
Subject: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Sat, 12 Oct 2013 22:11:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

I found a better way to find distribution version. I have replace function "GetOsInfo_CheckLsb" to "GetOsInfoCheckRelease".

```
static bool GetOsInfoCheckRelease(String &distro, String &distVersion) {
    StringParse release;
    for (FindFile ff(AppendFileName("/etc/", "*-release")); ff; ff.Next())
        release += LoadFile_Safe(ff.GetPath());

    if (release.IsEmpty())
        return false;
    if (!release.GoAfter("DISTRIB_ID="))
        return false;
    distro = release.GetText();
    if (distro.IsEmpty())
        return false;

    release.GoAfter_Init("VERSION=");
    distVersion = release.GetText();

    return true;
}
```

I enclose full source file with additional changes (SysInfo/os.cpp).

P.S.

Are you sure about this assignation "distro = ubuntu" or "distro=fedora" (Shouldn't it be "Ubuntu" or "Fedora")? BTW, my version uses big letter notation.

Sincerely,
Klugier

File Attachments

1) [os.cpp](#), downloaded 656 times

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Sat, 12 Oct 2013 22:38:36 GMT

Hello Klugier

Thank you for your work.

You have changed:

- /etc/lsb-release with /etc/*-release
- DISTRIB_RELEASE= with VERSION=

Please send me the distros that you have tested.

About the lowercase, I think it is more standard to use "ubuntu" than "Ubuntu" and I hate the uppercases like "UBUNTU".

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Sat, 12 Oct 2013 23:22:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Following two lines of code are tricky:

```
for (FindFile ff(AppendFileName("/etc/", "*-release")); ff; ff.Next())
  release += LoadFile_Safe(ff.GetPath());
```

This means, that find all files, that have "release" chain in their names. File "lsb-release" doesn't contain full version information, so we need to scan other files with "release" in their names.

```
*****
***** **
```

Here is the output (Ubuntu 13.04):
DISTRIB_VERSION=13.04
VERSION=13.04, Raring Ringtail

VERSION can be replaced with VERSION_ID (And even DISTRIB_VERSION to ID):
VERSION_ID=13.04
ID=ubuntu

```
*****
***** **
```

You can type following command in your terminal: "cat /etc/*-release":

DISTRIB_ID=Ubuntu

```
DISTRIB_RELEASE=13.04
DISTRIB_CODENAME=raring
DISTRIB_DESCRIPTION="Ubuntu 13.04"
NAME="Ubuntu"
VERSION="13.04, Raring Ringtail"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 13.04"
VERSION_ID="13.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
```

Personally, I think we can add more information about distribution to SysInfo.

```
*****
```

```
***** **
```

P.S.

I uploaded old file (It doesn't compile!!!).

*I enclose newer version with lowercase fix.

Sincerely,
Klugier

File Attachments

1) [os.cpp](#), downloaded 712 times

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Sat, 12 Oct 2013 23:58:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Here is the final version of the function:

```
static bool GetOsInfoCheckRelease(String &distro, String &distVersion)
{
    StringParse release;
    for (FindFile ff(AppendFileName("/etc/", "*-release")); ff; ff.Next())
        release += LoadFile_Safe(ff.GetPath());

    if (release.IsEmpty())
        return false;
    if (!release.GoAfter("DISTRIB_ID="))
        return false;
    distro = ToLower(release.GetText());
```

```
if (distro.IsEmpty())
    return false;

release.GoAfter_Init("DISTRIB_RELEASE=");
distVersion = release.GetText();

return true;
}
```

I don't want to change current functionality, so i decided to stay with old qualifiers. In the future we can get more informations about distribution (with my extension). It is up to you.

P.S.

