

NetOrder Client Application FAQ Rev 1.01.00

February 20, 2004

1. Table of Contents

1. Table of Contents	1
2. Revision History	1
3. Scope	1
4. FAQ	2

2. Revision History

<i>Date</i>	<i>Changes</i>	<i>Revision</i>
Feb. 12, 2003	Initial draft	1.00.00
Feb. 20, 2004	“4.9. The first QssTransmitFile API is failed” was added.	1.01.00
	“4.10. An error is given on QSS when QssTransmitFile API is called.” was added.	

3. Scope

In order to ensure the smooth development activity of the client application that uses NetOrderAPI (“Client Application”), this document lists the questions asked frequently in the form of Q&A.

4. FAQ

4.1. Developing environment

Question:

Is it possible to develop Client Application by using Visual Studio 97?

Answer:

We have not conducted testing of Visual Studio 97. We recommend using Visual Studio 6.0.

4.2. Compiling RPC

Question:

Functions defined in QssSvr.IDL are failed to link.

Answer:

Add the following functions to the source code.

- MIDL_user_allocate()
- MIDL_user_free()

For more details, refer to the sample application named “client_win32”.

4.3. Registry to be used with RPC

Question:

It seems QssSvr is not registered in the registry. Is it necessary to register it? If so, how?

I assume that HKEY_CLASSES_ROOT\INTERFACE\{6b4e94fe-5fc7-40a0-acf3-bae3063c9cfe} is required for UUID in QssSvr.IDL.

Answer:

Unlike COM, RPC does not require registration to the registry.

4.4. How to call RPC functions

Question:

I use Borland C++ Builder to develop Client Application. QssTransmitFile API does not work correctly from Client Application.

Answer:

Confirm that the function calling method is set to “_stdcall”. If not, it can be addressed by either the following 2 methods:

- 1) Change “Calling convention” to “Standard call” in Projects Options of C++ Builder.
- 2) Modify the QSS_FRAME_PIPE structure as follows and define “_stdcall” in Sample_PipeAlloc() and Sample_PipePull().

Define the QSS_FRAME_PIPE structure.

```
typedef struct pipe_QSS_FRAME_PIPE
{
    void (_stdcall * pull) (
        char __RPC_FAR * state,
```

```

    byte __RPC_FAR * buf,
    unsigned long esize,
    unsigned long __RPC_FAR * ecount );
void ( __RPC_FAR * push) (
    char __RPC_FAR * state,
    byte __RPC_FAR * buf,
    unsigned long ecount );
void ( _stdcall * alloc) (
    char __RPC_FAR * state,
    unsigned long bsize,
    byte __RPC_FAR * __RPC_FAR * buf,
    unsigned long __RPC_FAR * bcount );
char __RPC_FAR * state;
} QSS_FRAME_PIPE;

```

Define Sample_PipeAlloc() and Sample_PipePull().

```
void _stdcall Sample_PipeAlloc(char* pcStateInfo, ...);
```

```
void _stdcall Sample_PipePull(char* pcStateInfo, ...);
```

Implement Sample_PipeAlloc() and Sample_PipePull().

```

void _stdcall Sample_PipeAlloc(char* pcStateInfo, ...)
{
    (omitted)
}

void _stdcall Sample_PipePull(char* pcStateInfo, ...)
{
    (omitted)
}

```

4.5. Mac address

Question:

How can I get Mac address?

Answer:

Use Netbios API. For more detail, refer to “HOWTO: Get the MAC Address for an Ethernet Adapter” (Q118623) of Microsoft Knowledge Base.

4.6. Cannot get order status when a large number of orders are registered

Question:

When I register 40,000 orders to the QSS and attempt to get the order status of 50,000 orders by using QssGetOrderState API, an exceptional “Access is denied (5)” arises on RPC, and thus, I cannot get the order status.

Answer:

The NetOrder implemented on the QSS uses the RPC implemented on Windows 2000. There is such limitation with RPC of Windows 2000 that you can get up to 1Mbyte of messages at a time. If you try to get order status of 50,000 orders, it exceeds this limitation and an exception arises on RPC. In order to use QssGetOrderState API securely, please register orders not exceeding 10,000 when registering orders from Client Application.

TCP/IP interface also uses RPC internally, this problem may arise with command ID: 08H.

4.7. QSS_FRAME_PARAM of Version 1.0.4.1

Question:

What is "PRINT_SIZE_FREE_?" added to Version 1.0.4.1?

Answer:

When you set PRINT_SIZE_FREE_? to PrintSize of the QSS_FRAME_PARAM structure, you can specify the paper width, surface, and advance length for each frame in an order.

The reason that PRINT_SIZE_FREE_? has PRINT_SIZE_FREE_C, PRINT_SIZE_FREE_P, and PRINT_SIZE_FREE_H is that when counting total number of prints by using total counter, they are counted by print size (C, P, and H).

By using PRINT_SIZE_FREE_?, you can assign two different paper magazines and as many different paper advance lengths as you like to an order. When not using PRINT_SIZE_FREE_?, you can assign a paper magazine and up to 3 paper advance lengths per order.

Please note, however, the maximum number of paper magazine that you may assign an order is the same as that you may install on the QSS. For example, on the QSS-31 that can hold up to 2 paper magazines, when you assign 3 different paper magazines to an order, the QSS-31 will not accept the order.

4.8. Cannot get the status of or delete an order registered

Question:

Even though the order is registered correctly, I cannot get the status of or delete the order.

Answer:

It is not allowed to get the status of or delete the order when the client information specified when registering an order with QssTransmitFile API and QssSetOrder API, and that specified when getting status of the order with QssGetOrderState API or deleting the order with QssAbort API do not match.

When getting the status of or deleting an order is requested, it is confirmed that the User and Mac address, member variables of the client information, are consistent. Ensure that all the characters in the arrays of these member variables are exactly the same. If not, it is required to first initialize the client information structure to "0".

4.9. The first QssTransmitFile API is failed

Question:

When I attempt to call QssTransmitFile API from the generated thread and send an image file of bigger than 1,024 KByte, an exception "Access is denied (5)" is provided in RPC. This is not the case with the 2nd attempt.

Answer:

An exception is encountered if the thread priority of AfxBeginThread is high when generating a thread with MFC AfxBeginThread. Set the thread priority to THREAD_PRIORITY_NORMAL.

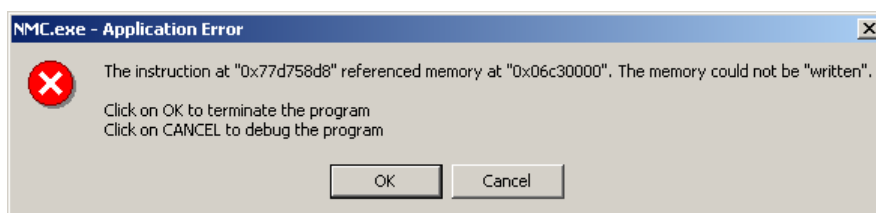
How to call AfxBeginThread:

```
CWinThread* pThread = AfxBeginThread(  
    (AFX_THREADPROC)ThreadEntry,  
    NULL,  
    THREAD_PRIORITY_NORMAL,  
    0,  
    0,  
    NULL  
);
```

4.10. An error is given on QSS when QssTransmitFile API is called.

Question:

When QssTransmitFile API is called the following error is given on QSS, which requires to restart QSS.



Answer:

Access breach takes place in RPC as there is no image file specified by QssTransmitFile API. Be sure to specify the correct path for the image file to send when calling QssTransmitFile API.