
Subject: DirectoryCreateMulti as an alternative for RealizeDirectory

Posted by [Sender Ghost](#) on Sun, 07 Dec 2008 19:15:49 GMT

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

In the "Core\Path.h" file we have useful function RealizeDirectory that creates not existent directories from the given path. I create alternative function with name DirectoryCreateMulti. Complexity: recursion versus memory management with Vector.

```
#ifndef PLATFORM_POSIX
bool DirectoryCreateMulti(const String& path, int mode = 0755)
#else
bool DirectoryCreateMulti(const String& path)
#endif
{
    Vector<String> dirs;

    for (int i=path.GetCount() - 1; ; --i)
    {
        i = path.ReverseFind(DIR_SEP, i);
#ifdef PLATFORM_POSIX
        if (i>0)
#else
        if (i>0 && path[i - 1] != ':')
#endif
        {
            const String tmp(path.Left(i));
            if (!DirectoryExists(tmp))
            {
                dirs.Add(tmp);
            }
            else break;
        }
        else break;
    }

    for (int i=dirs.GetCount() - 1; i>=0; --i)
    {
#ifdef PLATFORM_POSIX
        if (!DirectoryCreate(dirs[i], mode))
#else
        if (!DirectoryCreate(dirs[i]))
#endif
        {
            return false;
        }
    }
}

#ifdef PLATFORM_POSIX
```

```
return DirectoryCreate(path, mode);
#else
return DirectoryCreate(path);
#endif
}
```

Subject: Re: DirectoryCreateMulti as an alternative for RealizeDirectory
Posted by [mirek](#) on Mon, 15 Dec 2008 08:32:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks, I am afraid you are right that linear recursion is usually not a best thing to do....

Mirek

Subject: Re: DirectoryCreateMulti as an alternative for RealizeDirectory
Posted by [mirek](#) on Mon, 15 Dec 2008 08:42:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, I have ended with this code

```
#ifdef POSIX
void RealizeDirectory(String dir, int mode)
#else
void RealizeDirectory(String dir)
#endif
{
    Vector<String> p;
    while(dir.GetLength() > DIR_MIN) {
        p.Add(dir);
        dir = GetFileFolder(dir);
    }
    for(int i = p.GetCount() - 1; i >= 0; i--)
        if(!DirectoryExists(p[i]))
#ifdef POSIX
            DirectoryCreate(p[i], mode);
#else
            DirectoryCreate(p[i]);
#endif
    }
}
```

This is definitely a step in the right direction

Mirek

Subject: Re: DirectoryCreateMulti as an alternative for RealizeDirectory

Posted by [Sender Ghost](#) on Mon, 15 Dec 2008 14:59:46 GMT

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

luzr wrote on Mon, 15 December 2008 13:42OK, I have ended with this code

```
#ifdef POSIX
void RealizeDirectory(String dir, int mode)
#else
void RealizeDirectory(String dir)
#endif
{
    Vector<String> p;
    while(dir.GetLength() > DIR_MIN) {
        p.Add(dir);
        dir = GetFileFolder(dir);
    }
    for(int i = p.GetCount() - 1; i >= 0; i--)
        if(!DirectoryExists(p[i]))
#ifdef POSIX
            DirectoryCreate(p[i], mode);
#else
            DirectoryCreate(p[i]);
#endif
    }
}
```

This is definitely a step in the right direction

Mirek

Without NormalizePath function your code still not working with "a\b\c" directory. As stated in the comments:

```
#ifdef PLATFORM_WIN32
#define DIR_MIN 3 ///! wrong! what about \a\b\c ?
#endif
```

This is my intention.

It's not wrong with recursion, if they not too long for the stack overflow . I think it's not a problem (MAX_PATH == 260 in the stdlib.h).

So, you prefer:

```
dir.GetLength() > DIR_MIN
```

I use another more adequate method (in my opinion of course):

```
i>0 && path[i - 1] != ':'
```

To note:

- DirectoryCreate returns variable about success of directory creation. DirectoryCreateMulti do this also.
- DirectoryCreateMulti doesn't keep all directories for verification of existent directory. They use backward search with ReverseFind function.

Subject: Re: DirectoryCreateMulti as an alternative for RealizeDirectory

Posted by [mirek](#) on Mon, 15 Dec 2008 16:08:19 GMT

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

Sender Ghost wrote on Mon, 15 December 2008 09:59luzr wrote on Mon, 15 December 2008 13:42OK, I have ended with this code

```
#ifdef POSIX
void RealizeDirectory(String dir, int mode)
#else
void RealizeDirectory(String dir)
#endif
{
    Vector<String> p;
    while(dir.GetLength() > DIR_MIN) {
        p.Add(dir);
        dir = GetFileFolder(dir);
    }
    for(int i = p.GetCount() - 1; i >= 0; i--)
        if(!DirectoryExists(p[i]))
#ifdef POSIX
            DirectoryCreate(p[i], mode);
#else
            DirectoryCreate(p[i]);
#endif
    }
}
```

This is definitely a step in the right direction

Mirek

Without NormalizePath function your code still not working with "a\b\c" directory. As stated in the comments:

```
#ifdef PLATFORM_WIN32
#define DIR_MIN 3 ///! wrong! what about \a\b\c ?
```

#endif

This is my intention.

It's not wrong with recursion, if they not too long for the stack overflow . I think it's not a problem (MAX_PATH == 260 in the stdlib.h).

So, you prefer:

dir.GetLength() > DIR_MIN

I use another more adequate method (in my opinion of course):

i>0 && path[i - 1] != ':'

To note:

- DirectoryCreate returns variable about success of directory creation. DirectoryCreateMulti do this also.
- DirectoryCreateMulti doesn't keep all directories for verification of existent directory. They use backward search with ReverseFind function.

Good points, thanks.

```
#ifdef POSIX
bool RealizeDirectory(const String& d, int mode)
#else
bool RealizeDirectory(const String& d)
#endif
{
    String dir = NormalizePath(d);
    Vector<String> p;
    while(dir.GetLength() > DIR_MIN) {
        p.Add(dir);
        dir = GetFileFolder(dir);
    }
    for(int i = p.GetCount() - 1; i >= 0; i--)
        if(!DirectoryExists(p[i]))
#ifdef POSIX
            if(!DirectoryCreate(p[i], mode))
#else
            if(!DirectoryCreate(p[i]))
#endif
        return false;
    return true;
}
```

Mirek

Subject: Re: DirectoryCreateMulti as an alternative for RealizeDirectory
Posted by [Sender Ghost](#) on Mon, 15 Dec 2008 16:47:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Mon, 15 December 2008 21:08
Good points, thanks.

```
#ifdef POSIX
bool RealizeDirectory(const String& d, int mode)
#else
bool RealizeDirectory(const String& d)
#endif
{
    String dir = NormalizePath(d);
    Vector<String> p;
    while(dir.GetLength() > DIR_MIN) {
        p.Add(dir);
        dir = GetFileFolder(dir);
    }
    for(int i = p.GetCount() - 1; i >= 0; i--)
        if(!DirectoryExists(p[i]))
#ifdef POSIX
            if(!DirectoryCreate(p[i], mode))
#else
            if(!DirectoryCreate(p[i]))
#endif
        return false;
    return true;
}
```

Mirek

Thank you, Mirek.

Subject: Re: DirectoryCreateMulti as an alternative for RealizeDirectory
Posted by [mirek](#) on Thu, 18 Dec 2008 10:19:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Now even fixed to work in POSIX (Just a typo, must be "PLATFORM_POSIX", not "POSIX").

Mirek
