

```

```

menuWidth += 4 + 3;
menuHeight += 2;

```

```
usLeft= 0; usTop= 0;
```

```
/* Draw the top border */
```

```
abCell[1]=ATTR_BORDER;  
abCell[0]='█'; VioWrtNCell(abCell, 1, usTop, usLeft, 0);  
abCell[0]='█'; VioWrtNCell(abCell, 1, usTop, usLeft+menuWidth-1, 0);  
abCell[1]=ATTR_NORMAL;  
abCell[0]='█'; VioWrtNCell(abCell, menuWidth-2, usTop, usLeft+1, 0);  
usTop++;
```

```
/* Draw the switch entries */
```

```
for (i=0; i < count(); i++)  
{
```

```
/* left border */
```

```
bAttr=ATTR_BORDER;  
VioWrtCharStrAtt("█", 1, usTop, usLeft, &bAttr, 0);
```

```
/* Draw the application name */
```

```
bAttr=(BYTE)((i==current) ? ATTR_SELECT : ATTR_NORMAL);  
sprintf(temp, " %-*s %2d ", menuWidth-4-3,  
         pswb->aswentry[i].swctl.szSwttitle, i );  
VioWrtCharStrAtt(temp, menuWidth-2, usTop, usLeft+1, &bAttr, 0);
```

```
/* right border */
```

```
bAttr=ATTR_BORDER;  
VioWrtCharStrAtt("█", 1, usTop, usLeft+menuWidth-1, &bAttr, 0);
```

```
/* draw shadow */
```

```
bAttr=7;  
VioWrtNAttr(&bAttr, 1, usTop, usLeft+menuWidth, 0);
```

```
usTop++;  
}
```

```
/* bottom border */
```

```
abCell[0]='█'; VioWrtNCell(abCell, menuWidth-2, usTop, usLeft+1, 0);  
abCell[1]=ATTR_BORDER;  
abCell[0]='█'; VioWrtNCell(abCell, 1, usTop, usLeft, 0);  
abCell[0]='█'; VioWrtNCell(abCell, 1, usTop, usLeft+menuWidth-1, 0);
```

```
/* Draw bottom of shadow */
```

```
abCell[1]=7;
```

```
abCell[0]=' '; VioWrtNAttr(&bAttr, 1, usTop, usLeft+menuWidth, 0);
```

```
bAttr=7; VioWrtNAttr(&bAttr, menuWidth, usTop+1, usLeft+1, 0);
```

```
}
```

```
/* for switching to the task in list use ... WinSwitchToProgram( pswb→aswentry[current].hswitch ); ...  
*/
```

```
void main(void) {
```

```
fill_session_list();  
draw_list();  
delete pswb;
```

```
} ----- >% swl_exam.cpp %< -----
```

From:

<https://ftp.osfree.org/doku/> - **osFree wiki**

Permanent link:

<https://ftp.osfree.org/doku/doku.php?id=ru:os2faq:os2prog:os2prog.047>

Last update: **2014/06/20 05:08**

