

```

<?php

/**
 * @file
 * Holds theme functions and template preprocesses.
 * Other style related functions are in style.inc
 */

/**
 * Theme function to display a link, optionally buttonized.
 */
function theme_advanced_forum_l(&$variables) {
  $text = $variables['text'];
  $path = empty($variables['path']) ? NULL : $variables['path'];
  $options = empty($variables['options']) ? array() : $variables['options'];
  $button_class = empty($variables['button_class']) ? NULL : $variables['button_class'];

  $l = '';
  if (!isset($options['attributes'])) {
    $options['attributes'] = array();
  }
  if (!is_null($button_class)) {
    // Buttonized link: add our button class and the span.
    if (!isset($options['attributes']['class'])) {
      $options['attributes']['class'] = array("af-button-$button_class");
    }
    else {
      $options['attributes']['class'][] = "af-button-$button_class";
    }
  }
  $options['html'] = TRUE;
  // @codingStandardsIgnoreStart
  $l = l('<span>' . $text . '</span>', $path, $options);
  // @codingStandardsIgnoreEnd
}
else {
  // Standard link: just send it through l().
  $l = l($text, $path, $options);
}

return $l;
}

/**
 * Theme function to show list of types that can be posted in forum.
 */
function theme_advanced_forum_node_type_create_list(&$variables) {
  $forum_id = $variables['forum_id'];

  // Get the list of node types to display links for.
  $type_list = advanced_forum_node_type_create_list($forum_id);

  $output = '';
  if (is_array($type_list)) {
    foreach ($type_list as $type => $item) {
      $output .= '<div class="forum-add-node forum-add-' . $type . '">';
      $output .= theme('advanced_forum_l', array(
        'text' => t('New @node_type', array('@node_type' => $item['name'])),

```

```

        'path' => $item['href'],
        'options' => NULL,
        'button_class' => 'large',
    ));
    $output .= '</div>';
}
}
else {
    // User did not have access to create any node types in this forum so
    // we just return the denial text / login prompt.
    $output = $type_list;
}

return $output;
}

/**
 * Theme function to show simple author pane when not using Author Pane.
 */
function theme_advanced_forum_simple_author_pane(&$variables) {
    $context = $variables['context'];
    // Sending the context rather than the account makes it work for anon comments.
    $name = theme('username', array('account' => $context));

    $account = user_load($context->uid);
    $picture = theme('user_picture', array('account' => $account));

    return '<div class="author-pane">' . $name . $picture . '</div>';
}

/**
 * Theme function to format the reply link at the top/bottom of topic.
 */
function theme_advanced_forum_reply_link(&$variables) {
    $node = $variables['node'];

    // Get the information about whether the user can reply and the link to do
    // so if the user is allowed to.
    $reply_link = advanced_forum_get_reply_link($node);

    if (is_array($reply_link)) {
        // Reply is allowed. Variable contains the link information.
        $output = '<div class="topic-reply-allowed">';
        $output .= theme('advanced_forum_l', array(
            'text' => $reply_link['title'],
            'path' => $reply_link['href'],
            'options' => $reply_link['options'],
            'button_class' => 'large',
        ));
        $output .= '</div>';
        return $output;
    }
    elseif ($reply_link == 'reply-locked') {
        // @TODO: The double span here is icky but I don't know how else to get
        // around the fact that there's no "a" to put the button class on.
        return '<div class="topic-reply-locked"><span class="af-button-large"><span>' . t('Topic
        locked') . '</span></span></div>';
    }
}

```

```

}
elseif ($reply_link == 'reply-forbidden') {
  // User is not allowed to reply to this topic.
  return theme('comment_post_forbidden', array('node' => $node));
}
}

/**
 * Theme function to a formatted list of subforums.
 *
 * @param array $variables
 *   Array of subforums.
 *
 * @return string
 *   Formatted list of subforums.
 */
function theme_advanced_forum_subforum_list(&$variables) {
  $subforums = array();
  foreach ($variables['subforum_list'] as $tid => $subforum) {
    // Note: $subforum->name has not been run through check_plain because
    // it ends up going through there when l() is called without the HTML
    // option. If you change this to set HTML to TRUE, you must sanitize it.
    $text = l($subforum->name, "forum/$tid");
    $text .= ' (' . $subforum->total_posts;

    if (empty($subforum->new_posts)) {
      $text .= ')';
    }
    else {
      $text .= ' - ' . l($subforum->new_posts_text, $subforum->new_posts_path,
        array('fragment' => 'new')) . ')';
    }

    $subforums[] = $text;
  }
  return implode(', ', $subforums);
}

/**
 * Theme function to a formatted list of subcontainers.
 *
 * @param array $variables
 *   Array of subcontainers.
 *
 * @return string
 *   Formatted list of subcontainers.
 */
function theme_advanced_forum_subcontainer_list(&$variables) {
  $subcontainers = array();
  foreach ($variables['subcontainer_list'] as $tid => $subcontainer) {
    // Note: $subcontainer->name has not been run through check_plain because
    // it ends up going through there when l() is called without the HTML
    // option. If you change this to set HTML to TRUE, you must sanitize it.
    $text = l($subcontainer->name, "forum/$tid");
    $text .= ' (' . $subcontainer->total_posts;

    if (empty($subcontainer->new_posts)) {

```

147 2014 12 14 10:38

```
    $text .= ' ' . $text;
  }
  else {
    $text .= ' - ' . l($subforum->new_posts_text, $subforum->new_posts_path,
      array('fragment' => 'new')) . ' ' . $text;
  }

  $subcontainers[] = $text;
}

return implode(' ', $subcontainers);
}
```