

```

}
return $form;
}

function islandora_book_purge_pages_form_submit($form, &$form_state) {
  module_load_include('inc', 'fedora_repository', 'ConnectionHelper');
  $pid = $form_state['values']['pid'];
  if (!isset($form_state['storage']['confirm'])) {
    $form_state['storage']['confirm'] = TRUE; // this will cause the form to be rebuilt, entering the confirm
    part of the form
    $form_state['rebuild'] = TRUE; // along with this
  }
  else {
    $query = <<<XML
select \Object \Title from <#ri>
  where (\Object <dc:title> \Title
        and \Object <fedora-rels-ext:isMemberOf> <info:fedora/$pid>
        and \Object <fedora-model:state> <info:fedora/fedora-system:def/model#Active>)
order by \Title
XML;

    // this is where you do your processing after they have pressed the confirm button
    module_load_include('inc', 'fedora_repository', 'CollectionClass');
    $collection = new CollectionClass($pid);
    $contents = $collection->getRelatedItems($pid, $query);
    $results_xml = simplexml_load_string($contents);
    $resultsarray = array();
    foreach ($results_xml->results->result as $result) {
      $pid_to_delete = str_replace('info:fedora/', '', $result->object->attributes()->uri);

      //parse contents

      $params = array(
        "pid" => $pid_to_delete,
        "logMessage" => "Purged",
        "force" => ""
      );
      try {
        $soapHelper = new ConnectionHelper();
        $client = $soapHelper->getSoapClient(variable_get('fedora_soap_manage_url',
'http://localhost:8080/fedora/services/management?wsdl'));
        $object = $client->__soapCall('purgeObject', array($params));
        unset($form_state['storage']['confirm']);
      } catch (exception $e) {
        if (preg_match('/org\.fedora\.server\.security\.xacml\.pep\.AuthzDeniedException/', $e->getMessage())) {
          drupal_set_message(t('Error: Insufficient permissions to purge object.'), 'error');
        }
        else {
          drupal_set_message(t($e->getMessage()), 'error');
        }
      }
      return;
    }
  }
  if (!empty($form_state['values']['referrer'])) {
    $form_state['redirect'] = $form_state['values']['referrer'];
  }
  elseif (empty($collectionPid) && !empty($_SESSION['fedora_collection']) && $_SESSION['fedora_collection'] !=
$pid) {
    $collectionPid = $_SESSION['fedora_collection'];
  }
  $form_state['redirect'] = "fedora/repository/$collectionPid/";
}
else {
  $form_state['redirect'] = 'fedora/repository/';
}
}
}

function islandora_book_help($path, $arg) {
  switch ($path) {
    case 'admin/help#islandora_book':
      return t(
        '<p>
The Islandora Book Module allows users to create digital representations of books within the Fedora
Repository.
A book object is created using !mods data. A collection of associated page objects may then be created
created. Page objects
are created by ingesting a zipped folder of TIFF images, each representing a distinct page.
Page TIFFs are used to create a series of Fedora datastreams on ingestion. These streams include
derivative images
for display - JPEGs, Thumbnails, and JP2s, and one ore more OCR streams for text display and analysis.
</p>
<p>
<h3>Configuration</h3>
The page datastreams may be created on this server, or on an external processing server with !
microservices.<br/>
To process images from TIFFS on this server check <strong>Create derivative images locally?</strong> on
the admin page. (Link at the bottom of this page).<br />
TO create OCR streams on this server check <strong>Perform OCR on incoming TIFF images?</strong>. <br
/><br />
',

```