

```
<?
/*
 * @author KAD::Systems
 * Импортер спарсенных товаров v 4.4
 * 16-03-15
 *
 * 1.0.4 от 16-03-15
 - Поиск существующего доп. свойства с учетом сайта
 - Изменен разделитель на ~
 - Добавлена обработка кавычек ' &quot; '
 * 1.0.5 от 24-02-2016
 - Исправлено дублирование доп. свойств
 - Теперь в список попадают все доп. свойства, а не первое
 */

//@set_time_limit(90000);

require_once('bootstrap.php');

// Идентификатор сайта
$site_id = 3;
// Идентификатор магазина
$shop_id = 5;
// Идентификатор родительской группы
$start_group_id = ;
// Идентификатор родительской группы дополнительных свойств
$start_property_dir_id = ;

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

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

$curr = "py6.";
// Разделитель для CSV
$delim = "~";

$log = "";
$errors = "";
$properties_encoding = "";
```

```

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

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

if (Core_Array::getPost('cancel'))
{
    if (isset($_SESSION))
    {
        importReset();
    }
}
if (isset($_GET['step']))
{
    $_SESSION['importCurrentStep'] = $_GET['step'];
}
?>
<!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='/import.php'>
<?

if (!$start)
{

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

function importReset()
{
    unset($_SESSION['importCurrentStep']);
    unset($_SESSION['importList']);
    unset($_SESSION['conformation']);
    $_SESSION['vendor'] = "";
}

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

```

```

$propertiesImportFileName = CMS_FOLDER . "properties.csv";

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

// Импортируем доп. свойства магазину
function importProperties()
{
    global $shop_id, $aProperties, $aPropertiesIdentified,
    $propertiesImportFileName, $aPropertiesValues, $start_property_dir_id,
    $properties_encoding;

    $oShop_Item_Property_List =
    Core_Entity::factory('Shop_Item_Property_List', $shop_id);

    $pe = "";

    foreach ($aProperties as $id => $name)
    {
        $nativeName = $name;
        if (empty($properties_encoding))
        {
            $properties_encoding = mb_detect_encoding($name, array('utf-8',
            'cp-1251'));
        }

        // Перекодировать
        if ($properties_encoding != 'utf8')
        {
            $name = trim(mb_convert_encoding($name, 'utf8',
            $properties_encoding));
        }

        $tag_name = Core_Str::transliteration(strval($name));

        // Поиск существующего доп. свойства
        $oProperties = Core_Entity::factory('Property');
        $oProperties->queryBuilder()
            ->where('shop_id', '=', $shop_id)
            ->where('tag_name', '=', $tag_name);
        $aoProperties = $oProperties->findAll();
        $oProperty = null;
        if(count($aoProperties))
        {
            $oProperty = $aoProperties[];
        }

        if (isset($oProperty))
        {

```

```

        $oShopItemProperties =
Core_Entity::factory('shop_item_property');
        $oShopItemProperties->queryBuilder()
            //->where('property_id', '=', $oProperty->id)
            ->where('shop_id', '=', $shop_id);
        $oShopItemProperties = $oShopItemProperties->findAll();
    }
    if (!$oProperty && !isset($oShopItemProperties[]))
    {
        $oProperty = Core_Entity::factory('property');
        $oProperty->name = strval($name);
        $oProperty->shop_id = $shop_id;
        // Тип доп. свойств: 3 - список, 1 - строка
        $oProperty->type = 1;

        if ($oProperty->type == 3)
        {
            $oList = Core_Entity::factory('list');
            $oList->name = strval($name);
            $oList->save();

            $oProperty->list_id = $oList->id;
        }

        $oProperty->tag_name = $tag_name;
        // Каталог доп. свойств
        $oProperty->property_dir_id = $start_property_dir_id;
        $oShop_Item_Property_List->add($oProperty);
    }

    $aPropertiesIdentified[$nativeName] = $oProperty->id;
}

// Генерируем файл импорта
$file = buildPropertiesImportFile();
// Сохраняем файл импорта
Core_File::write($propertiesImportFileName, $file, 0644);
// Генерируем массив соответствий
$params = array();
$params['fname'] = $propertiesImportFileName;
$params['conformity'] = getPropertiesConformity();
$params['shop_id'] = $shop_id;
$params['encoding'] = $properties_encoding;

// Импортируем
if ($imp = import($params))
{
    showStat($imp);
}
// Открываем доступность для текущей группы
reset($aPropertiesValues);

