

```

<?
/*
 * @author KAD
 * Скрипт парсинга v.5
 *
 * - Добавлено кэширование страниц сайта-донора
 * - Функции для конфигурации вынесены вверх файла
 * - Экспорт теперь осуществляется в виде отдельных файлов
 */

// @set_time_limit(90000);

require_once('bootstrap.php');
@session_start();

// Идентификатор сайта
$site_id = 2;

// URL донора
$url = "http://videoglaz.ru";

// Количество элементов за шаг
$stepCnt = 50;
$curr = "руб.";

// Кодировки (если пустые, то строка парсинга обрабатывается стандартным способом)
// $encryption_from = "CP1251";
$encryption_from = "";
$encryption_to = "UTF-8";

// Массив соответствий
// для доп. свойства prop-69 где 69 это id свойства
$arr_conf = array(
    "shop_groups_value",
    "shop_groups_value",
    "shop_groups_value",
    "shop_groups_value",
    "shop_groups_value",

    "shop_items_catalog_marking",
    "shop_items_catalog_name",
    "shop_items_catalog_text",
    "shop_items_catalog_description",
    "shop_producers_list_value",
    "shop_items_catalog_image",
);
//
=====
=====
// ===== Функции для заточки под сайт
=====

```

```

=====
//
=====
=====

/*
* создает лист ссылок каталога
*/
function createParseList()
{
    global $url, $delim;

    $out = "";

    $groupsList = array(
        '//catalog.php?id=1419' => 'Безопасность здоровья',
        '//catalog.php?id=118' => 'Системы охранно-пожарной сигнализации',
        '//catalog.php?id=816' => 'Системы видеонаблюдения',
        '//catalog.php?id=1160' => 'Модули порошкового пожаротушения',
        '//videoregistrator-avtomobilnyi' => 'Автомобильные видеорегистраторы',
        '//catalog.php?id=1193' => 'Комплекты беспроводной GSM-сигнализации',
        '//catalog.php?id=1189' => 'Комплекты охранной сигнализации (проводные)',
        '//catalog.php?id=299' => 'Беспроводная GSM-сигнализация',
        '//catalog.php?id=717' => 'Радиоканальные системы',
        '//catalog.php?id=729' => 'Системы охраны периметра',
        '//catalog.php?id=316' => 'Извещатели охранные для помещений',
        '/catalog.php?id=531' => 'Поворотные Wi-Fi-камеры'

    );
    echo "Начали";
    $parseList = array();
    foreach($groupsList as $key => $value)
    {
        $group = $value;
        $data = loadCachePage($url . $key);
        $groupsData = $data->find('div.content', );
        $groupsData = $groupsData->find('ul.ul0', );
        if(isset($groupsData))
        {
            foreach($groupsData->find('li') as $group1Block)
            {
                $href1 = getHref('a', $group1Block);
                $group1 = getValue('a', $group1Block);

                $data1 = loadCachePage($url . $href1);
                $groups1Data = $data1->find('div.content', );
                $groups1Data = $groups1Data->find('ul.ul0', );
                if(isset($groups1Data))
                {
                    foreach($groups1Data->find('li') as $group2Block)

```

```

    {
        $href2 = getHref('a', $group2Block);
        $group2 = getValue('a', $group2Block);

        $data2 = loadCachePage($url . $href2);
        $groups2Data = $data2->find('div.content', );
        $groups2Data = $groups2Data->find('ul.ul0', );
        if(isset($groups2Data))
        {
            foreach($groups2Data->find('li') as
$group3Block)
            {
                $href3 = getHref('a', $group3Block);
                $group3 = getValue('a', $group3Block);

                $data3 = loadCachePage($url . $href3);
                $groups3Data = $data3->find('div.content',
);
                $groups3Data = $groups3Data->find('ul.ul0',
);
                if(isset($groups3Data))
                {
                    foreach($groups3Data->find('li') as
$group4Block)
                    {
                        $href4 = getHref('a', $group4Block);
                        $group4 = getValue('a',
$group4Block);

                        $data4 = loadCachePage($url .
$href4);
                        $groups4Data =
                        $groups4Data =
                        $groups4Data->find('ul.ul0', );
                        if(isset($groups4Data))
                        {
                            echo "<br/><b>есть еще
подгруппы</b>";

                            $groups4Data->clear();
                            unset($groups4Data);
                        }

                        $items4data =
                        $data4->find('.goodtable', );
                        if(isset($items4data))
                        {
                            $pref = $group . "~" . $group1
. "~" . $group2 . "~" . $group3 . "~" . $group4 . "~";
                            $parseList[] = array( 'href' =>
$href4, 'pref' => $pref);

