

[Q]: wait/cwait не умеет работать с сессиями

[A]: Unknown author

This small program will start any program synchronously using DosStartSession(). The important thing is the queue. When you specify SSF_RELATED_CHILD and a TermQ name, OS/2 will write the return code to the specified queue when the session terminates. I use this in an event scheduler by creating a separate thread that does reads from the queue but you can just as easily block on the main thread to catch the return code. That will, in effect, provide for synchronous execution. Note that one problem with SSF_RELATED_CHILD is that if the program that started the child dies, so does the child.

```
#define INCL_DOSERRORS
#define INCL_DOSPROCESS
#define INCL_DOSQUEUES
#define INCL_DOSSESMGR
#include <os2.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#define QUEUE_NAME    "\\QUEUES\\STRTSYNC.QUE"
```

```
int main( int argc, char *argv[] );
```

```
int main( int argc, char *argv[] )
{
    APIRET rc;
    HQUEUE hque;
```

```
    if( argc < 2 )
        return 1;
```

```
    rc = DosCreateQueue( &hque, QUE_FIFO | QUE_CONVERT_ADDRESS, QUEUE_NAME
);
```

```
    if( !rc )
    {
        STARTDATA    stdata;
        PID           pidSession;
        CHAR          szObjFail[ 50 ];
        ULONG         ulLength, idSession;
        REQUESTDATA  rd;
        PUSHORT       pusInfo = NULL;
        BYTE          bPriority;
```

```
        (void) memset( &stdata, 0, sizeof( stdata ) );
```

```
        stdata.Length      = sizeof( STARTDATA );
        stdata.FgBg        = SSF_FGBG_FORE;
        stdata.TraceOpt    = SSF_TRACEOPT_NONE;
        stdata.PgmTitle    = "Rick's Program";
        stdata.InheritOpt  = SSF_INHERTOPT_SHELL;
```

```
stdata.SessionType = SSF_TYPE_DEFAULT;
stdata.PgmControl  = SSF_CONTROL_VISIBLE;
stdata.ObjectBuffer = szObjFail;
stdata.ObjectBuffLen= sizeof( szObjFail );
stdata.Related     = SSF_RELATED_CHILD;
stdata.TermQ       = QUEUE_NAME;
stdata.PgmName     = argv[ 1 ];
```

```
rc = DosStartSession( &stdata, &idSession, &pidSession );
```

```
if( rc && rc != ERROR_SMG_START_IN_BACKGROUND )
{
    printf( "DosStartSession RC(%u)\n", rc );
    return (INT) rc;
}
```

```
rc = DosReadQueue( hque, &rd, &ulLength, (PPVOID) &pusInfo, 0,
                  DCWW_WAIT, &bPriority, 0 );
```

```
if( rc && rc != ERROR_QUE_EMPTY )
{
    printf( "DosReadQueue RC(%u)\n", rc );
    return (INT) rc;
}
```

```
printf( "RetCode from Session %u: %u\n",
        pusInfo[ 0 ], pusInfo[ 1 ]);
```

```
DosCloseQueue( hque );
}
else
{
    printf( "DosCreateQueue RC(%u)\n", rc );
    return (INT) rc;
}
```

```
return 0;
}
```

From:

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

Permanent link:

<https://osfree.org/doku/doku.php?id=ru:os2faq:os2prog:os2prog.025>

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

