## 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.