```

```

groups4Data" . $url . $href4;

//echo "<br/>есть блок товаров в

$items4data->clear();
unset($items4data);
}

$data4->clear();
unset($data4);
}
$groups3Data->clear();
unset($groups3Data);
}

$items3data = $data3->find('.goodtable', );
if(isset($items3data))
{
    $pref = $group . "~" . $group1 . "~" .
$group2 . "~" . $group3 . "~" . "~";
    $parseList[] = array( 'href' => $href3,
'pref' => $pref);
    //echo "<br/>есть блок товаров в
groups3Data" . $url . $href3;

    $items3data->clear();
    unset($items3data);
}

    $data3->clear();
    unset($data3);
}
$groups2Data->clear();
unset($groups2Data);
}

$items2data = $data2->find('.goodtable', );
if(isset($items2data))
{
    $pref = $group . "~" . $group1 . "~" . $group2
. "~" . "~" . "~";
    $parseList[] = array( 'href' => $href2, 'pref'
=> $pref);
    //echo "<br/>есть блок товаров в groups2Data" .
$url . $href2;\

    $items2data->clear();
    unset($items2data);
}

    $data2->clear();
    unset($data2);
}
$groups1Data->clear();
unset($groups1Data);

```

```

    }

    $itemsldata = $data1->find('.goodtable', );
    if(isset($itemsldata))
    {
        $pref = $group . "~" . $group1 . "~" . "~" . "~" . "~";
        $parseList[] = array( 'href' => $href1, 'pref' =>
$pref);
        //echo "<br/>есть блок товаров в groups1Data" . $url .
$href1;

        $itemsldata->clear();
        unset($itemsldata);
    }

    $data1->clear();
    unset($data1);
}
$groupsData->clear();
unset($groupsData);
}

$itemsData = $data->find('.goodtable', );
if(isset($itemsData))
{
    $pref = $group . "~" . "~" . "~" . "~" . "~";
    $parseList[] = array( 'href' => $key, 'pref' => $pref);
    //echo "<br/>есть блок товаров в groupsData" . $url . $key;
    $itemsData->clear();
    unset($itemsData);
}

$data->clear();
unset($data);
}
echo "Парслист готов:" . count($parseList) . " страниц";
uLog("Parse list created!");
return array('parselist' => $parseList, 'out' => $out);
}

/*
* парсим со страницы каталога товаров, когда в подразделе произвольное число страниц
*/
function loadPage($pref, $href, $start)
{
    global $url, $delim, $stepCnt, $imgDir;

    $i = ;
    uLog("Loading page... {$pref} -> {$url}{$href}");
    $cOut = "";
    $aItems = array();
    $errcnt = ;

```

```

$send = false;
$finPage = (isset($_SESSION['finPage']))?$_SESSION['finPage'];;
$page_url = $url . $href;
if ($start != )
{
    $page = $start+1;
    $page_url .= "&page=" . $page;
}
echo $page_url."<br>";
//$page_url = 'http://zoo-galereya.ru/category/shlejki-dlja-sobak/';
$cData = loadCachePage($page_url);
if($start == )
{
    $pagination = $cData->find('td.nav div.pages', );
    if(isset($pagination))
    {
        $pagination = $pagination->find('a');

        $out = $pagination[count($pagination)-2]->plaintext;
        $out = str_replace("\n", "", str_replace("\r", "",
str_replace(';', ' ', $out)));

        $finPage = intval($out);
        $_SESSION['finPage'] = $finPage;
    }
    else
    {
        $_SESSION['finPage'] = ;
    }
}
$block = $cData->find("table.goodtable", );
//$rows = $block->find("tr");
//берем со всей страницы блоки tr
$rows = $cData->find("tr");
foreach ($rows as $row)
{
    // Разбор товаров
    $itemHref = getHref('a', $row);
    //если блок tr содержит ссылку с good то это товар
    $strpos = strpos($itemHref,"good.php?id=");
    if ($itemHref==' ' || $itemHref=='#' || $strpos===FALSE){
continue;
}
    // идем на страницу товара
    $itemHref = $url."/". $itemHref;
    uLog("Try to load first time... {$itemHref}");
    $iData = loadItem($itemHref, $pref);
    if (!$iData) //исправляет проблему с быстрой загрузкой ошибочной страницы
    {
        usleep(0.5 * 1000000);
        uLog("Try to load again... {$itemHref}");
    }
}

