

PDH 네트워크 이더넷 목록 얻어 사용하기

```
#define df_PDH_ETHERNET_MAX      8

//-----
// 이더넷 하나에 대한 Send,Recv PDH 쿼리 정보.
//-----
struct st_ETHERNET
{
    bool                _bUse;
    WCHAR               _szName[128];

    PDH_HCOUNTER        _pdh_Counter_Network_RecvBytes;
    PDH_HCOUNTER        _pdh_Counter_Network_SendBytes;
};

st_ETHERNET            _EthernetStruct[df_PDH_ETHERNET_MAX]; // 랜카드 별 PDH 정보
double                _pdh_value_Network_RecvBytes;          // 총 Recv Bytes 모든 이더넷의 Recv 수치 합산
double                _pdh_value_Network_SendBytes;          // 총 Send Bytes 모든 이더넷의 Send 수치 합산

////////////////////////////////////

int iCnt = 0;
bool bErr = false;
WCHAR *szCur = NULL;
WCHAR *szCounters = NULL;
WCHAR *szInterfaces = NULL;
DWORD dwCounterSize = 0, dwInterfaceSize = 0;
WCHAR szQuery[1024] = {0,};

//      PDH enum Object 를 사용하는 방법.
//      모든 이더넷 이름이 나오지만 실제 사용중인 이더넷, 가상이더넷 등등을 확인불가 함.

//-----
// PdhEnumObjectItems 을 통해서 "NetworkInterface" 항목에서 얻을 수 있는
// 측정항목(Counters) / 인터페이스 항목(Interfaces) 를 얻음. 그런데 그 개수나 길이를 모르기 때문에
// 먼저 버퍼의 길이를 알기 위해서 Out Buffer 인자들을 NULL 포인터로 넣어서 사이즈만 확인.
//-----
PdhEnumObjectItems(NULL, NULL, L"Network Interface", szCounters, &dwCounterSize, szInterfaces, &dwInterfaceSize, PERF_DETAIL_WIZARD, 0);

szCounters = new WCHAR[dwCounterSize];
szInterfaces = new WCHAR[dwInterfaceSize];

//-----
// 버퍼의 동적할당 후 다시 호출!
//
// szCounters 와 szInterfaces 버퍼에는 여러개의 문자열이 쭉쭉 들어온다. 2차원 배열도 아니고,
// 그냥 NULL 포인터로 끝나는 문자열들이 dwCounterSize, dwInterfaceSize 길이만큼 줄줄이 들어있음.
// 이를 문자열 단위로 끊어서 개수를 확인 해야 함.   aaaW0bbbW0cccW0ddd 이딴 식
//-----
```

```

if ( PdhEnumObjectItems (NULL, NULL, L"Network Interface", szCounters, &dwCounterSize, szInterfaces, &dwInterfaceSize, PERF_DETAIL_WIZARD,
0) != ERROR_SUCCESS)
{
    delete[] szCounters;
    delete[] szInterfaces;
    return false;
}

iCnt = 0;
szCur = szInterfaces;

//-----
// szInterfaces 에서 문자열 단위로 끊으면서 , 이름을 복사받는다.
//-----
for (; *szCur != L'\\0' && iCnt < df_PDH_ETHERNET_MAX; szCur += wcslen(szCur) + 1, iCnt++)
{
    _EthernetStruct[iCnt]._bUse = true;
    _EthernetStruct[iCnt]._szName[0] = L'\\0';

    wcscpy_s(_EthernetStruct[iCnt]._szName, szCur);

    szQuery[0] = L'\\0';
    StringCbPrintf(szQuery, sizeof(WCHAR) * 1024, L"\\Network Interface(%s)\\Bytes Received/sec", szCur);
    PdhAddCounter(_pdh_Query, szQuery, NULL, &_EthernetStruct[iCnt]._pdh_Counter_Network_RecvBytes);

    szQuery[0] = L'\\0';
    StringCbPrintf(szQuery, sizeof(WCHAR) * 1024, L"\\Network Interface(%s)\\Bytes Sent/sec", szCur);
    PdhAddCounter(_pdh_Query, szQuery, NULL, &_EthernetStruct[iCnt]._pdh_Counter_Network_SendBytes);
}

////////////////////////////////////
위에서 만들어진
_EthernetStruct[iCnt]._pdh_Counter_Network_SendBytes
_EthernetStruct[iCnt]._pdh_Counter_Network_RecvBytes

PDH 카운터를 다른 PDH 카운터와 같은 방법으로 사용 해주면 됨.
//-----
// 이더넷 개수만큼 돌면서 총 합을 뽑음.
//-----
for ( int iCnt = 0; iCnt < df_PDH_ETHERNET_MAX; iCnt++ )
{
    if ( _EthernetStruct[iCnt]._bUse )
    {
        Status = PdhGetFormattedCounterValue(_EthernetStruct[iCnt]._pdh_Counter_Network_RecvBytes,
PDH_FMT_DOUBLE, NULL, &CounterValue);
        if ( Status == 0 ) _pdh_value_Network_RecvBytes += CounterValue.doubleValue;
        Status = PdhGetFormattedCounterValue(_EthernetStruct[iCnt]._pdh_Counter_Network_SendBytes,
PDH_FMT_DOUBLE, NULL, &CounterValue);
        if ( Status == 0 ) _pdh_value_Network_SendBytes += CounterValue.doubleValue;
    }
}

```