```

```

$art = key($aPropertiesValues);
//foreach ($aPropertiesValues as $art => $value)
//{
    $oItem = Core_Entity::factory('shop_item')->getBymarking($art);
    if ($oItem)
    {
        $group_id = $oItem->shop_group_id;
        openProperties($group_id);
    }
//}
}

```

// Получить массив соответствий доп. свойств

```

function getPropertiesConformity()
{
    global $aPropertiesIdentified;

    $aPropertiesConformity = array('shop_items_catalog_marking');

    // Перечисляем доп. свойства
    foreach ($aPropertiesIdentified as $name => $id)
    {
        $aPropertiesConformity[] = "prop-" . $id;
    }
    return $aPropertiesConformity;
}

```

// Генерация файла импорта для доп. свойств

```

function buildPropertiesImportFile()
{
    global $aPropertiesValues, $aPropertiesIdentified, $delim;

    $out = "";

    // Перечисляем товары
    foreach ($aPropertiesValues as $art => $aItemValues)
    {
        $str = $art . $delim;

        // Перечисляем доп. свойства
        foreach ($aPropertiesIdentified as $name => $id)
        {
            if (isset($aItemValues[$name]))
            {
                $value = $aItemValues[$name];
                $str .= $value . $delim;
            }
            else
            {
                $str .= "" . $delim;;
            }
        }
    }
}

```

```

    }

    $str .= "\n";
    $out .= $str;
}

return $out;
}

// Открываем доступность свойств группе $group_id
function openProperties($group_id)
{
    global $shop_id, $aPropertiesIdentified;

    $oShop = Core_Entity::factory('shop', $shop_id);

    foreach ($aPropertiesIdentified as $name => $id)
    {
        $oProperty = Core_Entity::factory('property', $id);
        $oShop->Shop_Item_Property_For_Groups->allowAccess($oProperty->Shop_Item_Property->id, $group_id);

        // Открываем доступность родительской группе
        $oGroup = Core_Entity::factory('shop_group', $group_id);
        $oShop->Shop_Item_Property_For_Groups->allowAccess($oProperty->Shop_Item_Property->id, $oGroup->parent_id);
    }
}

/*
 * импорт в интернет -магазин
 */
function import($params)
{
    global $imgDir, $start_group_id, $delim;

    $shop_id = (isset($params["shop_id"]))?$params["shop_id"]:1;
    $oShop = Core_Entity::factory('Shop', $shop_id);

    $sSeparator = $delim;
    $sLimiter = '';
    $sLocale = (isset($params["encoding"]))?$params["encoding"]:"UTF-8";

    $sFileName = $params['fname'];
    $sTmpFileName = CMS_FOLDER . TMP_DIR . 'file_' . date("U") . '.csv';
    Core_File::upload($sFileName, $sTmpFileName);

    $max_time = 999;
    $max_count = 999;
    $imagesPath = "";

