

## eXtplorer - PHP-based File Manager - Fehler #162

### Reporting wrong size for files larger than 2GB using default "filesize" PHP function

05/03/2014 07:18 PM - Venicios NC

<b>Status:</b>	Neu	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	Sören Eberhardt-Biermann	
<b>Category:</b>	Bug	
<b>Target version:</b>		
<b>eXtplorer Version:</b>	2.1.4	<b>Joomla! Version:</b> none
<b>Description</b> In "site\include\functions.php":  <pre>function get_file_size( \$abs_item) { // file size return @\$GLOBALS['ext_File']-&gt;filesize( \$abs_item ); }</pre> This function return wrong file sizes. There are many complaints. One user in php.net suggests a solution:  This functions returns the exact file size for file larger than 2 GB on 32 bit OS:  <pre>function file_get_size(\$file) { //open file \$fh = fopen(\$file, "r"); //declare some variables \$size = "0"; \$char = ""; //set file pointer to 0; I'm a little bit paranoid, you can remove this fseek(\$fh, 0, SEEK_SET); //set multiplicator to zero \$count = 0; while (true) { //jump 1 MB forward in file fseek(\$fh, 1048576, SEEK_CUR); //check if we actually left the file if ((\$char = fgetc(\$fh)) !== false) { //if not, go on \$count +; } else { //else jump back where we were before leaving and exit loop fseek(\$fh, -1048576, SEEK_CUR); break; } } //we could make \$count jumps, so the file is at least \$count * 1.000001 MB large //1048577 because we jump 1 MB and fgetc goes 1 B forward too \$size = bcmul("1048577", \$count); //now count the last few bytes; they're always less than 1048576 so it's quite fast \$fine = 0; while(false !== (\$char = fgetc(\$fh))) { \$fine +; } //and add them \$size = bcadd(\$size, \$fine); fclose(\$fh); return \$size; } ?&gt;</pre> I have tested it on X86, X64 and even o ARM based OS Raspbian (very light even with a 16 GB file in a external USB HDD) with no issues.		