```

```

        $iData = loadItem($itemHref, $pref);
    }

    if (!$iData)
    {
        $errcnt++;
        error("Error: {$itemHref}");
    } else
    {
        $cOut .= $iData;
    }
}

$cData->clear();
unset($cData);

if($finPage != && $page < $finPage)
{
    $end = false;
}
else
{
    $end = true;
}

/*if ($start*$stepCnt >= count($rows))
{
    $end = true;
}*/

$return_arr = array('out' => $cOut, 'end' => $end, 'err' => $errcnt);

return $return_arr;
}

```

```

/*
* парсим со страницы товара
*/
function loadItem($href, $pref/*, $aProps*/)
{
    global $imgDir, $curr, $url;
    $pData = loadCachePage( $href );

    // Тут была проблема, если убрать трим, то все работает
    //if ($pData && trim($pData) != "")
    if ($pData)
    {
        $pData = $pData->find('.context', );
    }
}

```

```

//имя
$name = $pData->find('h1.pageHeader', );
if(isset($name))
{
    $name = $name->innertext;
    $name = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;', ' ', str_replace('&quot;', '"', str_replace('&bull;',
'.', str_replace('&rsaquo;', ' ', str_replace('~', '-', $name))))))));
}
else
{
    //echo "не нашли блок с именем";
    $name = '';
}
// echo "<br>Название: ".$name;

//краткое описание
$description = '';
$descr_mass = $pData->find('p');
foreach ($descr_mass as $descr_elem) {
if (strstr($descr_elem->prev_sibling(), "Описание")) {
    $descr = $descr_elem;
}
}
if(isset($descr))
{
    $elements = $descr->find('h2');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $elements = $descr->find('a');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $elements = $descr->find('table');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }
    $elements = $descr->find('div');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }

    //$descr = $descr->innertext;
    $descr = trim(str_replace("\n", "", str_replace("\r", "",

```

```

str_replace('&nbsp;',' ',' ', str_replace('&quot;','"', str_replace('&bull;','',
'.', str_replace('&rsaquo;', '', str_replace('~', '-', $descr))))));
    }
    else
    {
        //echo "не нашли блок с кратким описанием";
        $descr = '';
    }
// echo "<br>Краткое описание: ".$descr;

//полное описание
$text_mass = $pData->find('ul');
foreach ($text_mass as $text_elem) {
    if (strstr($text_elem->prev_sibling(),"Технические характеристики")) {
        $text = $text_elem;
    }
}
if(isset($text))
{
    $elements = $text->find('p');
    foreach ($elements as $elem)
    {
        $elem->outertext = '';
    }

    $text = $text->innertext;
    $text = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;',' ',' ', str_replace('&quot;','"', str_replace('&bull;','',
'.', str_replace('&rsaquo;', '', str_replace('~', '-', $text))))));
    }
    else
    {
        $text = '';
    }
}
//echo "<br>Полное описание: ".$text;

//артикул
$art = $pData->find('.sertext', );
if(isset($art))
{
    $art = $art->innertext;
    $art = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;',' ',' ', str_replace('&quot;','"', str_replace('&bull;','',
'.', str_replace('&rsaquo;', '', str_replace('~', '-', $art))))));
    $art = substr($art, 11);
}
else
{
    //echo "не нашли блок с артикулом";
}

```

```

        $art = '';
    }
    //echo "<br>Артикул: ".$art;

    //производитель
    $producer = $pData->find('td.desc div a', );
    if(isset($art))
    {
        $producer = $producer->innertext;
        $producer = trim(str_replace("\n", "", str_replace("\r", "",
str_replace('&nbsp;', ' ', str_replace('&quot;', '"', str_replace('&bull;',
'.', str_replace('&rsaquo;', ' ', str_replace('~', '-', $producer))))))));
    }
    else
    {
        //echo "не нашли блок с роизводителем";
        $producer = '';
    }
    //echo "<br>Производитель: ".$producer;

    //изображение
    $img_href = $pData->find('a.fancybox img', );
    $img_href = $img_href->getAttribute('src');
    if(isset($img_href))
    {
        $img = saveImg($img_href);
    }
    else
    {
        //echo "не нашли блок с роизводителем";
        $img='';
    }
    //echo "<br>Изображение: ".$img;

    //дополнительные свойства
    $property_mass = $pData->find('table.bbt');
    foreach ($property_mass as $property_elem)
    {
        if (strstr($property_elem->prev_sibling(), 'Параметры'))
        {
            $property = $property_elem;
        }
    }
    if(isset($property))
    {
        $property_setes = $property->find('tr');
        foreach ($property_setes as $property_set)

