index.phtml

<?

Header('Content-Type: text/html; charset=utf-8');

include "./data/client.init.php";

include CMS\_BASE."data/cms.init.php";

if($access->status==ACCESS\_ADMIN){

$struct = & $root->create("admin\_end");

$struct->setFromBranch();}

echo $root->get($template);

?>

client.init.php

<?

define("DEBUG",false);define("MYSQL\_HOSTNAME","localhost");define("MYSQL\_DBUSER","kmk");define("MYSQL\_DBPASS","BFvnpTmxyDNQnVzJ");define("MYSQL\_DATABASE","kmk");define("MYSQL\_LOCAL","utf8");define("CMS\_BASE","/home/www/system/cmsUTF/");$LOGINS\_ARRAY = array("admin"=>"admin");define("CMS\_PREFIX","kmk");define("CMS\_DEVELOPMENT\_MODE",true);define("CLIENT\_OBJECTS\_PATH","data/objects/");define("CLIENT\_CLASSES\_PATH","data/classes/");define("CLIENT\_SHABLONS\_PATH","data/shablons/");define("CMS\_SHORT\_TITLE",1); define("CLIENT\_MENU\_DEPTH",0);define("CLIENT\_SHOW\_CHILDREN",0);define("CLIENT\_MAP\_DEPTH",3);define("CLIENT\_SEARCH\_RESULTS",15);define("CLIENT\_DATE\_FORMAT","d.m.Y");define("CLIENT\_DATE\_FORMAT\_D","d");define("CLIENT\_DATE\_FORMAT\_M","m"); # если отлично от mdefine("CLIENT\_DATE\_FORMAT\_Y","Y"); # если отлично от Ydefine("TO\_MAIL","");

$CMS\_INCLUDE\_CLASSES = array("cms.node.text","cms.node.news","cms.node.feedback","cms.node.link");

$CLIENT\_INCLUDE\_CLASSES = array();

?>

cmsUTF/conf.php

<?

if(!defined('CMS\_HTTP\_BASE')) define("CMS\_HTTP\_BASE","http://system.vm010.xcp.r52.ru/cmsUTF/");

if(!defined('CMS\_BASE')) define("CMS\_BASE","/home/www/system/cmsUTF/");

cmsUTF/data/classes/cms.config.php

<?

class cms\_config{ var $data;var $sdata;var $is\_main;

function cms\_config($sdata){

$this->sdata = $sdata;

$this->data = array();}

function get100($sid,$chid=false){

global $access,$CMS\_LAST\_MODIFIED;

$configFile = CLIENT\_OBJECTS\_PATH.$sid."/nconfig".(($chid!==false)?".".$chid:"").".html";

if($access->status!=ACCESS\_ADMIN){

$CMS\_LAST\_MODIFIED = max($CMS\_LAST\_MODIFIED,(is\_file($configFile)?filemtime($configFile):0));}

$this->data = file2str($configFile);

if($this->data!=""){

$this->data = unserialize($this->data);}else{$this->data = array();}

$this->data = array\_merge($this->sdata,$this->data);}

function get($sid,$chid=false){

global $access,$CMS\_LAST\_MODIFIED;

$configFile = CLIENT\_OBJECTS\_PATH.$sid."/nconfig".(($chid!==false)?".".$chid:"").".html";

if($access->status!=ACCESS\_ADMIN){

$CMS\_LAST\_MODIFIED = max($CMS\_LAST\_MODIFIED,(is\_file($configFile)?filemtime($configFile):0));}

$this->data = file2str($configFile);

if($this->data!=""){

$this->data = unserialize($this->data);}else{

$this->data = array();}

$this->data = array\_merge($this->sdata,$this->data);}

function set100($sid,$chid=false,$strip=true){

dir\_make(CLIENT\_OBJECTS\_PATH.$sid,true);

if(isset($\_POST)){

$this->get($sid,$chid);

$data = str2save($\_POST);

foreach($this->sdata as $key => $val){

if($key!="felements"){

if(isset($data[$key])) $this->data[$key] = $data[$key];

}else{

if(isset($data["felements"])){

$this->data["felements"] = $this->renum($data["felements"]);}}}

foreach($this->data as $key => $val){

$this->data[$key] = str\_replace(chr(13),'',str\_replace(chr(10),'',$val));}

str2file(CLIENT\_OBJECTS\_PATH.$sid."/nconfig".(($chid!==false)?".".$chid:"").".html",serialize($this->data));}}

function set($sid,$chid=false,$strip=true){

dir\_make(CLIENT\_OBJECTS\_PATH.$sid,true);

if(isset($\_POST)){

$this->get($sid,$chid);

$data = str2save($\_POST);

foreach($this->sdata as $key => $val){

if($key!="felements"){

if(isset($data[$key])) $this->data[$key] = $data[$key];}else{

if(isset($data["felements"])){

$this->data["felements"] = $this->renum($data["felements"]);}}}

foreach($this->data as $key => $val){

$this->data[$key] = str\_replace(chr(13),'',str\_replace(chr(10),'',$val));}

str2file(CLIENT\_OBJECTS\_PATH.$sid."/nconfig".(($chid!==false)?".".$chid:"").".html",serialize($this->data));}}

function save($sid,$chid=false){

dir\_make(CLIENT\_OBJECTS\_PATH.$sid,true);

$data = $this->data;

unset($this->data);

foreach($this->sdata as $key => $val){

if($key!="felements"){

if(isset($data[$key])) $this->data[$key] = $data[$key];}else{

if(isset($data["felements"])){

$this->data["felements"] = $this->renum($data["felements"]);}else{

$this->data["felements"] = $val;}}}

foreach($this->data as $key => $val){

$this->data[$key] = str\_replace(chr(13),'',str\_replace(chr(10),'',$val));}

str2file(CLIENT\_OBJECTS\_PATH.$sid."/nconfig".(($chid!==false)?".".$chid:"").".html",serialize($this->data));}

function check($sid,$chid=false){

return is\_file(CLIENT\_OBJECTS\_PATH.$sid."/nconfig".(($chid!==false)?".".$chid:"").".html");}

function remove($sid,$chid=false){

$file=CLIENT\_OBJECTS\_PATH.$sid."/nconfig".(($chid!==false)?".".$chid:"").".html";

@rename($file,$file.".delete");}

function renum($elements){

$elements\_ = array();

if(is\_array($elements) && count($elements)>0){

foreach($elements as $i => $element){

$elements\_[] = $element;}}

return $elements\_;}}

?>

cmsUTF/data/classes/cms.node.feedback\_ext.php

<?

$CMS\_NODE\_TYPES[99] = array(

"\_class"=>"cms\_node\_feedback\_ext",

"\_name"=>"раздел обратной связи (расширенный)",

"\_type"=>(99 << CMS\_TYPE\_BITS),

"\_permissions" => 7,

"\_development"=>false,

"\_pnames" => array(array("просмотр узла","show"),array("добавление записей","add"),array("просмотр списка записей","list")),

"\_actions" => array("show"=>0,"\_entry.add"=>1,"entries.list"=>2,"entry.show"=>2),

"\_config" => array("nblocks"=>"2,1","nmap"=>"","nroot"=>"","nsort"=>0,"nshow"=>1,"natpage"=>0,"npages"=>0,"ntype"=>0,"nfields"=>0,"ndate"=>0,"nmail"=>0,"fmail"=>"","fsubmit"=>"","msubmit"=>"","nantispam"=>0, "nclass"=>"","nshowtemplate"=>"","felements"=>array(

array("fname"=>"Имя","ftype"=>1,"ffieldtype"=>"input","fdefault"=>""),

array("fname"=>"Email","ftype"=>1,"ffieldtype"=>"email","fdefault"=>""),

array("fname"=>"Текст","ftype"=>1,"ffieldtype"=>"textarea","fdefault"=>"")

)));

class cms\_node\_feedback\_ext extends cms\_node\_text{ var $virtual\_inself = false;var $identificator = 99;var $editable = true;var $config = true;

function & cms\_node\_feedback\_ext($data){

parent::cms\_node\_text($data);

return $this;}

function handler(& $root, $event){

global $access,$cms\_alert,$template,$sql,$CMS\_ALERTS,$CMS\_LAST\_MODIFIED,$log\_id;

$struct\_bits = array(

1=>array("name"=>"required","title"=>"обязательно для заполнения"),

2=>array("name"=>"adminonly","title"=>"только для администратора"));

$struct\_bitsReserved = 5;

$struct\_types = array(

array("name"=>"input","title"=>"строка текста"),array("name"=>"email","title"=>"email адрес"),array("name"=>"textarea","title"=>"текст"),

array("name"=>"checkbox","title"=>"флажок"),array("name"=>"select","title"=>"поле со списком"),array("name"=>"file","title"=>"загрузка файла"),array("name"=>"hr","title"=>"горизонтальный разделитель"),array("name"=>"text","title"=>"статичный текст"),);

if($event != ''){

if(parent::handler($root, $event)){

return true;}}else{

$this->load();}

if($access->status!=ACCESS\_ADMIN && !$this->config->data["nantispam"]){

include\_once CMS\_CLASSES\_PATH."cms.antispam.php";

$AS = new AntiSpam(CLIENT\_OBJECTS\_PATH.$this->id."/");}

switch($event):

case "edit":

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("feedback\_ext.edit"));

$struct->set($this->data);

if(!empty($this->config->data['nmap'])) $this->config->data['nmap'] = str2html($this->config->data['nmap']);

$struct->set($this->config->data);

$struct->setFromBranch();

for($i=0;$i<sizeof($this->config->data["felements"]);$i++){

if(!empty($this->config->data["felements"][$i]["fvalue"])){

$this->config->data["felements"][$i]["fvalue"] \*= 1;

$this->config->data["felements"][$i]["fdefault"] = str\_replace("\r\n","\\n",$this->config->data["felements"][$i]["fdefault"]);}

$struct\_data = & $struct->create("form\_data");

$struct\_data->set($this->config->data["felements"][$i]);

$struct\_data->setFromBranch();}

$struct->set("bitsReserved",$struct\_bitsReserved);

foreach($struct\_types as $type => $values){

$available\_types = & $struct->create("available\_types",$values);}

foreach($struct\_bits as $type => $values){

$preavailable\_types = & $struct->create("preavailable\_types",$values);

$preavailable\_types->set("type",$type);}

return false; break;

case "\_entry.edit":

$eid = $\_GET["eid"]\*1;

if(!$eid || !$this->entry\_node->check($this->id,$eid)){

$cms\_alert->set("er","form.entry.edit");

$cms\_alert->redirect($access->self.$this->get\_full\_path());}

$this->entry\_node->set($this->id,$eid,false);

$sql->send("UPDATE ".CMS\_PREFIX."\_node\_feedback SET epublished='".$this->entry\_node->data["epublished"]."' WHERE eid=".$eid);

if($this->entry\_node->data["epublished"]==0){

cms\_tree("indexes",$this->id.":".$eid);

}else{

$sql->send("DELETE FROM ".CMS\_PREFIX."\_indexes WHERE sid=".$this->id." AND entryid='".$eid."'");

clear\_cash();}

update\_logs($log\_id,$this->id,$eid);

$cms\_alert->set("ok","form.entry.edit");

$cms\_alert->redirect($access->self.$this->get\_full\_path());

return true; break;

case "\_entry.remove":

$eid = $\_GET["eid"]\*1;

if(!$eid || !$this->entry\_node->check($this->id,$eid)){

$cms\_alert->set("er","form.entry.remove");

$cms\_alert->redirect($access->self.$this->get\_full\_path());}

$sql->send("INSERT INTO ".CMS\_PREFIX."\_garbage\_node\_feedback SELECT \* FROM ".CMS\_PREFIX."\_node\_feedback WHERE eid=".$eid);

$sql->send("DELETE FROM ".CMS\_PREFIX."\_node\_feedback WHERE eid=".$eid);

$this->entry\_node->get($this->id,$eid);

for($i=0;$i<sizeof($this->config->data["felements"]);$i++){

if($this->config->data["felements"][$i]["ftype"]==8 ){

$file = substr($this->entry\_node->data["udata\_".$i],strlen($access->base));

if(is\_file($file)) @rename($file,$file.".delete");}}

$this->entry\_node->remove($this->id,$eid);

$sql->send("DELETE FROM ".CMS\_PREFIX."\_indexes WHERE sid=".$this->id." AND entryid='".$eid."'");

clear\_cash();

update\_logs($log\_id,$this->id,$eid);

$cms\_alert->set("ok","form.entry.remove");

$cms\_alert->redirect($access->self.$this->get\_full\_path());

return true; break;

case "\_entry.add":

if($access->status==ACCESS\_ADMIN || $this->config->data["nantispam"] || (isset($AS) && $AS->check($\_POST['asword'],$this->id))){

$formsend = "";

include\_once CMS\_BASE."scripts/mail.php";

$mailstructure = new structure();

$mailroot = & $mailstructure->create('cms\_mail\_texts');

$mailroot->set\_file(cms\_files\_check("feedback\_ext.mail"));

$mailtext = & $mailroot->create("form\_send");

$mailtext->set($this->data);

$mailtext->set($root->sdata);

$data = str2save($\_POST,true);

$m = new Mail("u","html");

$m->from(CMS\_SYSTEM\_MAIL,'CMS Brane Mail Module');

$m->subj(CLIENT\_SITE\_TITLE.': Feedback data');

for($i=0;$i<sizeof($this->config->data["felements"]);$i++){

$mailrows = & $mailtext->create("form\_rows");

$type = $this->config->data["felements"][$i]["ffieldtype"];

if($type=="hr"){

$mailrow = & $mailrows->create("form\_hr");}elseif($type=="text"){

$mailrow = & $mailrows->create("form\_text",str2html($this->config->data["felements"][$i]));}else{

if($type=="file"){

if($file = file\_upload("udata\_".$i,CLIENT\_OBJECTS\_PATH.$this->id."/")){

$m->attach\_file($file);

$\_POST["udata\_".$i] = $access->base.$file;

$data["udata\_".$i] = str2save($\_POST["udata\_".$i],true);}}

if(strlen($data["udata\_".$i])>0){

$mailrow = & $mailrows->create("form\_row",str2html($this->config->data["felements"][$i]));

$mailrow->set("fvalue",str2html($data["udata\_".$i],false,true,true));}

if (isset($data["udata\_".$i]) && $type=="checkbox"){

$mailrow = & $mailrows->create("form\_row",str2html($this->config->data["felements"][$i]));

$mailrow->set("fvalue","отмечен");}}}

$m->text($mailroot->get($template));

if($array = explode(";",$this->config->data["fmail"])){

for($j=0;$j<sizeof($array);$j++){

$array[$j]=trim($array[$j]);

if($array[$j]!=""){

$m->to($array[$j]); $m->send();}}}

if($this->config->data["ntype"]!=2){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_node\_feedback SET edate='".$this->entry\_node->sdata["edate"]."',epublished='".$this->config->data["ntype"]."',sid=".$this->id);

$eid = $sql->get\_id();

update\_logs($log\_id,$this->id,$eid);

$this->entry\_node->set($this->id,$eid);

if($this->config->data["ntype"]==0) cms\_tree("indexes",$this->id.":".$eid); }

$formsend = "?formsend=1";

$cms\_alert->set("ok","form.entry.add"); }else{

$cms\_alert->set("er","form.antispam"); }

$cms\_alert->redirect($access->self.$this->get\_full\_path().$formsend);

return true; break;

case "entry.edit":

$eid = $\_GET["eid"]\*1;

if(!$eid || !$this->entry\_node->check($this->id,$eid)){

$cms\_alert->set("er","form.entry.edit");

$cms\_alert->redirect($access->self.$this->get\_full\_path());}

$this->entry\_node->get($this->id,$eid);

$this->entry\_node->data["eid"] = $eid;

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("feedback\_ext.entry.edit"));

$struct->set($this->data);

$struct->set($this->config->data);

$struct->set($this->entry\_node->data);

$struct->setFromBranch();

$tinymce = & $struct->create("tinymce");

$tinymce->set\_file(cms\_files\_check("tinymce"));

$tinymce->setFromBranch();

for($i=0;$i<sizeof($this->config->data["felements"]);$i++){

$struct\_data = & $struct->create("form\_element");

$struct\_data->set($this->config->data["felements"][$i]);

$struct\_data->set("fsort",$i);

$struct\_data->setFromBranch();

$type = $this->config->data["felements"][$i]["ffieldtype"];

$required = $this->config->data["felements"][$i]["ftype"]&1;

$struct\_element = & $struct\_data->create("type\_".$type);

$struct\_element->setFromBranch($struct\_data->id());

$struct\_element->set("fvalue",str2html($this->entry\_node->data["udata\_".$i]));

switch($type){

case "select":

if($array = explode(";",$this->config->data["felements"][$i]["fdefault"])){

for($j=0;$j<sizeof($array);$j++){

$array[$j]=trim($array[$j]);

if($array[$j]!=""){

$struct\_element\_item = & $struct\_element->create("type\_select\_item");

$struct\_element\_item->set("fvalue",$array[$j]);

if($array[$j]==$this->entry\_node->data["udata\_".$i]) $struct\_element\_item->set("selected","selected");

$struct\_element\_item->setFromBranch($struct\_data->id());}}} break;

case "checkbox":

if($this->config->data["felements"][$i]["fdefault"]==$this->entry\_node->data["udata\_".$i])

$struct\_element->set("selected","checked"); break;

case "input":

case "textarea":

if($required) {

$check\_data = & $struct->create("check\_data",array("check\_type"=>-1));

$check\_data->setFromBranch($struct\_data->id());} break;

case "email":

if($required) {

$check\_data = & $struct->create("check\_data",array("check\_type"=>1));

$check\_data->setFromBranch($struct\_data->id());} break;}} return false; break;

case "entry.show":

$eid = $\_GET["eid"]\*1;

if(!$eid || !$this->entry\_node->check($this->id,$eid)){

$cms\_alert->set("er","form.entry.show");

$cms\_alert->redirect($access->self.$this->get\_full\_path());}

$this->entry\_node->get($this->id,$eid);

$this->entry\_node->data["eid"] = $eid;

if($access->status!=ACCESS\_ADMIN){

$CMS\_LAST\_MODIFIED = max($CMS\_LAST\_MODIFIED,$data["edate"]);}

$this->entry\_node->data["edate"] = int2date($data["edate"],CLIENT\_DATE\_FORMAT);

$this->entry\_node->data["edate\_d"] = int2date($data["edate"],defined("CLIENT\_DATE\_FORMAT\_D")?CLIENT\_DATE\_FORMAT\_D:"d");

$this->entry\_node->data["edate\_m"] = int2date($data["edate"],defined("CLIENT\_DATE\_FORMAT\_M")?CLIENT\_DATE\_FORMAT\_M:"m");

$this->entry\_node->data["edate\_y"] = int2date($data["edate"],defined("CLIENT\_DATE\_FORMAT\_Y")?CLIENT\_DATE\_FORMAT\_Y:"Y");

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("feedback\_ext.entry".($this->config->data["nshowtemplate"]!=""?".".$this->config->data["nshowtemplate"]:"").".show"));

$struct->set($this->data);

$struct->set($this->config->data);

$struct->set($this->entry\_node->data);

$struct->setFromBranch();

if($access->status==ACCESS\_ADMIN || !$this->config->data["ndate"]){

$detail\_node = & $struct->create("cms\_entry\_date");

$detail\_node->setFromBranch($struct->id());}

for($i=0;$i<sizeof($this->config->data["felements"]);$i++){

$type = $this->config->data["felements"][$i]["ffieldtype"];

if($type=="hr"){

$cms\_entry\_fields = & $struct->create("cms\_entry\_fields");

$struct\_node\_item = & $cms\_entry\_fields->create("cms\_entry\_hr");}elseif($type=="text"){

$cms\_entry\_fields = & $struct->create("cms\_entry\_fields");

$struct\_node\_item = & $cms\_entry\_fields->create("cms\_entry\_text",str2html($this->config->data["felements"][$i]));}else{

if(strlen($this->entry\_node->data["udata\_".$i])>0 && ($access->status==ACCESS\_ADMIN || !$this->config->data["nmail"] || $this->config->data["felements"][$i]["ffieldtype"]!="email")){

$cms\_entry\_fields = & $struct->create("cms\_entry\_fields");

$struct\_node\_item = & $cms\_entry\_fields->create("cms\_entry\_field",str2html($this->config->data["felements"][$i]));

switch($type){

case "textarea":

$struct\_node\_item->set("fvalue",str2html($this->entry\_node->data["udata\_".$i]));break;

default:

$struct\_node\_item->set("fvalue",str2html($this->entry\_node->data["udata\_".$i],false,true,true));}}}}return false; break;

default:

$tmp = cms\_files\_check("feedback\_ext".($access->status!=ACCESS\_ADMIN && $this->config->data["nshowtemplate"]!=""?".".$this->config->data["nshowtemplate"]:"").".show");

if(trim($this->config->data["nblocks"]) == '') $this->config->data["nblocks"] = '0';

$blocks = explode(',',$this->config->data["nblocks"]);

foreach($blocks as $key => $val){

switch($val):

case 0:

parent::handler($root, $event);

case 1:

$struct = & $root->create("cms\_node\_content");

$struct->set\_file($tmp);

$struct->set($this->data);

$struct->set($this->config->data);

$struct->setFromBranch();

if(cms\_permission($this,"entries.list")){

$query\_where = "sid = ".$this->id." ".(($access->status==ACCESS\_ADMIN)?"":"AND epublished=0 ");

$current = cms\_pages($struct,"SELECT COUNT(eid) as count FROM ".CMS\_PREFIX."\_node\_feedback WHERE ".$query\_where,$this->config->data["natpage"],!$this->config->data["npages"]);

$query = "SELECT \* FROM ".CMS\_PREFIX."\_node\_feedback WHERE ".$query\_where." ";

$query .= "ORDER BY edate ".(($this->config->data["nsort"])?"DESC":"ASC")." ";

if($this->config->data["natpage"]>0) $query .= " LIMIT ".($current\*$this->config->data["natpage"]).",".$this->config->data["natpage"]."";

$res = $sql->send($query);

while($data = $sql->get($res)){

$this->entry\_node->get($this->id,$data["eid"]);

if($access->status!=ACCESS\_ADMIN){

$CMS\_LAST\_MODIFIED = max($CMS\_LAST\_MODIFIED,$data["edate"]);}

$this->entry\_node->data["edate"] = int2date($data["edate"],CLIENT\_DATE\_FORMAT);

$this->entry\_node->data["edate\_d"] = int2date($data["edate"],defined("CLIENT\_DATE\_FORMAT\_D")?CLIENT\_DATE\_FORMAT\_D:"d");

$this->entry\_node->data["edate\_m"] = int2date($data["edate"],defined("CLIENT\_DATE\_FORMAT\_M")?CLIENT\_DATE\_FORMAT\_M:"m");

$this->entry\_node->data["edate\_y"] = int2date($data["edate"],defined("CLIENT\_DATE\_FORMAT\_Y")?CLIENT\_DATE\_FORMAT\_Y:"Y");

$this->entry\_node->data["eid"] = $data["eid"];

$struct\_node = & $struct->create("cms\_entry");

$struct\_node->set(str2html($this->entry\_node->data));

$struct\_node->setFromBranch($struct->id());

$struct\_node1 = & $struct->create("cms\_entry1");

$struct\_node1->set(str2html($this->entry\_node->data,false,true,true));

$struct\_node1->setFromBranch($struct->id());

if(strlen($this->entry\_node->data["udata\_0"])>0 && ($access->status==ACCESS\_ADMIN || !$this->config->data["nmail"] || $this->config->data["felements"][0]["ffieldtype"]!="email")){

$struct\_node->set("fvalue",str2html($this->entry\_node->data["udata\_0"],false,true,true));}

$fsize = sizeof($this->config->data["felements"]);

$fsize = ($this->config->data["nfields"]==0)?$fsize:min($this->config->data["nfields"],$fsize);

for($i=1;$i<$fsize;$i++){

if($access->status!=ACCESS\_ADMIN && $this->config->data["nmail"]==1 && $this->config->data["felements"][$i]["ffieldtype"]=="email") continue;

$type = $this->config->data["felements"][$i]["ffieldtype"];

if($type=="hr"){

$cms\_entry\_fields = & $struct\_node->create("cms\_entry\_fields");

$struct\_node\_item = & $cms\_entry\_fields->create("cms\_entry\_hr");}elseif($type=="text"){

$cms\_entry\_fields = & $struct\_node->create("cms\_entry\_fields");

$struct\_node\_item = & $cms\_entry\_fields->create("cms\_entry\_text",str2html($this->config->data["felements"][$i]));}else{

if(strlen($this->entry\_node->data["udata\_".$i])>0){

$cms\_entry\_fields = & $struct\_node->create("cms\_entry\_fields");

$struct\_node\_item = & $cms\_entry\_fields->create("cms\_entry\_field",str2html($this->config->data["felements"][$i]));

switch($type){

case "textarea":

$struct\_node\_item->set("fvalue",str2html($this->entry\_node->data["udata\_".$i]));break;

default:

$struct\_node\_item->set("fvalue",str2html($this->entry\_node->data["udata\_".$i],false,true,true));}}}}

if($fsize!=sizeof($this->config->data["felements"])){

$detail\_node = & $struct\_node->create("cms\_entry\_detail");

$detail\_node->setFromBranch($struct\_node->id());}

if($access->status==ACCESS\_ADMIN || !$this->config->data["ndate"]){

$detail\_node = & $struct\_node->create("cms\_entry\_date");

$detail\_node->setFromBranch($struct\_node->id());}

if($access->status==ACCESS\_ADMIN){

$struct\_node->set("cms\_class",($data["epublished"]==1)?"cms\_hidden":"");

$struct\_node\_admin = & $struct\_node->create("cms\_feedback\_admin");

$struct\_node\_admin->set\_file(cms\_files\_check("feedback\_ext.admin"));

$struct\_node\_admin->setFromBranch($struct\_node->id());}}}

if(cms\_permission($this,"\_entry.add") && (!isset($\_GET["formsend"]) || $this->config->data["nshow"]==1)){

$struct\_add = & $struct->create("cms\_entry\_add");

if($access->status!=ACCESS\_ADMIN && !$this->config->data["nantispam"]){

$imageAntiSpam = $AS->create(4,$this->id);

$detail\_node = & $struct\_add->create("cms\_antispam");

$detail\_node->setFromBranch($struct->id());

$detail\_node->set("imageAntiSpam",$imageAntiSpam);}

for($i=0;$i<sizeof($this->config->data["felements"]);$i++){

if($access->status!=ACCESS\_ADMIN && ($this->config->data["felements"][$i]["ftype"]&2)) continue;

$struct\_node\_items = & $struct\_add->create("form\_elements");

$type = $this->config->data["felements"][$i]["ffieldtype"];

if($type=="hr" OR $type=="text"){

$struct\_data = & $struct\_node\_items->create("form\_simple");}else{

$struct\_data = & $struct\_node\_items->create("form\_element");}

$struct\_data->set($this->config->data["felements"][$i]);

$struct\_data->set("fsort",$i);

$struct\_data->set("sid",$this->id);

$struct\_data->setFromBranch();

$required = $this->config->data["felements"][$i]["ftype"]&1;

$struct\_element = & $struct\_data->create("type\_".$type);

$struct\_element->setFromBranch($struct\_data->id());

switch($type){

case "select":

if($array = explode(";",$this->config->data["felements"][$i]["fdefault"])){

for($j=0;$j<sizeof($array);$j++){

$array[$j]=trim($array[$j]);

if($array[$j]!=""){

$struct\_element\_item = & $struct\_element->create("type\_select\_item");

