

# **USERS GUIDE**

## **for**

# **AS MEMCPU WIN32**

## **Library**

**version 1.1**  
**December 2005.**



**First, we want to thank you for your interesting in our little library. How we choose functions for this library? We look at documentation of many programming tools and we just want to make it easy to do things that you can if you only knows win32 API. So, all this functions (well, not all) are using win32 API. We love it, and now we want you to use it without digging throw windows documentation. We put all this calls in one small fast library so you can use it with minimal effort.**

**We want functions that tell you what happened by return value and with logical parameters. Names must be descriptive, so you can easy remember it. Type of parameters and return value must be supported for large range of compilers. Library must be small and with functions that programmer REALLY NEED. Check functions and see if we are successful or not.**

**We believe that users want library to use not to have. Changing your applications to use this library is VERY ESAY and will not take century. We think of minutes, not hours. Tell us if we are wrong.**

**What is the best? You will have NEW APPLICATION, new release in very, very short time., with many big changes from users point , but with little source code changes from programmers point of view.**

**Now, we stop. It's your turn. Start using AS MEMCPU library and you will love it.**

**Your  
AS Computer Software team**

# List of AS MEMCPU library FUNCTIONS

**In this library you can find these functions:**

**asGetMemorySize  
asProcessorName  
asProcessorIdentifier  
asProcessorVendor  
asProcSpeed  
asFrequencyCounter  
asHighCounterStart  
asHighCounterEnd  
asGetIdleTime  
asGetKernelTime  
asGetUserTime  
asSetPriorityReal  
asSetPriorityHigh  
asSetPriorityNormal  
asSetPriorityIdle**

## Description of AS MEMCPU library FUNCTIONS

**BOOL long asGetMemorySize(int nWhat,int how);**

Supported OS: all windows versions

We get current memory configuration.

First parameter is in range from 1 to 7

1= Overall memory use (you will receive number 0-100)

2= Get total Physical memory

3= Get available Physical memory

4= Get total Virtual memory

5= Get available Virtual memory

6= Get total number of pages

7= Get available number of pages

Second parameter is 0,1 or 2. You want number in bytes, KB or MB. When you send 1 as first parameter, second parameter is not used, but you must send it to.

**void asProcessorName(LPSTR pName);**  
**void asProcessorIdentifier(LPSTR pIdentifier);**  
**void asProcessorVendor(LPSTR pVendor);**

Supported OS: all windows versions

Parameter in all these function is pointer to string buffer 255 bytes long. There you will receive your requested values.

**LONG asProcSpeed (void);**  
**DWORD asFrequencyCounter (void);**

Supported OS: all windows versions

Get some information about speed. First you will get value in MHZ and from second function you will get number of ticks per second.

**BOOL asHihgCounter Start (void);**  
**DWORD asHighCounterEnd(void);**

Supported OS: Windows 2000/XP

Processor make tiks all the time. With first function we will get and remember current value of processor counter and when you call second function we will use that processor

counter and we will return difference to you. This will be number of ticks. If you use frequency counter you can calculate time in seconds (better say parts of seconds).

```
DWORD asGetIdleTime (int how);  
DWORD asGetKernelTime(int how);  
DWORD asGetUserTime(int how);
```

Supported OS; Windows XP SP1 and later

Lest see how you system is running. You will get time in seconds, but if you send 1 as parameter, you will get value in minutes.

```
BOOL asSetPriorityReal(HWND);  
BOOL asSetPriorityHigh(HWND);  
BOOL asSetPriorityNormal(HWND);  
BOOL asSetPriorityIdle(HWND);
```

Supported OS: all windows versions

Very powerful staff, if you want speed, gets it, or slow it down. PriorityReal is max you can get, but please notice, IF YOUR APPLICATION GET Priority, other system activity will slow down. So, you must carefully choose when to use these functions. Use it for some tasks and return to normal ALWAYS.

That will be all.

As you can see, using these functions is very easy.

We will try to answer to every question you have.

AS Computer Software  
Aleksandar Stefanovic

