```php
<?php

/**
 * LaTeX Rendering Class
 * Copyright (C) 2003  Benjamin Zeiss <zeiss@math.uni-goettingen.de>
 * ----------------------------------------------------------------------
 * Changed to MathLatexRender Class for HAWK HHG by Gunnar Werner (C) 2016
 * to be used with MW 1.23 LTS and extension Math REL-1_23...
 * for Windows with GhostScript+ImageMagick+MikTeX instead of Texvc
 * Notice 1: In file Math.php loading this class has to be added:
 * $wgAutoloadClasses['MathLatexRender'] = $dir . 'MathLatexRender.php';
 * Notice 2: In file MathRenderer.php new default object $renderer has to be changed:
 * $renderer = new MathLatexRender( $tex, $params ); #$renderer = new MathTexvc( $tex, $params );
 * ----------------------------------------------------------------------
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2.1 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 * ----------------------------------------------------------------------
 * @author Benjamin Zeiss <zeiss@math.uni-goettingen.de>
 * => @author Gunnar Werner for HAWK HHG <gunnar.werner@hawk-hhg.de>
 * @version v0.8 => v0.8.2016.05
 * @package latexrender => class MathLatexRender for extension Math
 *
*/

/**
 * Error code:
 * 1. Too long formulas
 * 2. formula contains tags in the blacklist
 * 3. formula incorrect, can't be render
 * 4. can't exec tex command
 *    maybe directory unwritable,can't create temporary files
 * 5. formula image too big
 * 6. can't copy image file to cahed formula directory
 *    maybe ImageMagick fail
*/

// "extends MathRenderer" new by changes 2016
class MathLatexRender extends MathRenderer {

    // ================================================================================
    // Variable Definitions
    // ================================================================================
    var $_picture_path = '';
    var $_picture_path_httpd = '';
    var $_tmp_dir = '';
    // i was too lazy to write mutator functions for every single program used
    // just access it outside the class or change it here if nescessary
    var $_latex_path    = 'latex.exe';
    var $_dvips_path    = 'dvips.exe';
    var $_convert_path  = 'convert.exe';
    var $_identify_path = 'identify.exe';

    var $_formula_density = 120;
    var $_xsize_limit = 700;
    var $_ysize_limit = 700;
    var $_string_length_limit = 800;
    var $_font_size = 10;
    var $_latexclass = "article"; //install extarticle class if you wish to have smaller font sizes
    var $_tmp_filename;
    var $_image_format = "png"; //change to gif if you prefer but it's not clear
    // this most certainly needs to be extended. in the long term it is planned to use
    // a positive list for more security. this is hopefully enough for now. i'd be glad
    // to receive more bad tags !
    var $_latex_tags_blacklist = array(
        "include","def","command","loop","repeat","open","toks","output","input",
        "catcode","name","^^",
        "\\every","\\errhelp","\\errorstopmode","\\scrollmode","\\nonstopmode","\\batchmode",
        "\\read","\\write","csname","\\newhelp","\\uppercase", "\\lowercase","\\relax","\\aftergroup",
        "\\afterassignment","\\expandafter","\\noexpand","\\special"
```

```
    );
 var $_errorcode = 0;
 var $_errorextra = '';
// new by changes 2016
var $url = '';
var $text = '';

 // ================================================================================
 // constructor
 // ================================================================================
 /**
 * Initializes the class
 *
 * @param string path where the rendered pictures should be stored
 * @param string same path, but from the httpd chroot
 *//* replaced with function __constructor() below by changes 2016
 function LatexRender($picture_path,$picture_path_httpd,$tmp_dir) {
    $this->_picture_path = $picture_path;
    $this->_picture_path_httpd = $picture_path_httpd;
    $this->_tmp_dir = $tmp_dir;
    $this->_tmp_filename = md5(rand());
}
 */
// new constructor combining old function LatexRender()
// and function renderMath() from old class MathRenderer
// by changes 2016
public function __construct($latex_formula, $params = array() ) {

        global $wgMathDirectory,
                $wgMathPath,
                $wgTmpDirectory,
                $wgLaTexCommand,
                $wgDvipsCommand,
                $wgImageMagickConvertCommand,
                $wgImageMagickIdentifyCommand;

        $latex_formula = '\displaystyle ' . $latex_formula;

        // from old function LatexRender()
        $this->_picture_path = $wgMathDirectory;
        $this->_picture_path_httpd = $wgMathPath;
        $this->_tmp_dir = $wgTmpDirectory;
        $this->_tmp_filename = md5(rand());

        // check Math dir
