

# Smart Device: rilevare la memoria di un palmare con Visual Basic .NET e C#

**Autore:** Emanuele Mattei <http://blog.shareoffice.it/emanuele/>

*Utilizzando Visual Basic .NET o C# è possibile rilevare, tramite le API (Application Program Inteface) GlobalMemoryStatus e GetSystemMemoryDivision, i dati relativi alla memoria, in particolare, la memoria totale, quella fisica, quella virtuale, il file page, e altro ancora. Vediamo insieme come fare con entrambi i linguaggi.*

## I namespace per la gestione delle API

Inseriamo i **namespace** per la gestione delle **API**, ed uno per il testo (il testo indicherà le informazioni nella messagebox). Di seguito si riportano tali spazi di nomi.

### Vb.NET:

```
Imports System
Imports System.Runtime.InteropServices
Imports System.Text
Imports System.Windows.Forms
```

### C#:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
```

## Gestione dei dati informativi sulla memoria

Dopo aver aggiunto gli opportuni spazi di nomi, si crea una struttura per la gestione dei vari dati informativi sulla memoria. Di seguito si riporta i campi di tale struttura.

### Vb.NET:

```
Public Structure MEMORYSTATUS
    Public dwLength As UInt32
    Public dwMemoryLoad As UInt32
    Public dwTotalPhys As UInt32
    Public dwAvailPhys As UInt32
    Public dwTotalPageFile As UInt32
    Public dwAvailPageFile As UInt32
    Public dwTotalVirtual As UInt32
    Public dwAvailVirtual As UInt32
End Structure
```

**C#:**

```
//Struttura per il rilevamento della memoria (caricata, totale, ecc)
public struct MEMORYSTATUS
{
    public uint dwLength;
    public uint dwMemoryLoad;
    public uint dwTotalPhys;
    public uint dwAvailPhys;
    public uint dwTotalPageFile;
    public uint dwAvailPageFile;
    public uint dwTotalVirtual;
    public uint dwAvailVirtual;
}
```

## Inserimento della dichiarazione di API

A questo punto inseriamo la dichiarazione di **API**, nella nostra **form**, come riportato qui di seguito.

**Vb.NET:**

```
'API per la gestione della memoria
Public Declare Function GlobalMemoryStatus Lib "CoreDll.Dll" _
(ByRef ms As MEMORYSTATUS) As Integer

'API per la gestione della memoria
Public Declare Function GetSystemMemoryDivision Lib "CoreDll.Dll" _
(ByRef lpdwStorePages As UInt32, _
ByRef lpdwRamPages As UInt32, _
ByRef ldpwPageSize As UInt32) As Integer
```

**C#:**

```
//API per la gestione della memoria
[System.Runtime.InteropServices.DllImport("CoreDll.Dll")]
public static extern int GlobalMemoryStatus(ref MEMORYSTATUS ms);

//API per la gestione della memoria
[System.Runtime.InteropServices.DllImport("CoreDll.Dll")]
public static extern int GetSystemMemoryDivision(ref uint lpdwStorePages, ref
uint lpdwRamPages, ref uint ldpwPageSize);
```

## Creare una funzione da utilizzare nel form

Il programma è quasi completato, non resta che creare una funzione da utilizzare nel form, il quale fa uso della **funzione API**. In particolare per l'API **globalmemoryStatus**, viene passata la struttura creata precedentemente. Di seguito si riporta tale funzione.

**Vb.Net:**

```
Public Shared Sub ShowMemory()
Dim storePages As UInt32
Dim ramPages As UInt32
Dim pageSize As UInt32
Dim Intres As Integer = GetSystemMemoryDivision(storePages, ramPages, pageSize)

' tramite l'API globalmemorystatus rilevo la memoria
Dim memStatus As New MEMORYSTATUS
GlobalMemoryStatus(memStatus)

' compongo il testo per il messaggio
Dim StrbMemoryInfo As New StringBuilder
StrbMemoryInfo.Append("Memoria Caricata: " & _
& memStatus.dwMemoryLoad.ToString() & vbCrLf)
StrbMemoryInfo.Append("Totale Fisica: " & _
& memStatus.dwTotalPhys.ToString() & vbCrLf)
StrbMemoryInfo.Append("Aviata Fisica: " & _
& memStatus.dwAvailPhys.ToString() & vbCrLf)
StrbMemoryInfo.Append("Totale Page File: " & _
& memStatus.dwTotalPageFile.ToString() & vbCrLf)
StrbMemoryInfo.Append("Avviato Page File: " & _
& memStatus.dwAvailPageFile.ToString() & vbCrLf)
StrbMemoryInfo.Append("Totale Virtuale: " & _
& memStatus.dwTotalVirtual.ToString() & vbCrLf)
StrbMemoryInfo.Append("Avviata Virtuale: " & _
& memStatus.dwAvailVirtual.ToString() & vbCrLf)

MsgBox(StrbMemoryInfo.ToString())
End Sub
```

**C#:**

```
public void ShowMemory()
{
uint storePages = 0;
uint ramPages = 0;
uint pageSize = 0;
int res = GetSystemMemoryDivision(ref storePages,
ref ramPages, ref pageSize);

// tramite l'API globalmemorystatus rilevo la memoria
MEMORYSTATUS memStatus = new MEMORYSTATUS();
GlobalMemoryStatus(ref memStatus);

// compongo il testo per il messaggio
StringBuilder StbMemoryInfo = new StringBuilder();
StbMemoryInfo.Append("Memoria Caricata: "
+ memStatus.dwMemoryLoad.ToString() + "\n");
StbMemoryInfo.Append("Totale Fisica: "
+ memStatus.dwTotalPhys.ToString() + "\n");
```

```
StbMemoryInfo.Append("Avviata Fisica: "
+ memStatus.dwAvailPhys.ToString() + "\n");
StbMemoryInfo.Append("Totale Page File: "
+ memStatus.dwTotalPageFile.ToString() + "\n");
StbMemoryInfo.Append("Avviata Page File: "
+ memStatus.dwAvailPageFile.ToString() + "\n");
StbMemoryInfo.Append("Totale Virtuale: "
+ memStatus.dwTotalVirtual.ToString() + "\n");
StbMemoryInfo.Append("Avviata Virtuale: "
+ memStatus.dwAvailVirtual.ToString() + "\n");

MessageBox.Show(StbMemoryInfo.ToString());
}
```

Dall'evento **click di un pulsante**, possiamo invocare tale funzione, in questo modo verrà visualizzato a video un messaggio contenente le informazioni alla memoria.

#### **Vb.NET:**

```
Private Sub BtnVisualizza_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnVisualizza.Click
```

```
'Per visualizzare la memoria avviare la funzione
ShowMemory()
End Sub
```

#### **C#:**

```
private void btnVisualizza_Click(object sender, EventArgs e)
{
//Per visualizzare la memoria avviare la funzione
ShowMemory();
}
```

## **Conclusioni**

In questo articolo, si è visto come creare un applicazione per dispositivo mobile, per visualizzare a video un messaggio relativo alla memoria. Il tutto tramite le API di Windows che possono aiutare di molto lo sviluppo in ambiente mobile.