$struct\_element\_item->set("fvalue",$array[$j]);

$struct\_element\_item->setFromBranch($struct\_data->id());}}} break;

case "checkbox":

if(!empty($this->entry\_node->data["udata\_".$i]) && $this->config->data["felements"][$i]["fdefault"]==$this->entry\_node->data["udata\_".$i])

$struct\_element->set("selected","checked"); break;

case "input":

case "textarea":

if($required) {

$check\_data = & $struct\_add->create("check\_data",array("check\_type"=>-1));

$check\_data->setFromBranch($struct\_data->id());} break;

case "email":

if($required) {

$check\_data = & $struct\_add->create("check\_data",array("check\_type"=>1));

$check\_data->setFromBranch($struct\_data->id());} break;}}

$struct\_add->setFromBranch($struct->id());} break;

case 2:

$struct = & $root->create("cms\_node\_content");

$struct->set\_file($tmp);

$cms\_text = & $struct->create("cms\_text");

$cms\_text->set("fpath",$this->data["fpath"]);

$cms\_text->set("parentpath",$this->parent->data["fpath"]);

$cms\_text->set($this->data);

$cms\_text->setFromBranch(); break;

case 3:

$this->config->data["nmap"] = str2html($this->config->data["nmap"]);

$struct = & $root->create("cms\_node\_content");

$struct->set\_file($tmp);

$cms\_text = & $struct->create("cms\_map");

$cms\_text->set($this->data);

$cms\_text->set("nmap",$this->config->data['nmap']);

$cms\_text->set("fpath",$this->data["fpath"]);

$cms\_text->set("parentpath",$this->parent->data["fpath"]);

$cms\_text->setFromBranch(); break;

endswitch;}

return false; break;

endswitch;

return false;}

function search($with\_entries=false){

global $sql;

$strings = array();

$strings[0] = (parent::search());

if($with\_entries){

$query = "SELECT eid FROM ".CMS\_PREFIX."\_node\_feedback WHERE sid = ".$this->id." AND epublished=0";

$res = $sql->send($query);

while($data = $sql->get($res)){

$strings[$data["eid"]] = $this->entry\_search($data["eid"]);}}

return $strings;}

function entry\_search($entryid=0){

global $sql;

$entry\_string = "";

if($entryid\*1>0){

$this->entry\_node->get($this->id,$entryid);

$this->entry\_node->data["eid"] = $entryid;

if(true){

for($i=0;$i<sizeof($this->config->data["felements"]);$i++){

if(strlen($this->entry\_node->data["udata\_".$i])>0 && ($this->config->data["felements"][$i]["ffieldtype"]=="textarea" || $this->config->data["felements"][$i]["ffieldtype"]=="text")){

$entry\_string .= " ".$this->entry\_node->data["udata\_".$i];}}}}

return $entry\_string;}

function entry\_load($entryid=0){

global $sql;

$data = array();

if($entryid\*1>0){

$query = "SELECT edate FROM ".CMS\_PREFIX."\_node\_feedback WHERE eid = ".$entryid."";

$res = $sql->send($query);

if($data = $sql->get($res)){

$data["sname"] = int2date($data["edate"],CLIENT\_DATE\_FORMAT);

$data["fpath"] = $this->get\_full\_path();

$data["entrypath"] = "eid=".$entryid."&a=entry.show&";}}

return $data;}

function load(){

global $sql,$current\_node\_extends\_vars,$CMS\_NODE\_TYPES;

parent::load();

$this->config->get($this->id);

$current\_node\_extends\_vars = cms\_node\_extended\_vars($this->id);

foreach ($current\_node\_extends\_vars as $key => $value) {

if (array\_key\_exists($key,$CMS\_NODE\_TYPES[$this->identificator]["\_config"])) {

$this->config->data[$key] = $value;}}

$entry\_config\_array = array("edate"=>time(), "epublished"=>$this->config->data["ntype"]);

for($i=0;$i<sizeof($this->config->data["felements"]);$i++) $entry\_config\_array["udata\_".$i] = "";

$this->entry\_node = new cms\_config($entry\_config\_array); }

function set($data){

global $sql;

parent::set($data);

$this->config->set($this->id);}

function remove(){

global $sql;

parent::remove();

$sql->send("INSERT INTO ".CMS\_PREFIX."\_garbage\_node\_feedback SELECT \* FROM ".CMS\_PREFIX."\_node\_feedback WHERE sid=".$this->id."");

$sql->send("DELETE FROM ".CMS\_PREFIX."\_node\_feedback WHERE sid=".$this->id."");}

function get\_entry\_info(& $root,$entryid,$operation){

global $sql,$access;

$query1 = "SELECT edate as entryname, 'eid' as entrynameid, eid as entryid FROM ".CMS\_PREFIX.(strpos($operation,"remove")!==false?"\_garbage":"")."\_node\_feedback WHERE sid = '".($this->id)."' AND eid='".$entryid."'";

$res1 = $sql->send($query1);

while($data1 = $sql->get($res1)){

$data1["entryname"] = "Запись от ".int2date($data1["entryname"]);

$cms\_entry\_more = & $root->create("cms\_entry\_more",$data1);

$cms\_entry\_more->set($data1);

$cms\_entry\_more->set($this->data);

if(strpos($operation,"remove")===false){

$cms\_entry\_more\_ref = & $cms\_entry\_more->create("cms\_entry\_more\_ref");

$cms\_entry\_more\_ref->set($data1);

$cms\_entry\_more\_ref->set($this->data);

$cms\_entry\_more\_ref->set("self",$access->self);}else{

$cms\_entry\_more->create("cms\_entry\_more\_noref");}}}}

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_node\_feedback(";

$query .= "eid int(10) unsigned NOT NULL auto\_increment,";

$query .= "sid int(11) NOT NULL default '0',";

$query .= "epublished tinyint(3) NOT NULL default '0',";

$query .= "edate int(11) unsigned NOT NULL default '0',";

$query .= "PRIMARY KEY (eid),";

$query .= "KEY sid (sid),";

$query .= "KEY edate (edate)";

$query .= ")";

$CMS\_NODE\_SQL\_QUERY[] = $query;

?>

cmsUTF/data/classes/cms.node.news.php

<?

$CMS\_NODE\_TYPES[2] = array(

"\_class"=>"cms\_node\_news",

"\_name"=>"новостной раздел",

"\_type"=>(2 << CMS\_TYPE\_BITS),

"\_development"=>false,

"\_permissions" => 1,

"\_pnames" => array(array("просмотр узла","show")),

"\_actions" => array("show"=>0,"entry.show"=>0,"entries.show"=>0),

"\_config" => array("nmult"=>1,"nchildssort"=>0,"nchildsshow"=>0,"nmethod"=>0,"norder"=>0,"nw"=>120,"nh"=>90,"nroot"=>"","ntoshort"=>150,"nsort"=>0,"nmultiplicity"=>1,"natpage"=>0,"nfield"=>"ndate","ndate"=>0,"npages"=>0,"nh1"=>0,"nphoto"=>0,"nphotomethod"=>1,"nprevwidth"=>120,"nprevheight"=>90,"nmdlwidth"=>240,"nmdlheight"=>180,"nphotowidth"=>600,"nphotoheight"=>600,"nclass"=>"","nshowtemplate"=>""),

);

class cms\_node\_news extends cms\_node\_text{ var $identificator = 2;var $editable = true;var $config = true;var $entry = false;var $ext\_array = array("gif","jpg","png");

function cms\_node\_news($data){

parent::cms\_node\_text($data);

return $this;}

function handler(& $root, $event, $params=false){

global $access,$template,$sql,$cms\_alert,$CMS\_LAST\_MODIFIED,$log\_id,$cms\_current\_node;

if(!parent::handler($root, $event)){

switch($event):

case "\_entry.remove":

$nid = !empty($\_GET["nid"]) ? intval($\_GET["nid"]) : 0;

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE nid=".$nid." ");

if($ndata = $sql->get()){

$this->entry\_remove($nid);

$cms\_alert->set("ok","news.entry.remove");

update\_logs($log\_id,$this->id,$nid);}else{

$cms\_alert->set("er","news.entry.remove");}

$cms\_alert->redirect($access->self.$this->get\_full\_path());

return false; break;

case "\_entry.save":

$nid = !empty($\_GET["nid"]) ? intval($\_GET["nid"]) : 0;

$data["skeywords"] = str2save($\_POST["skeywords"],true);

$data["stitle"] = str2save($\_POST["stitle"],true);

$data["sdescription"] = str2save($\_POST["sdescription"],true);

$data["ntitle"] = str2save($\_POST["ntitle"],true);

$data["stitle"] = $data["stitle"]=="" ? $data["ntitle"] : $data["stitle"];

$data["ndate"] = str2date($\_POST["ndate"]);

$data["nurgencydate"] = str2date($\_POST["nurgencydate"]);

$data["sid"] = $this->id;

$query\_data = "ntitle='".$data["ntitle"]."',ndate='".$data["ndate"]."',nurgencydate='".$data["nurgencydate"]."',npublished=".($\_POST["npublished"]\*1).",sid=".$data["sid"].", stitle='".$data["stitle"]."', sdescription='".$data["sdescription"]."', skeywords='".$data["skeywords"]."'";

$res = $sql->send("SELECT nid FROM ".CMS\_PREFIX."\_node\_news WHERE nid=".$nid." ");

if($sql->get($res)){

$sql->send("UPDATE ".CMS\_PREFIX."\_node\_news SET ".$query\_data." WHERE nid=".$nid." ");

update\_logs($log\_id,$this->id,$nid);}else{

$sql->send("INSERT INTO ".CMS\_PREFIX."\_node\_news SET ".$query\_data."");

$nid = $sql->get\_id();

if($nid==0){

$cms\_alert->set("er","nosave");

$cms\_alert->redirect($access->self.$this->get\_full\_path());

return false; break;}

update\_logs($log\_id,$this->id,$nid,$act='\_entry.add');}

str2file(CLIENT\_OBJECTS\_PATH.$this->id."/npreview.".$nid.".html",str2save($\_POST["npreview"]));

str2file(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.".$nid.".html",str2save($\_POST["ntext"]));

if($\_POST["npublished"]\*1==0){

cms\_tree("indexes",$this->id.":".$nid);

}else{

$sql->send("DELETE FROM ".CMS\_PREFIX."\_indexes WHERE sid=".$this->id." AND entryid='".$nid."'");

clear\_cash();}

if(isset($this->config->data["nphoto"]) && $this->config->data["nphoto"]==0){

if ($\_POST["delete\_photo"]==1) {

foreach($this->ext\_array as $key => $ext){

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/img/".$nid.".".$ext);

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$nid.".".$ext);

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$nid.".".$ext); }}

$img\_name = file\_upload("nphotofile",CLIENT\_OBJECTS\_PATH.$this->id."/img/",$nid);

if($img\_name!=""){

$src\_ext = get\_extension($img\_name);

foreach($this->ext\_array as $key => $ext){

if($ext!=$src\_ext){

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/img/".$nid.".".$ext);

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$nid.".".$ext);

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$nid.".".$ext); }}

$dest\_ext = ($src\_ext=="gif" ? "png" : $src\_ext);

if($this->config->data["nphotomethod"]==0){

image\_resize($img\_name, $this->config->data["nprevwidth"], $this->config->data["nprevheight"],CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$nid.".".$dest\_ext);

image\_resize($img\_name, $this->config->data["nmdlwidth"], $this->config->data["nmdlheight"],CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$nid.".".$dest\_ext);

image\_resize($img\_name,$this->config->data["nphotowidth"],$this->config->data["nphotoheight"],CLIENT\_OBJECTS\_PATH.$this->id."/img/".$nid."\_original.".$dest\_ext);

image\_resize($img\_name,$this->config->data["nphotowidth"],$this->config->data["nphotoheight"],$img\_name);}else{

image\_preview($img\_name,$this->config->data["nprevwidth"], $this->config->data["nprevheight"],1, 1,CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$nid.".".$dest\_ext);

image\_preview($img\_name,$this->config->data["nmdlwidth"], $this->config->data["nmdlheight"],1, 1,CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$nid.".".$dest\_ext);

image\_resize($img\_name,$this->config->data["nphotowidth"],$this->config->data["nphotoheight"],CLIENT\_OBJECTS\_PATH.$this->id."/img/".$nid."\_original.".$dest\_ext);

image\_preview($img\_name,$this->config->data["nphotowidth"], $this->config->data["nphotoheight"],1, 1,$img\_name);}}}

$cms\_alert->set("ok","news.entry.save");

$cms\_alert->redirect($access->self.$this->get\_full\_path());

return false; break;

case "edit":

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("news.edit"));

$struct->set($this->data);

$struct->set($this->config->data);

$struct->setFromBranch();

$tinymce = & $struct->create("tinymce");

$tinymce->set\_file(cms\_files\_check("tinymce"));

$tinymce->setFromBranch();

return false; break;

case "entry.edit":

$nid = !empty($\_GET["nid"]) ? intval($\_GET["nid"]) : 0;

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE nid=".$nid." ");

if($data = $sql->get()){

$data["ndate"] = int2date($data["ndate"]);

$data["nurgencydate"] = ($data["nurgencydate"]==0)?"":int2date($data["nurgencydate"]);

$data["npreview"] = str2html(file2str(CLIENT\_OBJECTS\_PATH.$this->id."/npreview.".$nid.".html"),true);

$data["ntext"] = str2html(file2str(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.".$nid.".html"),true); }else{

$data["sid"] = $this->id;

$data["nid"] = 0;

$data["ndate"] = int2date();

$data["nurgencydate"] = "";

$data["npublished"] = 0;

$data["npreview"] = "";

$data["ntext"] = "";

$data["stitle"] = "";

$data["sdescription"] = "";

$data["skeywords"] = "";}

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("news.entry.edit"));

$struct->set($this->data);

$struct->set($this->config->data);

$struct->set($data);

$tinymce = & $struct->create("tinymce");

$tinymce->set\_file(cms\_files\_check("tinymce"));

$tinymce->setFromBranch();

$struct->set("client\_objects\_path",CLIENT\_OBJECTS\_PATH);

$struct->setFromBranch();

if(isset($this->config->data["nphoto"]) && $this->config->data["nphoto"]==0){

$cms\_node\_content\_photo = & $struct->create("cms\_node\_content\_photo");

$cms\_node\_content\_photo->setFromBranch();

$img = false;

foreach($this->ext\_array as $key => $ext){

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$data["nid"].".".$ext)) $img = CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$data["nid"].".".$ext;}

if($img){

$w=@getimagesize($img);

$data["bsizes"] = ($w[0]>=$w[1]?" height=".$this->config->data["nprevheight"]:" width=".$this->config->data["nprevwidth"]);

$data["previmg"]=$img;

$cms\_node\_content\_photo\_prev = & $cms\_node\_content\_photo->create("cms\_node\_content\_photo\_prev",$data);

$cms\_node\_content\_photo\_prev->setFromBranch($struct->id());}}

return false; break;

case "entry.show":

$nid = !empty($\_GET["nid"]) ? intval($\_GET["nid"]) : 0;

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE nid=".$nid." AND sid=".$this->id."");

if($data = $sql->get()){

$previewFile = CLIENT\_OBJECTS\_PATH.$this->id."/npreview.".$data["nid"].".html";

$textFile = CLIENT\_OBJECTS\_PATH.$this->id."/ntext.".$nid.".html";

$data = str2html($data);

if($access->status!=ACCESS\_ADMIN){

$CMS\_LAST\_MODIFIED = max($CMS\_LAST\_MODIFIED,$data["ndate"],(is\_file($textFile)?filemtime($textFile):0),(is\_file($previewFile)?filemtime($previewFile):0));}

$data["ndate\_d"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_D")?CLIENT\_DATE\_FORMAT\_D:"d");

$data["ndate\_m"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_M")?CLIENT\_DATE\_FORMAT\_M:"m");

$data["ndate\_y"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_Y")?CLIENT\_DATE\_FORMAT\_Y:"Y");

$data["ndate"] = int2date($data["ndate"],CLIENT\_DATE\_FORMAT);

$data["npreview"] = str2html(file2str($previewFile));

$data["ntext"] = str2html(file2str($textFile));

$data["ntitle"] = htmlspecialchars($data["ntitle"]);

$data["sid"] = $this->id;

$data["ppath"] = $this->parent->get\_full\_path();

$this->entry = ($data["stitle"]=="" ? $data["ntitle"] : $data["stitle"]);

$struct = & $root->create("cms\_node\_content");

$tmp = $this->config->data["nshowtemplate"];

if(!empty($\_GET['nshowtemplate'])) $tmp = $\_GET['nshowtemplate'];

$struct->set\_file(cms\_files\_check("news.entry".($access->status!==ACCESS\_ADMIN && $tmp!=""?".".$tmp:"").".show"));

$struct->set($this->data);

$struct->set($this->config->data);

$struct->set("client\_objects\_path",CLIENT\_OBJECTS\_PATH);

$struct->set("page",(isset($\_GET["p"])?$\_GET["p"]\*1:0));

$struct->set($data);

$struct->setFromBranch();

if(!$this->config->data["ndate"]){

$detail\_node = & $struct->create("cms\_entry\_date");

$detail\_node->setFromBranch($struct->id());}

if($this->config->data["nh1"]\*1==1) $root->set('content\_title',$data["ntitle"]);

if(isset($this->config->data["nphoto"]) && $this->config->data["nphoto"]==0){

$data["sid"] = $this->id;

$img = false;

$previmg = false;

$mdlimg = false;

$origimg = false;

foreach($this->ext\_array as $key => $ext){

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"].".".$ext)) $img = CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"].".".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"]."\_original.".$ext)) $origimg = CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"]."\_original.".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$data["nid"].".".$ext)) $previmg = CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$data["nid"].".".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$data["nid"].".".$ext)) $mdlimg = CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$data["nid"].".".$ext;}

if($mdlimg){

if($w = @getimagesize($mdlimg)){

$data["mdlwidth"]=$w[0];

$data["mdlheight"]=$w[1];

$data["mdlsizes"]=$w[3];

$data["mdlimg"]=$mdlimg;}}

if($origimg){

if($w = @getimagesize($origimg)){

$data["origwidth"]=$w[0];

$data["origheight"]=$w[1];

$data["origsizes"]=$w[3];

$data["origvolume"]=size\_convert(filesize("./".$origimg));

$data["origimg"]=$origimg;}}

if($img){

if($w = @getimagesize($img)){

$data["width"]=$w[0];

$data["height"]=$w[1];

$data["sizes"]=$w[3];

$data["volume"]=size\_convert(filesize("./".$img));

$data["img"]=$img;}}

if($previmg){

if($w = @getimagesize($previmg)){

$data["previmg"]=$previmg;

$data["prevsizes"]=$w[3];

$cms\_entry\_photo = & $struct->create("cms\_entry\_photo",$data);

$cms\_entry\_photo->setFromBranch($struct->id());}}}

if($access->status==ACCESS\_ADMIN){

$struct\_node\_admin = & $struct->create("cms\_news\_admin");

$struct\_node\_admin->set\_file(cms\_files\_check("news.admin"));

$struct\_node\_admin->setFromBranch($struct->id());}}

return false; break;

case "entries.show":

$struct = & $root->create("cms\_node\_content");

if(!isset($params["nshowtemplate"])) $params["nshowtemplate"] = '';

if(!isset($params["natpage"])) $params["natpage"] = 0;

if(!isset($params["npages"])) $params["npages"] = 0;

if(!isset($params["nphoto"])) $params["nphoto"] = 0;

$struct->set\_file(cms\_files\_check("news".($params["nshowtemplate"]==''?"":".".$params["nshowtemplate"]).".show"));

$query\_where = $params["query"];

$current = cms\_pages($struct,"SELECT COUNT(nid) as count FROM ".CMS\_PREFIX."\_node\_news WHERE ".$query\_where,$params["natpage"],!$params["npages"]);

$params["nfield"] = ($params["nfield"]=="nrand")?" RAND() ":$params["nfield"];

$query = "SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE ".$query\_where." ";

if($params["natpage"]>0) $query .= " LIMIT ".($current\*$params["natpage"]).",".$params["natpage"]."";

$res = $sql->send($query);

if($sql->num($res)>0){

$struct\_list = & $struct->create("cms\_entries\_list");

$struct\_list->setFromBranch($struct->id());}

$iii = 0;

while($data = $sql->get($res)){

$data = str2html($data);

$data["number"] = ++$iii;

$previewFile = CLIENT\_OBJECTS\_PATH.$data["sid"]."/npreview.".$data["nid"].".html";

$data["ndate\_d"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_D")?CLIENT\_DATE\_FORMAT\_D:"d");

$data["ndate\_m"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_M")?CLIENT\_DATE\_FORMAT\_M:"m");

$data["ndate\_y"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_Y")?CLIENT\_DATE\_FORMAT\_Y:"Y");

$data["ntitle"] = htmlspecialchars($data["ntitle"]);

$data["ndate"] = int2date($data["ndate"],CLIENT\_DATE\_FORMAT);

$data["npreview"] = str2html(file2str($previewFile));

$data["nshort"] = substr(strip\_tags($data["npreview"]),0,100);

$struct\_node = & $struct\_list->create("cms\_entry");

$ppp = cms\_tree("getbyid",$data["sid"]);

$struct\_node->set($data);

if($iii==1) $struct\_list->set($data);

$struct\_node->set("fpath",$ppp->get\_full\_path());

$struct\_node->set("self",$access->self);

$struct\_node->set("base",$access->base);

$struct\_node->set("client\_objects\_path",CLIENT\_OBJECTS\_PATH);

$struct\_node->setFromBranch();

if(!isset($params["ndate"]) || (isset($params["ndate"]) && $params["ndate"]==0)){

$detail\_node = & $struct\_node->create("cms\_entry\_date");

$detail\_node->setFromBranch($struct\_node->id());}

if($params["nphoto"]==1){

$img = false;

$previmg = false;

$mdlimg = false;

$origimg = false;

foreach($this->ext\_array as $key => $ext){

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/".$data["nid"]."\_original.".$ext)) $origimg = CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/".$data["nid"]."\_original.".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/".$data["nid"].".".$ext)) $img = CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/".$data["nid"].".".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/preview/".$data["nid"].".".$ext)) $previmg = CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/preview/".$data["nid"].".".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/middle/".$data["nid"].".".$ext)) $mdlimg = CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/middle/".$data["nid"].".".$ext;}

if($mdlimg){

if($w = @getimagesize($mdlimg)){

$data["mdlwidth"]=$w[0];

$data["mdlheight"]=$w[1];

$data["mdlsizes"]=$w[3];

$data["mdlimg"]=$mdlimg;}}

if($img){

if($w = @getimagesize($img)){

$data["width"]=$w[0];

$data["height"]=$w[1];

$data["volume"]=size\_convert("./".filesize($img));

$data["img"]=$img;

if($iii==1) $struct\_list->set('first\_img',$img);}}

if($previmg){

if($w = @getimagesize($previmg)){

$data["previmg"]=$previmg;

$data["prevsizes"]=$w[3];

$cms\_entry\_photo = & $struct\_node->create("cms\_entry\_photo",$data);

$cms\_entry\_photo->setFromBranch($struct\_node->id());}}}}

return false; break;

default:

$struct = & $root->create("cms\_node\_content");

$tmp = $this->config->data["nshowtemplate"];

if(!empty($\_GET['nshowtemplate'])) $tmp = $\_GET['nshowtemplate'];

$struct->set\_file(cms\_files\_check("news".($access->status!==ACCESS\_ADMIN && $tmp!=""?".".$tmp:"").".show"));

$struct->set($this->data);

$struct->set($this->config->data);

$struct->setFromBranch();

$query\_where = "sid = ".$this->id;

if($access->status==ACCESS\_ADMIN){

$query\_where .= ""; }else{

$query\_where .= " AND npublished=0 AND ".time().">=ndate AND ( nurgencydate=0 OR ".time()."<=nurgencydate)";}

$current = cms\_pages($struct,"SELECT COUNT(nid) as count FROM ".CMS\_PREFIX."\_node\_news WHERE ".$query\_where,$this->config->data["natpage"],!$this->config->data["npages"]);

$this->config->data["nfield"] = ($this->config->data["nfield"]=="nrand")?" RAND() ":$this->config->data["nfield"];

$query = "SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE ".$query\_where." ";

$query .= "ORDER BY ".$this->config->data["nfield"]." ".(($this->config->data["nsort"])?"DESC":"ASC")." ";

if($this->config->data["natpage"]>0) $query .= " LIMIT ".($current\*$this->config->data["natpage"]).",".$this->config->data["natpage"]."";

$res = $sql->send($query);

$\_cnt = 0;

if($sql->num($res)>0 || $access->status==ACCESS\_ADMIN){

$struct\_list = & $struct->create("cms\_entries\_list");

$struct\_list->setFromBranch($struct->id());

if($access->status==ACCESS\_ADMIN){

$struct\_admin = & $struct\_list->create("cms\_entries\_list\_admin");

$struct\_admin->setFromBranch($struct->id());}

$\_cnt = $sql->num($res);}

$\_nid = 0;

if(!empty($\_GET["a"]) && $\_GET["a"]=='entry.show' && !empty($\_GET["nid"]) && $cms\_current\_node->id==$this->id){

$\_nid = intval($\_GET["nid"]); }

$iii = 0;

while($data = $sql->get($res)){

if($data['nid'] != $\_nid){

$data = str2html($data);

$data["number"] = ++$iii;

$data["number\_prev"] = ($data["number"]-1)<1 ? $\_cnt : $data["number"]-1;

$data["number\_next"] = ($data["number"]+1)>$\_cnt ? 1 : $data["number"]+1;

$data["page"] = isset($\_GET["p"])?$\_GET["p"]\*1:0;

$data["nshowtemplate"] = $this->config->data["nshowtemplate"];

$entry\_attributes = array();

if ($\_nid == $data["nid"]) $entry\_attributes[] = "cms\_entry\_open";

$nmultiplicity = intval($this->config->data["nmultiplicity"]);

if($nmultiplicity>1 && fmod($iii,$nmultiplicity) == 0) $entry\_attributes[] = "lastcolumn";

$data["entry\_attributes"] = implode(' ', $entry\_attributes);

$data["nmultiplicity"] = $nmultiplicity;

$previewFile = CLIENT\_OBJECTS\_PATH.$this->id."/npreview.".$data["nid"].".html";

$textFile = CLIENT\_OBJECTS\_PATH.$this->id."/ntext.".$data["nid"].".html";

if($access->status!=ACCESS\_ADMIN){

$CMS\_LAST\_MODIFIED = max($CMS\_LAST\_MODIFIED,$data["ndate"],(is\_file($previewFile)?filemtime($previewFile):0));

}

$data["ndate\_d"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_D")?CLIENT\_DATE\_FORMAT\_D:"d");

$data["ndate\_m"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_M")?CLIENT\_DATE\_FORMAT\_M:"m");

$data["ndate\_y"] = int2date($data["ndate"],defined("CLIENT\_DATE\_FORMAT\_Y")?CLIENT\_DATE\_FORMAT\_Y:"Y");

$data["ndate"] = int2date($data["ndate"],CLIENT\_DATE\_FORMAT);

$data["ntitle"] = htmlspecialchars($data["ntitle"]);

$data["npreview"] = str2html(file2str($previewFile));

$data["nshort"] = strip\_tags($data["npreview"]);

$num = $this->config->data["ntoshort"];

if(mb\_strlen($data["nshort"]) > $num){

$len = mb\_strripos(mb\_substr($data["nshort"], 0, $num), ' ');