```

```

$aConformity = $params["conformity"];

if ($fInputFile = fopen($sTmpFileName, 'rb'))
{
    $oShop_Item_Import_Csv_Controller = new
Shop_Item_Import_Csv_Controller($oShop->id, $start_group_id);
    $oShop_Item_Import_Csv_Controller
        ->file($sTmpFileName)
        ->encoding($sLocale)
        ->csv_fields($aConformity)
        ->time($max_time)
        ->step($max_count)
        ->separator($sSeparator)
        ->limiter($sLimiter)
        ->imagesPath($imgDir)
        ->importAction(1)
        ->searchIndexation()
    ;
    $oShop_Item_Import_Csv_Controller->import();
    return $oShop_Item_Import_Csv_Controller;
}
}

function showStat($Shop_Item_Import_Csv_Controller)
{
    /*
    echo Core::_('Shop_Item.count_insert_item') . ' &#151; <b>' .
$Shop_Item_Import_Csv_Controller->getInsertedItemsCount() . '</b><br/>';
    echo Core::_('Shop_Item.count_update_item') . ' &#151; <b>' .
$Shop_Item_Import_Csv_Controller->getUpdatedItemsCount() . '</b><br/>';
    echo Core::_('Shop_Item.create_catalog') . ' &#151; <b>' .
$Shop_Item_Import_Csv_Controller->getInsertedGroupsCount() . '</b><br/>';
    echo Core::_('Shop_Item.update_catalog') . ' &#151; <b>' .
$Shop_Item_Import_Csv_Controller->getUpdatedGroupsCount() . '</b><br/>';
    */

    uLog(Core::_('Shop_Item.count_insert_item') . ' - ' .
$Shop_Item_Import_Csv_Controller->getInsertedItemsCount());
    uLog(Core::_('Shop_Item.count_update_item') . ' - ' .
$Shop_Item_Import_Csv_Controller->getUpdatedItemsCount());
    uLog(Core::_('Shop_Item.create_catalog') . ' - ' .
$Shop_Item_Import_Csv_Controller->getInsertedGroupsCount());
    uLog(Core::_('Shop_Item.update_catalog') . ' - ' .
$Shop_Item_Import_Csv_Controller->getUpdatedGroupsCount());
}

// Логирование > -----

function uLog($txt)
{

```

```

    global $log;

    $log .= "[" . date('H:i:s') . "] " . $txt . "\n";
    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($list);
}
function getKexFiles($dir)
{
    $aImportList = myscandir($dir);
    foreach($aImportList as $key => $value)
    {
        $ext = (($p = strrpos($value, '.')) !== false) ? substr($value,$p+1)
: '';
        if ($ext == "kex")
        {
            $list[] = $value;
        }
    }
    return array_values($list);
}
// MAIN -----

$csv = "";

if (!isset($_SESSION['importList']))
{
    if ($start)
    {
        clearLogErrorFiles();

        $aImportList = unserialize(Core_File::read($configFile));

        $_SESSION['importList'] = getKexFiles($fileDir);
        $_SESSION['conformation'] = $aImportList['conformation'];
    }
}

if (isset($_SESSION['importList']))
{
    $steps = count( $_SESSION['importList'] );
    $currentStep =
(isset($_SESSION['importCurrentStep']))?$_SESSION['importCurrentStep'];;
    $istart =
(isset($_SESSION['importItemStart']))?$_SESSION['importItemStart'];;

    if ($start)
    {
        if ( $currentStep < $steps )

```

```

{
    // Читаем из файла

    $sFile = Core_File::read(CMS_FOLDER . $fileDir .
DIRECTORY_SEPARATOR . $_SESSION['importList'][$currentStep]);

    $aKex = unserialize($sFile);

    $csv = $aKex['csv'];
    $aProperties = $aKex['properties'];
    $aPropertiesValues = $aKex['property_values'];

    $csv = str_replace('&quot;', '"', $csv);
    //var_dump($aKex);

    //пишем в файл
    Core_File::write($tmpItemName, $csv, 0644);

    // Импортируем
    $params = array();
    $params['fname'] = $tmpItemName;

    $params['conformity'] = $_SESSION['conformation'];
    $params['shop_id'] = $shop_id;

    if ($imp = import($params))
    {
        showStat($imp);
    }

    // Импортируем доп. свойства
    importProperties();

    $_SESSION['importCurrentStep'] = $currentStep + 1;

    // Редирект
    $RedirectTime = 500;
    Core::factory('Core_Html_Entity_Script')
        ->type('text/javascript')
        ->value('setTimeout(function (){ document.parse.submit(); },
' . $RedirectTime . ')')
        ->execute();

    echo "<input type='hidden' name='start' value='1'/>";
}
}

uLog("{ $currentStep } / { $steps } " . " " .
$_SESSION['importList'][$currentStep]);

saveLogFile();

```

```
    saveErrorFile();
```

```
}
```

```
?>
```

```
</form>
```