```

```

    {
        $id = $property_set->find('td',)->innertext;
        $val = $property_set->find('b',)->innertext;
        if ($val != "-"){
            setPropertiesValues($id, $val, $art);
        }
    }
}

else
{
    $property = '';
}

// Модификации
$mod = "";

uLog("Item loaded!<br>");

$name = trim($name);

$pData->Clear();
unset($pData);

return $pref . csvStr(array($art, $name, $text, $descr, $producer,
$img)) . $mod;
} else
{
    return false;
}
exit();
}

/*
* Извлекаем информацию о группах, для размещения в конце файла
*/
function loadGroup($pref, $cData)
{
    $groupDescr = getHTML("div.category-top-text p", $cData, 1);

    uLog("Group info loaded!");

    return $pref . csvStr(array('', '', '', '', '', '', '', '', '', '',
    $groupDescr));
}

//

```

```

=====
// ===== Функции движка
=====
=====
//
=====
=====

// Разделитель для CSV
$delim = "~";

// Директория изображений
$imgDir = 'parse/imgs/';
// Директория файлов
$fileDir = 'parse';
// Директория кэширования
$cacheDir = 'parsecache';

// Имя временного файла
$tmpItemName = CMS_FOLDER . $fileDir . DIRECTORY_SEPARATOR . "export";
// Имя лог файла
$logFile = "log.txt";
// Имя файла ошибок
$errorFile = "errors.txt";
// Имя конфиг файла
$configFile = CMS_FOLDER . $fileDir . DIRECTORY_SEPARATOR . "config.inf";

$log = "";
$errors = "";

define('CURRENT_SITE', $site_id);
$oSite = Core_Entity::factory('Site', CURRENT_SITE);
Core::initConstants($oSite);

include 'simple_html_dom.php';

$start = Core_Array::getPost('start');

if (Core_Array::getPost('cancel'))
{
    if (isset($_SESSION))
    {
        parseReset();
    }
}

if (isset($_GET['step']))
{
    $_SESSION['currentStep'] = $_GET['step'];
    $_SESSION['itemStart'] = ;
    unset($_SESSION['aItems']);
}

```

```

}
if (isset($_GET['items']))
{
    $_SESSION['itemscount'] = $_GET['items'];
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<head>
<meta content="text/html; charset=UTF-8" http-equiv="Content-Type"></meta>
</head>
<form name="parse" method='POST' action='/parse.php'>
<?

if (!$start)
{

    if (isset($_SESSION['parseList']))
    {?>
        <input type='submit' name='cancel' value ='Сбросить'><br>
        <input type="submit" name="start" value="Продолжить парсинг"/>
    <?> else {?>
        <input type="submit" name="start" value="Начать парсинг"/>
    <?
    }
}

function parseReset()
{
    global $imgDir, $fileDir;

    unset($_SESSION['parseList']);
    unset($_SESSION['currentStep']);
    unset($_SESSION['aItems']);
    unset($_SESSION['properties']);
    unset($_SESSION['config']);
    unset($_SESSION['finPage']);
    unset($_SESSION['itemStart']);

    //clearDir($fileDir . DIRECTORY_SEPARATOR);

    $_SESSION['producer'] = "";
}

function delim($str = '')
{
    global $delim;
    return $str . $delim;
}

