

```

<?php

defined('HOSTCMS') || exit('HostCMS: access denied.');
```

/\*\*
 \*
 \* Версия для HostCMS v.6x
 \* @author KAD
 \* http://www.artemkuts.ru/
 \* artem.kuts@gmail.com
 \*
 \* Копирование и использование файлов модуля
 \* в коммерческих целях ЗАПРЕЩЕНО
 \*
 Core\_Event::attach('Shop\_Group\_Model.onBeforeGetXml',
 array('Kad\_Shop\_Group\_Observers\_Minmaxpriceob', 'onBeforeGetXml'));
 \*
 \*/

```

class Kad_Shop_Group_Observers_Minmaxpriceob
{

    // Кэш
    static public $aCache = NULL;
    /**
     * The singleton instances.
     * @var mixed
     */
    static public $instance = NULL;
    public function __construct()
    {
        return self::instance();
    }
    static public function instance()
    {
        if (is_null(self::$instance))
        {
            self::$instance = new self();
        }

        return self::$instance;
    }

    // Добавляем информацию о минимальной и максимальной цене товара в группе
    static public function onBeforeGetXml($object, $operation)
    {
        /*if (CURRENT_SITE != 2)
        {
            return;
        }*/
        if (!isset(self::$aCache[$object->id]))

```

```

    {
    // Получаем подгруппы
    $aGroupsId = array();
    $currGroupId = $object->id;
    $aGroupsId[] = $currGroupId;
    $aoCurGroups = array(Core_Entity::factory('Shop_group',
$currGroupId));
    do{
        $aoParGroups = $aoCurGroups;
        $aCurGroupsId = array();
        $aoCurGroups = array();
        foreach($aoParGroups as $oParGroup)
        {
            $aCurGroupsId[] = $oParGroup->id;
            $aGroupsId[] = $oParGroup->id;
        }
        $aGroups = Core_Entity::factory('Shop_group');
        $aGroups->queryBuilder()
            ->where('parent_id', 'IN', $aCurGroupsId);
        $aoCurGroups = $aGroups->findAll();
    }while(count($aoCurGroups) != 0);
    $oQB = Core_QueryBuilder::select(array('max(price)', 'max'),
array('min(price)', 'min'))
        ->from('shop_items')
        ->where('shop_group_id', 'IN', $aGroupsId);
    $aRes = $oQB->execute()->asAssoc()->result();
    $min = $aRes[0]['min'];
    $max = $aRes[0]['max'];
    self::$aCache[$object->id]['min'] = $min;
    self::$aCache[$object->id]['max'] = $max;
    } else
    {
        $min = self::$aCache[$object->id]['min'];
        $max = self::$aCache[$object->id]['max'];
    }
    $object
        ->addEntity(
            Core::factory('Core_Xml_Entity')
                ->name('min_price')
                ->value($min)
        )
        ->addEntity(
            Core::factory('Core_Xml_Entity')
                ->name('max_price')
                ->value($max)
        );
    }
}

```