

# New Technical Notes

Macintosh

®

---

Developer Support

## NW 26 - ARA GetUserPortGlobalsPtr Call Networking

Written by: Scott Kuechle

May 1993

This Technical Note documents the Remote Access Manager (RAM) `GetUserPortGlobalsPtr` call. In order to make the `Status` call on a machine that is setup to answer calls, you must first make the `GetUserPortGlobalsPtr` call to retrieve a pointer to the globals for the user port.

---

### Introduction

The Apple Remote Access (ARA) 1.0 client software supports dial-out and answering capabilities through a single port called the “user” port (the modem or printer port on your Mac). This means that when you setup your machine to answer calls, you can answer only one call at a time on the user port. However, the underlying ARA architecture was designed so that in the future multiple ports may be supported (in a dial-in server for example).

When you dial-out on your Mac and establish an ARA connection, ARA internally allocates the data structures for the user port - but not until a connection is actually made. This is why, for example, the `Status` call will return the `-5833 ERR_PORTDOESNOTEXIST` error on the originating machine if there is no active connection. Once a connection has been made on a machine that is the originator of the connection, the `Status` call should return no error, because the data structures for the port will have been created.

When you make the `Status` call on a machine that is setup to answer calls (the answer calls box is checked in the Remote Access Setup control panel), you will find that you always get the `-5833 ERR_PORTDOESNOTEXIST` error. The reason for this is that when ARA is in answer mode it needs to be able to uniquely identify each connection with a separate `portGlobalsPtr`, because in the future there may be support for more than one connection on a single machine. Therefore, on a machine that is setup to answer calls, you must first retrieve the `portGlobalsPtr` for the user port before the `Status` call can be made. This is accomplished with the `GetUserPortGlobalsPtr` call. This call makes use of the familiar `TRemoteAccessParmHeader` structure, which is defined in the *AppleTalk Remote Access Developer's Toolkit*. The pointer to the port globals for the user port is returned in the `portGlobalsPtr` field upon completion of the call. Here are the data structures used by this call:

```
#define DControlParamHeader \  
    QElem      *qLink;          /*next queue entry*/\  
    short      qType;          /*queue type*/\  
    short      ioTrap;         /*routine trap*/\  
    Ptr        ioCmdAddr;      /*routine address*/
```

```
ProcPtr   ioCompletion;      /*completion routine*/\
OSErr     ioResult;         /*result code*/\
long      userData;         /*for use by the user */\
short     unused;          /*unused field */\
short     ioRefNum;         /*driver reference number*/\
short     csCode;          /*Call command code*/

#define DExtendedParam \
    DControlParamHeader \
    Ptr     hReserved1; \
    Ptr     hReserved2; \
    Ptr     resultStrPtr; \
    Ptr     extendedType;    /* pointer to identifier string */

#define DRemoteAccessParmHeader \
    DExtendedParam \
    short   extendedCode;    /* for use by extended call proc */\
    Ptr     portGlobalsPtr;  /* pointer to globals for this port (0=userport) */\

struct TRemoteAccessParmHeader
{
    DRemoteAccessParmHeader
};
typedef struct TRemoteAccessParmHeader TRemoteAccessParmHeader;
```

The fields in the `TRemoteAccessParmHeader` structure used for the `GetUserPortGlobalsPtr` call to the Remote Access Manager are defined as follows:

→	12	<code>ioCompletion</code>	long	pointer to completion routine
←	16	<code>ioResult</code>	word	result code
→	20	<code>userData</code>	long	for use by the user
→	26	<code>ioRefNum</code>	word	driver reference number
→	28	<code>csCode</code>	word	call command code
→	40	<code>extendedType</code>	long	pointer to identifier string
→	42	<code>extendedCode</code>	word	for use by extended call procedure
↔	44	<code>portGlobalsPtr</code>	long	pointer to globals for this port (0 = return user port globals)

Here are the detailed descriptions of the parameter block fields used by this call:

<code>ioCompletion</code>	pointer to completion routine.
<code>ioResult</code>	result code returned by the call.
<code>userData</code>	user data for use by the user.
<code>ioRefNum</code>	driver reference number.
<code>csCode</code>	command code, normally set to <code>RAM_EXTENDED_CALL</code> .
<code>extendedType</code>	should be set to <code>REMOTEACCESSNAME</code> .
<code>extendedCode</code>	set to 54 to indicate the <code>GetUserPortGlobalsPtr</code> call.
<code>portGlobalsPtr</code>	returns a pointer to the globals for the port. On input pass 0 to indicate you want the user port globals.

The following result codes can be returned by the `GetUserPortGlobalsPtr` call:

<code>noErr</code>	0	no error.
<code>ERR_PORTDOESNOTEXIST</code>	-5833	port does not exist.
<code>ERR_PORTSHUTDOWN</code>	-5832	port is shutting down.

The following is an example of how you would use the `GetUserPortGlobalsPtr` call prior to making a `Status` call.

```
#include      "RemoteAccessInterface.h"

Str255      ResultStr;
Str255      UserName;
Str255      LastMessage;
Str255      ConnectedTo;

#define CmdRemoteAccess_GetUserPortGlobalsPtr 54

void DoStatus()
{
    TRemoteAccessParamBlock      pb;

    /* Ask LTM driver for PortGlobals address */

    pb.HDR.csCode = RAM_EXTENDED_CALL;          /* extended call */
    pb.HDR.extendedType = (Ptr)REMOTEACCESSNAME; /* to Netshare */
    pb.HDR.extendedCode = CmdRemoteAccess_GetUserPortGlobalsPtr; /* get user port globals */
    pb.HDR.portGlobalsPtr = nil;                /* 0 = return user port globals */
    PRemoteAccess( &pb, false );
    if (pb.HDR.ioResult != noErr)
        HandleError(pb.HDR.ioResult);
    else
    {

        /* Issue ARA Status Call */

        ResultStr[0] = 0;
        pb.STATUS.resultStrPtr = (Ptr)ResultStr; /* put results here */
        pb.STATUS.extendedCode = CmdRemoteAccess_Status; /* status command */
        pb.STATUS.userNamePtr = UserName;
        pb.STATUS.connectedToNamePtr = ConnectedTo;
        pb.STATUS.theLastStatusMsgPtr = LastMessage;
        pb.STATUS.statusUserNamePtr = 0;
        pb.STATUS.statusMsgSeqNum = 0;
        PRemoteAccess( &pb, false );
        if (pb.STATUS.ioResult != noErr)
            HandleError(pb.STATUS.ioResult);

    }
}
```

## Conclusion

To make sure that the `RAM Status` call will work on machines that are setup to answer calls, you will need to first make the `GetUserPortGlobalsPtr` call so that the Remote Access Manager knows which port to return status information for.

## Further Reference:

---

- *AppleTalk Remote Access Developer's Toolkit*