$data["nshort"] = mb\_substr($data["nshort"], 0, $len).'&#8230;';}

$data["ntext"] = str2html(file2str($textFile));

$data["sid"] = $this->id;

$struct\_node = & $struct\_list->create("cms\_entry");

$struct\_node->set($this->data);

if($iii==1) $struct\_list->set($data);

$struct\_node->set($data);

$struct\_node->set("client\_objects\_path",CLIENT\_OBJECTS\_PATH);

$struct\_node->setFromBranch();

if(!$this->config->data["ndate"]){

$detail\_node = & $struct\_node->create("cms\_entry\_date");

$detail\_node->setFromBranch($struct\_node->id());}

if(isset($this->config->data["nphoto"]) && $this->config->data["nphoto"]==0){

$data["sid"] = $this->id;

$img = false;

$previmg = false;

$mdlimg = false;

$origimg = false;

foreach($this->ext\_array as $key => $ext){

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"].".".$ext)) $img = CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"].".".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"]."\_original.".$ext)) $origimg = CLIENT\_OBJECTS\_PATH.$this->id."/img/".$data["nid"]."\_original.".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$data["nid"].".".$ext)) $previmg = CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$data["nid"].".".$ext;

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/middle/".$data["nid"].".".$ext)) $mdlimg = CLIENT\_OBJECTS\_PATH.$data["sid"]."/img/middle/".$data["nid"].".".$ext;}

if($mdlimg){

if($w = @getimagesize($mdlimg)){

$data["mdlwidth"]=$w[0];

$data["mdlheight"]=$w[1];

$data["mdlsizes"]=$w[3];

$data["mdlimg"]=$mdlimg;}}

if($origimg){

if($w = @getimagesize($origimg)){

$data["origwidth"]=$w[0];

$data["origheight"]=$w[1];

$data["origsizes"]=$w[3];

$data["origvolume"]=size\_convert(filesize("./".$origimg));

$data["origimg"]=$origimg;}}

if($img){

if($w = @getimagesize($img)){

$data["width"]=$w[0];

$data["height"]=$w[1];

$data["sizes"]=$w[3];

$data["volume"]=size\_convert(filesize("./".$img));

$data["img"]=$img;

if($iii==1) $struct\_list->set('first\_img',$img);}}

if($previmg){

if($w = @getimagesize($previmg)){

$data["previmg"]=$previmg;

$data["prevsizes"]=$w[3];

$cms\_entry\_photo = & $struct\_node->create("cms\_entry\_photo",$data);

$cms\_entry\_photo->setFromBranch($struct\_node->id());}}}

$struct\_node->set($data);

if($access->status==ACCESS\_ADMIN){

$struct\_node->set("cms\_class",($data["npublished"]==1)?"cms\_hidden":"");

$struct\_node\_admin = & $struct\_node->create("cms\_news\_admin");

$struct\_node\_admin->set\_file(cms\_files\_check("news.admin"));

$struct\_node\_admin->setFromBranch($struct\_node->id());}}}

return false; break;

endswitch;

return false; }else{ return true;}}

function entry\_remove($entryid=0){

global $sql;

$sql->send("INSERT INTO ".CMS\_PREFIX."\_garbage\_node\_news SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE nid=".$entryid."");

$sql->send("DELETE FROM ".CMS\_PREFIX."\_node\_news WHERE nid=".$entryid." ");

@rename(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.".$entryid.".html",CLIENT\_OBJECTS\_PATH.$this->id."/ntext.".$entryid.'.html.delete');

@rename(CLIENT\_OBJECTS\_PATH.$this->id."/npreview.".$entryid.".html",CLIENT\_OBJECTS\_PATH.$this->id."/npreview.".$entryid.'.html.delete');

foreach($this->ext\_array as $key => $ext){

@rename(CLIENT\_OBJECTS\_PATH.$this->id."/img/".$entryid."\_original.".$ext,CLIENT\_OBJECTS\_PATH.$this->id."/img/".$entryid."\_original.".$ext.".delete");

@rename(CLIENT\_OBJECTS\_PATH.$this->id."/img/".$entryid.".".$ext,CLIENT\_OBJECTS\_PATH.$this->id."/img/".$entryid.".".$ext.".delete");

@rename(CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$entryid.".".$ext,CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/".$entryid.".".$ext.".delete");

@rename(CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$entryid.".".$ext,CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/".$entryid.".".$ext.".delete"); }

$sql->send("DELETE FROM ".CMS\_PREFIX."\_indexes WHERE sid=".$this->id." AND entryid='".$entryid."'");

clear\_cash();}

function counter($is\_admin=false){

global $sql;

$count = 0;

$query = "SELECT count(\*) as cnt FROM ".CMS\_PREFIX."\_node\_news WHERE sid = ".$this->id.($is\_admin?"":" AND npublished=0");

$res = $sql->send($query);

if($data = $sql->get($res)) $count = $data["cnt"];

return $count;}

function search($with\_entries=false){

global $sql;

$strings = array();

$strings[0] = (parent::search());

if($with\_entries){

$query = "SELECT nid FROM ".CMS\_PREFIX."\_node\_news WHERE sid = ".$this->id." AND npublished=0";

$res = $sql->send($query);

while($data = $sql->get($res)){

$strings[$data["nid"]] = $this->entry\_search($data["nid"]);}}

return $strings;}

function entry\_search($entryid=0){

global $sql;

$entry\_string = "";

if($entryid\*1>0){

$query = "SELECT ntitle FROM ".CMS\_PREFIX."\_node\_news WHERE nid='".$entryid."' AND npublished=0";

$res = $sql->send($query);

if($data = $sql->get($res)){

$entry\_string = $data["ntitle"]." ".file2str(CLIENT\_OBJECTS\_PATH.$this->id."/npreview.".$entryid.".html")." ".file2str(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.".$entryid.".html");}}

return $entry\_string;}

function entry\_load($entryid=0){

global $sql;

$data = array();

if($entryid\*1>0){

$query = "SELECT ntitle as sname FROM ".CMS\_PREFIX."\_node\_news WHERE nid='".$entryid."' AND npublished=0";

$res = $sql->send($query);

if($data = $sql->get($res)){

$data["fpath"] = $this->get\_full\_path();

$data["entrypath"] = "nid=".$entryid."&amp;a=entry.show&amp;";}}

return $data;}

function load(){

global $sql,$CMS\_NODE\_TYPES,$current\_node\_extends\_vars;

parent::load();

$this->config->get($this->id);

$current\_node\_extends\_vars = cms\_node\_extended\_vars($this->id);

foreach ($current\_node\_extends\_vars as $key => $value) {

if (array\_key\_exists($key,$CMS\_NODE\_TYPES[$this->identificator]["\_config"])) {

$this->config->data[$key] = $value;}}}

function get\_meta($tag){

global $sql;

if(isset($\_GET["nid"]) && $\_GET["nid"]\*1>0){

$query = "SELECT ".$tag." as tag FROM ".CMS\_PREFIX."\_node\_news WHERE nid='".($\_GET["nid"]\*1)."' AND npublished=0";

$res = $sql->send($query);

if($data = $sql->get($res)){

$this->data[$tag] = $data["tag"];}}

if($this->data[$tag]!=""){

return $this->data[$tag]; }else{

if($this->id!=0){

return $this->parent->get\_meta($tag); }else{ return "";}}}

function set($data){

global $sql;

parent::set($data);

$this->config->set($this->id);

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id."/img/preview/",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id."/img/middle/",true);}

function remove(){

global $sql;

$query = "SELECT nid FROM ".CMS\_PREFIX."\_node\_news WHERE sid = ".$this->id." ";

$res = $sql->send($query);

while($data = $sql->get($res)){

$this->entry\_remove($data["nid"]);}

$sql->send("INSERT INTO ".CMS\_PREFIX."\_garbage\_node\_news SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE sid=".$this->id."");

$sql->send("DELETE FROM ".CMS\_PREFIX."\_node\_news WHERE sid=".$this->id."");

parent::remove();}

function get\_entry\_info(& $root,$entryid,$operation){

global $sql,$access;

$query1 = "SELECT ntitle as entryname, 'nid' as entrynameid, nid as entryid FROM ".CMS\_PREFIX.(strpos($operation,"remove")!==false?"\_garbage":"")."\_node\_news WHERE sid = '".($this->id)."' AND nid='".$entryid."'";

$res1 = $sql->send($query1);

while($data1 = $sql->get($res1)){

$cms\_entry\_more = & $root->create("cms\_entry\_more",$data1);

$cms\_entry\_more->set($data1);

$cms\_entry\_more->set($this->data);

if(strpos($operation,"remove")===false){

$cms\_entry\_more\_ref = & $cms\_entry\_more->create("cms\_entry\_more\_ref");

$cms\_entry\_more\_ref->set($data1);

$cms\_entry\_more\_ref->set($this->data);

$cms\_entry\_more\_ref->set("self",$access->self); }else{

$cms\_entry\_more->create("cms\_entry\_more\_noref");}}}}

function cms\_show\_news($shablone,$par=false,$ss=false,$ns=false,$nfield=false,$nsort=false,$nphoto=false,$nshowtemplate=false){

global $root,$node\_extends\_vars,$access,$sql;

$sids=array();

$nids=array();

if($ss!==false){

$sids = explode(",",$ss);

}elseif(isset($node\_extends\_vars["sids"])){

$sids = explode(",",$node\_extends\_vars["sids"]);}

if($ns!==false){

$nids = explode(",",$ns);

}elseif(isset($node\_extends\_vars["nids"])){

$nids = explode(",",$node\_extends\_vars["nids"]);}

if(is\_object($par)){

$parent = & $par; }else{

$parent = & $root;}

$shablons = explode("/",$shablone);

for($i=0;$i<count($shablons);$i++){

$tnode = & $parent->create($shablons[$i]);

$tnode->setFromBranch();

$parent = & $tnode;}

$query\_where = "";

if (is\_array($nids) && count($nids)>0){

$query\_where .= " nid IN ('".implode("','",$nids)."')";

}elseif(is\_array($sids) && count($sids)>0){

$query\_where .= " sid IN ('".implode("','",$sids)."')";}

if($query\_where!=""){

if($nshowtemplate!==false) $node\_extends\_vars["nshowtemplate"] = $nshowtemplate;

if($nphoto!==false) $node\_extends\_vars["nphoto"] = $nphoto;

if($nfield!==false) $node\_extends\_vars["nfield"] = $nfield;

if(!isset($node\_extends\_vars["nfield"])) $node\_extends\_vars["nfield"] = "ndate";

if($nsort!==false) $node\_extends\_vars["nsort"] = $nsort;

if(!isset($node\_extends\_vars["nsort"])) $node\_extends\_vars["nsort"] = 0;

$query = $query\_where." AND npublished=0 AND ".time().">=ndate AND ( nurgencydate=0 OR ".time()."<=nurgencydate) ";

$query .= "ORDER BY ".(isset($node\_extends\_vars["nfield"])?$node\_extends\_vars["nfield"]:"ndate")." ".($node\_extends\_vars["nsort"]?"ASC":"DESC");

$res = $sql->send("SELECT \* FROM ".CMS\_PREFIX."\_node\_news WHERE ".$query." LIMIT 0,1");

if($data = $sql->get($res)) $node\_id = $data["sid"];

if($node\_id>0){

$extend\_node = cms\_tree("getbyid",$node\_id);

$node\_extends\_vars["query"] = $query;

$extend\_node->handler($tnode,"entries.show",$node\_extends\_vars);}}

foreach($node\_extends\_vars as $k => $v) unset($node\_extends\_vars[$k]);}

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_node\_news(";

$query .= "nid int(10) unsigned NOT NULL auto\_increment,";

$query .= "sid int(11) NOT NULL default '0',";

$query .= "npublished tinyint(3) NOT NULL default '0',";

$query .= "ntitle varchar(255) NOT NULL default '',";

$query .= "ndate int(11) unsigned NOT NULL default '0',";

$query .= "nurgencydate INT NOT NULL default '0',";

$query .= "sdescription varchar(255) NOT NULL default '',";

$query .= "skeywords varchar(255) NOT NULL default '',";

$query .= "stitle varchar(255) NOT NULL default '',";

$query .= "PRIMARY KEY (nid),";

$query .= "KEY sid (sid),";

$query .= "KEY ndate (ndate),";

$query .= "KEY nurgencydate (nurgencydate)";

$query .= ") ENGINE=MyISAM ";

$CMS\_NODE\_SQL\_QUERY[] = $query;

?>

cmsUTF/data/classes/cms.node.php

<?

$CMS\_NODE\_TYPES[0] = array(

"\_class"=>"cms\_node",

"\_name"=>"языковая версия сайта",

"\_type"=>CMS\_TYPE\_LINK,

"\_development"=>true,

"\_permissions" => 1,

"\_pnames" => array(array("просмотр узла","show")),

"\_actions" => array("show"=>0),);

class cms\_node{ var $data;var $identificator = 0;var $virtual\_inself = false;var $editable = false;var $config = false;var $is\_loaded = false;var $has\_children = 0;var $children;var $parent;var $prev\_node=0;var $next\_node=0;var $permission\_special = false;var $id;var $type;var $access;var $level;

function cms\_node($data=array("sid"=>0,"pid"=>0,"stype"=>0,"saccess"=>0,"slevel"=>0,"stitle"=>"","sdescription"=>"","skeywords"=>"")){

global $access,$CMS\_NODE\_TYPES;

$this->data = $data;

if (!empty($data["sdate"])) $this->data["sdate"] = ($access->status!=ACCESS\_ADMIN)?int2date($data["sdate"],CLIENT\_DATE\_FORMAT):int2date($data["sdate"]);

$this->data["surgencydate"] = (!isset($data["surgencydate"]) || $data["surgencydate"]==0)?"":int2date($data["surgencydate"]);

if (!empty($data["sname"])) $this->data["sname"] = ($access->status!=ACCESS\_ADMIN)?str2html($data["sname"]):$data["sname"];

$this->id = & $this->data["sid"];

$this->type = & $this->data["stype"];

$this->access = & $this->data["saccess"];

$this->set\_parent();

if($this->parent) $this->level = $this->parent->level+1;

if($this->config) $this->config = new cms\_config($CMS\_NODE\_TYPES[$this->identificator]["\_config"]);

if (!empty($data["sname"])) if ($data["stitle"]=="") $this->data["stitle"] = $data["sname"];

if (!empty($data["sname"])) if ($data["smenu"]=="") $this->data["smenu"] = $data["sname"];

return $this;}

function check(){

global $CMS\_NODE\_TYPES;

if(is\_object($this->parent) && $this->parent->id!=0){

$this->access = $this->access | ($this->parent->access&CMS\_ACCESS\_HIDDEN);

$this->level = $this->parent->level+1;}

$this->type = ($this->type & CMS\_TYPE\_MASK) | $CMS\_NODE\_TYPES[$this->identificator]["\_type"];}

function get\_meta($tag){

global $access,$cms\_root\_node;

if($this->data[$tag]!=""){

return $this->data[$tag]; }else{

if($this->id!=0){

if($tag=='skeywords' || $tag=='sdescription'){

switch (CMS\_KW\_DSC\_EMPTY):

case 1:

return $this->parent->get\_meta($tag); break;

case 2:

return $cms\_root\_node->get\_meta($tag); break;

case 0: endswitch; }else{

return $this->parent->get\_meta($tag);

} }else{

return "";}}}

function set\_meta($data){

$data = str2save($data);

$this->data["stitle"] = $data["stitle"]=="" ? $data["sname"] : $data["stitle"];

$this->data["sdescription"]=$data["sdescription"];

$this->data["skeywords"]=$data["skeywords"];}

function set\_parent(){

if($this->id!=0){

$parent = & cms\_tree("getbyid",$this->data["pid"]);

if($parent->has\_children){

$prev\_node = & end($parent->children);

$prev\_node = & cms\_tree("getbyid",$prev\_node->id);

$this->prev\_node = $prev\_node->id;

$prev\_node->next\_node = $this->id;}

$parent->has\_children = (($this->type&CMS\_TYPE\_VIRTUAL)==0)?$parent->has\_children|1:$parent->has\_children|2;

$parent->children[$this->data["spath"]] = & $this;

$this->parent = & $parent;

$this->data["pid"] = $parent->id;

$this->data["fpath"] = $this->get\_full\_path();}

$this->check(); }

function move\_node(& $parent,& $path){

$path = preg\_replace("[^a-z0-9\_]","",strtolower($path));

if($path=="") $path = $this->id;

if(isset($parent->children[$path])) return false;

unset($this->parent->children[$this->data["spath"]]);

$this->parent->has\_children = (sizeof($this->parent->children)>0);

$this->data["spath"] = $path;

$this->data["pid"] = $parent->id;

$this->set\_parent();}

function get\_child($path){

if($this->has\_children){

if(sizeof($path)>=1){

if(isset($this->children[$path[0]])){

$cpath = array\_shift($path);

return $this->children[$cpath]->get\_child($path);}}}

return $this;}

function get\_full\_path(){

if($this->id!=0){

return $this->parent->get\_full\_path().(!empty($this->data["spath"])?$this->data["spath"]."/":""); }else{

return "";}}

function get\_root(){

if(($this->type&CMS\_TYPE\_ROOT)==CMS\_TYPE\_ROOT || $this->id==0){

return $this; }else{

return $this->parent->get\_root();}}

function handler(& $root, $event){

global $access,$cms\_alert,$sql,$template,$CMS\_NODE\_TYPES,$LOGINS\_ARRAY,$to\_log\_text,$to\_log;

if($event == "\_plogin"){

$\_SESSION["cms\_special"]["plogin"]= str2save($\_POST["plogin"],true);

$\_SESSION["cms\_special"]["ppass"] = str2save($\_POST["ppass"],true);

$cms\_alert->redirect($access->self.$this->get\_full\_path());}

if(!$this->is\_loaded) $this->load();

if(!cms\_permission($this,$event)){

if($event=="show" || $event==""){

if($this->permission\_special){

$\_SESSION["cms\_special"] = false;

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("node.login"));

$struct->set($this->data);

return true; }else{

$cms\_alert->set("er","node.forbid");

$cms\_alert->set\_text(file2str(CLIENT\_OBJECTS\_PATH.$this->id."/forbid.text.html"));} }else{

$cms\_alert->set("er","node.action.forbid");}

return true;}

switch($event):

case "copy":

$struct = & $root->create("cms\_node\_content");

$struct->set("sid",$this->data["sid"]);

$struct->set\_file(cms\_files\_check("node.copy"));

$struct->set("fpath",$this->data["fpath"]);

$struct->set("sdate",int2date(time()));

foreach($CMS\_NODE\_TYPES as $key => $types){

if(CMS\_DEVELOPMENT\_MODE || !$types["\_development"]) $struct->create("cms\_types",$types);}

$struct->setFromBranch();

$this->get\_nodes(0,"",$struct,true);

return true; break;

case "create":

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("node.create"));

$struct->set("sid",$this->data["sid"]);

$struct->set("fpath",$this->data["fpath"]);

$struct->set("sdate",int2date(time()));

foreach($CMS\_NODE\_TYPES as $key => $types){

if(CMS\_DEVELOPMENT\_MODE || !$types["\_development"]) {

if($types["\_type"]!=(10 << CMS\_TYPE\_BITS) || ($this->data["stype"] & (2 << CMS\_TYPE\_BITS))==(2 << CMS\_TYPE\_BITS)){

$struct->create("cms\_types",$types);}}}

$struct->setFromBranch();

return true; break;

case "help":

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("node.system"));

$struct->set($CMS\_NODE\_TYPES[$this->identificator]);

$struct->set($this->data);

$struct->setFromBranch();

return true; break;

case "edit":

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id."/images",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id."/images/preview",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id."/images/icons",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id."/files",true);

return false; break;

case "system":

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("node.system"));

$struct->set($CMS\_NODE\_TYPES[$this->identificator]);

$struct->set($this->data);

$struct->setFromBranch();

$this->get\_nodes(0,"",$struct);

return true; break;

case "\_up":

cms\_tree("change",$this->id,$this->prev\_node);

return true; break;

case "\_down":

cms\_tree("change",$this->id,$this->next\_node);

return true; break;

case "\_upTo":

cms\_tree("changeTo",$this->id,"up");

return true; break;

case "\_downTo":

cms\_tree("changeTo",$this->id,"down");

return true; break;

case "\_set":

$VARS = $\_POST;

if (empty($VARS["sname"]) && !empty($VARS["smenu"])) $VARS["sname"] = $VARS["smenu"];

if (empty($VARS["smenu"]) && !empty($VARS["sname"])) $VARS["smenu"] = $VARS["sname"];

if (empty($VARS["stitle"]) && !empty($VARS["sname"])) $VARS["stitle"] = $VARS["sname"];

cms\_tree("set",$this->id,$VARS);

return true; break;

case "\_remove":

cms\_tree("remove",$this->id);

return true;

break; case "\_copy":

cms\_tree("copy",$this->id,$\_POST);

return true; break;

case "\_create":

$VARS = $\_POST;

if (empty($VARS["sname"]) && !empty($VARS["smenu"])) $VARS["sname"] = $VARS["smenu"];

if (empty($VARS["smenu"]) && !empty($VARS["sname"])) $VARS["smenu"] = $VARS["sname"];

if (empty($VARS["stitle"]) && !empty($VARS["sname"])) $VARS["stitle"] = $VARS["sname"];

cms\_tree("create",$this->id,$VARS);

return true; break;

case "\_set\_preferences":

cms\_tree("\_set\_preferences",$this->id,$\_POST);

return true; break;

case "\_permission":

$this->set\_permission();

return true; break;

case "edit\_preferences":

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("preferences.edit"));

$struct->set($this->data);

$struct->setFromBranch();

$moderators = array();

$res = $sql->send("SELECT \* FROM ".CMS\_PREFIX."\_moderators");

while($data = $sql->get($res)){

$moderators[] = $data["login"].":".$data["passwd"];}

$struct->set("moderators",implode(";\n",$moderators));

return false; break;

case "show\_actions":

include CMS\_BASE."data/cms.log.php";

show\_logs("show\_actions");

return false; break;

case "show\_entry":

include CMS\_BASE."data/cms.log.php";

if(isset($\_GET["id"]) && $\_GET["id"]\*1!=0) show\_logs("show\_entry",$\_GET["id"]\*1);

return false; break;

case "permission":

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("node.permission"));

$struct->set($CMS\_NODE\_TYPES[$this->identificator]);

$struct->set($this->data);

$struct->set("forbid\_text",file2str(CLIENT\_OBJECTS\_PATH.$this->id."/forbid.text.html"));

$struct->setFromBranch();

$tinymce = & $struct->create("tinymce");

$tinymce->set\_file(cms\_files\_check("tinymce"));

$tinymce->setFromBranch();

$current\_type = $CMS\_NODE\_TYPES[$this->identificator];

$current\_psize = sizeof($current\_type["\_pnames"]);

for($i=0;$i<$current\_psize;$i++){

$actions\_name = & $struct->create("cms\_actions\_name");

$actions\_name->set(array("pname"=>$current\_type["\_pnames"][$i][0],"pico"=>$current\_type["\_pnames"][$i][1]));

$actions\_name->setFromBranch();}

$groups[] = array("sname"=>"Гости (Все)","sid"=>0);

$sql->send("SELECT sid,sname FROM ".CMS\_PREFIX."\_structure WHERE (stype - (stype&".CMS\_TYPE\_MASK."))=".(4 << CMS\_TYPE\_BITS)."");

while($data = $sql->get()) $groups[] = $data;

for($j=0;$j<sizeof($groups);$j++){

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_permission WHERE sid='".$this->id."' AND gid='". $groups[$j]["sid"]."'");

$groups[$j]["permission"] = ($data = $sql->get())?$data["permissions"]:$current\_type["\_permissions"];

$groups[$j]["psize"] = $current\_psize;

$group = & $struct->create("cms\_group",$groups[$j]);

$group->setFromBranch();

for($i=0;$i<$current\_psize;$i++){

$actions\_name = & $group->create("cms\_actions",array("pbit"=>$i));

$actions\_name->setFromBranch($group->id());}}

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_permission\_special WHERE sid='".$this->id."'");

if($data = $sql->get()){

$struct->set("pspecial",1);

$struct->set($data); }else{

$struct->set("pspecial",0);}

$moderators = array();

$res = $sql->send("SELECT \* FROM ".CMS\_PREFIX."\_moderators\_access WHERE sid='".$this->id."'");

while($data = $sql->get($res)){

$moderators[] = $data["login"];}

$struct->set("moderators",implode(",",$moderators));

return true; break;

endswitch;

return false;}

function search($condition=false){

$this->load();

return (!empty($this->data["sname"]) ? $this->data["sname"]." " : "");}

function load(){

$this->is\_loaded = true;}

function set($data){

$this->load();

if(isset($data["spath"])){

$this->move\_node($this->parent,$data["spath"]);}

if(isset($data["slink"])){

$node = & cms\_tree("getbyid",$data["slink"]\*1);

$this->data["slink"] = $node->id; }

$this->set\_meta($data);

if(isset($data["pid"])) $this->data["pid"] = $data["pid"]\*1;

if(isset($data["stype"])) $this->data["stype"] = $data["stype"]\*1;

if(isset($data["saccess"])) $this->data["saccess"]= $data["saccess"]\*1;

if(isset($data["ssort"])) $this->data["ssort"] = $data["ssort"]\*1;

if($this->data["ssort"]==0) $this->data["ssort"] = $this->id;

if(isset($data["surgencydate"])) $this->data["surgencydate"] = str2date($data["surgencydate"]);

if(isset($data["sdate"])) $this->data["sdate"] = $data["sdate"];

if(isset($data["sname"])) $this->data["sname"] = str2save($data["sname"]);

if(isset($data["smenu"])){

$this->data["smenu"] = str2save($data["smenu"]); }else{

$this->data["smenu"] = $this->data["sname"]; }

if(isset($data["sextend"])) $this->data["sextend"]= str2save($data["sextend"],true);

$this->check();

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id,true);}

function remove(){

if($this->has\_children){

foreach($this->children as $key => $node){

cms\_tree("remove",$node->id,true);}}

@rename(CLIENT\_OBJECTS\_PATH.$this->id,CLIENT\_OBJECTS\_PATH.$this->id.'.delete');}

function get\_nodes($current\_id,$pref,& $struct,$withself=false,$withredirect=false){

global $sql,$CMS\_NODE\_TYPES;

if($current\_id==0){

$struct->create("cms\_nodes",array("sid"=>0,"sname"=>"корень сайта","stype"=>"-"));

$pref = "...";}

$res = $sql->send("SELECT sid,sname,stype FROM ".CMS\_PREFIX."\_structure WHERE pid=".$current\_id." ORDER BY ssort");

while ($data=$sql->get($res)){

if($this->id==$data["sid"] && !$withself) continue;

if($data["stype"]==(15 << CMS\_TYPE\_BITS) && !$withredirect) continue;

$node = cms\_tree("getbyid",$data["sid"]);

$data["stype"] = $CMS\_NODE\_TYPES[$node->identificator]["\_name"];

$data["sname"] = $pref.$data["sname"];

$struct->create("cms\_nodes",$data);

$this->get\_nodes($data["sid"],$pref."...",$struct,$withself);}}

function get\_parents($current\_id,& $parents){

global $sql;

$res = $sql->send("SELECT pid FROM ".CMS\_PREFIX."\_structure WHERE id=".$current\_id."");

if ($data=$sql->get($res)){

if($data["pid"]>0){

$parents[] = $data["pid"];