Moreover, you can change LoadFile_Safe function (Functions4U) to:

```
String LoadFile_Safe(const String& fileName);
```

insted of:

```
String LoadFile_Safe(String fileName);
```

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Sun, 13 Oct 2013 14:40:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Klugier

I have included

```
for (FindFile ff("/etc/*-release"); ff; ff.Next())
    release += LoadFile_Safe(ff.GetPath()); because it does not hurt.
```

However you have to imagine that this function has to work with many different distros, not only Ubuntu. You can see SysInfo testing here:

[http://www.ultimatepp.org/srcimp\\$SysInfo\\$SysInfo\\$en-us.html](http://www.ultimatepp.org/srcimp$SysInfo$SysInfo$en-us.html)

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Fri, 18 Oct 2013 20:32:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Did you forget to add reference in LoadFile_Safe function? In my opinion declaration of this function should look like:

```
String LoadFile_Safe(const String& fileName);
```

* Negligible change which can increase memory performance of yours packages (Including SysInfo)

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Fri, 18 Oct 2013 22:37:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Moreover, I wrote better handling for measuring cpu temperature on GNU/Linux (POSIX?). Now, SysInfo should use "hwmon" when "acpi" fails. Let's look at code:

```
static double GetCpuTemperatureViaAcpi()
{
    FindFile ff;
    if(ff.Search("/proc/acpi/thermal_zone/*")) {
        do {
            if (ff.IsDirectory()) {
                String name = ff.GetName();
                if (name != "." && name != "..") {
                    StringParse str =
LoadFile_Safe(AppendFileName(AppendFileName("/proc/acpi/thermal_zone", name),
"temperature"));
                    str.GoAfter("temperature:");
                    return str.GetDouble();
                }
            }
        } while(ff.Next());
    }
}
```

```

return Null;
}

static double GetCpuTemperatureViaHwmon()
{
    Vector <double> temps;
    for (FindFile ff(AppendFileName("/sys/class/hwmon/hwmon0/device", "**input")); ff; ff.Next()) {
        if (!ff.IsHidden()) {
            String temp = LoadFile_Safe(ff.GetPath());
            if (!temp.IsEmpty())
                temps.Add((double)StrToInt(temp) / 1000.0);
        }
    }

    double sumTemps = 0.0;
    for (int i = 0; i < temps.GetCount(); i++)
        sumTemps += temps[i];

    if (!temps.IsEmpty())
        return sumTemps / (double)temps.GetCount();
    else
        return Null;
}

double GetCpuTemperature()
{
    double temp = Null;

    if (temp == (double)Null) temp = GetCpuTemperatureViaAcpi();
    if (temp == (double)Null) temp = GetCpuTemperatureViaHwmon();

    return temp;
}

```

* in this case "static" means that function should be use only in file where it was declared.

P.S.

Toomorrow, I will enclose source file with above improvement.

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Sat, 19 Oct 2013 20:29:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

BTW, I noticed that LoadFile_Safe function is really useful for system files (like hwmon files). What is not good in Core package is that LoadFile doesn't provide such functionality. I tried loading hwmon files using LoadFile function, but in this case it doesn't work. I think we should discuss transferring LoadFile_Safe (LoadFileSafe? or LoadSystemFile?) function to Ultimate++ mainstream package (Core). What do you think guys?

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Sun, 20 Oct 2013 08:00:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Klugier

LoadSystemFile() could be a good name.

These files (many in Linux) cannot be opened by LoadFile because it is not possible to know their size in advance, they are like pipes.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Sun, 20 Oct 2013 08:04:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Klugier

It seems good.

For me to check it, could you tell me some distros where acpi is not used and Hwmon is used?

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Zbych](#) on Sun, 20 Oct 2013 10:36:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

klugier wrote on Sat, 19 October 2013 00:37
static double GetCpuTemperatureViaHwmon()

```
{  
Vector <double> temps;  
for (FindFile ff(AppendFileName("/sys/class/hwmon/hwmon0/device", "**input")); ff; ff.Next()) {
```

Unfortunately it will not work on all systems. I checked some computers in my office:

1. intel, ubuntu 12.04, kernel 3.2, temperature location: /sys/class/hwmon/hwmon0/temp1_input (no device sub-dir)
2. intel, ubuntu 12.04, kernel 3.8, temperature location: /sys/devices/platform/coretemp.0/temp*_input
3. AMD, ubuntu 12.04, kernel 3.2, temperature location: /sys/class/hwmon/hwmon0/device/temp1_input

Maybe we should check libsensors to find out how it locates cpu temperature files.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Tue, 22 Oct 2013 14:24:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Zbych

Following your advice I am checking it.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Wed, 23 Oct 2013 20:13:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello

There is a libsensors test in the beginning of SysInfo_demo_console.

