<?php/\*\* \* ------------------------------------------------------------------------ \* JUDownload for Joomla 2.5, 3.x \* ------------------------------------------------------------------------ \* \* @copyright Copyright (C) 2010-2014 JoomUltra Co., Ltd. All Rights Reserved. \* @license GNU General Public License version 2 or later; see LICENSE.txt \* @author JoomUltra Co., Ltd \* @website http://www.joomultra.com \* @----------------------------------------------------------------------@ \*/defined('\_JEXEC') or die('Restricted access');class plgJUDownloadOSMembership extends JPlugin{ /\* \* Check if current user can download $document \* Return true if can download \* Return error message if can not download \* \*/ public function canDownload($document) { $user = JFactory::getUser(); if (!$user->id) { return null; } if (!$document) { return null; } $catId = JUDownloadFrontHelperCategory::getMainCategoryId($document->id); if (!$catId) { return null; } // Membership plans that user need to subscribe to download the document $plans = JUDownloadFrontHelperPluginParams::getDocPluginParamRecursive($this->\_name, $document->id, 'plans', array(), 'plan\_type', 'plan\_type'); // If has \* in plans, means all plans, so just need to check \* if (in\_array('\*', $plans)) { $plans = array('\*'); } // Membership plan statuses that user can download $statuses = (array) $this->params->get('statuses', array(1)); $allowDownload = false; foreach ($statuses as $status) { foreach ($plans as $plan) { if ($this->checkMembership($plan, $status)) { $allowDownload = true; break 2; } } } if ($allowDownload) { return true; } return $this->params->get('message', 'Please subscribe to download'); } /\* \* Check if current user has a specified plan with status \* \*/ public function checkMembership($plan, $status) { $db = JFactory::getDbo(); $user = JFactory::getUser(); $date = JFactory::getDate(); $now = $date->toSql(); $query = $db->getQuery(true); // Limit download field\_id $field\_id = $this->params->get('field\_id', 15); $query->select('COUNT(\*)') ->from('#\_\_osmembership\_subscribers AS sub') ->join('LEFT', '#\_\_osmembership\_field\_value AS fv ON sub.id = fv.subscriber\_id') ->where('sub.published = ' . $db->quote($status)) ->where('sub.user\_id =' . $user->id) ->where('sub.from\_date <= ' . $db->quote($now)) ->where('(sub.to\_date = ' . $db->quote($db->getNullDate()) . ' OR sub.to\_date >= ' . $db->quote($now) . ')') ->join('', '#\_\_osmembership\_plans AS p ON p.id = sub.plan\_id') ->where('p.published = 1') // Subsciber need to have download limit > 0 ->where('fv.field\_id = ' . $field\_id) ->where('fv.field\_value > 0'); if ($plan && $plan != '\*') { $query->where('p.id = ' . $db->quote($plan)); } $db->setQuery($query); return $db->loadResult(); } /\* \* Check if user can edit this rule \* Return true if user can edit this rule \* Return false if user can not edit this rule \* \*/ public function canEdit() { //TODO set permission for user edit this rule return true; } public function onAfterDownloadDocument($documentId, $fileIdArray, $size) { $document = JUDownloadFrontHelperDocument::getDocument($documentId); // No not deduct point of document owner if (JUDownloadFrontHelperPermission::isDocumentOwner($documentId)) { return; } // No not deduct point of moderator has download permission if (JUDownloadFrontHelperModerator::isModerator()) { if (JUDownloadFrontHelperModerator::checkModeratorCanDoWithDocument($document->cat\_id, 'document\_download')) { return; } } if ($document) { $db = JFactory::getDbo(); $user = JFactory::getUser(); $date = JFactory::getDate(); $now = $date->toSql(); // Membership plans that user need to subscribe to download the document $plans = JUDownloadFrontHelperPluginParams::getDocPluginParamRecursive($this->\_name, $document->id, 'plans', array(), 'plan\_type', 'plan\_type'); // If has \* in plans, means all plans, so just need to check \* if (in\_array('\*', $plans)) { $plans = array('\*'); } // Membership plan statuses that user can download $statuses = (array) $this->params->get('statuses', array(1)); // Limit download field\_id $field\_id = $this->params->get('field\_id', 15); foreach ($statuses as $status) { foreach ($plans as $plan) { $query = $db->getQuery(true); $query->select('sub.id') ->from('#\_\_osmembership\_subscribers AS sub') ->where('sub.published = ' . $db->quote($status)) ->where('sub.user\_id =' . $user->id) ->where('sub.from\_date <= ' . $db->quote($now)) ->where('(sub.to\_date = ' . $db->quote($db->getNullDate()) . ' OR sub.to\_date >= ' . $db->quote($now) . ')') ->join('', '#\_\_osmembership\_plans AS p ON p.id = sub.plan\_id') ->where('p.published = 1'); if ($plan && $plan != '\*') { $query->where('p.id = ' . $db->quote($plan)); } $db->setQuery($query); $subIds = $db->loadColumn(); foreach($subIds AS $subId) { // Update download limit(field\_value) of subscriber id that has download limit > 0 $query = "SELECT COUNT(\*) FROM #\_\_osmembership\_field\_value WHERE subscriber\_id = " . $subId . " AND field\_id = " . $field\_id . " AND field\_value > 0"; $db->setQuery($query); if($db->loadResult()) { $query = "UPDATE #\_\_osmembership\_field\_value SET field\_value = field\_value - 1 WHERE subscriber\_id = " . $subId . " AND field\_id = " . $field\_id . " AND field\_value > 0"; $db->setQuery($query); $db->execute(); break 2; } } } } } return true; }}