```

```

}

function endl($str = '')
{
    return $str . "\n";
}

function csvStr(array $array)
{
    global $arr_conf;

    $out = "";
    foreach ($arr_conf as $key => $value)
    {
        $out .= delim(isset($array[$key]) ? $array[$key] : "");
    }

    return endl($out);
}

/*
 * получаем html элемента
 */
function getHTML($pattern, $data, $num = )
{
    if ($out = $data->find($pattern, $num))
    {
        $out = $out->innertext;
        //$out = str_replace("\n", "", str_replace("\r", "",
str_replace(';', ' ', str_replace('\"', '\"', $out)));
        $out = str_replace("\n", "", str_replace("\r", "", str_replace(';', ' ',
',', str_replace('&nbsp;', ' ', $out))));

        return $out;
    }
}

/*
 * получаем значение элемента
 */
function getValue($pattern, $data, $num = )
{
    if ($out = $data->find($pattern, $num))
    {
        $out = $out->plaintext;
        //$out = str_replace("\n", "", str_replace("\r", "",
str_replace(';', ' ', str_replace('\"', '\"', $out)));
        $out = str_replace("\n", "", str_replace("\r", "", str_replace(';', ' ',
',', $out)));

        return $out;
    }
}

```

```

    }
}

/*
* получаем ссылку элемента
*/
function getHref($pattern, $data, $num = )
{
    if ($out = $data->find($pattern, $num))
    {
        $out = $out->href;
        $out = str_replace("\n", "", str_replace("\r", "",
str_replace('&','&', $out)));

        return $out;
    }
}

/*
* получаем source элемента
*/
function getSrc($pattern, $data)
{
    if ($out = $data->find($pattern,))
    {
        $out = $out->src;
        $out = str_replace("\n", "", str_replace("\r", "", $out));

        return $out;
    }
}

function saveImg($img_h, $img_l = false)
{
    global $imgDir, $url;

    $img = '';
    //uLog( $url . $img_h );

    if ($img_h)
    {
        $img = substr_replace($img_h, '', strrpos($img_h, '/') , 1);
        $img = basename($img);

        // Если изображение не было скачено ранее
        if (!is_file(CMS_FOLDER . $imgDir . $img))
        {
            $oImage = @file_get_contents( $url . $img_h);
            if (!$oImage && $img_l)
            {
                $oImage = @file_get_contents( $url . $img_l);
            }
        }
    }
}

```

```

        }

        if ($oImage)
        {
            file_put_contents( CMS_FOLDER . $imgDir . $img ,
$oImage);
        }
    }

    return $img;
}

/* Обработчики доп. свойств ===== */
//lib

$aProperties = array();
$aPropertiesIdentified = array();
$aPropertiesValues = array();

// Поместить значение в массив, если такого нет
function insert($arr, $value)
{
    if (!in_array($value, $arr))
    {
        $arr[] = $value;
    }
    return $arr;
}

//user
// Добавляем доп. свойства и значения товара
function setPropertiesValues($id, $val, $art)
{
    global $aPropertiesValues, $aProperties;

    $aPropertiesValues[$art][$id] = $val;
    $aProperties = insert($aProperties, $id);
}

/* /Обработчики доп. свойств ===== */

function loadCachePage($url)
{
    global $cacheDir, $encryption_to, $encryption_from;
    $hash = md5($url);
    $filePath = $cacheDir . DIRECTORY_SEPARATOR . $hash . ".html";

```

```

if (!is_file($filePath))
{
    $sData = @file_get_contents($url);
    if($encryption_from != '' && $encryption_to != '')
    {
        $sData = iconv($encryption_to, $encryption_from . "//IGNORE",
$sData);
    }
    Core_File::Write($filePath, $sData);
    $data = str_get_html($sData);
} else
{
    $fileData = Core_File::Read($filePath);
    if($encryption_from != '' && $encryption_to != '')
    {
        $fileData = iconv($encryption_to, $encryption_from . "//IGNORE",
$fileData);
    }
    $data = str_get_html($fileData);
}
return $data;
}

```

// Логирование > -----

```

function setStatus($message)
{
    $token = "0A==";
    //$response =
file_get_contents("http://crm.kad.pw/service/set.status/?token=" . $token .
"&message=" . urlencode($message));
}

```

```

function uLog($txt)
{
    global $log;

    $new = "[" . date('H:i:s') . "] " . $txt . "\n";
    $log .= $new;
    echo "<br/>" . $txt;
}

```

```

function error($txt)
{
    global $error;

    $error .= $txt . "\n";
    uLog($txt);
}

```

```

function saveLogFile()