Please run it and report the results (Do not forget to install libsensors before).

In my case they are:

LibSensors version: 3.3.2

Results:

Chip: acpitz-virtual-0

temperature: temp1

input(RM) temp1_input = 47

critical(RM) temp1_crit = 104

Chip: coretemp-isa-0000

```
temperature: Core 0
input(RM) temp2_input = 52
max(RM) temp2_max = 85
critical(RM) temp2_crit = 85
crit alarm(R) temp2_crit_alarm = 0
temperature: Core 1
input(RM) temp3_input = 48
max(RM) temp3_max = 85
critical(RM) temp3_crit = 85
crit alarm(R) temp3_crit_alarm = 0
```

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Wed, 23 Oct 2013 20:46:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Koldo, please noticed that libsensors is GPL library. It means that all applications which are linking to SysInfo should also use GPL license. Only the latest version of libsensors use LGPL license. I do not think that people linking their programs with libsensors want to risk so much.

To use my trick with temperature you don't need to link with libsensors, beacuse I read only system file. In this way, you can also get more information about the computer sensors state. The main problem of this solution is how to find appropriate path on certain linux version.

More information about libsensors licence you can find on official site.

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Tue, 29 Oct 2013 17:52:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

Oh! Although it is LGPL I think we can get the information by our means .

libsensors is now removed from SysInfo.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Tue, 29 Oct 2013 21:19:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Can you add also following if statement at the end of GetCpuTemperatureHWMON function?
Dividing by zero can be painful. This situation can occur when hwmon subsystem does not exist!

```
if (!temps.IsEmpty())  
    return sumTemps / (double)temps.GetCount();  
else  
    return Null;
```

P.S.

Now, I am looking for optimal path for hwmon.

Sincerley,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Wed, 30 Oct 2013 08:17:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thank you Klugier

Quote:Now, I am looking for optimal path for hwmon.
I was going to do this so thank you for relieving me of this duty.

About the division by zero, I think actual code avoids that possibility.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Wed, 30 Oct 2013 14:12:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Quote>About the division by zero, I think actual code avoids that possibility.

No it doesn't, cause (double)temps.GetCount() can return zero if input files don't exist!!!

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Wed, 30 Oct 2013 20:37:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Klugier

I think there is a misunderstanding. See actual code:

```
static double GetCpuTemperatureHWMON() {
    Vector <double> temps;
    for (FindFile ff("/sys/class/hwmon/hwmon0/device/*input"); ff; ff.Next()) {
        if (!ff.IsHidden()) {
            String temp = LoadFile_Safe(ff.GetPath());
            if (!temp.IsEmpty())
                temps.Add((double)StrInt(temp) / 1000.0);
        }
    }
    if (temps.IsEmpty())
        return Null;
    double sumTemps = 0.;
    for (int i = 0; i < temps.GetCount(); i++)
        sumTemps += temps[i];

    return sumTemps/double(temps.GetCount());
}
```

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Wed, 30 Oct 2013 21:16:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Now, I see:

```
if (temps.IsEmpty())
    return Null;
```

I am sorry for the mess which i have made.

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Wed, 30 Oct 2013 22:15:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

Anyway, now it is like this:

```
static double GetCpuTemperatureHWMON() {
    double sumTemps = 0.;
    int count = 0;
    for (FindFile ff("/sys/class/hwmon/hwmon0/device/*input"); ff; ff.Next()) {
        if (!ff.IsHidden()) {
            String temp = LoadFile_Safe(ff.GetPath());
            if (!temp.IsEmpty()) {
                sumTemps += double(StrInt(temp))/1000.;
                count++;
            }
        }
    }
    if (count == 0)
        return Null;
    return sumTemps/count;
}
```