$this->get\_parents($data["pid"], $parents);}}}

function set\_permission(){

global $sql,$access,$cms\_alert,$CMS\_NODE\_TYPES;

if(is\_array($\_POST["permission"])){

foreach($\_POST["permission"] as $gid => $value){

$sql->send("DELETE FROM ".CMS\_PREFIX."\_permission WHERE sid='".$this->id."' AND gid='".$gid."'");

if($CMS\_NODE\_TYPES[$this->identificator]["\_permissions"]!=$value)

$sql->send("INSERT INTO ".CMS\_PREFIX."\_permission SET sid='".$this->id."',gid='".$gid."',permissions='".$value."'");}

if(isset($\_POST["forbid\_text"])){

dir\_make(CLIENT\_OBJECTS\_PATH.$this->id,true);

str2file(CLIENT\_OBJECTS\_PATH.$this->id."/forbid.text.html",str2save($\_POST["forbid\_text"]));}

$cms\_alert->set("ok","node.permission");}

$data = str2save($\_POST,true);

$moderators\_ = explode(",",$data["moderators"]);

$moderators = array();

foreach($moderators\_ as $login){

$login = preg\_replace("/[^A-z0-9]/","",trim($login));

if(strlen($login)>=3){

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_moderators WHERE login='".$login."'");

if($sql->get()) $moderators[] = $login;}}

$this->set\_moderators($moderators, $this->id, (empty($data['moderators\_for\_childs']) ? 0 : 1));

$sql->send("DELETE FROM ".CMS\_PREFIX."\_permission\_special WHERE sid='".$this->id."'");

if($\_POST["pspecial"]==1){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_permission\_special SET sid='".$this->id."',plogin='".$data["plogin"]."',ppass='".$data["ppass"]."'");}

$cms\_alert->redirect($access->self.$this->get\_full\_path());}

function set\_moderators($moderators, $sid, $moderators\_for\_childs){

global $sql;

$sql->send("DELETE FROM ".CMS\_PREFIX."\_moderators\_access WHERE sid='".$sid."'");

foreach($moderators as $login){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_moderators\_access SET sid='".$sid."',login='".$login."'");}

if($moderators\_for\_childs==1) {

$res = $sql->send("SELECT sid FROM ".CMS\_PREFIX."\_structure WHERE pid='".$sid."'");

while ($data=$sql->get($res)){

$this->set\_moderators($moderators, $data['sid'], $moderators\_for\_childs);}}}}

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_logs (";

$query .= "id int(10) NOT NULL auto\_increment,";

$query .= "ip varchar(15) NOT NULL default '',";

$query .= "operation varchar(20) NOT NULL default '',";

$query .= "sid varchar(255) NOT NULL default '0',";

$query .= "timer timestamp NOT NULL,";

$query .= "PRIMARY KEY (id)";

$query .= ") ENGINE=MyISAM COMMENT='Таблица для логов'";

$CMS\_NODE\_SQL\_QUERY\_ADD[] = $query;

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_indexes (";

$query .= "iword varchar(32) NOT NULL default '',";

$query .= "sid int(11) unsigned NOT NULL default '0',";

$query .= "entryid int(10) NOT NULL default '0',";

$query .= "entrypref char(2) NOT NULL default '',";

$query .= "KEY iword (iword),";

$query .= "KEY sid (sid)";

$query .= ") ENGINE=MyISAM ";

$CMS\_NODE\_SQL\_QUERY\_ADD[] = $query;

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_search\_cash (";

$query .= " scid int(10) unsigned NOT NULL default '0',";

$query .= " iwords varchar(255) NOT NULL default '',";

$query .= " iword\_count varchar(255) NOT NULL default '0',";

$query .= " KEY scid (scid),";

$query .= " KEY iwords (iwords)";

$query .= ") ENGINE=MyISAM ";

$CMS\_NODE\_SQL\_QUERY\_ADD[] = $query;

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_structure (";

$query .= "sid int(10) unsigned NOT NULL auto\_increment,";

$query .= "pid int(10) unsigned NOT NULL default '0',";

$query .= "spath varchar(128) NOT NULL default '',";

$query .= "slink int(10) unsigned NOT NULL default '0',";

$query .= "stype int(10) unsigned NOT NULL default '2',";

$query .= "saccess int(10) unsigned NOT NULL default '0',";

$query .= "ssort int(10) NOT NULL default '0',";

$query .= "sdate int(10) unsigned NOT NULL default '0',";

$query .= "surgencydate INT(10) unsigned NOT NULL default '0',";

$query .= "sname varchar(255) NOT NULL default '',";

$query .= "sextend text default '',";

$query .= "sdescription varchar(255) NOT NULL default '',";

$query .= "skeywords varchar(255) NOT NULL default '',";

$query .= "stitle varchar(255) NOT NULL default '',";

$query .= "smenu varchar(255) NOT NULL default '',";

$query .= "PRIMARY KEY (sid),";

$query .= "KEY pid (pid),";

$query .= "UNIQUE KEY `pid\_2` (`pid`,`ssort`),";

$query .= "KEY spath (spath),";

$query .= "KEY stype (stype),";

$query .= "KEY ssort (ssort),";

$query .= "KEY saccess (saccess),";

$query .= "KEY surgencydate (surgencydate),";

$query .= "KEY sdate (sdate)";

$query .= ") ENGINE=MyISAM COMMENT='Таблица со структурой'";

$CMS\_NODE\_SQL\_QUERY[] = $query;

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_permission (";

$query .= "sid int(11) unsigned NOT NULL default '0',";

$query .= "gid int(11) unsigned NOT NULL default '0',";

$query .= "permissions int(11) NOT NULL default '0',";

$query .= "KEY sid (sid),";

$query .= "KEY permissions (permissions),";

$query .= "KEY gid (gid)";

$query .= ") ENGINE=MyISAM ";

$CMS\_NODE\_SQL\_QUERY[] = $query;

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_permission\_special(";

$query .= "sid int(11) unsigned NOT NULL default '0',";

$query .= "plogin varchar(128) NOT NULL,";

$query .= "ppass varchar(128) NOT NULL,";

$query .= "KEY sid (sid),";

$query .= "KEY plogin (plogin),";

$query .= "KEY ppass (ppass)";

$query .= ") ENGINE=MyISAM ";

$CMS\_NODE\_SQL\_QUERY\_ADD[] = $query;

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_moderators (";

$query .= "login varchar(64) NOT NULL default '',";

$query .= "passwd varchar(64) NOT NULL default ''";

$query .= ") ENGINE=MyISAM COMMENT='Таблица для модераторов';";

$CMS\_NODE\_SQL\_QUERY\_ADD[] = $query;

$query = "CREATE TABLE IF NOT EXISTS ".CMS\_PREFIX."\_moderators\_access (";

$query .= "sid int(10) NOT NULL default '0',";

$query .= "login varchar(64) NOT NULL default '',";

$query .= "KEY `sid` (`sid`),";

$query .= "KEY login (login)";

$query .= ") ENGINE=MyISAM COMMENT='Таблица для модераторов';";

$CMS\_NODE\_SQL\_QUERY\_ADD[] = $query;

?>

cmsUTF/data/classes/cms.node.text.php

<?

$CMS\_NODE\_TYPES[1] = array(

"\_class"=>"cms\_node\_text",

"\_name"=>"текстовый раздел",

"\_type"=>(1 << CMS\_TYPE\_BITS),

"\_development"=>false,

"\_permissions" => 1,

"\_pnames" => array(array("просмотр узла","show")),

"\_actions" => array("show"=>0),

"\_config" => array("nmult"=>1,"nchildssort"=>0,"nchildsshow"=>0,"nmethod"=>0,"norder"=>0,"nw"=>120,"nh"=>90,"nclass"=>"","nroot"=>"","nshowtemplate"=>""),

);

class cms\_node\_text extends cms\_node{var $identificator = 1;var $editable = true;var $config = true;var $ext\_array = array("gif","jpg","png");

function cms\_node\_text($data){

parent::cms\_node($data);

return $this;}

function handler(& $root, $event){

global $access,$template,$current\_node\_extends\_vars,$cms\_current\_node;

if(!parent::handler($root, $event)){

switch($event):

case "edit":

$this->data["ntext"] = str2html(file2str(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.html"), true);

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("text.edit"));

$struct->set($this->data);

$struct->set("client\_objects\_path",CLIENT\_OBJECTS\_PATH);

$struct->setFromBranch();

$tinymce = & $struct->create("tinymce");

$tinymce->set\_file(cms\_files\_check("tinymce"));

$tinymce->setFromBranch($struct->id());

if(isset($this->config->data["nmult"]) && isset($this->config->data["norder"]) && isset($this->config->data["nchildssort"]) && isset($this->config->data["nchildsshow"]) && isset($this->config->data["nmethod"]) && isset($this->config->data["nw"]) && isset($this->config->data["nh"])){

$cms\_node\_sort = & $struct->create("cms\_node\_sort");

$cms\_node\_sort->set($this->data);

$cms\_node\_sort->set($this->config->data);

$img = false;

foreach($this->ext\_array as $key => $ext){

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/npic.".$ext)) $img = CLIENT\_OBJECTS\_PATH.$this->id."/npic.".$ext;}

if($img){

$w=@getimagesize($img);

$data["img"]=$img;

$cms\_node\_content\_photo\_prev = & $cms\_node\_sort->create("cms\_node\_content\_photo\_prev",$data);

$cms\_node\_content\_photo\_prev->setFromBranch($struct->id());} }else{

$struct->create("cms\_node\_nosort");}

return false; break;

default:

if ($this->identificator==1 || !isset($\_GET["a"])){

$struct\_photo = false;

if($access->status!=ACCESS\_ADMIN && isset($this->config->data["nchildsshow"]) && $this->config->data["nchildsshow"]==1){

if(isset($this->config->data["norder"]) && $this->config->data["norder"]==1){

$struct\_photo = & $root->create("cms\_node\_content");

$struct = & $root->create("cms\_node\_content"); }else{

$struct = & $root->create("cms\_node\_content");

$struct\_photo = & $root->create("cms\_node\_content");} }else{

$struct = & $root->create("cms\_node\_content");}

$tmp = cms\_files\_check("text".($access->status!==ACCESS\_ADMIN && $this->config->data["nshowtemplate"]!=""?".".$this->config->data["nshowtemplate"]:"").".show");

if(!is\_file($tmp)) $tmp = cms\_files\_check("text.show");

$struct->set\_file($tmp);

$struct->set($current\_node\_extends\_vars);

$struct->setFromBranch();

$struct->set("fpath",$this->data["fpath"]);

$struct->set("parentpath",$this->parent->data["fpath"]);

$struct->set($this->data);

if(is\_object($struct\_photo)){

$tmp = cms\_files\_check("text.ext".($access->status!==ACCESS\_ADMIN && $this->config->data["nshowtemplate"]!=""?".".$this->config->data["nshowtemplate"]:"").".show");

if(!is\_file($tmp)) $tmp = cms\_files\_check("text.ext.show");

$struct\_photo->set\_file($tmp);

$struct\_photo->set($current\_node\_extends\_vars);

$struct\_photo->setFromBranch();

$struct\_photo->set("fpath",$this->data["fpath"]);

$struct\_photo->set("parentpath",$this->parent->data["fpath"]);

$struct\_photo->set($this->data);

$struct\_photo->set($this->config->data);

cms\_node\_tree($this,$struct\_photo,"imagemenu",0);}}

return false; break;

endswitch;

return false; }else{

return true;}}

function search($condition=false){

return (parent::search($condition)).$this->data["ntext"]." ";}

function more(){

global $access;

$data = array("nshort"=>str2html(file2str(CLIENT\_OBJECTS\_PATH.$this->id."/nshort.html")),"npic"=>CMS\_HTTP\_BASE."img/sp.gif");

$img = false;

foreach($this->ext\_array as $key => $ext){

if(is\_file("./".CLIENT\_OBJECTS\_PATH.$this->id."/npic.".$ext)) $img = $access->base.CLIENT\_OBJECTS\_PATH.$this->id."/npic.".$ext;}

if($img) $data["npic"] = $img;

return $data;}

function load(){

global $CMS\_LAST\_MODIFIED,$CMS\_NODE\_TYPES;

parent::load();

$this->config->get($this->id);

$current\_node\_extends\_vars = cms\_node\_extended\_vars($this->id);

foreach ($current\_node\_extends\_vars as $key => $value) {

if (array\_key\_exists($key,$CMS\_NODE\_TYPES[$this->identificator]["\_config"])) {

$this->config->data[$key] = $value;

$this->config->sdata[$key] = $value;}}

$this->data["nshort"] = str2html(file2str(CLIENT\_OBJECTS\_PATH.$this->id."/nshort.html"));

$this->data["ntext"] = str2html(file2str(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.html"));

if(is\_file(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.html")) $CMS\_LAST\_MODIFIED = max($CMS\_LAST\_MODIFIED,filemtime(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.html"));}

function set($data){

parent::set($data);

$this->config->set($this->id);

if(isset($data["nshort"])){

$this->data["nshort"] = str2save($data["nshort"]);

str2file(CLIENT\_OBJECTS\_PATH.$this->id."/nshort.html",$this->data["nshort"]);}

if(isset($data["ntext"])){

$this->data["ntext"] = str2save($data["ntext"]);

str2file(CLIENT\_OBJECTS\_PATH.$this->id."/ntext.html",$this->data["ntext"]);}

if (!empty($data["delete\_photo"]) && $data["delete\_photo"]==1) {

foreach($this->ext\_array as $key => $ext){

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/npic.".$ext); } }else{

$img\_name = file\_upload("nphotofile",CLIENT\_OBJECTS\_PATH.$this->id."/","npic");

if($img\_name!=""){

$src\_ext = get\_extension($img\_name);

foreach($this->ext\_array as $key => $ext){

if($ext!=$src\_ext){

@unlink(CLIENT\_OBJECTS\_PATH.$this->id."/npic.".$ext); }}

$dest\_ext = ($src\_ext=="gif" ? "png" : $src\_ext);

if($this->config->data["nmethod"]==0){

image\_resize($img\_name,$this->config->data["nw"],$this->config->data["nh"],$img\_name); }else{

image\_preview($img\_name,$this->config->data["nw"], $this->config->data["nh"],1, 1,$img\_name);}}}}

function remove(){

parent::remove();}}

?>

cmsUTF/data/cms.functions.php

<?

function set\_meta($root\_id){

global $sql,$root;

if(isset($root\_id)){

$query = "SELECT sdescription as description,skeywords as keywords,stitle as sitetitle FROM ".CMS\_PREFIX."\_structure WHERE sid='".($root\_id)."'";

$res = $sql->send($query);

if($data = $sql->get($res)) $root->set($data);}}

function is\_404(){

global $access,$cms\_current\_node,$CMS\_NODE\_TYPES;

$path\_orig = $access->path;

$cnt\_orig = count($access->path\_map);

$node\_orig = cms\_tree("getbypath",$access->path\_map);

$path\_orig1 = $node\_orig->get\_full\_path();

$path\_orig1 = (substr($path\_orig1,-1)=="/")?substr($path\_orig1,0,-1):$path\_orig1;

if($path\_orig != $path\_orig1){

$node\_orig = false;

$sid\_orig = false;

}else{

$sid\_orig = $node\_orig->id;}

$path\_show = $cms\_current\_node->get\_full\_path();

$path\_show = (substr($path\_show,-1)=="/")?substr($path\_show,0,-1):$path\_show;

$cnt\_show = count(explode('/',$path\_show));

$sid\_show = $cms\_current\_node->id;

if($path\_orig == $path\_show){

if(!empty($\_GET['a'])){

if(array\_key\_exists($\_GET['a'], $CMS\_NODE\_TYPES[$cms\_current\_node->identificator]['\_actions'])){

if(strpos($\_GET['a'], 'entry') !== false){

$ids = array();

foreach($\_GET as $k =>$v){

if(strpos($k, 'id') !== false) $ids[] = $k;}

if(!empty($ids) && count($ids) == 1){

$entry = $node\_orig->entry\_load($\_GET[$ids[0]]);

if(empty($entry)){

return true;} }else{} }else{} }else{

return true;}} }else{

if($sid\_orig == $sid\_show){

return true; }else{

if($path\_orig == ""){ }else{

if(!empty($node\_orig->data['slink'])){

if($node\_orig->data['slink'] == $sid\_show){ }else{} }else{

return true;}}}}

return false;}

function show\_error($error){

global $template;

switch ($error) :

case 404 :

header($\_SERVER['SERVER\_PROTOCOL'] .' 404 Not Found');

header("Status: 404 Not Found");

$template = new template(cms\_files\_check("404"));

return true; break;

default: break;

endswitch;

return false;}

function insert\_logs($sid=0,$action,$status,$user,$entryid=0){

global $sql,$to\_log,$to\_log\_text;

$id = 0;

if ($action=='cms.map' || $action=='cms.search') return $id;

$res = $sql->send("show columns from ".CMS\_PREFIX."\_logs like 'entryid'");

if($sql->num($res)==0){

$sql->send("ALTER TABLE ".CMS\_PREFIX."\_logs ADD `status` TINYINT NOT NULL DEFAULT '0',ADD `user` VARCHAR( 100 ) NOT NULL DEFAULT 'admin',ADD `entryid` INT NOT NULL DEFAULT '0'");}

$res = $sql->send("show columns from ".CMS\_PREFIX."\_logs like 'id'");

if($sql->num($res)==0){

$sql->send("ALTER TABLE ".CMS\_PREFIX."\_logs ADD `id` BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY FIRST");}

if(array\_search($action, $to\_log)!==false){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_logs SET ip='".$\_SERVER['REMOTE\_ADDR']."',`operation`='".$action."',sid='".$sid."',`status`='".$status."',`user`='".$user."',entryid='".$entryid."'");

$id = $sql->get\_id();}

return $id;}

function update\_logs($id,$sid,$entryid=0,$act=''){

global $sql;

$sql->send("UPDATE ".CMS\_PREFIX."\_logs SET sid='".$sid."',entryid='".$entryid."'".($act!=''?",operation='".$act."'":"")." WHERE id='".$id."'");}

function clear\_cash(){

global $sql;

$query = "SELECT scid FROM ".CMS\_PREFIX."\_search\_cash";

$res = $sql->send($query);

while($data = $sql->get($res)){

$sql->send("DROP TABLE ".CMS\_PREFIX."\_search\_cash\_".$data["scid"]);}

$sql->send("TRUNCATE TABLE ".CMS\_PREFIX."\_search\_cash");}

function set\_indexes($id=0,$search\_string,$full=false){

global $sql;

if(is\_array($search\_string)){

$search\_strings = $search\_string; }else{

$search\_strings = array("0"=>$search\_string);}

foreach($search\_strings as $entryid =>$entrystring){

$searchString = $entrystring." ";

if($search\_array = get\_search\_array($searchString)){

if($full) { echo "\t<script>document.getElementById('progress').innerHTML =' node #".$id." - entry #".$entryid." indexed';</script>";

flush();}

foreach($search\_array as $key =>$val){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_indexes SET iword='".$val."',sid=".$id.",entryid='".$entryid."'");}}}

return true;}

function cms\_user\_groups(){

global $sql,$access;

$user\_groups = array();

$query = "SELECT sid FROM ".CMS\_PREFIX."\_structure WHERE stype-(stype&".CMS\_TYPE\_MASK.")"."=".CMS\_TYPE\_GROUP." ";

if(($access->status!=ACCESS\_ADMIN)) $query .= "AND sdate<".TODAY." AND (surgencydate=0 OR surgencydate>".TODAY.") AND saccess&".CMS\_ACCESS\_HIDDEN."=0 ";

$res = $sql->send($query);

while($data = $sql->get($res)){

$user\_groups[] = $data['sid'];}

return $user\_groups;}

function cms\_level\_tree(& $nodes\_by\_id,& $nodes\_root,$\_pid,$\_level){

global $sql,$access,$CMS\_NODE\_TYPES,$CMS\_ACCESS\_GROUP,$cms\_nodes\_root;

$query = "SELECT \* FROM ".CMS\_PREFIX."\_structure WHERE pid='".$\_pid."' ";

if(($access->status!=ACCESS\_ADMIN)) $query .= "AND sdate<".TODAY." AND (surgencydate=0 OR surgencydate>".TODAY.") AND saccess&".CMS\_ACCESS\_HIDDEN."=0 ";

$query .= "ORDER BY ssort";

$res = $sql->send($query);

while($data = $sql->get($res)){

$class\_name = !empty($CMS\_NODE\_TYPES[$data["stype"]-($data["stype"]&CMS\_TYPE\_MASK)>>CMS\_TYPE\_BITS]) ? $CMS\_NODE\_TYPES[$data["stype"]-($data["stype"]&CMS\_TYPE\_MASK)>>CMS\_TYPE\_BITS]["\_class"] : 'error';

if(!class\_exists($class\_name)) $class\_name = "cms\_node";

if(isset($nodes\_by\_id[$data["pid"]])){

$node = @new $class\_name;

$node->id = $data["sid"];

$nodes\_by\_id[$data["sid"]] = new $class\_name($data);

if(($data["stype"]&CMS\_TYPE\_ROOT)==CMS\_TYPE\_ROOT){

$nodes\_root[] = & $nodes\_by\_id[$data["sid"]];

$cms\_nodes\_root[$data["sid"]] = $data["slink"];}}

cms\_level\_tree($nodes\_by\_id, $nodes\_root,$data["sid"],$\_level+1);}}

function cms\_tree($action,$param=0,$data=false){

global $sql,$access,$CMS\_NODE\_TYPES,$CMS\_ACCESS\_GROUP,$log\_id,$cms\_alert;

static $nodes\_by\_id,$nodes\_root;

switch($action):

case "load":

$nodes\_by\_id[0] = new cms\_node;

$nodes\_by\_id[0]->type = $nodes\_by\_id[0]->type | CMS\_TYPE\_LINK;

$nodes\_by\_id[0]->level = 0;

cms\_level\_tree($nodes\_by\_id,$nodes\_root,0,0);

$nodes\_by\_id[0]->data["fpath"] = "";

$cms\_current\_node = & cms\_tree("getbypath",$access->path\_map);

if (!isset($\_GET["a"])) $cms\_current\_node->load();

if($access->status!=ACCESS\_ADMIN){

if($cms\_current\_node->id==0 && is\_array($nodes\_root)){

$node = & current($nodes\_root);

$cms\_current\_node = & $node;}

$node = & $cms\_current\_node;

do{

$ids\_path[$node->id] = true;

$cms\_current\_node = & $node;

if(($node->type&CMS\_TYPE\_LINK)!=0){

if(isset($node->data["slink"])){

$node = & cms\_tree("getbyid",$node->data["slink"]);

$next = (!array\_key\_exists($node->id, $ids\_path) && $node->id); }else{

$next = false;} }else{

$next = false;}

}while($next);}

return $cms\_current\_node; break;

case "getroots": return $nodes\_root; break;

case "getbypath": return $nodes\_by\_id[0]->get\_child($param); break;

case "getbyid":

$param=$param\*1;

if(!isset($nodes\_by\_id[$param])) return $nodes\_by\_id[0];

return $nodes\_by\_id[$param]; break;

case "change":

$param=$param\*1;

$data = $data\*1;

if(!isset($nodes\_by\_id[$param]) || !isset($nodes\_by\_id[$data])) return NULL;

$parent = & $nodes\_by\_id[$param]->parent;

$sql->send("UPDATE ".CMS\_PREFIX."\_structure SET ssort=0 WHERE sid=".($param)."");

$sql->send("UPDATE ".CMS\_PREFIX."\_structure SET ssort=".($nodes\_by\_id[$param]->data["ssort"])." WHERE sid=".($data)."");

$sql->send("UPDATE ".CMS\_PREFIX."\_structure SET ssort=".($nodes\_by\_id[$data]->data["ssort"])." WHERE sid=".($param)."");

header("location:".$access->self.$parent->get\_full\_path()); exit(0);

case "changeTo":

$param=$param\*1;

if(!isset($nodes\_by\_id[$param]) || !($data=="up" || $data=="down")) return NULL;

$parent = & $nodes\_by\_id[$param]->parent;

$ssort = $nodes\_by\_id[$param]->data["ssort"];

$sql->send("ALTER TABLE ".CMS\_PREFIX."\_structure DROP INDEX pid\_2");

$res1 = $sql->send("SELECT sid,ssort FROM ".CMS\_PREFIX."\_structure WHERE pid=".($parent->id)." AND ssort".($data=="up"?"<":">").($nodes\_by\_id[$param]->data["ssort"])." ORDER BY ssort ".($data=="up"?"DESC":"ASC"));

while($data1=$sql->get($res1)){

$sql->send("UPDATE ".CMS\_PREFIX."\_structure SET ssort = ".$ssort." WHERE sid=".$data1["sid"]);

$ssort = $data1["ssort"];

}

$sql->send("UPDATE ".CMS\_PREFIX."\_structure SET ssort = ".$ssort." WHERE sid=".$param);

$sql->send("ALTER TABLE ".CMS\_PREFIX."\_structure ADD UNIQUE pid\_2 (pid,ssort)");

header("location:".$access->self.$parent->get\_full\_path()); exit(0);

case "indexes":

$search\_array = array();

clear\_cash();

if($param==-1){

for($ii=0;$ii<1024\*5;$ii++){

echo " ";}

echo "<html><body><P style='font:14px verdana;color:#000000;'><b>Переиндексация контента</b></P><div id='progress' style='width:300px;background:#eeeeee;text-align:center;font:10px/12px verdana;color:#000000;'></div></body></html>" ;

flush();

set\_time\_limit(300);

ignore\_user\_abort(true);

$sql->send("TRUNCATE TABLE ".CMS\_PREFIX."\_indexes");

foreach($nodes\_by\_id as $id => $object){

if(($object->access&CMS\_ACCESS\_HIDDEN)==0){

set\_indexes($id,$object->search(true),true);}}

echo "<script language='JavaScript'>alert('Переиндексация завершена');window.location.href = '".$access->self."';</script>";

exit(); }else{

$params = explode(":",$param);

$sql->send("DELETE FROM ".CMS\_PREFIX."\_indexes WHERE sid=".($params[0])." AND entryid=".(isset($params[1])?$params[1]:0)."");

if(!isset($nodes\_by\_id[$params[0]]) || ($nodes\_by\_id[$params[0]]->access&CMS\_ACCESS\_HIDDEN)!=0)

return false;

if(count($params)==1){

set\_indexes($params[0],$nodes\_by\_id[$params[0]]->search()); }else{

set\_indexes($params[0],array($params[1]=>$nodes\_by\_id[$params[0]]->entry\_search($params[1])));}}

return true; break;

case "set":

$param=$param\*1;

if(!isset($nodes\_by\_id[$param])) return NULL;

if(isset($data["spath"]) && strlen(trim($data["spath"]))>0) $data["spath"] = str2path(trim($data["spath"]));

$new\_date = false;

$change\_parent = isset($data["pid"]) && $nodes\_by\_id[$param]->data["pid"]!=$data["pid"] ? true : false;

$change\_spath = isset($data["spath"]) && $nodes\_by\_id[$param]->data["spath"]!=$data["spath"] ? true : false;

if($change\_parent || $change\_spath){

$suff="";

$\_pid = $change\_parent ? $data["pid"] : $nodes\_by\_id[$param]->data["pid"];

$\_spath = $data["spath"];

do{

$data["spath"] = $\_spath.$suff;

$suff=$suff\*1+1;

$res1 = $sql->send("SELECT sid FROM ".CMS\_PREFIX."\_structure WHERE pid='".$\_pid."' AND spath='".$data["spath"]."'");