```

```

{
    global $log, $logFile;

    $tLog = $log;
    if (file_exists(CMS_FOLDER . $logFile))
    {
        $tLog = Core_File::read(CMS_FOLDER . $logFile);
        $tLog .= $log;
    }
    Core_File::write(CMS_FOLDER . $logFile, $tLog, 0644);
}

function saveErrorFile()
{
    global $error, $errorFile;

    $tError = $error;
    if (file_exists(CMS_FOLDER . $errorFile))
    {
        $tError = Core_File::read(CMS_FOLDER . $errorFile);
        $tError .= $error;
    }
    Core_File::write(CMS_FOLDER . $errorFile, $tError, 0644);
}

function clearLogErrorFiles()
{
    global $logFile, $errorFile;

    $fLogFile = CMS_FOLDER.$logFile;
    $fErrorFile = CMS_FOLDER.$errorFile;

    @unlink($fLogFile);
    @unlink($fErrorFile);
    uLog("Log and Error files deleted!");
}

// < Логирование -----

// упрощенная функция scandir
function myscandir($dir)
{
    $list = scandir($dir);
    unset($list[], $list[1]);
    return array_values($list);
}

// функция очищения папки
function clearDir($dir)
{
    $list = myscandir($dir);

```

```

foreach ($list as $file)
{
    if (is_dir($dir . $file))
    {
        clearDir($dir.$file.'/');
        rmdir($dir.$file);
    }
    else
    {
        unlink($dir.$file);
    }
}
}

// MAIN -----

$csv = "";

if (!isset($_SESSION['parseList']))
{
    if ($start)
    {
        clearLogErrorFiles();
        if (!is_dir($fileDir))
        {
            mkdir($fileDir);
        }
        if (!is_dir($imgDir))
        {
            mkdir($imgDir);
        }
        if (!is_dir($cacheDir))
        {
            mkdir($cacheDir);
        }

        $_SESSION['parseList'] = createParseList();
        $csv = $_SESSION['parseList']['out'];
        $_SESSION['parseList'] = $_SESSION['parseList']['parselist'];

        $_SESSION['itemStart'] = ;
        $_SESSION['producer'] = "";
        $_SESSION['finPage'] = ;
        $_SESSION['config']['conformation'] = $arr_conf;
    }
}

if (isset($_SESSION['parseList']))
{
    $steps = count( $_SESSION['parseList'] );
    $currentStep =

```

```

(isset($_SESSION['currentStep']))?$_SESSION['currentStep'];;
$start = (isset($_SESSION['itemStart']))?$_SESSION['itemStart'];;
if(!isset($_SESSION['finPage']))
{
    $_SESSION['finPage'] = ;
}
if ($start)
{
    if ( $currentStep < $steps )
    {
        // $return_arr =
loadBigPage($_SESSION['parseList'][$currentStep]['pref'],
$_SESSION['parseList'][$currentStep]['href'], $start);
        $return_arr =
loadPage($_SESSION['parseList'][$currentStep]['pref'],
$_SESSION['parseList'][$currentStep]['href'], $start);
        $csv .= $return_arr['out'];
        if (!empty($csv))
        {
            $aFile = array($aProperties);
            $aFile['csv'] = $csv;
            $aFile['properties'] = $aProperties;
            $aFile['property_values'] = $aPropertiesValues;

            $sFile = serialize($aFile);

            $sFilePath = $tmpItemName . "-" . $currentStep . "-" .
$start . ".kex";

            $_SESSION['config']['files'][] = basename($sFilePath);
            Core_File::write(
                $configFile
                , serialize($_SESSION['config'])
                , 0644
            );

            Core_File::write(
                $sFilePath
                , $sFile
                , 0644
            );
        } else
        {
            // $return_arr['end'] = true;
        }

        $RedirectTime = 2000;
        Core::factory('Core_Html_Entity_Script')
            ->type('text/javascript')
            ->value('setTimeout(function (){ document.parse.submit(); },
' . $RedirectTime . ')')
    }
}

```

```

->execute();

if ($return_arr['end'])
{
    $_SESSION['itemStart'] = ;
    $_SESSION['currentStep'] = (int)$currentStep + 1;
    $_SESSION['finPage'] = ;
} else
{
    $_SESSION['itemStart']++;
}
echo "<input type='hidden' name='start' value='1'/>";
} else
{
    //parseReset();
}
}

saveLogFile();
saveErrorFile();

setStatus("{currentStep} / {steps} ".$istart." -
".((int)$istart+(int)$stepCnt));
uLog("{currentStep} / {steps} ".$istart*$stepCnt." -
".(($istart+1)*$stepCnt));
}
?>
</form>

```