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Thu, 31 Oct 2013 16:01:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Alternative paths, can be done in this way:

```
static double GetCpuTemperatureHWMON()
{
    FindFile ff;
    double sumTemps = 0.0;
    int count = 0;

    String path = "/sys/class/hwmon/";

    if (ff.Search(AppendFileName(path, "device"))) {
        path += ff.GetName() + "/";
    }
    else {
        String tempPath;
```

```

for (ff.Search(AppendFileName(path, "hwmon*")); ff; ff.Next()) {
    FindFile hwmonFF;

    tempPath = path;
    tempPath += ff.GetName() + "/";
    if (hwmonFF.Search(AppendFileName(tempPath, "device"))) {
        tempPath += hwmonFF.GetName() + "/";
    }

    if (hwmonFF.Search(AppendFileName(tempPath, "*input"))) {
        if (!hwmonFF.IsDirectory()) {
            path = tempPath;
            break;
        }
    }
}
if (!tempPath.IsEmpty() && tempPath != path)
    path = tempPath;
}

for (ff.Search(AppendFileName(path, "*input")); ff; ff.Next()) {
    if (!ff.IsHidden()) {
        String temp = LoadFile_Safe(ff.GetPath());
        if (!temp.IsEmpty()) {
            sumTemps += static_cast <double> ((StrInt(temp))) / 1000.0;
            count++;
        }
    }
}
if (count == 0)
    return Null;
return sumTemps / count;
}

```

* Additional changes:

- change casting method to more safer.
- changed "0." to "0.0" (Are you sure that 0. is more readable than 0.0???)

* TODO:

- create function static String "findHWMONInputFilesPath()" - ALTERNATIVE METHOD

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version

& more

Posted by [koldo](#) on Fri, 01 Nov 2013 21:51:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello all

To get CPU temperature is more complicated than expected, and more difficult than checking folders.

So it is necessary to install either acpi or lm-sensors packages. There is no problem with the licenses because now the libraries are not used.

GetCpuTemperature() has been adapted to this and documentation has been updated.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Fri, 01 Nov 2013 22:37:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

Using sensors command is the easiest solution and safe solution, but not efficient one. Calling external program always associated with the overhead time. In addition we can check where are input files only once per program to optimize the algorithm.

Quote:

To get CPU temperature is more complicated than expected, and more difficult than checking folders.

I am almost sure that we can do this in above way (Open sources projects like "Conky" do this in similar way).

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Sat, 02 Nov 2013 21:26:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Klugier

IMHO calling external programs is a solution very used in open source projects, SysInfo included. For example it permits GPL code to be called from any program.

In the other side, I do not consider that registering CPU temperature requires a super fast response.

acpi and lm-sensors are well respect solutions, and there are so many different devices that it seems difficult to support them.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Sun, 03 Nov 2013 00:09:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thank you for your answer Koldo. I fully understand your position and I agree with most of theses. If I have any ideas about SysInfo package I will posted about it on the forum.

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [kasome](#) on Tue, 20 May 2014 10:59:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

There is a bug in upp\bazaar\SysInfo\SysInfo.cpp

Here is the original code

```
Array<NetAdapter> GetAdapterInfo()
{
    Array<NetAdapter> res;
    Index<String> macs;

    int sck = socket(PF_INET, SOCK_DGRAM, 0);
    if(sck < 0)
        return res;

    int bufSize = 4096;
    struct ifconf ifc = { 0 };
    byte buf[bufSize];
    for(int iTry = 0; iTry < 4; iTry++) {
        ifc.ifc_len = bufSize;
        ifc.ifc_buf = (__caddr_t)buf;
```

```

if (ioctl(sck, SIOCGIFCONF, &ifc) < 0) {
    close(sck);
    return res;
}

if(ifc.ifc_len != bufSize)
    break;

bufSize *= 2;
if(iTry >= 3) {
    close(sck);
    return res;
}
}

int nIfaces = ifc.ifc_len / sizeof(struct ifreq);
struct ifreq *iface = ifc.ifc_req;
for(int ilface = 0; ilface < nIfaces; ilface++) {
    String MAC;

#ifdef SIOCGIFHWADDR
    if(ioctl(sck, SIOCGIFHWADDR, iface) < 0)
        continue;

    MAC = Format("%02x:%02x:%02x:%02x:%02x:%02x",
        (byte)iface->ifr_hwaddr.sa_data[0],
        (byte)iface->ifr_hwaddr.sa_data[1],
        (byte)iface->ifr_hwaddr.sa_data[2],
        (byte)iface->ifr_hwaddr.sa_data[3],
        (byte)iface->ifr_hwaddr.sa_data[4],
        (byte)iface->ifr_hwaddr.sa_data[5]
    );
#endif

#ifdef SIOCGENADDR
    if (ioctl(sck, SIOCGENADDR, iface) < 0)
        continue;

    MAC = Format("%02x:%02x:%02x:%02x:%02x:%02x",
        (byte)iface->ifr_enaddr[0],
        (byte)iface->ifr_enaddr[1],
        (byte)iface->ifr_enaddr[2],
        (byte)iface->ifr_enaddr[3],
        (byte)iface->ifr_enaddr[4],
        (byte)iface->ifr_enaddr[5]
    );
#endif

#ifdef __MACH__ || __NetBSD__ || __OpenBSD__ || __FreeBSD__
    int mib[6];

