

An onPrepareContent Mambot

Last Updated Thursday, 15 September 2005

Here is a code framework for a mambot that it triggered by the onPrepareContent event (this is the traditional Mambot).

```
<?php
/**
 * @version $Id $
 * @package Joomla
 * @copyright Copyright (C) 2005 Open Source Matters. All rights reserved.
 * @license http://www.gnu.org/copyleft/gpl.html GNU/GPL, see LICENSE.php
 * Joomla! is free software and parts of it may contain or be derived from the
 * GNU General Public License or other free or open source software licenses.
 * See COPYRIGHT.php for copyright notices and details.
 */

/** ensure this file is being included by a parent file */
defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' );

$_PLUGINS->registerFunction( 'onPrepareContent', 'botMosLink' );

/**
 * Link bot
 *
 * Usage:
 * <code>{moslink id="the_id"}</code>
 */
function botMosLink( $published, &$row, $mask=0, $page=0Â ) {
Â global $mosConfig_absolute_path;

Â if (!$published) {
Â Â Â return true;
Â }

Â require_once( $mosConfig_absolute_path . '/includes/domit/xml_saxy_lite_parser.php' );

Â // define the regular expression for the bot
Â $regex = "#{moslinks*(.*?)}#s";

Â // perform the replacement
Â $row->text = preg_replace_callback( $regex, 'botMosLink_replacer', $row->text );

Â return true;
}
/** 
 * Replaces the matched tags an image
 * @param array An array of matches (see preg_match_all)
 * @return string
 */
function botMosLink_replacer( &$matches ) {
Â $attribs = @SAXY_Lite_Parser::parseAttributes( $matches[1] );

Â $id = @$attribs['id'];

Â return '<a href="'.selfRelToAbs( 'index.php?option=com_content&task=view&id=' . $id ).'">Link</a>';
}
?>
```

Functions called by the onPrepareContent event have a required arguments list:function function_name(int \$published, object &\$row, int \$mask=0, int \$page=0)

published1 if the mambot is published, 0 if not
rowA variable reference to the content object
maskThe current mask, default is 0
pageThe current page number, default is 0

The trigger for the `onPrepareContent` event allows for unpublished mambots to still be processed.

The use of the built in `preg_replace_callback` function is a very efficient way of replacing the link tags. Once you define your regular expression, the nominated callback function is called. You simply return the string you want as a replacement for the regular expression.

You may wonder why we pass a 'published' argument. Some Mambots will need to do something if they are not published. For example, the mosimage Mambot needs to remove all of the `` tags from the text if it is not published.