}while($data1=$sql->get($res1));

if($change\_parent){

$new\_date = true;

$nodes\_by\_id[$param]->parent = & $nodes\_by\_id[$data["pid"]];}}

$old\_path = substr($nodes\_by\_id[$param]->get\_full\_path(),0,-1);

$old\_path\_cnt = count(explode("/", $old\_path));

if (!isset($data["sname"])) $data["sname"] = $nodes\_by\_id[$param]->data["sname"];

if (!isset($data["stitle"])){

$data["stitle"] = $nodes\_by\_id[$param]->data["stitle"];

$data["skeywords"] = $nodes\_by\_id[$param]->data["skeywords"];

$data["sdescription"] = $nodes\_by\_id[$param]->data["sdescription"];}

if (!isset($data["smenu"])) $data["smenu"] = $nodes\_by\_id[$param]->data["smenu"];

$nodes\_by\_id[$param]->set($data);

$udata = "";

$udata .= "spath ='".$nodes\_by\_id[$param]->data["spath"]."', ";

$udata .= "pid ='".$nodes\_by\_id[$param]->data["pid"]."', ";

$udata .= "smenu ='".$nodes\_by\_id[$param]->data["smenu"]."', ";

$udata .= "sname ='".$nodes\_by\_id[$param]->data["sname"]."', ";

$udata .= "stitle ='".$nodes\_by\_id[$param]->data["stitle"]."', ";

$udata .= "sdescription ='".$nodes\_by\_id[$param]->data["sdescription"]."', ";

$udata .= "skeywords ='".$nodes\_by\_id[$param]->data["skeywords"]."', ";

$udata .= "stype = ".$nodes\_by\_id[$param]->data["stype"].", ";

$udata .= "slink = ".$nodes\_by\_id[$param]->data["slink"].", ";

$udata .= "saccess= ".$nodes\_by\_id[$param]->data["saccess"].", ";

$udata .= "ssort = ".$nodes\_by\_id[$param]->data["ssort"].", ";

$udata .= "sdate = '".($new\_date?time():str2date($nodes\_by\_id[$param]->data["sdate"]))."', ";

$udata .= "surgencydate = '".str2date($nodes\_by\_id[$param]->data["surgencydate"])."', ";

$udata .= "sextend= '".$nodes\_by\_id[$param]->data["sextend"]."' ";

$sql->send("UPDATE ".CMS\_PREFIX."\_structure SET ".$udata." WHERE sid=".$param."");

cms\_tree("indexes",$param);

if(isset($data["parser"]) && $data["parser"]==1){

return; }else{

$new\_path = substr($nodes\_by\_id[$param]->get\_full\_path(),0,-1);

$new\_path\_cnt = count(explode("/", $new\_path));

if($change\_spath && $old\_path\_cnt == $new\_path\_cnt){

$res = $sql->send("SHOW TABLES FROM ".MYSQL\_DATABASE." like '%\_node\_announcements'");

if($sql->num($res)>0){

$sql->send("UPDATE ".CMS\_PREFIX."\_node\_announcements SET aurl=REPLACE(aurl,'/".$old\_path."/','/".$new\_path."/') WHERE aurl like './%'");

$sql->send("UPDATE ".CMS\_PREFIX."\_node\_announcements SET aurl=REPLACE(aurl,'/".$old\_path."','/".$new\_path."/') WHERE aurl like './%' AND aurl like '%/".$old\_path."'");}}

header("location:".$access->self.$nodes\_by\_id[$param]->get\_full\_path()); exit(0);} break;

case "\_set\_preferences":

$param=$param\*1;

if(!isset($nodes\_by\_id[$param])) return NULL;

$moderators = explode(";",str\_replace("\n","",str\_replace("\r","",$data["moderators"])));

$sql->send("DELETE FROM ".CMS\_PREFIX."\_moderators");

foreach($moderators as $value){

list($login,$passwd) = explode(":",$value);

$login = preg\_replace("/[^A-z0-9]/","",trim($login));

$passwd = preg\_replace("/[^A-z0-9]/","",trim($passwd));

if(strlen($login)>=3 && strlen($passwd)>=3)

$sql->send("INSERT INTO ".CMS\_PREFIX."\_moderators SET login='".$login."',passwd='".$passwd."'");}

unset($data["moderators"]);

header("location:".$access->self.$nodes\_by\_id[$param]->get\_full\_path()); exit(0); break;

case "copy":

$data["parser"] = 1;

if(isset($data["parser"])){

cms\_tree("load");

$parser = 1;

}else{

$parser = 0;}

$param=$param\*1;

if(!isset($nodes\_by\_id[$param])) return NULL;

$nodeId=$data["nodeId"]\*1;

if(!isset($nodes\_by\_id[$nodeId])) return NULL;

if(empty($data["with\_childs"])) $data["with\_childs"] = 0;

if(empty($data["with\_text"])) $data["with\_text"] = 0;

if(empty($data["with\_moderators"])) $data["with\_moderators"] = 0;

if(empty($data["with\_special"])) $data["with\_special"] = 0;

if(empty($data["with\_ann"])) $data["with\_ann"] = 0;

$sid = copyNode($nodes\_by\_id, $param,$nodeId,$parser,$data["sname"],$data["with\_childs"],$data["with\_text"],$data["with\_moderators"],$data["with\_special"],$data["with\_ann"]);

update\_logs($log\_id,$nodeId);

header("location:".$access->self.$nodes\_by\_id[$param]->get\_full\_path());

exit(0); break;

case "create":

$param=$param\*1;

if(!isset($nodes\_by\_id[$param])) return NULL;

$type\_redirect = $sql->send("SELECT \* FROM ".CMS\_PREFIX."\_structure WHERE stype= 15 << ".CMS\_TYPE\_BITS."");

if(($data["stype"]==(15 << CMS\_TYPE\_BITS)) && $sql->num($type\_redirect)>0){

$cms\_alert->set("er","redirect.exists");

$cms\_alert->redirect($access->self.$nodes\_by\_id[$param]->get\_full\_path());exit(0);

break;}

$sql->send("INSERT INTO ".CMS\_PREFIX."\_structure SET pid=".$param.",stype=".intval($data["stype"])."");

$sid = $sql->get\_id();

dir\_make(CLIENT\_OBJECTS\_PATH.$sid."/images/preview",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$sid."/images/icons",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$sid."/files",true);

update\_logs($log\_id,$sid);

$sql->send("SELECT login FROM ".CMS\_PREFIX."\_moderators\_access WHERE sid=".$param."");

if($logindata = $sql->get()){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_moderators\_access SET sid=".$sid.",login='".($logindata["login"])."'");}

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_structure WHERE sid=".$sid."");

if($ndata = $sql->get()){

$class\_name = $CMS\_NODE\_TYPES[$data["stype"]-($data["stype"]&CMS\_TYPE\_MASK)>>CMS\_TYPE\_BITS]["\_class"];

if(!class\_exists($class\_name)) $class\_name = "cms\_node";

$nodes\_by\_id[$ndata["sid"]] = new $class\_name($ndata);

cms\_tree("set",$ndata["sid"],$data);}

header("location:".$access->self.$nodes\_by\_id[$param]->get\_full\_path()); exit(0); break;

case "remove":

$param=$param\*1;

if(!isset($nodes\_by\_id[$param])) return NULL;

$nodes\_by\_id[$param]->remove();

if($param!=0){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_garbage\_structure SELECT \* FROM ".CMS\_PREFIX."\_structure WHERE sid=".$param."");

$sql->send("INSERT INTO ".CMS\_PREFIX."\_garbage\_permission SELECT \* FROM ".CMS\_PREFIX."\_permission WHERE sid=".$param."");

$sql->send("DELETE FROM ".CMS\_PREFIX."\_structure WHERE sid=".$param."");

$sql->send("DELETE FROM ".CMS\_PREFIX."\_permission WHERE sid=".$param."");

$sql->send("DELETE FROM ".CMS\_PREFIX."\_indexes WHERE sid=".$param."");

clear\_cash();

$parent = & $nodes\_by\_id[$param]->parent;

unset($nodes\_by\_id[$param]);

}else{

$parent = & $nodes\_by\_id[0];}

if(!$data){

header("location:".$access->self.$parent->get\_full\_path()); exit(0);} break;

endswitch;

return NULL;}

function cmp($a, $b) {

return strcmp(mb\_strtoupper($a->data["smenu"]), mb\_strtoupper($b->data["smenu"]));}

function cms\_node\_tree(& $parent\_node,& $root\_node,$shablone="mainmenu",$depth=0,$from\_node=0,$menu\_length=0){

global $access,$cms\_current\_node,$sql,$CMS\_ACCESS\_GROUP,$ADMIN\_LOGIN;

if($parent\_node && (($parent\_node->has\_children&1) && (!$CMS\_ACCESS\_GROUP || cms\_permission($parent\_node)) ) || (($parent\_node->has\_children&2) && $access->status==ACCESS\_ADMIN)){

if($access->status!=ACCESS\_ADMIN && $shablone=="node\_menu" && (isset($parent\_node->config->data["nchildsshow"]) && $parent\_node->config->data["nchildsshow"]==1)){

return;}

$menu\_node = & $root\_node->create("cms\_".$shablone,$parent\_node->data);

$full\_path = $cms\_current\_node->get\_full\_path();

if(strlen($full\_path)>0 && strlen($parent\_node->get\_full\_path())>0 && strpos($full\_path,$parent\_node->get\_full\_path())===0){

$menu\_node->set("node\_attributes","cms\_tree\_current");}

$menu\_node->setFromBranch();

$node\_number = 0;

$iii = 0;

$\_children = $parent\_node->children;

if(isset($parent\_node->config->data["nchildssort"]) && $parent\_node->config->data["nchildssort"]==1){

uasort ($\_children, "cmp");}

foreach($\_children as $url => $node){

if((($node->type&CMS\_TYPE\_VIRTUAL)==0 || ($access->status==ACCESS\_ADMIN && !$parent\_node->virtual\_inself && $from\_node+$menu\_length==0)) && ($node\_number>=$from\_node) && ($menu\_length==0 || $node\_number<$from\_node+$menu\_length)){

$data = $node->data;

$menu\_item\_node = & $menu\_node->create("cms\_".$shablone."\_item",$data);

$attributes = array();

if(++$iii==1)

$attributes[] = "cms\_tree\_first";

$menu\_item\_node->set("i",$iii);

if(strpos($cms\_current\_node->get\_full\_path(),$node->get\_full\_path())===0)

$attributes[] = "cms\_tree\_open";

if($cms\_current\_node->id==$node->id)

$attributes[] = "cms\_tree\_current";

if ((($node->has\_children&1) && (!$CMS\_ACCESS\_GROUP || cms\_permission($node)) ) || (($node->has\_children&2) && $access->status==ACCESS\_ADMIN))

$attributes[] = "cms\_has\_children";

if(isset($parent\_node->config->data["nmult"]) && $parent\_node->config->data["nmult"]>1){

$nmult = intval($this->config->data["nmult"]);

if(fmod($iii,$amultiplicity) == 0) $attributes[]="lastcolumn";}

if($access->status==ACCESS\_ADMIN){

if(($node->access&CMS\_ACCESS\_HIDDEN)==CMS\_ACCESS\_HIDDEN){ $attributes[]="cms\_hidden"; }

if(($node->type&CMS\_TYPE\_VIRTUAL)==CMS\_TYPE\_VIRTUAL){ $attributes[]="cms\_virtual";}

if(($node->type&CMS\_TYPE\_LINK)==CMS\_TYPE\_LINK){ $attributes[]="cms\_link";}

if(($node->type&CMS\_TYPE\_ROOT)==CMS\_TYPE\_ROOT){ $attributes[]="cms\_root";}

if(in\_array($access->user,$ADMIN\_LOGIN)){

cms\_node\_admin($node,$menu\_item\_node,"node\_admin"); }else{

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_moderators\_access WHERE login='".$access->user."' AND sid=".$node->id."");

if($sql->get()) $attributes[]="cms\_editable";} }else{

if(($node->type&CMS\_TYPE\_LINK)==CMS\_TYPE\_LINK && $data["slink"]>0){

$sl = cms\_tree("getbyid",$data["slink"]);

$menu\_item\_node->set("fpath",$sl->get\_full\_path());

}elseif($node->type==(8 << CMS\_TYPE\_BITS)){

$node->attribute($menu\_item\_node);}}

if(is\_array($attributes)) $menu\_item\_node->set("node\_attributes",implode(" ",$attributes));

$menu\_item\_node->set("orderid",$iii);

$menu\_item\_node->setFromBranch();

if($shablone=="node\_menu" && $node->type==(12 << CMS\_TYPE\_BITS)){

$node->config->get($node->id);

$menu\_node->set("with\_images","with\_images");

$node->show\_image($menu\_item\_node,"cms\_node\_menu\_item\_photo",1);}

if($shablone=="imagemenu"){

$menu\_item\_node->set($node->more());}

if(method\_exists($node, 'counter')){

$cnt = $node->counter($access->status==ACCESS\_ADMIN);

$menu\_item\_node->create("entries\_counter",array('cnt'=>$cnt));}

if($depth>0) cms\_node\_tree($node,$menu\_item\_node,$shablone,$depth-1);}

$node\_number++;}

if(isset($menu\_item\_node)){

$attributes[] = "cms\_tree\_last";

$menu\_item\_node->set("node\_attributes",implode(" ",$attributes));

$menu\_item\_node->set("orderid",$iii);}}}

function cms\_navigation($parent\_node,& $root\_node,$shablone="navigation",$reverse=true){

global $access;

$navigation\_node = & $root\_node->create("cms\_".$shablone);

$navigation\_node->set('base', $access->base);

$navigation\_node->set('self', $access->self);

$navigation\_node->reverse($reverse);

$func = "cms\_navigation\_".($shablone=='title' ? "title" : "other");

return $func($parent\_node,$navigation\_node,$shablone);}

function cms\_navigation\_other($node,& $navigation\_node,$shablone="navigation"){

global $access;

$iii = 0;

while($node->id>0 && $iii++<15){

if(($access->status==ACCESS\_ADMIN || ($node->type & CMS\_TYPE\_VIRTUAL)==0) && (!($node->parent->id>0 && ($node->parent->type&CMS\_TYPE\_ROOT)==CMS\_TYPE\_ROOT && $node->parent->data['slink']==$node->id && $shablone!="navigation"))){

$data\_ = $node->data;

$navigation\_item\_node = & $navigation\_node->create("cms\_".$shablone."\_item",$data\_);

if($iii==1 && !empty($\_GET["a"]) && $\_GET["a"]=="entry.show" && !empty($\_GET["nid"])){

$entry\_data = $node->entry\_load(intval($\_GET["nid"]));

$navigation\_end\_node = & $navigation\_node->create("cms\_".$shablone."\_end",$entry\_data);}

if($access->status==ACCESS\_ADMIN){

$attributes = "";

if(($node->access&CMS\_ACCESS\_HIDDEN)==CMS\_ACCESS\_HIDDEN) $attributes .= "cms\_hidden ";

if(($node->type&CMS\_TYPE\_VIRTUAL)==CMS\_TYPE\_VIRTUAL) $attributes .= "cms\_virtual ";

if(($node->type&CMS\_TYPE\_LINK)==CMS\_TYPE\_LINK) $attributes .= "cms\_link ";

if(($node->type&CMS\_TYPE\_ROOT)==CMS\_TYPE\_ROOT) $attributes .= "cms\_root ";

$navigation\_item\_node->set("node\_attributes",$attributes);}

$navigation\_item\_node->setFromBranch($navigation\_node->id());}

$node = $node->parent;}

return $navigation\_node;}

function cms\_navigation\_title($node,& $navigation\_node,$shablone="navigation"){

global $access,$cms\_current\_node,$cms\_nodes\_root;

$iii = 0;

$jjj = 0;

$is\_entry = false;

if(!empty($node->entry)){

if(CMS\_SHORT\_TITLE==0){

$navigation\_item\_node = & $navigation\_node->create("cms\_".$shablone."\_item");

$navigation\_item\_node->set("stitle",$node->entry);

$jjj++; }else{

$is\_entry = true;}}

while($node->id>0 && $iii++<15){

$show\_title = false;

if($access->status==ACCESS\_ADMIN){

$show\_title = true; }else{

if(($node->type & CMS\_TYPE\_ROOT) == CMS\_TYPE\_ROOT){

if($cms\_current\_node->id != $node->id && $cms\_current\_node->id != $node->data['slink']){

if((CMS\_SITE\_TITLE & 1)==1) $show\_title = true; }else{

if((CMS\_SITE\_TITLE & 2)==2) $show\_title = true; } }else{

if($cms\_current\_node->id == $node->id){

if(!in\_array($node->id, $cms\_nodes\_root)){

if((CMS\_PAGE\_TITLE & 1)==1) $show\_title = true; }else{

if((CMS\_PAGE\_TITLE & 2)==2) $show\_title = true; } }else{

if(CMS\_SHORT\_TITLE==0){

$show\_title = true;}}}}

if($show\_title){

$data\_ = $node->data;

if ($iii==1 && $is\_entry) $data\_["stitle"] = $node->entry;

$navigation\_item\_node = & $navigation\_node->create("cms\_".$shablone."\_item",$data\_);

$jjj++;

if ($jjj>1) $navigation\_item\_node->create("cms\_".$shablone."\_sep");

$navigation\_item\_node->setFromBranch($navigation\_node->id());}

$node = $node->parent;}

return $navigation\_node;}

function cms\_permission(& $node,$action=""){

global $CMS\_NODE\_TYPES,$access,$sql,$cms\_alert;

static $permission;

if($access->status==ACCESS\_ADMIN) return true;

if($action=="") $action = "show";

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_permission\_special WHERE sid='".$node->id."'");

if($data=$sql->get()){

if(!$\_SESSION["cms\_special"] || $data["plogin"]!=$\_SESSION["cms\_special"]["plogin"] || $data["ppass"]!=$\_SESSION["cms\_special"]["ppass"]){

$node->permission\_special = true;

if($\_SESSION["cms\_special"]){

$cms\_alert->set("er","group.login");}

return false;}}

if(!isset($permission[$node->id])){

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_permission WHERE sid='".$node->id."' AND gid='".$\_SESSION['cms\_gid']."'");

if($data = $sql->get())

$permission[$node->id] = $data["permissions"];

elseif(isset($CMS\_NODE\_TYPES[$node->identificator]["\_permissions"]))

$permission[$node->id] = $CMS\_NODE\_TYPES[$node->identificator]["\_permissions"];

else $permission[$node->id] = 0;}

if(isset($CMS\_NODE\_TYPES[$node->identificator]["\_actions"][$action])){

return ($permission[$node->id] & (1 << $CMS\_NODE\_TYPES[$node->identificator]["\_actions"][$action])); }else{

return false;}}

function cms\_show\_new($root,$shablone,$node\_id=0,$child=false,$with\_menu=false){

global $action;

global $node\_extends\_vars,$access;

if ($child==='false') $child=false;

if($access->status!=ACCESS\_ADMIN){

$need\_show = true;

if($child===false || $child\*1==0 || $child\*1==2 || $node\_id==0){

if($child!==false && $child\*1==0 && (!isset($action) || $action!="entry.show")) $need\_show = false;

if($child\*1==2 && $action!="") {

$need\_show = false;}

if($need\_show){

$parent = & $root;

$shablons = explode("/",$shablone);

for($i=0;$i<count($shablons);$i++){

$tnode = & $parent->create($shablons[$i]);

$tnode->setFromBranch();

$parent = & $tnode;}}

if($node\_id){

if($need\_show){

$extend\_node = cms\_tree("getbyid",$node\_id);

if(!isset($node\_extends\_vars["nshowtemplate"])) $node\_extends\_vars["nshowtemplate"] = '';

$extend\_node->load();

$extend\_node->config->data = array\_merge($extend\_node->config->data,$node\_extends\_vars);

$extend\_node->handler($tnode,"");

$tnode->set($extend\_node->data);

$tnode->set($extend\_node->config->data);

if($with\_menu!==false){

cms\_node\_tree($extend\_node,$parent,$with\_menu);} }else{

$root->set($node\_extends\_vars);}}

foreach($node\_extends\_vars as $k => $v) unset($node\_extends\_vars[$k]); }else{

$extend\_node = cms\_tree("getbyid",$node\_id);

if($extend\_node->has\_children>0){

$arr = array\_keys($extend\_node->children);

if($child==1){

$chld = $arr[0]; }else{

$chld = $arr[count($arr)-1]; }

cms\_show\_new($root,$shablone,$chld,false,$with\_menu);}}}}

function cms\_show($shablone,$node\_id=0,$child=false,$with\_menu=false){

global $action;

global $root,$node\_extends\_vars,$access;

if ($child==='false') $child=false;

if($access->status!=ACCESS\_ADMIN){

$need\_show = true;

if($child===false || $child\*1==0 || $child\*1==2 || $node\_id==0){

if($child!==false && $child\*1==0 && (!isset($action) || $action!="entry.show")) $need\_show = false;

if($child\*1==2 && $action!="") {

$need\_show = false;}

if($need\_show){

$parent = & $root;

$shablons = explode("/",$shablone);

for($i=0;$i<count($shablons);$i++){

$tnode = & $parent->create($shablons[$i]);

$tnode->setFromBranch();

$parent = & $tnode;}}

if($node\_id){

if($need\_show){

$extend\_node = cms\_tree("getbyid",$node\_id);

if(!isset($node\_extends\_vars["nshowtemplate"])) $node\_extends\_vars["nshowtemplate"] = '';

$extend\_node->load();

if($extend\_node->config) $extend\_node->config->data = array\_merge($extend\_node->config->data,$node\_extends\_vars);

$extend\_node->handler($tnode,"");

$tnode->set($extend\_node->data);

if($extend\_node->config) $tnode->set($extend\_node->config->data);

if($with\_menu!==false){

cms\_node\_tree($extend\_node,$parent,$with\_menu);} }else{

$root->set($node\_extends\_vars);}}

$node\_extends\_vars = array();

}else{

$extend\_node = cms\_tree("getbyid",$node\_id);

if($extend\_node->has\_children>0){

$arr = array\_keys($extend\_node->children);

if($child==1){

$chld = $arr[0]; }else{

$chld = $arr[count($arr)-1]; }

cms\_show($shablone,$chld,false,$with\_menu);}}}}

function cms\_show\_extended($node\_id=0,$clear=false){

global $node\_extends\_vars, $current\_node\_extends\_vars, $access;

$extend\_node = & cms\_tree("getbyid",$node\_id);

$extends = explode(";",str\_replace("\n","",str\_replace("\r","",(!empty($extend\_node->data["sextend"])?$extend\_node->data["sextend"]:""))));

if(is\_array($extends)){

if($clear){

unset($\_GET);

unset($\_POST); }

for($i=0;$i<sizeof($extends);$i++){

if(strpos($extends[$i],"=")){

list($key,$var) = explode("=",$extends[$i]);

$key = trim($key);

if(substr($key,0,5)=="this\_"){

$current\_node\_extends\_vars[str\_replace("this\_","",$key)] = trim($var); }else{

$node\_extends\_vars[$key] = trim($var);}

}elseif(strpos($extends[$i],":")){

if($access->status!=ACCESS\_ADMIN){

list($key,$vars) = explode(":",$extends[$i]);

$key = trim($key);

if(function\_exists($key)){

call\_user\_func\_array($key,explode(",",$vars));}}} }}

return $current\_node\_extends\_vars;}

function cms\_get\_node\_extended\_vars($node\_id=0){

$extends\_vars = array();

$extend\_node = & cms\_tree("getbyid",$node\_id);

$extend\_node->load();

$extends = explode(";",str\_replace("\n","",str\_replace("\r","",$extend\_node->data["sextend"])));

if(is\_array($extends)){

for($i=0;$i<sizeof($extends);$i++){

if(strpos($extends[$i],"=")){

list($key,$var) = explode("=",$extends[$i]);

$key = trim($key);

$extends\_vars[str\_replace("this\_","",$key)] = trim($var);}}}

return $extends\_vars; }

function cms\_node\_extended\_vars($node\_id=0){

global $current\_node\_extends\_vars;

$extend\_node = & cms\_tree("getbyid",$node\_id);

$extends = explode(";",str\_replace("\n","",str\_replace("\r","",$extend\_node->data["sextend"])));

if(is\_array($extends)){

for($i=0;$i<sizeof($extends);$i++){

if(strpos($extends[$i],"=")){

list($key,$var) = explode("=",$extends[$i]);

$key = trim($key);

if(substr($key,0,5)=="this\_") $current\_node\_extends\_vars[str\_replace("this\_","",$key)] = trim($var);}}}

return $current\_node\_extends\_vars; }

function cms\_root\_menu($shablone="roots"){

global $root,$access,$cms\_current\_node;

$nodes = cms\_tree("getroots");

if(is\_array($nodes)){

$roots\_node = & $root->create("cms\_".$shablone);

$roots\_node->setFromBranch();

foreach($nodes as $url => $node){

$roots\_item\_node = & $roots\_node->create("cms\_".$shablone."\_item",$node->data);

$attributes = "";

if(strpos($cms\_current\_node->get\_full\_path(),$node->get\_full\_path())===0) $attributes .= "cms\_tree\_open ";

if($access->status==ACCESS\_ADMIN){

if(($node->access&CMS\_ACCESS\_HIDDEN)==CMS\_ACCESS\_HIDDEN){ $attributes .= "cms\_hidden "; }

if(($node->type&CMS\_TYPE\_VIRTUAL)==CMS\_TYPE\_VIRTUAL){ $attributes .= "cms\_virtual ";}

if(($node->type&CMS\_TYPE\_LINK)==CMS\_TYPE\_LINK){ $attributes .= "cms\_link ";}

if(($node->type&CMS\_TYPE\_ROOT)==CMS\_TYPE\_ROOT){ $attributes .= "cms\_root ";}}

$roots\_item\_node->set("node\_attributes",$attributes);

$roots\_item\_node->setFromBranch();}}}

function cms\_pages(& $struct,$query,$atpage,$need\_pages=true){

global $sql, $cms\_current\_node;

$current = false;

if($atpage>0&&$need\_pages){

$sql->send($query);

if($data = $sql->get()){

$counter = $data["count"];

$pages = ceil($counter/$atpage);

$\_SESSION['pages\_'.($cms\_current\_node->id)] = $pages;

$current = (!empty($\_GET["p"]) && $\_GET["p"]<$pages)?$\_GET["p"]\*1:0;

if($pages>1 && $pages){

$pages\_node = & $struct->create("cms\_pages");

$pages\_node->set("current\_page",$current+1);

$pages\_node->set("page\_count",$pages);

$pages\_node->set("node\_count",$counter);

$pages\_node->setFromBranch($struct->id());

$sp = max($current-5,0);

$ep = min($current+5,$pages-1);

for($i=$sp;$i<=$ep;$i++){

$page\_node = & $pages\_node->create("cms\_page");

$page\_node->set("page",$i);

if($i==$current){

$page\_node->set("number",($i+1));

$page\_node->set("current","cms\_current");

}else if($i==$ep && $i!=$pages-1){

$page\_node->set("number","»");

$page\_node->set("class","right");

}elseif($i==$sp && $i!=0){

$page\_node->set("number","«");

$page\_node->set("class","left");

}else{ $page\_node->set("number",($i+1));}

$page\_node->setFromBranch($struct->id());}}}

return $current; }else{

return 0;}}

function cms\_node\_admin(& $node,& $root\_node,$shablone="node\_admin"){

global $CMS\_NODE\_TYPES;