```

```

mib[0] = CTL_NET;
mib[1] = AF_ROUTE;
mib[2] = 0;
mib[3] = AF_LINK;
mib[4] = NET_RT_IFLIST;
mib[5] = if_nametoindex(iface->ifr_name);
if (mib[5] == 0) // nameless interface ? skip
    continue;

int len = 0;
if (sysctl(mib, 6, NULL, (size_t*)&len, NULL, 0) != 0)
    continue; // sysctl error, just leave MAC empty

char macbuf[len];
if (!macbuf) // can't allocat buffer, skip this MAC
    continue;

if (sysctl(mib, 6, macbuf, (size_t*)&len, NULL, 0) != 0)
    continue;

struct if_msghdr *ifm = (struct if_msghdr *)macbuf;
struct sockaddr_dl *sdl = (struct sockaddr_dl *)(ifm + 1);
byte ptr[] = (byte **)LLADDR(sdl);

MAC = Format("%02x:%02x:%02x:%02x:%02x:%02x",
    ptr[0], ptr[1], ptr[2], ptr[3], ptr[4], ptr[5]);

#else
#error OS Distribution Not Recognized
#endif

if(MAC == "00:00:00:00:00:00")
    MAC.Clear();

if(!MAC.IsEmpty() && macs.Find(MAC) >= 0)
    continue;
macs.Add(MAC);

NetAdapter &adapter = res.Add();

adapter.mac = MAC;

adapter.fullname = iface->ifr_name;

// set interface type from name
if(adapter.fullname.StartsWith("eth"))
    adapter.type = "ETHERNET";
else if(adapter.fullname.StartsWith("lo"))

```

```

    adapter.type = "SOFTWARE_LOOPBACK";
else if(adapter.fullname.StartsWith("ppp"))
    adapter.type = "MODEM";
else if(adapter.fullname.StartsWith("hci"))
    adapter.type = "BLUETOOTH";
else if(adapter.fullname.StartsWith("tr"))
    adapter.type = "TOKENRING";
else if(adapter.fullname.StartsWith("vbox") || adapter.fullname.StartsWith("wifi") ||
    adapter.fullname.StartsWith("ath"))
    adapter.type = "VIRTUALBOX";
else if(adapter.fullname.StartsWith("wlan"))
    adapter.type = "IEEE80211";
else if(adapter.fullname.StartsWith("vmnet"))
    adapter.type = "VMWARE";
else
    adapter.type = "OTHER";

adapter.description = adapter.fullname;

if (!ioctl(sck, SIOCGIFADDR, iface))
    continue;

    adapter.ip4 = inet_ntoa(((struct sockaddr_in *)&(iface->ifr_addr))->sin_addr);

    iface++;
}
close(sck);

return res;
}

```

Here is the fixed code

```

Array<NetAdapter> GetAdapterInfo()
{
    Array<NetAdapter> res;
    Index<String> macs;

    int sck = socket(PF_INET, SOCK_DGRAM, 0);
    if(sck < 0)
        return res;

    int bufSize = 4096;
    struct ifconf ifc = { 0 };
    byte buf[bufSize];
    for(int iTry = 0; iTry < 4; iTry++) {
        ifc.ifc_len = bufSize;