        if(!file_exists($wgMathDirectory)){@mkdir($wgMathDirectory);}
        if(!file_exists($wgTmpDirectory)){@mkdir($wgTmpDirectory);}

        // Objects $latex and $url replaced with $this by changes 2016
        $this->_latex_path = $wgLaTexCommand;
        $this->_dvips_path = $wgDvipsCommand;
        $this->_convert_path = $wgImageMagickConvertCommand;
        $this->_identify_path = $wgImageMagickIdentifyCommand;

        $this->url = $this->getFormulaURL($latex_formula);

        // htmlentities for newer PHP versions changed 2016
        $alt_latex_formula = htmlentities($latex_formula, ENT_QUOTES | ENT_XHTML, "UTF-8");
        $alt_latex_formula = str_replace("\r", "", $alt_latex_formula);
        $alt_latex_formula = str_replace("\n", "", $alt_latex_formula);
        $alt_latex_formula = str_replace('\displaystyle ', "", $alt_latex_formula);

        // Objects $latex, $url and $text replaced with $this and values of html-attributes with double quotes by changes 2016
        if($this->url != false){
                $this->text = '<img src="'.$this->url.'" title="'.$alt_latex_formula.'" alt="'.$alt_latex_formula.'" class="tex" />';
        }
        else{
                $this->text = '[Unparseable or potentially dangerous latex formula. Error '.$this->_errorcode.' '.$this->_errorextra.']';
        }
}

// ================================================================================
// public functions
// ================================================================================

/**
 * Picture path Mutator function
 *
 * @param string sets the current picture path to a new location
 */
function setPicturePath($name) {
```

```
        $this->_picture_path = $name;
    }

    /**
     * Picture path Mutator function
     *
     * @returns the current picture path
     */
    function getPicturePath() {
        return $this->_picture_path;
    }

    /**
     * Picture path HTTPD Mutator function
     *
     * @param string sets the current httpd picture path to a new location
     */
    function setPicturePathHTTPD($name) {
        $this->_picture_path_httpd = $name;
    }

    /**
     * Picture path HTTPD Mutator function
     *
     * @returns the current picture path
     */
    function getPicturePathHTTPD() {
        return $this->_picture_path_httpd;
    }

    /**
     * Tries to match the LaTeX Formula given as argument against the
     * formula cache. If the picture has not been rendered before, it'll
     * try to render the formula and drop it in the picture cache directory.
     *
     * @param string formula in LaTeX format
     * @returns the webserver based URL to a picture which contains the
     * requested LaTeX formula. If anything fails, the resultvalue is false.
     */
    function getFormulaURL($latex_formula) {
        // circumvent certain security functions of web-software which
        // is pretty pointless right here

        $latex_formula = preg_replace("/>/i", ">", $latex_formula);
        $latex_formula = preg_replace("/</i", "<", $latex_formula);

        $formula_hash = md5($latex_formula);

        $filename = 'math-' . $formula_hash.".".$this->_image_format;
        $full_path_filename = $this->getPicturePath()."/".$filename;

        if (is_file($full_path_filename)) {
            return $this->getPicturePathHTTPD()."/".$filename;
        } else {
            // security filter: reject too long formulas
            if (strlen($latex_formula) > $this->_string_length_limit) {
                $this->_errorcode = 1;
                return false;
            }

            // security filter: try to match against LaTeX-Tags Blacklist
            for ($i=0;$i<sizeof($this->_latex_tags_blacklist);$i++) {
                if (stristr($latex_formula,$this->_latex_tags_blacklist[$i])) {
                        $this->_errorcode = 2;
                    return false;
                }
            }

            // security checks assume correct formula, let's render it
            if ($this->renderLatex($latex_formula)) {
               return $this->getPicturePathHTTPD()."/".$filename;
            } else {
                $this->_errorcode = 3;
                return false;
            }
        }
    }

// new for abstract function in "MathRenderer.php" by changes 2016
public function render(){
            return $this->text."\n";
```

```
}

// =============================================================================
// private functions
// =============================================================================

    /**
     * wraps a minimalistic LaTeX document around the formula and returns a string
     * containing the whole document as string. Customize if you want other fonts for
     * example.