$menu\_node = & $root\_node->create("cms\_".$shablone,$node->data);

$menu\_node->set("identificator",$node->identificator);

$menu\_node->set("sclass",$CMS\_NODE\_TYPES[$node->identificator]["\_name"]);

$menu\_node->set\_file(cms\_files\_check("node.admin"));

$menu\_node->setFromBranch();

if($node->parent && $node->parent->id > 0) $node->parent->load();

if($node->prev\_node!=0 && !(isset($node->parent->config->data["nchildssort"]) && $node->parent->config->data["nchildssort"]==1)){

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_".$shablone."\_up");

$menu\_node\_item->setFromBranch($menu\_node->id());

}else{

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());}

if($node->next\_node!=0 && !(isset($node->parent->config->data["nchildssort"]) && $node->parent->config->data["nchildssort"]==1)){

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_".$shablone."\_down");

$menu\_node\_item->setFromBranch($menu\_node->id()); }else{

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());}

if($node->editable ){

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_".$shablone."\_edit",$node->data);

$menu\_node\_item->setFromBranch();

}elseif(!$node->editable && $node->identificator==0 && $node->id>0){

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_".$shablone."\_edit\_preferences",$node->data);

$menu\_node\_item->setFromBranch(); }else{

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());}

if($node->id!=0){

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_".$shablone."\_system",$node->data);

$menu\_node\_item->setFromBranch($menu\_node->id());

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_".$shablone."\_remove",$node->data);

$menu\_node\_item->setFromBranch($menu\_node->id()); }else{

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());

$menu\_button = & $menu\_node->create("button");

$menu\_node\_item = & $menu\_button->create("cms\_node\_space");

$menu\_node\_item->setFromBranch($menu\_node->id());}}

function cms\_content\_check($nshowtemplate\_base,$nshowtemplate\_add=""){

global $access;

if(!empty($\_GET['nshowtemplate'])) $nshowtemplate\_add = $\_GET['nshowtemplate'];

$nshowtemplate = $nshowtemplate\_base.($nshowtemplate\_add!=""?".".$nshowtemplate\_add:"");

$temp = cms\_files\_check($nshowtemplate);

if($access->status!==ACCESS\_ADMIN){

if(!is\_file($temp)) $temp = cms\_files\_check($nshowtemplate\_base); }else{

if(!is\_file($temp)) echo "<script>alert('Ошибка: не найден шаблон ".$nshowtemplate."')</script>";

$temp = cms\_files\_check($nshowtemplate\_base);}

return $temp;}

function cms\_files\_check($file,$alternativefile=""){

global $cms\_root\_node,$cms\_current\_node,$access;

static $files;

if(!isset($files[$file])){

if($alternativefile!=""){

if(is\_file(CLIENT\_SHABLONS\_PATH.$cms\_current\_node->data["fpath"]."cms.".$alternativefile.".html")){

$files[$file] = CLIENT\_SHABLONS\_PATH.$cms\_current\_node->data["fpath"]."cms.".$alternativefile.".html";

}elseif(is\_file(CLIENT\_SHABLONS\_PATH.$cms\_root\_node->data["fpath"]."cms.".$alternativefile.".html")){

$files[$file] = CLIENT\_SHABLONS\_PATH.$cms\_root\_node->data["fpath"]."cms.".$alternativefile.".html";

}elseif(is\_file(CLIENT\_SHABLONS\_PATH."cms.".$alternativefile.".html")){

$files[$file] = CLIENT\_SHABLONS\_PATH."cms.".$alternativefile.".html";

}elseif(CMS\_DEVELOPMENT\_MODE && is\_file(CMS\_SHABLONS\_PATH."development/cms.".$alternativefile.".html")){

$files[$file] = CMS\_SHABLONS\_PATH."development/cms.".$alternativefile.".html";

}elseif(is\_file(CMS\_SHABLONS\_PATH."cms.".$alternativefile.".html")){

$files[$file] = CMS\_SHABLONS\_PATH."cms.".$alternativefile.".html";}}

if(!isset($files[$file])){

if($file=="root" && $access->status!=ACCESS\_ADMIN){

$fff = $file;

$fff1 = "";

if(isset($cms\_current\_node->config->sdata["nroot"]) && $cms\_current\_node->config->sdata["nroot"]!=""){

$fff1 = ".".$cms\_current\_node->config->sdata["nroot"];

}elseif(isset($cms\_current\_node->config->data["nroot"]) && $cms\_current\_node->config->data["nroot"]!=""){

$fff1 = ".".$cms\_current\_node->config->data["nroot"];}

$fff = $file.$fff1;

$\_fpath = array\_diff(explode('/',$cms\_current\_node->data["fpath"]), array(''));

if(count($\_fpath)>0){

$\_\_fpath = $\_fpath;

for($iii=count($\_fpath);$iii>=0;$iii--){

$fp = (count($\_\_fpath)>0)?implode('/',$\_\_fpath)."/":"";

if(is\_file(CLIENT\_SHABLONS\_PATH.$fp."cms.".$fff.".html")){

$files[$file] = CLIENT\_SHABLONS\_PATH.$fp."cms.".$fff.".html";

break; }else{

if($iii>0)array\_pop($\_\_fpath);}}}}

if(!isset($files[$file])){

if($cms\_root\_node && is\_file(CLIENT\_SHABLONS\_PATH.$cms\_root\_node->data["fpath"]."cms.".$file.".html")){

$files[$file] = CLIENT\_SHABLONS\_PATH.$cms\_root\_node->data["fpath"]."cms.".$file.".html";

}else if(is\_file(CLIENT\_SHABLONS\_PATH."cms.".$file.".html")){

$files[$file] = CLIENT\_SHABLONS\_PATH."cms.".$file.".html";

}else if(CMS\_DEVELOPMENT\_MODE && is\_file(CMS\_SHABLONS\_PATH."development/cms.".$file.".html")){

$files[$file] = CMS\_SHABLONS\_PATH."development/cms.".$file.".html"; }else{

$files[$file] = CMS\_SHABLONS\_PATH."cms.".$file.".html";}}}}

return $files[$file];}

function cms\_include\_alerts(){

global $cms\_root\_node,$CMS\_ALERTS,$to\_log\_text,$to\_log;

if(is\_file(CLIENT\_SHABLONS\_PATH.$cms\_root\_node->data["fpath"]."cms.alerts.php")){

include\_once(CLIENT\_SHABLONS\_PATH.$cms\_root\_node->data["fpath"]."cms.alerts.php");

}else if(is\_file(CLIENT\_SHABLONS\_PATH."cms.alerts.php")){

include\_once(CLIENT\_SHABLONS\_PATH."cms.alerts.php"); }else{

include\_once(CMS\_SHABLONS\_PATH."cms.alerts.php");}}

function copyNode(& $nodes\_by\_id, $param,$nodeId,$parser,$sname,$with\_childs,$with\_text,$with\_moderators,$with\_special,$with\_ann){

global $log\_id, $sql, $CMS\_NODE\_TYPES, $access, $ADMIN\_LOGIN;

$ext\_array = array("gif","jpg","png");

$res = $sql->send("SELECT \* FROM ".CMS\_PREFIX."\_structure WHERE sid=".$nodeId."");

if($copydata = $sql->get($res)){

unset($copydata["sid"]);

unset($copydata["ssort"]);

$copydata["pid"] = $param;

$copydata["parser"] = $parser;}

$sql->send("INSERT INTO ".CMS\_PREFIX."\_structure SET pid=".$param.",stype=".($copydata["stype"]\*1)."");

$sid = $sql->get\_id();

insert\_logs($sid,'\_create',(in\_array($access->user,$ADMIN\_LOGIN))\*1,$access->user);

dir\_make(CLIENT\_OBJECTS\_PATH.$sid."/images/preview",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$sid."/images/icons",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$sid."/files",true);

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/nconfig.html",CLIENT\_OBJECTS\_PATH.$sid."/nconfig.html");

if($with\_text==1) {

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/ntext.html",CLIENT\_OBJECTS\_PATH.$sid."/ntext.html");

foreach($ext\_array as $key => $ext){

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/npic.".$ext,CLIENT\_OBJECTS\_PATH.$sid."/npic.".$ext); }}

$copydata["spath"] = $sid;

if(!empty($sname)){

$copydata["sname"] = $sname;

$copydata["smenu"] = $sname;

$copydata["stitle"] = $sname;}

if($with\_moderators==1){

$moder = $nodeId; }else{

$moder = $param;}

$sql->send("SELECT login FROM ".CMS\_PREFIX."\_moderators\_access WHERE sid=".$moder."");

if($logindata = $sql->get()){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_moderators\_access SET sid=".$sid.",login='".($logindata["login"])."'");}

if($with\_special==1){

$sql->send("SELECT gid,permissions FROM ".CMS\_PREFIX."\_permission WHERE sid=".$nodeId."");

if($logindata = $sql->get()){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_permission SET sid=".$sid.",gid='".($logindata["gid"])."',permissions='".($logindata["permissions"])."'");}

$sql->send("SELECT plogin,ppass FROM ".CMS\_PREFIX."\_permission\_special WHERE sid=".$nodeId."");

if($logindata = $sql->get()){

$sql->send("INSERT INTO ".CMS\_PREFIX."\_permission\_special SET sid=".$sid.",plogin='".($logindata["plogin"])."',ppass='".($logindata["ppass"])."'");}

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/forbid.text.html",CLIENT\_OBJECTS\_PATH.$sid."/forbid.text.html");}

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_structure WHERE sid=".$sid."");

if($ndata = $sql->get()){

$class\_name = $CMS\_NODE\_TYPES[$copydata["stype"]-($copydata["stype"]&CMS\_TYPE\_MASK)>>CMS\_TYPE\_BITS]["\_class"];

if(!class\_exists($class\_name)) $class\_name = "cms\_node";

$nodes\_by\_id[$ndata["sid"]] = new $class\_name($ndata);

if($parser==1) $copydata["parser"] = 1;

cms\_tree("set",$sid,$copydata);

if($with\_ann==1){

$res = $sql->send("SELECT aid,apublished FROM ".CMS\_PREFIX."\_node\_announcements WHERE sid=".$nodeId."");

while($adata = $sql->get($res)){

$qqq = "INSERT INTO ".CMS\_PREFIX."\_node\_announcements (SELECT '','".$sid."',apublished,atitle,aurl,adate,aurgencydate FROM ".CMS\_PREFIX."\_node\_announcements WHERE aid='".$adata['aid']."')";

$res1 = $sql->send($qqq);

$aid = $sql->get\_id($res1);

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/apreview.".$adata["aid"].".html",CLIENT\_OBJECTS\_PATH.$sid."/apreview.".$aid.".html");

foreach($ext\_array as $key => $ext){

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/img/".$adata["aid"]."\_orig.".$ext,CLIENT\_OBJECTS\_PATH.$sid."/img/".$aid."\_orig.".$ext);

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/img/".$adata["aid"].".".$ext,CLIENT\_OBJECTS\_PATH.$sid."/img/".$aid.".".$ext);

@copy(CLIENT\_OBJECTS\_PATH.$nodeId."/img/preview/".$adata["aid"].".".$ext,CLIENT\_OBJECTS\_PATH.$sid."/img/preview/".$aid.".".$ext); }

insert\_logs($sid,'\_entry.add',(in\_array($access->user,$ADMIN\_LOGIN))\*1,$access->user,$aid);

if($adata["apublished"]\*1==0) cms\_tree("indexes",$sid.":".$aid); }}

if($with\_childs==1){

$res = $sql->send("SELECT sid, sname FROM ".CMS\_PREFIX."\_structure WHERE pid=".$nodeId."");

while($ndata = $sql->get($res)){

insert\_logs($ndata['sid'],'\_copy',(in\_array($access->user,$ADMIN\_LOGIN))\*1,$access->user);

copyNode($nodes\_by\_id, $sid,$ndata['sid'],$parser,$ndata['sname'],$with\_childs,$with\_text,$with\_moderators,$with\_special,$with\_ann);}}}

return $sid;}

?>

cmsUTF/data/cms.init.php

<?

@session\_start();

setlocale(LC\_ALL,'ru\_RU.UTF-8');

setlocale(LC\_NUMERIC,".");

ini\_set('date.timezone', 'Europe/Moscow');define("CMS\_TYPE\_BITS", 3);define("CMS\_TYPE\_MASK", 7);define("CMS\_TYPE\_VIRTUAL", 1); define("CMS\_TYPE\_LINK", 2);define("CMS\_TYPE\_ROOT", 4);define("CMS\_TYPE\_GROUP", 32);define("CMS\_ACCESS\_BITS", 3);define("CMS\_ACCESS\_MASK", 7);define("CMS\_ACCESS\_HIDDEN", 1);define("CMS\_ACCESS\_SPEСIAL",2);define("CMS\_VERSION","CMS Brane 1.3.9 (utf8)");define("CMS\_SYSTEM\_MAIL","postmaster@r52.ru");include CMS\_BASE."conf.php";define("TYNIMCE\_PATH","data/tinymce/");define("COLORBOX\_PATH","data/colorbox/");define("CMS\_CLASSES\_PATH",CMS\_BASE."data/classes/");define("CMS\_SHABLONS\_PATH",CMS\_BASE."data/shablons/");

if(!defined('CLIENT\_THEME')) define('CLIENT\_THEME','');

if(CLIENT\_THEME != ''){

define('CLIENT\_THEME\_PATH',''.CLIENT\_THEME.'/');

$client\_css = 'themes/'.CLIENT\_THEME.'/css/'; }else{

define('CLIENT\_THEME\_PATH','data/');

$client\_css = 'data/styles/';}

define("TODAY",time());

if(!defined('CMS\_ALERT\_AJAX')) define('CMS\_ALERT\_AJAX',0);

if(!defined('CMS\_REVERSE\_NAV')) define('CMS\_REVERSE\_NAV',1);

if(!defined('CMS\_REVERSE\_TITLE')) define('CMS\_REVERSE\_TITLE',0);

if(!defined('CMS\_404')) define('CMS\_404',0);

if(!defined('CMS\_SHORT\_TITLE')) define('CMS\_SHORT\_TITLE',1);

if(!defined('CMS\_SITE\_TITLE')) define('CMS\_SITE\_TITLE',3);

if(!defined('CMS\_PAGE\_TITLE')) define('CMS\_PAGE\_TITLE',1);

if(!defined('CMS\_KW\_DSC\_EMPTY')) define('CMS\_KW\_DSC\_EMPTY',0);

if(!defined('CMS\_NEED\_SLASH')) define('CMS\_NEED\_SLASH',1);

if(!defined('CMS\_NEED\_WWW')) define("CMS\_NEED\_WWW",0);

$cms\_nodes\_root = array();

$CMS\_LAST\_MODIFIED = 0;

$log\_id = 0;

$to\_log\_text = array();

$to\_log = array();

include CMS\_BASE."scripts/functions.php";

include CMS\_BASE."scripts/mysql.php";

include CMS\_BASE."scripts/struct.php";

include CMS\_BASE."scripts/template.php";

$sql= new sql(MYSQL\_HOSTNAME, MYSQL\_DBUSER, MYSQL\_DBPASS);

$sql->set\_db(MYSQL\_DATABASE);

$sql->set\_locale(MYSQL\_LOCAL);

$changeAccess = false;

$ADMIN\_LOGIN = array\_keys($LOGINS\_ARRAY);

$res = $sql->send("SELECT \* FROM ".CMS\_PREFIX."\_moderators");

while($data = $sql->get($res)){

$LOGINS\_ARRAY[$data["login"]] = $data["passwd"];}

include CMS\_BASE."scripts/access.php";

if($access->status==ACCESS\_USER && empty($\_POST)){

if(CMS\_NEED\_SLASH\*1==1 && isset($\_SERVER["PATH\_INFO"]) && substr($\_SERVER["PATH\_INFO"],-1)!='/'){

header("HTTP/1.1 301 Moved Permanently");

header('Location:'.$access->base.$access->path.($access->path!=""?'/':"").($\_SERVER["QUERY\_STRING"]!=""?'?'.$\_SERVER["QUERY\_STRING"]:""));

exit; }

$server = explode('.',$access->server\_name);

if(CMS\_NEED\_WWW\*1==0 && $server[0]=='www'){

array\_shift($server);

$server = implode('.',$server);

header("HTTP/1.1 301 Moved Permanently");

header('Location:'.$access->protocol.'://'.$server.$access->path.($access->path!="" && CMS\_NEED\_SLASH\*1==1?'/':"").($\_SERVER["QUERY\_STRING"]!=""?'?'.$\_SERVER["QUERY\_STRING"]:""));

exit;

}elseif(CMS\_NEED\_WWW\*1==1 && $server[0]!='www'){

array\_unshift($server,'www');

$server = implode('.',$server);

header("HTTP/1.1 301 Moved Permanently");

header('Location:'.$access->protocol.'://'.$server.$access->path.($access->path!="" && CMS\_NEED\_SLASH\*1==1?'/':"").($\_SERVER["QUERY\_STRING"]!=""?'?'.$\_SERVER["QUERY\_STRING"]:""));

exit; }}

include CMS\_BASE."data/cms.functions.php";include CMS\_CLASSES\_PATH."cms.alert.php";include CMS\_CLASSES\_PATH."cms.node.php"; include CMS\_CLASSES\_PATH."cms.config.php";

for($i=0;$i<sizeof($CMS\_INCLUDE\_CLASSES);$i++){

include CMS\_CLASSES\_PATH.$CMS\_INCLUDE\_CLASSES[$i].".php";

}

for($i=0;$i<sizeof($CLIENT\_INCLUDE\_CLASSES);$i++){

include CLIENT\_CLASSES\_PATH.$CLIENT\_INCLUDE\_CLASSES[$i].".php"; }

$node\_extends\_vars = array();

$current\_node\_extends\_vars = array();

if($access->status!=ACCESS\_USER){

$DIR\_ROOT = explode("/",$\_SERVER["SCRIPT\_FILENAME"]);

array\_pop($DIR\_ROOT);

$\_SESSION["path\_root"] = implode("/",$DIR\_ROOT)."/";

$HTTP\_ROOT = explode("/",$\_SERVER["SCRIPT\_NAME"]);

array\_pop($HTTP\_ROOT);

$\_SESSION["path\_http"] = "http://".$\_SERVER["SERVER\_NAME"].implode("/",$HTTP\_ROOT)."/";}

if(!isset($\_SESSION['cms\_user']) || !$\_SESSION['cms\_user']){

$\_SESSION['cms\_user'] = false;

$\_SESSION['cms\_gid'] = 0; }

$nav = false;

$structure = new structure();

$root = & $structure->create('root');

$CMS\_USER\_GROUPS = cms\_user\_groups();

$CMS\_ACCESS\_GROUP = !empty($CMS\_USER\_GROUPS);

$root->set("self",$access->self);

$root->set("base",$access->base);

$root->set("path",$access->path);

$root->set("cmsbase",CMS\_HTTP\_BASE);

$root->set("objbase",CLIENT\_OBJECTS\_PATH);

$root->set("client\_ip",$\_SERVER["REMOTE\_ADDR"]);

$root->set("time",time());

$root->set("currentnode","0");

$root->set("mclass","");

$root->set("navclass","");

$root->set("lang","");

$root->set("langpath","");

$root->set("query\_string","");

$root->set("colorboxpath",COLORBOX\_PATH);

$root->set("client\_theme",CLIENT\_THEME);

$root->set("client\_theme\_path",CLIENT\_THEME\_PATH);

$root->set("client\_css",$client\_css);

$root->set("tyniMCEpath",TYNIMCE\_PATH);

$root->set("current\_year",date("Y"));

$root->set("content\_title",'');

$lang = "";

$cms\_head = & $root->create("cms\_head\_".($access->status==ACCESS\_ADMIN?"admin":"user"));

$cms\_head->setFromBranch();

$CMS\_DEBUG = "DEBUG CONTROL \n";

$CMS\_DEBUG .= " version: ".CMS\_VERSION."\n";

$CMS\_DEBUG .= " client ip: ".$\_SERVER["REMOTE\_ADDR"]."\n";

time\_control($CMS\_DEBUG,true);

$cms\_alert = new cms\_alert();

$action = isset($\_GET["a"])?$\_GET["a"]:"";

switch($action){

case "cms.export":

if(is\_file(CMS\_BASE."data/cms.export.php")) {

include CMS\_BASE."data/cms.export.php"; }else{

header("location:".$access->self); exit(0);} break;

case "cms.initialize":

if($access->status==ACCESS\_ADMIN){

for($i=0;$i<sizeof($CMS\_NODE\_SQL\_QUERY\_ADD);$i++){

$sql->send($CMS\_NODE\_SQL\_QUERY\_ADD[$i]); }

for($i=0;$i<sizeof($CMS\_NODE\_SQL\_QUERY);$i++){

$sql->send($CMS\_NODE\_SQL\_QUERY[$i]);

$sql->send(str\_replace("IF NOT EXISTS ".CMS\_PREFIX,"IF NOT EXISTS ".CMS\_PREFIX."\_garbage",$CMS\_NODE\_SQL\_QUERY[$i])); }

dir\_make(CLIENT\_OBJECTS\_PATH,true);

dir\_make(CLIENT\_CLASSES\_PATH,true);

dir\_make(CLIENT\_SHABLONS\_PATH,true);

if(@linkinfo(dirname($\_SERVER['SCRIPT\_FILENAME'])."/".substr(TYNIMCE\_PATH,0,-1))\*1==-1)

@symlink(CMS\_BASE.TYNIMCE\_PATH,dirname($\_SERVER['SCRIPT\_FILENAME'])."/".substr(TYNIMCE\_PATH,0,-1));

if(@linkinfo(dirname($\_SERVER['SCRIPT\_FILENAME'])."/".substr(COLORBOX\_PATH,0,-1))\*1==-1)

@symlink(CMS\_BASE.COLORBOX\_PATH,dirname($\_SERVER['SCRIPT\_FILENAME'])."/".substr(COLORBOX\_PATH,0,-1));

$query = "SELECT \* FROM ".CMS\_PREFIX."\_structure ";

$res = $sql->send($query);

while($data = $sql->get($res)){

if($access->status==ACCESS\_ADMIN){

dir\_make(CLIENT\_OBJECTS\_PATH.$data["sid"]."/images",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$data["sid"]."/images/preview",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$data["sid"]."/images/icons",true);

dir\_make(CLIENT\_OBJECTS\_PATH.$data["sid"]."/files",true);}}

header("location:".$access->self);

exit(0);} break;

case "cms.indexes":

if($access->status==ACCESS\_ADMIN){

cms\_tree("load");

cms\_tree("indexes",-1);

header("location:".$access->self); exit(0);} break;

case "cms.map":

$cms\_current\_node = & cms\_tree("load");

$cms\_root\_node = & $cms\_current\_node->get\_root();

if($cms\_root\_node) {

$lang = $cms\_root\_node->data["spath"];

$root->set("lang",$cms\_root\_node->data["spath"]);

$root->set("langpath",$cms\_root\_node->data["fpath"]);

define("CLIENT\_SITE\_TITLE",$cms\_root\_node->data["stitle"]); }else{

define("CLIENT\_SITE\_TITLE","");}

set\_meta($cms\_root\_node->id);

$cms\_current\_node->config->sdata["nroot"] = "map";

$need\_template = true;

$template = new template(cms\_files\_check("root"));

cms\_include\_alerts();

cms\_node\_tree($cms\_root\_node,$root,"map",CLIENT\_MAP\_DEPTH);

$cms\_current\_node = new cms\_node;

$cms\_current\_node->data["sname"] = $cms\_alert->get("ok","cms.map");

$cms\_current\_node->data["stitle"] = $cms\_alert->get("ok","cms.map");

$cms\_current\_node->data["fpath"] = ($cms\_root\_node?$cms\_root\_node->data["fpath"]:"")."?a=cms.map";

$cms\_current\_node->id = 99999999999;

$cms\_current\_node->parent = & $cms\_root\_node;

$system\_node\_title = & $root->create("cms\_system\_node\_title");

$system\_node\_title->set($cms\_current\_node->data);

$system\_node\_title->setFromBranch();

$root->set('map\_active','active'); break;

case "cms.search":

$cms\_current\_node = & cms\_tree("load");

$cms\_root\_node = & $cms\_current\_node->get\_root();

if($cms\_root\_node) {

$lang = $cms\_root\_node->data["spath"];

$root->set("lang",$cms\_root\_node->data["spath"]);

$root->set("langpath",$cms\_root\_node->data["fpath"]);

define("CLIENT\_SITE\_TITLE",$cms\_root\_node->data["stitle"]); }else{

define("CLIENT\_SITE\_TITLE","");}

set\_meta($cms\_root\_node->id);

$need\_template = true;

$template = new template(cms\_files\_check("root"));

cms\_include\_alerts();

include CMS\_BASE."data/cms.search.php";

$cms\_current\_node = new cms\_node;

$cms\_current\_node->data["sname"] = $cms\_alert->get("ok","cms.search");

$cms\_current\_node->data["stitle"] = $cms\_alert->get("ok","cms.search");

$cms\_current\_node->id = 99999999999;

$cms\_current\_node->parent = & $cms\_root\_node;

$system\_node\_title = & $root->create("cms\_system\_node\_title");

$system\_node\_title->set($cms\_current\_node->data);

$system\_node\_title->setFromBranch(); break;

default:

$cms\_current\_node = & cms\_tree("load");

$error404 = false;

if($access->status!=ACCESS\_ADMIN && CMS\_404 == 1){

$error404 = is\_404();

if($error404){

header($\_SERVER['SERVER\_PROTOCOL'] .' 404 Not Found');

header("Status: 404 Not Found");}}

if($access->status==ACCESS\_USER){

$res = $sql->send("show tables like '".CMS\_PREFIX."\_node\_redirect'");

if($sql->num($res)>0){

$query = "SELECT \* FROM ".CMS\_PREFIX."\_node\_redirect WHERE roldurl='".($access->base.$access->path)."' OR roldurl='".($access->base.$access->path."/")."' LIMIT 0,1";

$res = $sql->send($query);

if($data = $sql->get($res)){

$new = cms\_tree("getbyid",$data["rsid"]);

$data["newurl"] = $new->get\_full\_path();

if($data["rslash"]\*1==0) $data["newurl"] = (substr($data["newurl"],-1)=="/")?substr($data["newurl"],0,strlen($data["newurl"])-1):$data["newurl"];

$data["newurl"] .= ($data["rget"]!=""?"?".trim($data["rget"]):"");

header($\_SERVER['SERVER\_PROTOCOL'] ." 301 Moved Permanently");

header('Location: '.$access->base.$data["newurl"]);

exit;}}}

$root->set("mclass",(CLIENT\_SHOW\_CHILDREN\*1==0 && ($cms\_current\_node->has\_children&1\*1)==0 && $access->status!=ACCESS\_ADMIN?"hidden":""));

$cms\_root\_node = & $cms\_current\_node->get\_root();

if($cms\_root\_node && $cms\_root\_node->id!=0) {

$lang = $cms\_root\_node->data["spath"];

$root->set("lang",$cms\_root\_node->data["spath"]);

$root->set("langpath",$cms\_root\_node->data["fpath"]);

define("CLIENT\_SITE\_TITLE",$cms\_root\_node->data["stitle"]); }else{

define("CLIENT\_SITE\_TITLE","");}

if(($access->path=="" || (!empty($cms\_root\_node->data['spath']) && $cms\_root\_node && $access->path==$cms\_root\_node->data['spath']))) $root->set('home\_active','active');

set\_meta($cms\_root\_node->id);

if($error404){

$template = new template(cms\_files\_check("404")); }else{

$need\_template = true;