```

```

ifc.ifc_buf = (__caddr_t)buf;

if (ioctl(sck, SIOCGIFCONF, &ifc) < 0) {
    close(sck);
    return res;
}

if(ifc.ifc_len != bufSize)
    break;

bufSize *= 2;
if(iTry >= 3) {
    close(sck);
    return res;
}
}

int nlfaces = ifc.ifc_len / sizeof(struct ifreq);
struct ifreq *iface = ifc.ifc_req;
for(int ilface = 0; ilface < nlfaces; ilface++) {
    String MAC;

#ifdef SIOCGIFHWADDR
    if(ioctl(sck, SIOCGIFHWADDR, iface) < 0)
        continue;

    MAC = Format("%02x:%02x:%02x:%02x:%02x:%02x",
        (byte)iface->ifr_hwaddr.sa_data[0],
        (byte)iface->ifr_hwaddr.sa_data[1],
        (byte)iface->ifr_hwaddr.sa_data[2],
        (byte)iface->ifr_hwaddr.sa_data[3],
        (byte)iface->ifr_hwaddr.sa_data[4],
        (byte)iface->ifr_hwaddr.sa_data[5]
    );
#endif

#ifdef SIOCGENADDR
    if (ioctl(sck, SIOCGENADDR, iface) < 0)
        continue;

    MAC = Format("%02x:%02x:%02x:%02x:%02x:%02x",
        (byte)iface->ifr_enaddr[0],
        (byte)iface->ifr_enaddr[1],
        (byte)iface->ifr_enaddr[2],
        (byte)iface->ifr_enaddr[3],
        (byte)iface->ifr_enaddr[4],
        (byte)iface->ifr_enaddr[5]
    );
#endif
}

```

```

#elif __MACH__ || __NetBSD__ || __OpenBSD__ || __FreeBSD__
    int mib[6];
    mib[0] = CTL_NET;
    mib[1] = AF_ROUTE;
    mib[2] = 0;
    mib[3] = AF_LINK;
    mib[4] = NET_RT_IFLIST;
    mib[5] = if_nametoindex(iface->ifr_name);
    if (mib[5] == 0) // nameless interface ? skip
        continue;

    int len = 0;
    if (sysctl(mib, 6, NULL, (size_t*)&len, NULL, 0) != 0)
        continue; // sysctl error, just leave MAC empty

    char macbuf[len];
    if (!macbuf) // can't allocat buffer, skip this MAC
        continue;

    if (sysctl(mib, 6, macbuf, (size_t*)&len, NULL, 0) != 0)
        continue;

    struct if_msghdr *ifm = (struct if_msghdr *)macbuf;
    struct sockaddr_dl *sdl = (struct sockaddr_dl *)(ifm + 1);
    byte ptr[] = (byte **)LLADDR(sdl);

    MAC = Format("%02x:%02x:%02x:%02x:%02x:%02x",
        ptr[0], ptr[1], ptr[2], ptr[3], ptr[4], ptr[5]);

#else
#error OS Distribution Not Recognized
#endif

    if(MAC == "00:00:00:00:00:00")
        MAC.Clear();

    if(!MAC.IsEmpty() && macs.Find(MAC) >= 0)
        continue;
    macs.Add(MAC);

    NetAdapter &adapter = res.Add();

    adapter.mac = MAC;

    adapter.fullname = iface->ifr_name;

    // set interface type from name
    if(adapter.fullname.StartsWith("eth"))