     *
     * @param string formula in LaTeX format
     * @returns minimalistic LaTeX document containing the given formula
     */
    function wrap_formula($latex_formula) {
#       $string  = "\documentclass[".$this->_font_size."pt]{".$this->_latexclass."}\n";
#       $string .= "\usepackage[latin1]{inputenc}\n";
        $string  = "\documentclass{".$this->_latexclass."}\n";
        $string .= "\usepackage{amsmath}\n";
        $string .= "\usepackage{amsfonts}\n";
        $string .= "\usepackage{amssymb}\n";
        $string .= "\pagestyle{empty}\n";
        $string .= "\begin{document}\n";
        $string .= "$".$latex_formula."$\n";
        $string .= "\end{document}\n";

        return $string;
    }

    /**
     * returns the dimensions of a picture file using 'identify' of the
     * imagemagick tools. The resulting array can be adressed with either
     * $dim[0] / $dim[1] or $dim["x"] / $dim["y"]
     *
     * @param string path to a picture
     * @returns array containing the picture dimensions
     */
    function getDimensions($filename) {
        $output=exec($this->_identify_path." ".$filename);
            //For some reason this didn't work for me, I used
            //$commander = "identify ".$filename;
            //$output=exec($commander);
            //instead. This should work if Identify works on the commandline
        $result=explode(" ",$output);
        $dim=explode("x",$result[2]);
        $dim["x"] = $dim[0];
        $dim["y"] = $dim[1];

        return $dim;
    }

    /**
     * Renders a LaTeX formula by the using the following method:
     *  - write the formula into a wrapped tex-file in a temporary directory
     *    and change to it
     *  - Create a DVI file using latex (tetex)
     *  - Convert DVI file to Postscript (PS) using dvips (tetex)
     *  - convert, trim and add transparancy by using 'convert' from the
     *    imagemagick package.
     *  - Save the resulting image to the picture cache directory using an
     *    md5 hash as filename. Already rendered formulas can be found directly
     *    this way.
     *
     * @param string LaTeX formula
     * @returns true if the picture has been successfully saved to the picture
     *          cache directory
     */
    function renderLatex($latex_formula) {
        $latex_document = $this->wrap_formula($latex_formula);

        $current_dir = getcwd();

        chdir($this->_tmp_dir);

        // create temporary latex file
        $fp = fopen($this->_tmp_dir."/".$this->_tmp_filename.".tex","a+");
        fputs($fp,$latex_document);
        fclose($fp);

        // create temporary dvi file
        // The \"'s are used in case the path has spaces in it (same as below for dvi)
```

```
        $command = "\"".$this->_latex_path."\" --interaction=nonstopmode ".$this->_tmp_filename.".tex";
           //In my case this didn't output in the right directory (amongst other things)
           //so I hardcoded everything in (If you use this, adjust it to your directories)
           //$command = "\"C:\Program Files\MiKTeX\miktex\bin\latex\" --output-directory=D:\Wiki\www\images\\tmp\\ --
interaction=nonstopmode D:\Wiki\www\images\\tmp\\".$this->_tmp_filename.".tex";

        $status_code = exec($command);
         // added checking file by changes 2016
        if (!$status_code || ! file_exists($this->_tmp_filename.".dvi")) {
              $this->cleanTemporaryDirectory();
              chdir($current_dir);
              $this->_errorcode = 4;
              return false;
        }

        // convert dvi file to postscript using dvips
        $command = "\"".$this->_dvips_path."\" -q -E ".$this->_tmp_filename.".dvi"." -o ".$this->_tmp_filename.".ps";
        $status_code = exec($command);

        // imagemagick convert ps to image and trim picture
        $command = $this->_convert_path." -density ".$this->_formula_density.
                  " -trim -transparent \"#FFFFFF\" ".$this->_tmp_filename.".ps ".
                  $this->_tmp_filename.".".$this->_image_format;

        $status_code = exec($command);

        // test picture for correct dimensions
        $dim = $this->getDimensions($this->_tmp_filename.".".$this->_image_format);

        if ( ($dim["x"] > $this->_xsize_limit) or ($dim["y"] > $this->_ysize_limit)) {
           $this->cleanTemporaryDirectory();
           chdir($current_dir);
           $this->_errorcode = 5;
           $this->_errorextra = ": " . $dim["x"] . "x" . number_format($dim["y"],0,"","");
           return false;
        }

        // copy temporary formula file to cached formula directory
        $latex_hash = md5($latex_formula);
        $filename = $this->getPicturePath()."/math-".$latex_hash.".".$this->_image_format;

        $status_code = copy($this->_tmp_filename.".".$this->_image_format,$filename);

        $this->cleanTemporaryDirectory();

        if (!$status_code) {
              chdir($current_dir);
              $this->_errorcode = 6;
              return false;
        }
        chdir($current_dir);

        return true;
    }

    /**
     * Cleans the temporary directory
     */
    function cleanTemporaryDirectory() {

        $current_dir = getcwd();
        chdir($this->_tmp_dir);

         // check existence before deleting to prevent warnings added by changes 2016
         $file_types = array("tex", "aux", "log", "dvi", "ps", $this->_image_format);
         foreach($file_types as $f_typ){
           $tmp_file = $this->_tmp_dir."/".$this->_tmp_filename.".".$f_typ;
              if(file_exists($tmp_file)){
                unlink($tmp_file);
              }
              $tmp_file = "";
         }

        chdir($current_dir);
    }
}
?>
```