$template = new template(cms\_files\_check("root",(isset($\_GET["alternativefile"]) && $\_GET["alternativefile"]!=""?$\_GET["alternativefile"]:"")));}

cms\_include\_alerts();

if(isset($\_GET["sid"])){

$cmsSearchId = $\_GET["sid"]\*1;

$cmsSearchTable = CMS\_PREFIX."\_search\_cash\_".$cmsSearchId;

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_search\_cash WHERE scid=".$cmsSearchId);

if($data=$sql->get()){

$cmsSearchString = $data["iwords"];

$cmsSearchCountString = explode(",",$data["iword\_count"]);

$cmsSearchArray = explode(" ",$cmsSearchString);

$root->set("search\_string",$cmsSearchString);

$root->set("search\_id",$cmsSearchId);

$cms\_node\_highlight = & $root->create("cms\_node\_highlight", $action);

$cms\_node\_highlight->set\_file(cms\_files\_check("node.highlight"));

$cms\_node\_highlight->setFromBranch();}}

if($access->status==ACCESS\_ADMIN && !in\_array($access->user,$ADMIN\_LOGIN)){

if($cms\_current\_node->id==0){

$sql->send("SELECT sid FROM ".CMS\_PREFIX."\_moderators\_access WHERE login='".$access->user."' AND sid!=0 ORDER BY sid LIMIT 1");

if($data = $sql->get()){

$cms\_redirect\_node = & cms\_tree("getbyid",$data["sid"]);

header("location:./admin/".$cms\_redirect\_node->get\_full\_path());}}

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_moderators\_access WHERE sid='".$cms\_current\_node->id."' AND login='".$access->user."'");

if(!$sql->get()){

$access->status=ACCESS\_USER;

$changeAccess = true; }else{

$MODERATOR = true;}}

if($access->status==ACCESS\_ADMIN && (in\_array($access->user,$ADMIN\_LOGIN) || $MODERATOR)){

$log\_id = insert\_logs($cms\_current\_node->id,$action,(in\_array($access->user,$ADMIN\_LOGIN))\*1,$access->user);}

$cms\_current\_node->handler($root, $action);

if($access->status==ACCESS\_ADMIN && (in\_array($access->user,$ADMIN\_LOGIN) || $MODERATOR)){

cms\_node\_admin($cms\_current\_node,$root,"node\_admin");}

cms\_show\_extended($cms\_current\_node->id,false); break;}

if($changeAccess){

$access->status=ACCESS\_ADMIN;}

if($CMS\_ACCESS\_GROUP){

$gr = cms\_tree("getbyid",$CMS\_USER\_GROUPS[0]);

$root->set("loginpath",$gr->get\_full\_path());}

$root->set("keywords",$cms\_current\_node->get\_meta("skeywords"));

$root->set("description",$cms\_current\_node->get\_meta("sdescription"));

if((!isset($root->sdata["content\_title"]) || $root->sdata["content\_title"]=="") && !empty($cms\_current\_node->data["sname"])) $root->set("content\_title",$cms\_current\_node->data["sname"]);

$root->set("currentnode",$cms\_current\_node->data["sid"]);

$root->set("navclass",($access->status!=ACCESS\_ADMIN && ($access->path=="" || $cms\_current\_node->data["stype"]==9)?"hidden":""));

$query\_string = "";

if(is\_array($\_GET) && count($\_GET)>0) foreach ($\_GET as $key => $value) $query\_string .= ($query\_string==""?"":"&").$key."=".$value;

$root->set("query\_string",($query\_string==""?"":"?".$query\_string));

cms\_root\_menu("roots");

cms\_navigation($cms\_current\_node,$root,"title",(CMS\_REVERSE\_TITLE\*1==1));

if($access->status==ACCESS\_ADMIN || !defined('CLIENT\_SHOW\_NAVIGATION') || CLIENT\_SHOW\_NAVIGATION\*1==1) $nav = & cms\_navigation($cms\_current\_node,$root,"navigation",(CMS\_REVERSE\_NAV\*1==1));

cms\_node\_tree($cms\_root\_node,$root,"mainmenu",CLIENT\_MENU\_DEPTH);

if(($cms\_current\_node->level>$cms\_root\_node->level+CLIENT\_MENU\_DEPTH && empty($node\_extends\_vars["cms\_hide\_node\_menu"])) || ($access->status==ACCESS\_ADMIN && $action!="create")){

$menu\_node = & $cms\_current\_node;

if($access->status!=ACCESS\_ADMIN && ($menu\_node->has\_children&1)==0 && $menu\_node->parent->level>$cms\_root\_node->level+CLIENT\_MENU\_DEPTH && ($menu\_node->parent->type&CMS\_TYPE\_VIRTUAL)==0 ){

$menu\_node = & $cms\_current\_node->parent;}

cms\_node\_tree($menu\_node,$root,"node\_menu");}

$cms\_alert->show($root);

if($need\_template) $template = new template(cms\_files\_check("root",(isset($\_GET["alternativefile"]) && $\_GET["alternativefile"]!=""?$\_GET["alternativefile"]:"")));

?>

cmsUTF/data/cms.js

function dj\_alert(title,message){

$.gritter.add({

title: title,

text: message,});}

function cmsCheckRoot(val){ var e0 = $('#stype\_0');var el = $('#stype\_1');var e2 = $('#stype\_2');var o0 = document.getElementById('stype\_0');var o1 = document.getElementById('stype\_1');var o2 = document.getElementById('stype\_2');

e0.attr('checked', val==2);

el.attr('checked', val==2);

e2.attr('checked', val==2);

cmsChangeCheckbox(o0);

cmsChangeCheckbox(o1);

cmsChangeCheckbox(o2);}

function cmsSetHiddenBlockFor(checkboxName,blockName){

var oCheckbox,oBlock;

if((oBlock=document.getElementById(blockName)) &&

(oCheckbox=document.getElementById(checkboxName))){

oCheckbox.blockName = blockName;}}

function cmsShowHiddenBlock(oCheckbox){

var oBlock=document.getElementById(oCheckbox.blockName);

if(oCheckbox.blockName && oBlock){

oBlock.style.display = (oCheckbox.checked)?"block":"none";}}

function cmsSetBitCheckboxesFor(inputName,checkboxesBitSize){

var oInput,oBitCheckbox;

if(oInput=document.getElementById(inputName)){

oInput.checkboxesBitSize = checkboxesBitSize;

var value = oInput.value\*1;

for(var i=0; i<checkboxesBitSize;i++){

if(oBitCheckbox=document.getElementById(inputName+"\_"+i)){

oBitCheckbox.checked = ((value&(1<<i))>0);

cmsChangeCheckbox(oBitCheckbox);}}}}

function cmsSetSelectValue(selectName,selectValue){

var oSelect;

var value = (selectValue)?selectValue:false;

if(oSelect=document.getElementById(selectName)){

if(value){

if(oSelect.options){

for(var i=0; i<oSelect.options.length;i++){

if(oSelect.options[i].value == value){

oSelect.selectedIndex = i;}} }else{

alert("Ошибка: пустой селектор "+selectName);}}

cmsChangeSelect(oSelect);}}

function cmsChangeCheckbox(oCheckbox){

var oInput;

var names = oCheckbox.id.split("\_");

if(oInput=document.getElementById(names[0])){

var value = oInput.value\*1;

oInput.value = (oCheckbox.checked)?value|(1<<names[1]):value-(value&(1<<names[1]));}

cmsShowHiddenBlock(oCheckbox);}

function cmsChangeSelect(oSelect){

var oInput;var mask = 0;var names = oSelect.id.split("\_");

if((names.length)>1 && (oInput=document.getElementById(names[0]))){

for(var i=0;i<oInput.checkboxesBitSize;i++) mask += 1 << i;

oInput.value = (oInput.value & mask) | oSelect.value;

var value = oSelect.value & mask;

for(var i=0; i<oInput.checkboxesBitSize;i++){

if(oBitCheckbox=document.getElementById(names[0]+"\_"+i)){

oBitCheckbox.disabled = ((value&(1<<i))>0);}}

cmsSetBitCheckboxesFor(names[0],oInput.checkboxesBitSize);}}

function cmsCheckInputValue(oInputName,type){

var oInput = $('#'+oInputName);

var val\_ = oInput.val();

oInput.removeClass('cms\_errorfield');

switch(type){

case 0:

if(val\_ != val\_\*1 || !(val\_>=0)){

if($.gritter){

dj\_alert("Ошибка заполнения поля","Требуется положительное целое число"); }else{

alert("Infillerror: positiveintegeronly\nОшибка: Требуется положительное целое число");}

oInput.val(0);

oInput.focus();

oInput.addClass('cms\_errorfield');

return false; }else{

oInput.val(Math.abs(Math.round(val\_)));

return true;} break;

case 1:

var re\_mail = /([\w\.\-\_]+@[\w\.\-\_]+)/;

if(val\_.match(re\_mail)!=null){

return true; }else{

if($.gritter){

dj\_alert("Ошибка заполнения поля","Требуется e-mail"); }else{

alert("Infill error: email address needed\nОшибка заполнения поля:Требуется e-mail");

}

oInput.focus();

oInput.addClass('cms\_errorfield');

return false;} break;

case 2:

if(val\_.length>=1){

return true; }else{

if($.gritter){

dj\_alert("Ошибка заполнения поля","Требуется текст"); }else{

alert("Infillerror: textneeded\nОшибка заполнения поля: Требуется текст");

}

oInput.addClass('cms\_errorfield');

return false;} break;

case 6:

case 7:

case 8:

case 9:

if(val\_.length>=type){

return true; }else{

if($.gritter){

dj\_alert("Ошибка заполнения поля","Требуется не менее "+type+" символов"); }else{

alert("Infillerror: textneeded\nОшибка заполнения поля: Требуется не менее "+type+" символов");

}

oInput.addClass('cms\_errorfield');

return false;} break;

case -1:

default:

if(val\_.length>=3){

return true; }else{

if($.gritter){

dj\_alert("Ошибка заполнения поля","Требуется текст"); }else{

alert("Infillerror: textneeded\nОшибка заполнения поля: Требуется текст"); }

oInput.focus();

oInput.addClass('cms\_errorfield');

return false;} break;}}

var cmsPrintObject = null;

var cmsHideObject = null;

function cmsPrintVersion(){

if(cmsPrintObject==null){

cmsPrintObject = document.createElement("div");

cmsPrintObject.addClass("cms\_printversion");

cmsPrintObject.appendChild(document.getElementById("cms\_navigation").cloneNode(1));

cmsPrintObject.appendChild(document.getElementById("cms\_nodecontent").cloneNode(1));

cmsHideObject = document.createElement("div");

cmsHideObject.innerHTML = document.body.innerHTML;

document.body.innerHTML = "";

cmsHideObject.style.display = "none";

document.body.appendChild(cmsPrintObject);

document.body.appendChild(cmsHideObject); }else{

document.body.innerHTML = cmsHideObject.innerHTML;

cmsPrintObject = null;

cmsHideObject = null;}}

function searchInNode(node, word, word\_length){

var word\_position, skip, spannode, middlebit, endbit, middleclone;

skip=0;

word = word.toLowerCase();

if(node.nodeType==3){

var text = $('#cms\_nodecontent').html();

var text = text.replace(new RegExp('('+word+')', 'gi'), "<span class='cms\_illuminate' >$1</span>");

$('#cms\_nodecontent').html(text);

skip=1;

}else if(node.nodeType==1&& node.childNodes && node.tagName.toUpperCase()!="SCRIPT" && node.tagName.toUpperCase!="STYLE"){

for (var child=0; child < node.childNodes.length; ++child){

child=child+searchInNode(node.childNodes[child], word, word\_length);}}

return skip;}

var cmsFormAddedFields = new Array;

function cmsFormNewField(num){

var oFormCanvas =document.getElementById("cms\_formcanvas");

var oFormField =document.getElementById("cms\_formfield");

var oNumber = cmsFormAddedFields.length;

var tempElements;

if(num>=0) oNumber = num;

else cmsFormAddedFields[oNumber] = new Array(0,"заголовок","");

var oNewField = oFormField.cloneNode(true);

oNewField.removeAttribute("id");

oNewField.style.display = "block";

oFormCanvas.appendChild(oNewField);

tempElements = oNewField.getElementsByTagName("select");

if(tempElements.length>0){

tempElements[0].setAttribute("name","felements["+oNumber+"][ftype]");

for(var i=0;i<tempElements[0].options.length;i++){

if(tempElements[0].options[i].value==cmsFormAddedFields[oNumber][0]) tempElements[0].selectedIndex = i;}}

tempElements = oNewField.getElementsByTagName("input");

if(tempElements.length>0){

tempElements[0].setAttribute("name","felements["+oNumber+"][fname]");

tempElements[0].value = cmsFormAddedFields[oNumber][1];}

tempElements = oNewField.getElementsByTagName("textarea");

if(tempElements.length>0){

tempElements[0].setAttribute("name","felements["+oNumber+"][fdefault]");

tempElements[0].value = cmsFormAddedFields[oNumber][2];}}

function cmsFormUpdate(){

for(var i=0;i<cmsFormAddedFields.length;i++) cmsFormNewField(i);}

function cmsFormClear(){

var oFormCanvas =document.getElementById("cms\_formcanvas");

oFormCanvas.innerHTML = "";

cmsFormAddedFields = new Array;}

function cmsFormClearLast(){

var \_cmsFormAddedFields = cmsFormAddedFields;

cmsFormClear();

for(var i=0;i<(\_cmsFormAddedFields.length-1);i++){

cmsFormAddedFields[i] = \_cmsFormAddedFields[i];

cmsFormNewField(i);}}

var cmsPollAddedFields = new Array;

function cmsPollNewField(num){

var oFormCanvas =document.getElementById("cms\_pollcanvas");

var oFormField =document.getElementById("cms\_pollfield");

var oNumber = cmsPollAddedFields.length;

var tempElements;

if(num>=0) oNumber = num;

else cmsPollAddedFields[oNumber] = new Array("вариант","0");

var oNewField = oFormField.cloneNode(true);

oNewField.removeAttribute("id");

oNewField.style.display = "block";

oFormCanvas.appendChild(oNewField);

tempElements = oNewField.getElementsByTagName("input");

tempElements[0].setAttribute("name","felements["+oNumber+"][fname]");

tempElements[0].value = cmsPollAddedFields[oNumber][0];

tempElements[1].setAttribute("name","felements["+oNumber+"][fvalue]");

tempElements[1].value = cmsPollAddedFields[oNumber][1];}

function cmsPollUpdate(){

for(var i=0;i<cmsPollAddedFields.length;i++) cmsPollNewField(i);}

function cmsPollClear(){

var oFormCanvas =document.getElementById("cms\_pollcanvas");

oFormCanvas.innerHTML = "";

cmsPollAddedFields = new Array;}

function cmsPollClearLast(){

var \_cmsPollAddedFields = cmsPollAddedFields;

cmsPollClear();

for(var i=0;i<(\_cmsPollAddedFields.length-1);i++){

cmsPollAddedFields[i] = \_cmsPollAddedFields[i];

cmsPollNewField(i);}}

var cmsPollSummary = 0;

function cmsPollResult(num,value){

var oPollCanvas =document.getElementById("cms\_pollresult\_"+num);

var oPollField =document.getElementById("cms\_pollresult");

var oNumber = cmsPollAddedFields.length;

var tempElements;

if(oPollCanvas && oPollField){

var oNewField = oPollField.cloneNode(true);

oNewField.style.display = "inline";

oNewField.removeAttribute("id");

value = (cmsPollSummary>0)?value/cmsPollSummary:0.01;

oNewField.width = Math.ceil(oNewField.width \* value);

oPollCanvas.appendChild(oNewField);}}

function cnter(MaxLen,idText) {

var Otext = document.getElementById(idText);

if (Otext){

if(Otext.value.length > MaxLen) {

Otext.value = Otext.value.substring(0,MaxLen);

alert("Это поле не может быть длиннее "+MaxLen+" символов.");

Otext.focus();}}}

function showHideSettings(){

$('#text-more-header').next('#text-more').css({'display':'none'});

$('#text-more-header').append('<span><b>развернуть</b></span>');

$('#text-more-header span').click(function(){

$('#text-more').slideToggle(function(){

$('#text-more-header').toggleClass('minus');

});

$('#text-more-header').hasClass('minus') ? $('#text-more-header span').html('развернуть') : $('#text-more-header span').html('свернуть');});}

cmsUTF/data/cms.search.php

<?

$firstTime = false;

if(isset($\_GET["sid"])){

$cmsSearchId = $\_GET["sid"]\*1;

$cmsSearchTable = CMS\_PREFIX."\_search\_cash\_".$cmsSearchId;

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_search\_cash WHERE scid=".$cmsSearchId);

if($data=$sql->get()){

$cmsSearchString = $data["iwords"];

$cmsSearchCountString = explode(",",$data["iword\_count"]);

$cmsSearchArray = explode(" ",$cmsSearchString); }else{

header("location:".$access->self.$access->path);

exit(0);} }else{

$cmsSearchId = TODAY;

$cmsSearchString = urldecode($\_GET["s"]);

$cmsSearchArray = get\_search\_array($cmsSearchString);

if(!$cmsSearchArray){

$firstTime = true; }else{

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_search\_cash WHERE scid<".(TODAY-3\*3600)."");

unset($queries);

while($data = $sql->get()){

$queries[] = CMS\_PREFIX."\_search\_cash\_".$data["scid"];}

if(is\_array($queries)){

$sql->send("DROP TABLE ".implode(",",$queries)); }

$sql->send("SELECT \* FROM ".CMS\_PREFIX."\_search\_cash WHERE scid>".(TODAY-3\*3600)." AND iwords='".$cmsSearchString."'");

if($data = $sql->get()){

$cmsSearchId = $data["scid"]; }else{

$query = "CREATE TABLE ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId." (";

$query .= " sid int( 11 ) unsigned NOT NULL default '0',";

$query .= " entryid int(10) NOT NULL default '0',";

$query .= " iword varchar(255) NOT NULL default '0',";

$query .= " icount int( 10 ) unsigned NOT NULL default '0',";

$query .= " KEY sid ( sid ) ,";

$query .= " KEY sid\_2(sid,iword),";

$query .= " KEY icount ( icount )";

$query .= ") ENGINE=MyISAM ";

$sql->send($query);

$query = "CREATE TABLE ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId."\_temp (";

$query .= " sid int( 11 ) unsigned NOT NULL default '0',";

$query .= " entryid int(10) NOT NULL default '0',";

$query .= " iword varchar(255) NOT NULL default '0',";

$query .= " KEY sid ( sid ),";

$query .= " KEY sid\_2(sid,iword)";

$query .= ") ENGINE=MyISAM ";

$sql->send($query);

$query = "INSERT INTO ".CMS\_PREFIX."\_search\_cash SET scid=".$cmsSearchId.",iwords='".$cmsSearchString."'";

$sql->send($query);

$countString;

foreach($cmsSearchArray as $key => $val){

$strSQL = "SELECT COUNT(DISTINCT sid,entryid,iword) as ncount FROM ".CMS\_PREFIX."\_indexes WHERE iword LIKE '".$val."%'";

$sql->send($strSQL);

if(($data = $sql->get()) && $data["ncount"]>0){

$countString[] = $val."=".$data["ncount"];

$res = $sql->send("INSERT INTO ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId."\_temp SELECT sid,entryid,iword FROM ".CMS\_PREFIX."\_indexes WHERE LOCATE('".$val."',iword)>0 GROUP BY sid,entryid,iword"); }else{

$countString[] = $val."=0";}}

$query = "INSERT INTO ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId." SELECT sid,entryid,iword,count(\*) as icount FROM ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId."\_temp GROUP BY sid,entryid,iword";

$sql->send($query);

$query = "DROP TABLE ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId."\_temp";

$sql->send($query);

$query = "UPDATE ".CMS\_PREFIX."\_search\_cash SET iword\_count='".implode(",",$countString)."' WHERE scid=".$cmsSearchId."";

$sql->send($query); }

header("location:".$access->self.$access->path."?a=cms.search&sid=".$cmsSearchId);}}

$root->set("search\_string",$cmsSearchString);

$root->set("search\_id",$cmsSearchId);

$struct = & $root->create("cms\_node\_content");

$struct->set\_file(cms\_files\_check("search"));

$struct->setFromBranch();

$struct->set("search\_string",$cmsSearchString);

if($firstTime){

$cms\_node\_content\_word = & $struct->create("cms\_node\_content\_word");

$cms\_node\_content\_word->set("search\_string",$cmsSearchString); }else{

$current = cms\_pages($struct,"SELECT COUNT(DISTINCT sid,entryid) as count FROM ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId." ",CLIENT\_SEARCH\_RESULTS);

$query = "SELECT sid,entryid,sum(icount) as icount FROM ".CMS\_PREFIX."\_search\_cash\_".$cmsSearchId." GROUP BY sid,entryid ORDER BY icount DESC,sid,entryid,iword";

if(CLIENT\_SEARCH\_RESULTS>0) $query .= " LIMIT ".($current\*CLIENT\_SEARCH\_RESULTS).",".CLIENT\_SEARCH\_RESULTS."";

$res = $sql->send($query);

$count = 0;

while($data\_ = $sql->get($res)){

$cms\_child\_node = & cms\_tree("getbyid",$data\_["sid"]);

if(is\_object($cms\_child\_node)){

$cms\_child\_node->load();

$struct\_node = & $struct->create("cms\_node\_child");

if($data\_["entryid"]\*1==0){

$struct\_node->set($cms\_child\_node->data);

cms\_navigation($cms\_child\_node->parent,$struct\_node,"navigation"); }else{

if(method\_exists($cms\_child\_node,"entry\_load")){

$struct\_node->set($cms\_child\_node->entry\_load($data\_["entryid"]\*1));

cms\_navigation($cms\_child\_node,$struct\_node,"navigation"); }else{

die("Все плохо : ". $cms\_child\_node->id);}}

$struct\_node->setFromBranch();

$equal\_node = & $struct\_node->create((($data\_["icount"]==sizeof($cmsSearchArray))?"equal":"notequal"));

$equal\_node->set("icount",$data\_["icount"]);

$equal\_node->set("scount",sizeof($cmsSearchArray));

$count++;}}

if($count==0){

$cms\_alert->set("er","cms.search");}}

?>

cmsUTF/scripts/access.php

<?

define("ACCESS\_USER" ,0);

define("ACCESS\_ADMIN" ,1);

define("ACCESS\_UNKNOWN\_USER" ,"access\_unknown\_user");

function authenticate(){

Header("WWW-authenticate: basic realm='For admin'");

Header("HTTP/1.0 401 Unauthorized");

echo "Correct login and password needed !!";

exit;}

class access{ var $base; var $self; var $protocol; var $server\_name; var $path = false; var $path\_map = false; var $logins; var $user = ACCESS\_UNKNOWN\_USER; var $need\_auth = false; var $status = ACCESS\_USER;

function access($logins){

$this->read();

$this->logins = $logins;

$this->status = ACCESS\_USER;

if(!empty($this->path\_map[0]) && $this->path\_map[0]=="admin"){

$this->need\_auth= true;}}

function read(){

$base\_dir = dirname($\_SERVER['SCRIPT\_NAME']);

$base\_dir .= (substr($base\_dir,-1)=="/")?"":"/";

$this->protocol = ((!empty($\_SERVER["HTTPS"]) && $\_SERVER["HTTPS"]=="on") || (!empty($\_SERVER["REDIRECT\_HTTPS"]) && $\_SERVER["REDIRECT\_HTTPS"]=="on") || (!empty($\_SERVER["HTTP\_X\_FORWARDED\_PROTO"]) && $\_SERVER["HTTP\_X\_FORWARDED\_PROTO"]=="https")) ? "https" : "http";

$this->server\_name = $\_SERVER["SERVER\_NAME"].$base\_dir;

$this->self = $this->base = $this->protocol.'://'.$this->server\_name;

if(isset($\_SERVER['PATH\_INFO'])){

if($base\_dir=="/") $this->path = substr($\_SERVER['PATH\_INFO'],1);

else $this->path = str\_replace($base\_dir, '', $\_SERVER['PATH\_INFO']);

$this->path = preg\_replace("/\/$/","",$this->path);

$this->path = (substr($this->path,0,1)=="/")?substr($this->path,1):$this->path;

$this->path\_map = explode("/", $this->path); }}

function check(){

$auth\_done = false;

foreach($this->logins as $login => $pass){

if(!empty($\_SERVER['PHP\_AUTH\_USER']) && !empty($\_SERVER['PHP\_AUTH\_PW']) && $\_SERVER['PHP\_AUTH\_USER']==$login&&$\_SERVER['PHP\_AUTH\_PW']==$pass){

$auth\_done = true;

$this->user = $login;

$this->status = ACCESS\_ADMIN;

$this->need\_auth = false;

$this->self = $this->base."admin/";

unset($this->path\_map[0]);

$this->path\_map = array\_values($this->path\_map);

$this->path = implode("/",$this->path\_map);}}

return $auth\_done;}}

$access = new access($LOGINS\_ARRAY);

if($access->need\_auth){

if(!$access->check()) authenticate();}

?>

cmsUTF/scripts/functions.php

<?

function file\_upload1($name,$copyPath,$original=false,$suffix=false){

$result = array("tmp\_name"=>"","source"=>"","ext"=>"","dest"=>"","error"=>"");

$suffix = ($suffix)?$suffix:"";

if(is\_array($\_FILES[$name])){

$result["source"] = $\_FILES[$name]["name"];

if($\_FILES[$name]["tmp\_name"]!="none" && $\_FILES[$name]["tmp\_name"]!=""){

$result["tmp\_name"] = $\_FILES[$name]["tmp\_name"];

if(is\_uploaded\_file($\_FILES[$name]["tmp\_name"])){

$source = $\_FILES[$name]["name"];

if($extension = get\_extension($source)){

$result["ext"] = $extension;

if(!is\_bool($original))

$dest = $copyPath.$suffix.addslashes($original.".".$extension);

elseif($original)

$dest = $copyPath.$suffix.addslashes(basename($\_FILES[$name]["name"])); else

do{$dest= $copyPath.$suffix.substr(md5($source.rand(0,1000000)),0,10).".".$extension; }while(is\_file($dest));

if(move\_uploaded\_file($\_FILES[$name]["tmp\_name"],$dest)){

$result["dest"] = $dest;

return $result; }else{

$result["error"] = "nosave";

return $result;} }else{

$result["error"] = "noext";

return $result;} }else{

$result["error"] = "no\_is\_uploaded\_file";

return $result;} }else{

if($\_FILES[$name]["name"]!="" && $\_FILES[$name]["name"]!="none") $result["error"] = "no\_temp";

return $result;} }else{

return $result;}}

functionstr2path($string){

$match = array(" ","А","Б","В","Г","Д","Е","Ё","Ж","З","И","Й","К","Л","М","Н","О","П","Р","С","Т","У","Ф","Х","Ц","Ч","Ш","Щ","Ъ","Ы","Ь","Э","Ю","Я","а","б","в","г","д","е","ё","ж","з","и","й","к","л","м","н","о","п","р","с","т","у","ф","х","ц","ч","ш","щ","ъ","ы","ь","э","ю","я");

$replace = array("\_","A","B","V","G","D","E","E","Zh","Z","I","J","K","L","M","N","O","P","R","S","T","U","F","H","Tc","Ch","Sh","Sch","","Y","","E","Ju","Ya","a","b","v","g","d","e","e","zh","z","i","y","k","l","m","n","o","p","r","s","t","u","f","h","tc","ch","sh","sch","","y","","e","ju","ya");