```

```

    adapter.type = "ETHERNET";
else if(adapter.fullname.StartsWith("lo"))
    adapter.type = "SOFTWARE_LOOPBACK";
else if(adapter.fullname.StartsWith("ppp"))
    adapter.type = "MODEM";
else if(adapter.fullname.StartsWith("hci"))
    adapter.type = "BLUETOOTH";
else if(adapter.fullname.StartsWith("tr"))
    adapter.type = "TOKENRING";
else if(adapter.fullname.StartsWith("vbox") || adapter.fullname.StartsWith("wifi") ||
    adapter.fullname.StartsWith("ath"))
    adapter.type = "VIRTUALBOX";
else if(adapter.fullname.StartsWith("wlan"))
    adapter.type = "IEEE80211";
else if(adapter.fullname.StartsWith("vmnet"))
    adapter.type = "VMWARE";
else
    adapter.type = "OTHER";

adapter.description = adapter.fullname;

if (ioctl(sck, SIOCGIFADDR, iface) < 0) {
    iface++;
    continue;
}

    adapter.ip4 = inet_ntoa(((struct sockaddr_in *)&(iface->ifr_addr))->sin_addr);

    iface++;
}
close(sck);

return res;
}

```

Hope that helps.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Tue, 20 May 2014 15:27:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Yes you are right. Thank you.

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [kasome](#) on Mon, 14 Jul 2014 08:26:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

There is a few memory leak in function GetWMInfo of upp\bazaar\SysInfo\SysInfo.cpp

find

```
bool GetWMInfo(String system, Array <String> &data, Array <Value> *ret[], String nameSpace = "root\\cimv2")
```

and modify

```
for (int col = 0; col < data.GetCount(); ++col) {
    VARIANT vProp;
    BSTR strClassProp = SysAllocString(data[col].ToWString());
    hRes = pClassObject->Get(strClassProp, 0, &vProp, 0, 0);
    if(hRes != S_OK){
        pWbemServices->Release();
        pIWbemLocator->Release();
        pEnumerator->Release();
        pClassObject->Release();
        CoUninitialize();
        return false;
    }
    SysFreeString(strClassProp);
    ret[col]->Add(GetVARIANT(vProp));
    rt = true;
}
```

to

```
for (int col = 0; col < data.GetCount(); ++col) {
    VARIANT vProp;
    VariantInit( &vProp );
    BSTR strClassProp = SysAllocString(data[col].ToWString());
    hRes = pClassObject->Get(strClassProp, 0, &vProp, 0, 0);
    if(hRes != S_OK){
        pWbemServices->Release();
        pIWbemLocator->Release();
        pEnumerator->Release();
        pClassObject->Release();
        CoUninitialize();
        return false;
    }
}
```

```
SysFreeString(strClassProp);
ret[col]->Add(GetVARIANT(vProp));
VariantClear( &vProp );
rt = true;
}
```

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [kasome](#) on Mon, 14 Jul 2014 08:33:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

About the functiion Shutdown in linux platform,

is there missing sync() function call when action == "shutdown" or action == "reboot" >

```
#ifdef PLATFORM_POSIX
bool Shutdown(String action)
{
    if (action == "logoff") {
        kill(1, SIGTSTP);
        sync();
        signal(SIGTERM, SIG_IGN);
        setpgrp();
        kill(-1, SIGTERM);
        sleep(1);
        kill(-1, SIGHUP); /* Force PPPD's down, too */
        sleep(1);
        kill(-1, SIGKILL);
        sync();
        sleep(1);
    } else if (action == "shutdown") {
        sync();
        sleep(1);
#ifdef __GNU_LIBRARY__ > 5
        reboot(0xCDEF0123);
#else
        reboot(0xfeef1dead, 672274793, 0xCDEF0123);
#endif
    } else if (action == "reboot") { // LINUX_REBOOT_MAGIC1, LINUX_REBOOT_MAGIC2
        sync();
        sleep(1);
#ifdef __GNU_LIBRARY__ > 5
        reboot(0x01234567);
#else
        reboot(0xfeef1dead, 672274793, 0x01234567);
#endif
    }
}
```

```
exit(0);  
return true;  
}  
#endif
```

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Tue, 15 Jul 2014 10:05:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

All included. Thank you!

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [Klugier](#) on Sat, 19 Jul 2014 18:48:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo,

You should also replace C api function "exit(int)" by new U++ dedicated function "Exit(int code = 1)". This is maybe not important, but it makes your package more upful

Sincerely,
Klugier

Subject: Re: [SysInfo - Improvement - Koldo] Better way to find distribution version & more

Posted by [koldo](#) on Mon, 21 Jul 2014 08:13:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thank you. An additional advantage is to avoid memory leaks because of the exit().
