Subject: Get MAC addresses for windows and linux
Posted by tojocky on Tue, 24 May 2011 11:14:00 GMT
View Forum Message <> Reply to Message

Hello all,

Some time ago I needed to create a function to get MAC addresses fro windows and linux.

In the end I have this functions:
in header file:

```
Array<String> GetMACAddresses();
#ifdef PLATFORM_WIN32
Array<String> GetMACAddressesIPv4();
#endif
```

and cpp file

```
#ifdef PLATFORM_WIN32
#include <winsock2.h>
#include <Windows.h>
#include <Iphlpapi.h>
#include <Assert.h>
#pragma comment(lib, "iphlpapi.lib")

// Fetches the MAC addresses, valid only for IPV8, min vers. windows 200 client/server
Array<String> GetMACAddressesIPv4(){
 Array<String> result;
 IP_ADAPTER_INFO AdapterInfo[32];   // Allocate information for up to 32 NICs
 DWORD dwBufLen = sizeof(AdapterInfo);  // Save the memory size of buffer

 DWORD dwStatus = GetAdaptersInfo(   // Call GetAdapterInfo
  AdapterInfo,       // [out] buffer to receive data
  &dwBufLen);        // [in] size of receive data buffer
 ASSERT(dwStatus == ERROR_SUCCESS);   // Verify return value is valid, no buffer overflow
 String curr_mac_str;

 PIP_ADAPTER_INFO pAdapterInfo = AdapterInfo;// Contains pointer to current adapter info
 do {
  String &curr_uuid = result.Add();
  for(int i=0;i<8;++i)
   curr_uuid << Format("%02X", pAdapterInfo->Address[i]);
  pAdapterInfo = pAdapterInfo->Next;  // Progress through linked list
 } while(pAdapterInfo);      // Terminate if last adapter
 return result;
```

```
}

// Fetches the MAC addresses, minimum version windows XP, windows 2003 server
Array<String> GetMACAddresses(){
 Array<String> result;

 Buffer<IP_ADAPTER_ADDRESSES> pAddresses;

 uint32 buffer_size = 16;
 uint32 buffer_item_bytes = sizeof(IP_ADAPTER_ADDRESSES);
 uint32 buffer_bytes;
 DWORD dwRetVal = 0;
 ULONG family = AF_UNSPEC; //can be: AF_INET, AF_INET6
 ULONG flags = GAA_FLAG_INCLUDE_PREFIX;
 int tries = 0;

 do{
  buffer_bytes = buffer_size * buffer_item_bytes;
  pAddresses.Alloc(buffer_size);

  dwRetVal = GetAdaptersAddresses(family, flags, NULL, ~pAddresses, &buffer_bytes);
  if(dwRetVal==ERROR_BUFFER_OVERFLOW)
   buffer_size=buffer_bytes/buffer_item_bytes+1;
  tries++;
 }while((dwRetVal==ERROR_BUFFER_OVERFLOW)&&(tries<3));

 if(dwRetVal == NO_ERROR){
  IP_ADAPTER_ADDRESSES *pCurrAddresses = ~pAddresses;
  while(pCurrAddresses){
   if (pCurrAddresses->PhysicalAddressLength != 0){
    String &curr_uuid = result.Add();

    for (int i = 0; i < (int) pCurrAddresses->PhysicalAddressLength; i++) {
     curr_uuid << Format("%02X", pCurrAddresses->PhysicalAddress[i]);
    }
   }
   pCurrAddresses = pCurrAddresses->Next;
  }
 }else{
  if(dwRetVal == ERROR_NO_DATA)
  ASSERT_(0, "Error to get MAC address");
 }
 return result;
}
#elif defined(PLATFORM_POSIX)

#include <net/if.h>
#include <sys/ioctl.h>
```

```
// Fetches the MAC addresses
Array<String> GetMACAddresses(){
 Array<String> result;
 int nSD; // Socket descriptor
 struct ifreq sIfReq; // Interface request
 struct if_nameindex *pIfList; // Ptr to interface name index
 struct if_nameindex *pListSave; // Ptr to interface name index

 //
 // Initialize this function
 //
 pIfList = (struct if_nameindex *)NULL;
 pListSave = (struct if_nameindex *)NULL;
 #ifndef SIOCGIFADDR
  // The kernel does not support the required ioctls
  return result;
 #endif

 //
 // Create a socket that we can use for all of our ioctls
 //
 nSD = socket( PF_INET, SOCK_STREAM, 0 );
 ASSERT(nSD >= 0);

 //
 // Obtain a list of dynamically allocated structures
 //
 pIfList = pListSave = if_nameindex();

 //
 // Walk thru the array returned and query for each interface's
 // address
 //
 unsigned char cMacAddr[8]; // Server's MAC address

 for ( pIfList; *(char *)pIfList != 0; pIfList++ ){
  //pIfList->if_name // address name like etho, eth1, ...
  strncpy( sIfReq.ifr_name, pIfList->if_name, IF_NAMESIZE );

  //
  // Get the MAC address for this interface
  //
  ASSERT(ioctl(nSD, SIOCGIFHWADDR, &sIfReq) == 0); // We failed to get the MAC address for
the interface
  byte* curr_address = (byte*)(&(sIfReq.ifr_ifru.ifru_hwaddr.sa_data[0]));

  String &curr_uuid = result.Add();
```

```
  for(int i=0;i<8;++i)
   curr_uuid << Format("%02X", curr_address[i]);
 }

 //
 // Clean up things and return
 //
 if_freenameindex( pListSave );
 close( nSD );
 return result;
}
#else
Array<String> GetMACAddresses(){
 Array<String> result;
 return result;
}
#error GetMACaddresses not supported for this plathform.
#endif
```

It must to works fine for IPV4 and IPV6 (for windows only).

If somebody know how to get MAC addresses for IPV6 I will be glad.

Any comments are welcome.
