```
function convert_query_to_page_query($query, $pid) {
  $newQuery = substr($query, 0, strlen($query) - 23);
  $pid = str_replace(':', '?', $pid);
  $newQuery = $newQuery . " AND PID:$pid* AND dc.type:Text";
  return $newQuery;
}

/**
 * Custom form element to do our nice images.
 */
function islandora_book_elements() {  // Change this line
  $type['imagebutton'] = array(
    '#input' => TRUE,
    '#button_type' => 'submit',
    '#executes_submit_callback' => TRUE,
    '#name' => 'op',
    '#process' => array('hook_imagebutton_process' => array()),
  );
  return $type;
}

function theme_imagebutton($element) {
  return '<input  style="background:url(\'\'); border:0px; width:10px; padding:0px,0px,0px,0px;" type="image"
   class="form-' . $element['#button_type'] . '" name="' . $element['#name'] . '" id="' . $element['#id'] . '"
   value="' . check_plain($element['#default_value']) . '" ' . drupal_attributes($element['#attributes']) . '
   src="' . $element['#image'] . '" alt="' . $element['#title'] . '" title="' . $element['#title'] . "\" />\n";
}

/**
 * Implementation of hook_theme() to register how to theme image buttons.
 */
function islandora_book_theme() {
  return array(
    'imagebutton' => array(
      'arguments' => array('form' => NULL),
    ),
    'islandora_book_book_search_form' => array(
      'arguments' => array('form' => NULL),
    ),
    'islandora_book_simple_book_search_form' => array(
      'arguments' => array('form' => NULL),
    )
  );
}

/**
 * Implementation of hook_requirements().
 *
 * @return
 *   An array describing the status of the site regarding available updates.
 *   If there is no update data, only one record will be returned, indicating
 *   that the status of core can't be determined. If data is available, there
 *   will be two records: one for core, and another for all of contrib
 *   (assuming there are any contributed modules or themes enabled on the
 *   site). In addition to the fields expected by hook_requirements ('value',
 *   'severity', and optionally 'description'), this array will contain a
 *   'reason' attribute, which is an integer constant to indicate why the
 *   given status is being returned (UPDATE_NOT_SECURE, UPDATE_NOT_CURRENT, or
 *   UPDATE_UNKNOWN). This is used for generating the appropriate e-mail
 *   notification messages during update_cron(), and might be useful for other
 *   modules that invoke update_requirements() to find out if the site is up
 *   to date or not.
 *
 * @see _update_message_text()
 * @see _update_cron_notify()
 */
function islandora_book_requirements($phase) {
  global $base_url;
  if ($phase == 'runtime') {
    module_load_include('inc', 'fedora_repository', 'api/fedora_utils');
    $requirements['iiv-war'] = array();
    $requirements['iiv-war']['title'] = t("Islandora OpenLayers image viewer web app");
    if (!_islandora_book_viewer_available()) {
      $requirements['iiv-war']['value'] = ("Not available");
      $requirements['iiv-war']['severity'] = REQUIREMENT_ERROR;
      $requirements['iiv-war']['description'] = t('Ensure that Fedora is running and that the <a href="@iiv-
   home">IIV</a> app is deployed.',
              array('@iiv-home' => 'http://github.com/islandora/iiv'));
    }
    else {
      $requirements['iiv-war']['value'] = ("Available");
      $requirements['iiv-war']['severity'] = REQUIREMENT_OK;
    }
  }
  return $requirements;
}

function _islandora_book_viewer_available() {
  $url = parse_url(variable_get('fedora_base_url', 'http://localhost:8080/fedora'));
  $fedora_host = ("{$url['scheme']}://{$url['host']}" . (!empty($url['port']) ? ":{$url['port']}/" : '/'));
  $response = drupal_http_request("$fedora_host/iiv/images/loading.gif");
  return empty($response->error);
}

function islandora_book_required_fedora_objects() {
```