```php
<?php
/* WordPress debugging output */

/* Copyright 2017,2018,2019,2020  Tony Lewis  (email : tlewis@exelana.com)

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 3 of the License, or
   (at your option) any later version.

   This program 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 General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program; if not, write to the Free Software
   Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA

   https://www.gnu.org/licenses/gpl-3.0.txt
 */

/* ==================================================================================================
 * Debugging messages are enabled according to the contents of the file EXELANA_WORDPRESS. To
 * enable debugging of WordPress sources include:
 *   wordpress: true
 *
 * You must also indicate whether debugging messages from individual files are allowed; for
 * example:
 *   wp-includes/plugin.php: true
 *
 * Debugging messages for theme sources can be enabled at either the theme or individual
 * file level; for example:
 *   wp-content/themes/MyTheme: true
 *   wp-content/themes/MyTheme/template-Home.php: false
 *
 * ==================================================================================================
 * To include debugging in the Exelana debugging bar, add conditional calls in any WordPress
 * or them code to one or more of the following functions:
 *   - exelana_debug($message)  Adds "file: $message" to the Output bar
 *   - exelena_trace([$depth])  Adds a stack trace to the Output bar (with up to $depth lines.
 *
 * Content for the message passed to exelana_debug can be automatically generated by one of the
 * following functions:
 *   - exelana_describe_fn($fn)  Returns a description of the function (for example,
 *                               "MyClass::myMethod" or "myFunction")
 *   - exelana_include($file)    Returns 'include "$relfile"'
 *   - exelana_require($file)    Returns 'require "$relfile"'
 *
 * For exelana_include and exelana_require the file is shown relative to the root of Wordpress.
 *
 * ==================================================================================================
 * Examples:
 *  if (is_callable('exelana_debug'))
 *    exelana_debug(exelana_require(dirname(__FILE__).'/wp-blog-header.php'));
 *  Example: index.php: require "wp-blog-header.php"
 *
 *  if (is_callable('exelana_debug'))
 *    exelana_debug("add_filter('$tag',".exelana_describe_fn($function_to_add).')');
 *  Example: wp-includes/plugin.php: add_filter('the_content',MyPlugin::cbFilterContent
 *              (line 999 of wp-content/plugins/my-plugin/include/myplugin_class.php))
 *
 *  if (is_callable('exelana_debug'))
 *    exelana_trace(1);
 *  Example:
 *    add_filter from line 999 of file wp-content/plugins/my-plugin/include/myplugin_class.php
 */
#
```

```
global $exelana_wordpress;
$dbt = debug_backtrace();
$exelana_wordpress = array(
  'debug' => array(),
  'elide' => array(),
  'enable' => array(),
  'enabled' => false,
  'root' => dirname($dbt[count($dbt)-1]['file']),
);

if (!file_exists(dirname(__FILE__).'/EXELANA_WORDPRESS'))
{
  define('EXELANA_WORDPRESS', false);
} else {
  $load = false;
  if ($fh = fopen(dirname(__FILE__).'/EXELANA_WORDPRESS','r'))
  {
    while ($line = fgets($fh))
    {
      if (preg_match('/^([^:]*):\s*(\w+)\s*$/',$line,$matches))
        $exelana_wordpress['enable'][$matches[1]] =
          strtolower($matches[2]) == 'true' ? true : false;
      {
        $load = true;
        $exelana_wordpress['enabled'] = true;
      }
      if (preg_match('/^wp-content\/themes\//',$line)
       && $exelana_wordpress['enable'][$matches[1]])
        $load = true;
    }
    fclose($fh);
  }
  if ($load)
  {
    define('EXELANA_WORDPRESS', true);
#
```

```php
    /* private */ function _exelana_allowed(&$file)
    {
      global $exelana_wordpress;
      $dbt = debug_backtrace();
      $file = str_replace($exelana_wordpress['root'].'/','',$dbt[1]['file']);
      $what = $dbt[1]['function'] == 'exelana_trace' ? 'Trace' : 'Debug messages';

      // Check if theme debugging is enabled
      if (preg_match('/^wp-content\/themes\//',_exelana_dir($file)))
      {
        $suppressed = false;
        foreach ($exelana_wordpress['enable'] as $key => $value)
        {
          global $exksc;
          if (preg_match('/^'.str_replace('/','\\/',$key).'/',$file))
          {
            if ($exelana_wordpress['enable'][$key])
              return true;
            $suppressed = true;
          }