$string = str\_replace($match,$replace,$string);

$string = preg\_replace("/(\_+)/u","\_",$string);

$string = preg\_replace("/([^0-9a-zA-Z\_])/u","",$string);

return strtolower($string);}

function get\_extension($fname){

$path\_parts = pathinfo($fname);

$extension = strtolower($path\_parts["extension"]);

foreach(explode(",","php,php3,phtml") as $key => $val){

if($extension==$val) return false;}

return $extension;}

function file\_from\_url($source,$copyPath,$original=false,$suffix=false) {

$urlArray = parse\_url($source);

$urlArray["port"] = "80";

$sock = @fsockopen($urlArray["host"], $urlArray["port"]);

if($extension = get\_extension($source)){

if(!is\_bool($original))

$dest = $copyPath.$suffix.addslashes($original.".".$extension);

elseif($original)

$dest = $copyPath.$suffix.addslashes(basename($source)); else

do{$dest= $copyPath.$suffix.substr(md5($source.rand(0,1000000)),0,10).".".$extension; }while(is\_file($dest));

if($sock){

$dump .= "GET ".$urlArray["path"]." HTTP/1.1\r\n";

$dump .= "Host: ".$urlArray["host"]."\r\n";

$dump .= "Connection: close\r\n";

$dump .= "Connection: close\r\n\r\n";

fputs($sock, $dump);

while($str = fgets($sock, 1024)) $headers[] = $str;

fclose($sock);

flush();

if(substr($headers[0],9,3)=="200"){

if(str2file($dest,implode("",array\_slice($headers,10)))){

return $dest;}}}}

return false;}

function file\_upload($name,$copyPath,$original=false,$suffix=false){

$suffix = ($suffix)?$suffix:"";

if(!empty($\_FILES[$name]) && is\_array($\_FILES[$name]) && !empty($\_FILES[$name]["tmp\_name"]) && $\_FILES[$name]["tmp\_name"]!="none"){

if(is\_uploaded\_file($\_FILES[$name]["tmp\_name"])){

$source = $\_FILES[$name]["name"];

if($extension = get\_extension($source)){

if(!is\_bool($original))

$dest = $copyPath.$suffix.addslashes($original.".".$extension);

elseif($original)

$dest = $copyPath.$suffix.addslashes(basename($\_FILES[$name]["name"])); else

do{$dest= $copyPath.$suffix.substr(md5($source.rand(0,1000000)),0,10).".".$extension; }while(is\_file($dest));

if(move\_uploaded\_file($\_FILES[$name]["tmp\_name"],$dest)){

return $dest; }else

return false;}}}

return false;}

function file2str($fileName){

$string='';

if(is\_file($fileName)){

if($fd = @fopen($fileName, "r")){

$string .= @fread($fd, filesize($fileName));

fclose($fd);}}

return $string;}

function str2file($fileName,$string){

if($fw = @fopen($fileName, "w")){

fwrite($fw,$string);

fclose($fw);

return true; }else{

return false;}}

function image\_preview($src\_image,$dest\_width,$dest\_height,$portrait=1,$landscape=1,$dest\_image=false,$quality=80) {

if(!$dest\_image) $dest\_image = $src\_image;

$src\_ext = get\_extension($src\_image);

$dest\_ext = get\_extension($src\_image);

$dest\_ext = ($dest\_ext=="gif" ? "png" : $dest\_ext);

$src\_func = "imagecreatefrom".($src\_ext=="jpg"?"jpeg":$src\_ext);

$dest\_func = "image".($dest\_ext=="jpg"?"jpeg":$dest\_ext);

if(!($portrait==0 || $portrait==1 || $portrait==2)) $portrait = 1;

if(!($landscape==0 || $landscape==1 || $landscape==2)) $landscape = 1;

$src\_img = @$src\_func($src\_image);

if(!$src\_img) return false;

$src\_width = imagesx($src\_img);

$src\_height = imagesy($src\_img);

if($dest\_width>=$src\_width&&$dest\_height>=$src\_height){

$ar = $dest\_width/$dest\_height;

$ar1 = $src\_width/$src\_height;

if($ar1 == $ar){

if($dest\_image!=$src\_image) copy($src\_image,$dest\_image);

return true; }else{

if($ar1 > $ar){

$dest\_width = $ar\*$src\_height;

$dest\_height = $src\_height; }else{

$dest\_height = $src\_width/$ar;

$dest\_width = $src\_width;}}}

$sC = $src\_width/$src\_height;

$dC = $dest\_width/$dest\_height;

$dst\_img = imagecreatetruecolor($dest\_width,$dest\_height);

if($sC>=$dC){

$k = $dest\_height/$src\_height;

switch ($landscape\*1) {

case 0:

$fromX = 0;

break;

case 2:

$fromX = max(0,round($src\_width - $dest\_width/$k));

break;

default:

$fromX = max(0,round(($src\_width - $dest\_width/$k)/2));}

$fromY = 0;

$toY = $src\_height;

$toX = $fromX + round(($toY-$fromY)\*($dest\_width/$dest\_height)); }else{

$k = $dest\_width/$src\_width;

$fromX = 0;

$toX = $src\_width;

switch ($portrait\*1) {

case 0:

$fromY = 0; break;

case 2:

$fromY = max(0,round($src\_height - $dest\_height/$k)); break;

default:

$fromY = max(0,round(($src\_height - $dest\_height/$k)/2));}

$toY = $fromY + round(($toX-$fromX)\*($dest\_height/$dest\_width)); }

if($dest\_ext=="png") imageAlphaBlending($dst\_img, false);

imagecopyresampled($dst\_img, $src\_img, 0, 0, $fromX, $fromY, $dest\_width ,$dest\_height, $toX-$fromX , $toY-$fromY);

if($dest\_ext=="png") imageSaveAlpha($dst\_img, true);

$dest\_func($dst\_img, $dest\_image, ($dest\_ext=="jpg"?$quality:0));

imagedestroy($src\_img);

imagedestroy($dst\_img);

return true; }

function image\_resize($src\_image,$dest\_width,$dest\_height,$dest\_image=false,$quality=80) {

if(!$dest\_image) $dest\_image = $src\_image;

$src\_ext = get\_extension($src\_image);

$dest\_ext = get\_extension($src\_image);

$dest\_ext = ($dest\_ext=="gif" ? "png" : $dest\_ext);

$src\_func = "imagecreatefrom".($src\_ext=="jpg"?"jpeg":$src\_ext);

$dest\_func = "image".($dest\_ext=="jpg"?"jpeg":$dest\_ext);

$src\_img = @$src\_func($src\_image);

if(!$src\_img) return false;

$src\_width = imagesx($src\_img);

$src\_height = imagesy($src\_img);

$mc1 = $dest\_width/$src\_width;

$mc2 = $dest\_height/$src\_height;

$k = min($mc1,$mc2);

$dest\_width = round($src\_width \* $k);

$dest\_height = round($src\_height \* $k);

if($dest\_width>$src\_width&&$dest\_height>$src\_height){

if($dest\_image!=$src\_image) copy($src\_image,$dest\_image);

return true; }

$dst\_img = imagecreatetruecolor($dest\_width,$dest\_height);

if($dest\_ext=="png") imageAlphaBlending($dst\_img, false);

imagecopyresampled($dst\_img, $src\_img, 0, 0, 0, 0, $dest\_width, $dest\_height, $src\_width, $src\_height);

if($dest\_ext=="png") imageSaveAlpha($dst\_img, true);

$dest\_func($dst\_img, $dest\_image, ($dest\_ext=="jpg"?$quality:0));

imagedestroy($src\_img);

imagedestroy($dst\_img);

return true; }

$rus\_month = array("Jan"=>"января","Feb"=>"февраля","Mar"=>"марта","Apr"=>"апреля","May"=>"мая","Jun"=>"июня","Jul"=>"июля","Aug"=>"августа","Sep"=>"сентября","Oct"=>"октября","Nov"=>"ноября","Dec"=>"декабря");

$rus\_month1 = array("Jan"=>"Янв","Feb"=>"Фев","Mar"=>"Мар","Apr"=>"Апр","May"=>"Май","Jun"=>"Июн","Jul"=>"Июл","Aug"=>"Авг","Sep"=>"Сен","Oct"=>"Окт","Nov"=>"Ноя","Dec"=>"Дек");

$rus\_month2 = array("Jan"=>"Январь","Feb"=>"Февраль","Mar"=>"Март","Apr"=>"Апрель","May"=>"Май","Jun"=>"Июнь","Jul"=>"Июль","Aug"=>"Август","Sep"=>"Сентябрь","Oct"=>"Октябрь","Nov"=>"Ноябрь","Dec"=>"Декабрь");

$rus\_weekdays = array("Mon"=>"Понедельник","Tue"=>"Вторник","Wed"=>"Среда","Thu"=>"Четверг","Fri"=>"Пятница","San"=>"Суббота","Sun"=>"Воскресенье");

function int2date($timestamp=0,$format="d.m.Y H:i",$language="en") {

global $access;

global $rus\_month, $rus\_month1, $rus\_month2,$rus\_weekdays;

if($timestamp<=0) $timestamp= time();

$date = date($format,$timestamp);

if($language="ru"){

if(defined('CLIENT\_DATE\_FORMAT\_M\_SHORT') && CLIENT\_DATE\_FORMAT\_M\_SHORT\*1==1){

$date = str\_replace(array\_keys($rus\_month1),array\_values($rus\_month1),$date);

}elseif(defined('CLIENT\_DATE\_FORMAT\_M\_LONG') && CLIENT\_DATE\_FORMAT\_M\_LONG\*1==1){

$date = str\_replace(array\_keys($rus\_month2),array\_values($rus\_month2),$date); }else{

$date = str\_replace(array\_keys($rus\_month),array\_values($rus\_month),$date);}

$date = str\_replace(array\_keys($rus\_weekdays),array\_values($rus\_weekdays),$date);}

return $date;}

function str2date($date){

$timestamp = 0;

if(!empty($date)){

preg\_match("/(\d{2})[\.\-\/](\d{2})[\.\-\/](\d{2,4})( (\d{2}):(\d{2}))?/u",$date,$int);

if(checkdate($int[2],$int[1],$int[3])){

if(!empty($int[5]) && !empty($int[6]) && $int[5]>=0 && $int[5]<=23 && $int[6]>=0 && $int[6]<=59){

$timestamp = mktime($int[5],$int[6],0, $int[2],$int[1], $int[3]); }else{

$timestamp = mktime(0,0,0, $int[2],$int[1], $int[3]);}

}elseif(checkdate(date("m",$date),date("d",$date),date("Y",$date))){

$timestamp = $date;}}

return $timestamp;}

function strip\_properies($string,$allowable\_property ='type|onmouseover|onmouseout|style|rules|valign|cellspacing|cellpadding|bgcolor|colspan|onclick|rowspan|border|coords|height|valign|target|usemap|align|clear|class|shape|title|width|name|href|data-toggle|alt|src|id|classid|codebase|quality|allowScriptAccess|allowFullScreen|type|pluginspage|wmode|value|flashVars|movie|color|rel|type'){

$return = "";

if(preg\_match\_all("/(".$allowable\_property.")=((\"[^\"]\*\")|('[^']\*')|(\S+(\s|\Z)))/Uimu", $string,$matches)){

for($i=0;$i<sizeof($matches[0]);$i++){

$return .= " ".$matches[0][$i];}}

return rtrim($return);}

function strip\_base($message){

global $access;

if(isset($access->base) && $access->base!="") {

$message = str\_replace($access->base,"./",$message);}

return $message;}

function add\_base($message){

global $access;

if(isset($access->base) && $access->base!="") {

if(substr($message,0,2)=="./"){

$message = $access->base.substr($message,2); }else{

$message = str\_replace("'./","'".$access->base,$message);

$message = str\_replace('"./','"'.$access->base,$message);

$message = str\_replace('=./','='.$access->base,$message);

$message = str\_replace('>./','>'.$access->base,$message);}}

return $message;}

function str2save($message,$strip\_all\_tags=false){

if(is\_array($message)){

reset($message);

foreach($message as $key => $val)

$message[$key] = str2save($message[$key],$strip\_all\_tags); }else{

$message = strip\_base($message);

if($strip\_all\_tags){

$message = strip\_tags($message); }else{

$allowable\_tags = 'strong|embed|small|big|iframe|noindex|noscript|script|embed|param|object|blockquote|colgroup|strike|area|small|span|table|col|div|img|map|sub|sup|tr|td|th|br|hr|em|ul|ol|li|h\d|p|b|i|u|a|s|font|thead|colgroup|col';

$message = preg\_replace("/<(\/?)(".$allowable\_tags.")+([^>]\*)( ?)(\/?)>/Uieu",'"[\\1\\2".strip\_properies("\\3")."\\4\\5]"',$message);

$message = str\_replace("\\'","'",$message);

$message = strip\_tags($message);

$message = preg\_replace("/\[(\/?)(".$allowable\_tags.")+([^\]]\*)( ?)(\/?)\]/Uiu", "<\\1\\2\\3\\4\\5>", $message);}

$message = str\_replace("'","&#039",$message);

$message = str\_replace("\\","&#092",$message);

$message = str\_replace('"',"&quot;",$message);}

return $message;}

function str2html($message,$edit\_mode=false,$need\_replace\_endl=false,$need\_replace\_url=false){

if(is\_array($message)){

reset($message);

foreach($message as $key => $val)

if((string)(int)$key!=$key)

$message[$key] = str2html($message[$key],$edit\_mode,$need\_replace\_endl,$need\_replace\_url); }else{

$message = str\_replace("&#039","'",$message);

$message = str\_replace("&#092",'\\',$message);

$message = str\_replace("&quot;",'"',$message);

$message = add\_base($message);

if($need\_replace\_endl){

$message = strip\_tags($message);

$message = nl2br($message);

}

if($need\_replace\_url){

$message = preg\_replace("/((http(s?):\/\/)|(www\.))([\w\~\.\/\-=#\?&\_]+)/iu", "<a href=\"http://\\4\\5\" target='\_blank'>\\4\\5</a>", $message);

$message = preg\_replace("/(([\w\.\-\_]+)(@)([\w\.\-\_]+))/iu", "<a href=\"mailto:\\2@\\4\">\\2@\\4</a>", $message);}}

return $message;}

function str\_convert($in\_charset, $out\_charset, $string){

$charsets = array("w"=>"Windows-1251","u8"=>"UTF-8");

return iconv($charsets[$in\_charset],$charsets[$out\_charset],$string);}

function dir\_make($dirName,$write\_mode=false){

if(is\_dir($dirName)){

return true; }else{

$path\_url\_array = explode("/",$dirName);

$current\_path="";

for($i=0;$i<sizeof($path\_url\_array);$i++){

if($path\_url\_array[$i]!=""){

$current\_path .= $path\_url\_array[$i]."/";

if(!is\_dir($current\_path))

if(mkdir($current\_path)){

}else{

return false;}}}

return true;}}

function dir\_remove($dirName){

if(substr($dirName,-1)=="/") $dirName = substr($dirName,0,-1);

if($d = @dir($dirName)){

dir\_make($dirName."@".date('Ymd-His',time()),true);

while($entry = $d->read()) {

if ($entry != "." && $entry != "..") {

if (is\_dir($dirName."/".$entry)) {

dir\_remove($dirName."/".$entry); }else{

if (!copy($dirName."/".$entry, $dirName."@".date('Ymd-His',time())."/".$entry.'@'.date('Ymd-His',time()).'')) {//

} else {

@unlink($dirName."/".$entry);}}}}

$d->close(); }

@rmdir($dirName); }

function size\_convert($size,$need\_bytes=true){

$sign = ($size>0)?"":"-";

$size = abs($size);

if($size>=1073741824){

$psize = number\_format($size/1073741824,2)." Gb";

}elseif($size>=1048576){

$psize = number\_format($size/1048576,2)." Mb";

}elseif($size>=1024){

$psize = number\_format($size/1024,2)." Kb";

}elseif(!$need\_bytes){

$psize = "1Kb"; }else{

$psize = $size."bytes";}

if($size == 0) $psize = "0bytes";

return $sign.$psize;}

if( !function\_exists('memory\_get\_usage') ) {

function memory\_get\_usage() {

if ( substr(PHP\_OS,0,3) == 'WIN') {

$output = array();

exec( 'tasklist /FI "PID eq ' . getmypid() . '" /FO LIST', $output );

return preg\_replace( '/[\D]/', '', $output[5] ); }else{

$pid = getmypid();

exec("ps -eo%mem,rss,pid | grep $pid", $output);

$output = explode(" ", $output[0]);

return $output[1]; }}}

function time\_control(&$html,$reset=false,$massage=""){

static $ctime;

static $stime;

static $lmem;

$m=explode(" ", microtime());

$ltime=$ctime;

$ctime=(float)$m[1] + (float)$m[0];

$cmem = memory\_get\_usage();

if($reset){

$stime = $ctime;

$lmem = $cmem;

$html .= "time control: started\n"; }else{

$html .= "time control: ";

$html .= "lo: ".round($ctime-$ltime,5)."\t [".size\_convert($cmem-$lmem)."],\t";

$html .= "summ: ".round($ctime-$stime,5)."\t [".size\_convert($cmem)."]";

$html .= (strlen($massage)>0)?"\tmess: [".$massage."]":"";

$html .= "\n";

$lmem = $cmem;}}

function get\_search\_array($string){

$string = trim($string);

if(strlen($string)>0){

$string=html\_entity\_decode($string,null,'UTF-8');

$string=strip\_tags($string);

$string=preg\_replace('~[^A-Za-z0-9а-яА-ЯЁё \-]~ui',' ',$string);

$string=preg\_replace('~\s+~',' ',$string);

$wordsEnding=array('а','е','ё','и','о','у','ь','ы','э','ю','я','[ая][мях]','[ео][мйвеюх]','и[хмейяию]','у[юйя]','[ью][юя]','ы[хмей]','ьев','ья[хм]','[иь]я[мх]','[иу]ей','[ео]го','[ео]му','[иоуь]ем','[аиыя]ми','[ыиуь]ями');

$string=preg\_replace('~(?:'.implode('|',$wordsEnding).') ~iu',' ', $string.' ');

$array=explode(' ',$string);

array\_pop($array);

$array=array\_unique($array);

foreach($array as $k=>$v) if(3>mb\_strlen($array[$k],'utf8')) unset($array[$k]);

array\_values($array);

return $array; }else{

return false;}}

?>

cmsUTF/scripts/mail.php

<?

function cyrMailHeader($t){

return '=?utf-8?B?'.base64\_encode($t).'?=';}

class Mail{ var $to; var $from; var $subject; var $text; var $files; var $charset; var $boundary; var $error="";

function Mail($charset='u',$texttype="plain"){

$this->charset = ($charset=='w')?'w':'u';

$this->texttype = ($texttype=='plain')?'plain':'html';

$this->boundary = md5(date("YmdHis"));

$this->files = false;}

function from($mail,$name=""){

$this->from = "From: ".cyrMailHeader($name)." <".$mail.">\n"; }

function to($mail,$name=""){

if($name=="") {$this->to = ""." <".$mail.">";}

if($name!="") {$this->to = "".cyrMailHeader($name)." <".$mail.">";}}

function subj($subject){

$this->subject = $subject;}

function text($text){

$this->text = $text."\n\n\n";}

function attach\_file($filename){

$this->files[] = $filename;}

function send(){

$header = "X-Mailer: PHP/".phpversion()." (www.r52.ru)\n";

$header .= $this->from;

$mailtext = "";

if($this->files){

$header .= "MIME-Version: 1.0\n";

$header .= "Content-Type: multipart/mixed; boundary=\"----------".$this->boundary."\"\n\n";

$mailtext .= "------------".$this->boundary."\n";

$mailtext .= "Content-Type: text/".$this->texttype."; charset=".(($this->charset=='u')?"utf-8":"Windows-1251")."\n\n";

$mailtext .= $this->text;

foreach($this->files as $filename){

$fp = fopen($filename, "rb");

if($fp){

$mailtext .= "------------".$this->boundary."\n";

$mailtext .= 'Content-Disposition: attachment; filename="'.basename($filename).'"'."\n";

$mailtext .= 'Content-Type: application/octet-stream; name="'.basename($filename).'"'."\n";

$mailtext .= 'Content-Transfer-Encoding: base64'."\n\n";

$content = fread($fp,filesize($filename));

$mailtext .= chunk\_split(base64\_encode($content))."\n";

fclose($fp);}} }else{

$header .= "Content-Type: text/".$this->texttype."; charset=".(($this->charset=='u')?"utf-8":"Windows-1251")."\n\n";

$mailtext = $this->text;}

return mail($this->to, cyrMailHeader($this->subject), $mailtext, $header);}}

?>

cmsUTF/scripts/mysql.php

<?

class sql{ var $host; var $user; var $pass; var $dbname; var $dbconnect; var $result;

function sql($host,$user,$pass){

$this->host=$host;

$this->user=$user;

$this->pass=$pass;

$this->dbconnect = mysql\_connect($this->host,$this->user,$this->pass);

if(!$this->dbconnect){

if(DEBUG){ echo "<font color=#cccccc>sql error: ".mysql\_errno().": ". mysql\_error()."</font>\n";}

return false; }else{

return true;}}

function disconnect(){

@mysql\_close($this->dbconnect);

$this->dbconnect = false;}

function set\_locale($locale){

mysql\_query("SET character\_set\_client = '".$locale."'");

mysql\_query("SET character\_set\_connection = '".$locale."'");

mysql\_query("SET character\_set\_results = '".$locale."'");

mysql\_query("SET NAMES '".$locale."'"); }

function set\_db($dbname){

$db = @mysql\_select\_db($dbname);

if($db){

$this->dbname = $dbname;

return true; }else{

if(DEBUG){ echo "<font color=#cccccc>sql error: ".mysql\_errno().": ". mysql\_error()."</font>\n";}

return false;}}

function num($result=0){

if($result==0) $result = $this->result;

return @mysql\_num\_rows($result);}

function send($query,$dbname=""){

if($dbname==""){

$this->result = @mysql\_query($query); }else{

$this->result = @mysql\_db\_query($dbname,$query);}

if(!$this->result){

if(DEBUG){ echo "<font color=#cccccc>sql error: ".mysql\_errno().": ". mysql\_error()."<br />".$query."<br /><br /><br /></font>\n";}}

return $this->result;}

function get($result=0){

if($result==0) $result = $this->result;

$row = @mysql\_fetch\_array($result,MYSQL\_ASSOC);

if(!is\_array($row)){

return false; }else

return $row;}

function free($result=0){

if($result==0){

@mysql\_free\_result($this->result);

$this->result = 0; }else{

@mysql\_free\_result($result);}}

function affect($result=0){

return @mysql\_affected\_rows (($result==0)?$this->result:$result);}

function get\_id(){

if($id = @mysql\_insert\_id()) return $id;

else return 0;}

function lock($tables,$dbname=""){

$tables\_ar = explode(",",$tables);

if(sizeof($tables\_ar)>1) $tables = implode(" WRITE, ",$tables\_ar)." WRITE";

$query = "LOCK TABLES ".$tables." WRITE";

$result = $this->result;

if($this->send($query,$dbname)) $this->free();

$this->result = $result;}

function unlock($dbname=""){

$query = "UNLOCK TABLES";

$result = $this->result;

if($this->send($query,$dbname)) $this->free();

$this->result = $result;}

?>

cmsUTF/scripts/struct.php

<?

function structure\_collection(&$value){

static $st\_count = 0;

static $st\_structs;

if(@get\_class($value)=="structure"){

$st\_structs[$st\_count] = & $value;

return $st\_count++;

}elseif(is\_int($value)){

if(@get\_class($st\_structs[$value])=="structure"){

return $st\_structs[$value];

}elseif($value===0){}

}else{return $st\_structs;}}

class structure{

var $sid;

var $shablone\_name;

var $shablone\_file = false;

var $reverse = false;

var $sdata = array();

var $schilds = array();

function create($shablone\_name,$values=false){

$new = new structure;

$new->shablone\_name = $shablone\_name;

$new->sid = structure\_collection($new);

if(@get\_class($this)=="structure"){

if($this->reverse){

array\_unshift($this->schilds, $new);

}else{

$this->schilds[] = & $new;}

if(isset($this->sid)){

$new->pid = $this->sid;}

if($values) $new->set($values);}

return $new;}

function id(){

return $this->sid;}

function set($param\_1,$param\_2=false){

if((!$param\_2) && is\_array($param\_1)){

foreach($param\_1 as $key=>$val) $this->set($key,$val);}else{

$this->sdata[$param\_1]=$param\_2;}}

function set\_file($filename){

$this->shablone\_file = $filename;}

function reverse($reverse=false){

$this->reverse = $reverse;}

function setFromBranch($id=0){

$branch = & structure\_collection($id);

foreach($branch->sdata as $key=>$val){

if(!isset($this->sdata[$key])){

$this->sdata[$key] = & $branch->sdata[$key];}}}

function get(&$tpl){

return $tpl->replace($this);}}

?>

cmsUTF/scripts/template.php

<?

class template{

public $collection; public $filename=""; public $files; public function template($filename){

static $objs;

$this->ini();

if(isset($objs[$filename])){

$this->collection = & $objs[$filename]->collection;

$this->filename = & $objs[$filename]->filename;

$this->files = & $objs[$filename]->files;}else{

$objs[$filename] = & $this;

$this->load($filename);}}

public function ini($delim = '##'){

$this->delim = $delim;

$this->struct\_patterns = "/<\!--(\w+)(=)?-->(.\*)<\!--\/\\1-->/Us";

$this->struct\_replace = $this->delim."\\1".$this->delim;

$this->pointer\_patterns = "/".$this->delim."([^-]+)".$this->delim."/U";}

public function load($filename){

if(file\_exists($filename)){

$this->filename = $filename;

$this->parce(file2str($filename));

return true; }else{

if(DEBUG && $filename!=""){ echo "<font color=#cccccc>template error: файл шаблона ".$filename." не существует</font>\n";}

return false;}}

public function parce($block){

if(preg\_match\_all($this->struct\_patterns,$block,$matches)){

$size=sizeof($matches[1]);

for($i=0;$i<$size;$i++){

if(strlen($matches[2][$i])){

$this->files[$matches[1][$i]]=$matches[3][$i];

$this->collection[$matches[1][$i]]=""; }else{

$this->collection[$matches[1][$i]]=preg\_replace($this->struct\_patterns,$this->struct\_replace,$matches[3][$i]);

$this->parce($matches[3][$i]);}}}}

public function \_replace($pointers,$html){

foreach($pointers as $pointer=>$replace){

$html = str\_replace($this->delim.$pointer.$this->delim, $replace.$this->delim.$pointer.$this->delim, $html); }

return $html; }

public function replace($struct){

if(isset($this->files[$struct->shablone\_name])){

if($struct->shablone\_file){

$filename = $struct->shablone\_file; }else{

$filename = $this->files[$struct->shablone\_name];}

$template = new template($filename);

$html = $struct->get($template); }else{

if (isset($this->collection[$struct->shablone\_name])) {

$html = $this->\_replace($struct->sdata,$this->collection[$struct->shablone\_name]);

if(isset($struct->schilds)){

$size=sizeof($struct->schilds);

for($i=0;$i<$size;$i++){

$html = $this->\_replace(array($struct->schilds[$i]->shablone\_name => $this->replace($struct->schilds[$i])),$html); } }}}

return @preg\_replace($this->pointer\_patterns, '', $html);}}

?>