Subject: FormatBytes function
Posted by Oblivion on Thu, 20 Apr 2017 23:18:51 GMT
View Forum Message <> Reply to Message

Hello,

FormatBytes() is a very simple and useful formatter which can be used for displaying file/transfer size etc.

```
String FormatBytes(int64 bytes, bool kilo = false)
{
 const char* base1000[] = { "B", "KB", "MB", "GB", "TB", "PB", "EB", "YB", "ZB" };
 const char* base1024[] = { "B", "KiB", "MiB", "GiB", "TiB", "PiB", "EiB", "YiB", "ZiB" };

 int base = kilo ? 1000 : 1024;
 int exponent = floor(log2(bytes) / log2(base));

 return bytes < base
   ? FormatInt64(bytes)  << " " << (kilo ? base1000[0] : base1024[0])
   : Format("%.1f %s", bytes / pow(base, exponent), kilo ? base1000[exponent] :
base1024[exponent]);
}
```

Example output:

```
------------------------------ Base:1024(kibi)--Base:1000(kilo)

1) 1023 (bytes) ->   1023 B  1.0 KB
2) 2046 (bytes) ->   2.0 KiB  2.0 KB
3) 3069 (bytes) ->   3.0 KiB  3.1 KB
4) 4092 (bytes) ->   4.0 KiB  4.1 KB
5) 5115 (bytes) ->   5.0 KiB  5.1 KB
6) 6138 (bytes) ->   6.0 KiB  6.1 KB
7) 7161 (bytes) ->   7.0 KiB  7.2 KB
8) 8184 (bytes) ->   8.0 KiB  8.2 KB
9) 9207 (bytes) ->   9.0 KiB  9.2 KB
```

Since it has a simple and strict format (number/SPACE/unit) it can be parsed if needed.
Is it possible to add this, or something similar to this, to U++ formatters?
It is very handy if you work with files and data transfers.

Regards,

Oblivion

Subject: Re: FormatBytes function
Posted by wimpie on Sat, 05 Aug 2017 17:03:01 GMT
View Forum Message <> Reply to Message

A while back I made a similar thing, though with hardcoded constants for division.
I couldn't hardcode the divisions needed for ZB and YB because of the limit of int64.
So I guess int64 can't handle those

Subject: Re: FormatBytes function
Posted by Oblivion on Sat, 05 Aug 2017 20:22:32 GMT
View Forum Message <> Reply to Message

Hello,

Sure. This version isn't really meant to hande YB and ZB.
I'd reserved those constants for future use, for I was going to implement a custom formatter instead of FormatInt64 (and change the function signature).
However, IMO, TB is practically the limit.
While I personally use this function (it is handy), there is already an undocumented FormatFileSize() function in U++ (which I later discovered).

Best regards,
Oblivion.

Subject: Re: FormatBytes function
Posted by wimpie on Sat, 05 Aug 2017 22:16:10 GMT
View Forum Message <> Reply to Message

Well I picked the same identifiers as you did for as it seemed they exist and yes for now TB is (on consumers systems) probably enough for a while. When we're at the YB and ZB there's probably an uint256 or something