        }
        if (!array_key_exists($file,$exelana_wordpress['elide']))
        {
          if ($suppressed)
            array_push($exelana_wordpress['debug'],
              "$file: $what suppressed in EXELANA_WORDPRESS");
          else
            array_push($exelana_wordpress['debug'],
              "$file: No reference found in EXELANA_WORDPRESS");
          $exelana_wordpress['elide'][$file] = true;
        }
        return false;
      }

      // Check if file debugging is enabled
      if ($exelana_wordpress['enable']['wordpress']
       && array_key_exists($file,$exelana_wordpress['enable'])
       && $exelana_wordpress['enable'][$file])
      {
        return true;
      } else {
        if (!array_key_exists($file,$exelana_wordpress['elide']))
        {
          if ($is_theme)
          {
            if (array_key_exists($file,$exelana_wordpress['enable']))
              array_push($exelana_wordpress['debug'],
                "$file: $what suppressed in EXELANA_WORDPRESS");
            else
              array_push($exelana_wordpress['debug'],
                "$file: No reference found in EXELANA_WORDPRESS");
          } elseif (!$exelana_wordpress['enable']['wordpress'])
            array_push($exelana_wordpress['debug'],
              "$file: $what suppressed for Wordpress in EXELANA_WORDPRESS");
          elseif (array_key_exists($file,$exelana_wordpress['enable']))
            array_push($exelana_wordpress['debug'],
              "$file: $what suppressed in EXELANA_WORDPRESS");
          else
            array_push($exelana_wordpress['debug'],
              "$file: No reference found in EXELANA_WORDPRESS");
        }
        $exelana_wordpress['elide'][$file] = true;
        return false;
      }
    }
#
```

```
    /* public */ function exelana_debug($message)
    {
      global $exelana_wordpress;
      if (_exelana_allowed($file))
      {
        if (!preg_match('/'.basename($file).':/',$message))
          $message = _exelana_file($file).": $message";
        array_push($exelana_wordpress['debug'],$message);
      }
    }

    /* public */ function exelana_describe_fn($fn)
    {
      global $exelana_wordpress;
      if (is_object($fn) && get_class($fn) == 'Closure')
      {
        $rf = new ReflectionFunction($fn);
        return "anonymous function (line ".$rf->getStartLine().' of '.
          _exelana_file($rf->getFileName()).')';
      } elseif (is_array($fn) && array_key_exists(0,$fn) && array_key_exists(1,$fn)) {
        $class = is_object($fn[0]) ? get_class($fn[0]) : (string)$fn[0];
        $rf = new ReflectionMethod($class,$fn[1]);
        return $class."::$fn[1] (line ".$rf->getStartLine().' of '.
          _exelana_file($rf->getFileName()).')';
      } elseif (is_callable($fn)) {
        $rf = new ReflectionFunction("$fn");
        return "$fn (line ".$rf->getStartLine().' of '.
          _exelana_file($rf->getFileName()).')';
      } else {
        return "$fn [undefined]";
      }
    }

    /* private */ function _exelana_dir($file,$quote=false)
    {
      global $exelana_wordpress;

      $dir = dirname(str_replace($exelana_wordpress['root'].'/','',$file));
      if ($quote)
        $dir = '"'.$dir.'"';
      return $dir;
    }

    /* private */ function _exelana_file($file,$quote=false)
    {
      global $exelana_wordpress;

      $file = str_replace($exelana_wordpress['root'].'/','',$file);
      $file = str_replace('wp-content/themes/','',$file);
      if ($quote)
        $file = '"'.$file.'"';
      return $file;
    }

    /* public */ function exelana_include($file)
    {
      return 'include '._exelana_file($file,true);
    }

    /* public */ function exelana_require($file)
    {
      return 'require '._exelana_file($file,true);
    }
#
```

```
    /* public */ function exelana_trace($max=-1)
    {
      global $exelana_wordpress;

      if (!_exelana_allowed($file))
        return;
      $dbt = debug_backtrace();
      if ($max < 0)
        $max = count($dbt) - 1;
      for ($i = 1; $i <= $max; $i++)
      {
        $bt = $dbt[$i];
        $fn = (array_key_exists('class',$bt) ? $bt['class'].'::' : '').$bt['function'];
        $file = (array_key_exists('file',$bt) ? _exelana_file($bt['file']) : '');
        array_push($exelana_wordpress['debug'],
          "  ".$fn.(array_key_exists('line',$bt) ?
            ' from line '.$bt['line'].($file != '' ? ' of ' : '') : '').
          ($file != '' ? 'file '.$file : ''));
      }
    }
  } else {
    define('EXELANA_WORDPRESS', false);
  }
}
```