

Integrating DokuWiki with Mantis

Disclaimer

This document describes the steps needed to integrate DokuWiki with Mantis. This is just for experimenting with the concept, and it is not a supported feature of Mantis. A formal implementation may be available in future releases.

Integration Functionality

The integration between Mantis and DokuWiki so far achieves the following:

- Customises DokuWiki to honour Mantis authentication (i.e. single sign-on).
- Customises DokuWiki authorisation based on Mantis access levels (i.e. groups).
- Adds a Wiki link on the issue view simple/advanced pages to their corresponding Wiki issue page.
- Adds a Wiki menu option which maps to the corresponding project Wiki page, or general page in case of "All Projects".
- Provides a way for Wiki pages to link to Mantis issues, e.g. `~~Mantis:7075~~`.

Tips

Following are some tips that are of interest to Mantis / DokuWiki integration:

- To define a template for all Wiki pages within a namespace, create `_template.txt` in the namespace folder. All pages created within this namespace will use the contents of the templates as a start point. This will affect the way the Wiki is structure. For example, if all issues within a project have a namespace, then they can all share one template, if there is a namespace per category, then there can be a template per category. The exact structure to be used should be configurable through Mantis configuration.

DockuWiki Steps

Installation

Install DokuWiki [<http://www.dokuwiki.org>] and make sure it is working properly. This involves making sure that you can create pages, edit them, view them, etc.

Configuration

Modify the configuration values in `dokuwiki\conf\dokuwiki.php` to match the values below:

```
#-----
# Add the following lines:
#
define( 'MANTIS_ROOT', 'c:/inetpub/wwwroot/mantisbt/' ); // file path to Mantis, must terminate with /
define( 'MANTIS_URL', 'http://localhost/mantisbt/' );      // url to Mantis, must terminate with /
#
```

```
# Modify the following configuration options to match the values below:
#
$conf['useacl']          = 1;                // Use Access Control Lists to restrict access?
$conf['authtype']        = 'mantis';         // which authentication backend should be used
$conf['defaultgroup']    = 'VIEWER';         // Default groups new Users are added to
```

Authentication (Single Sign-On)

Create `dokuwiki\inc\auth\mantis.class.php` with the code below.

```
<?php
/**
 * Mantis auth backend
 *
 * Uses external Trust mechanism to check against Mantis'
 * user cookie.
 *
 * @author Victor Bector (http://www.futureware.biz)
 */

require_once( MANTIS_ROOT . 'core.php' );

#dbg($GLOBALS);

class auth_mantis extends auth_basic {
/**
 * Constructor.
 *
 * Sets additional capabilities and config strings
 */
function auth_mantis(){
    $this->cando['external'] = true;
}

/**
 * Authenticates the user using Mantis APIs.
 */
function trustExternal($user,$pass,$sticky=false){
    global $USERINFO;
    global $conf;

    if ( auth_is_user_authenticated() ) {
        // okay we're logged in - set the globals
        $USERINFO['pass'] = current_user_get_field( 'password' );
        $USERINFO['name'] = current_user_get_field( 'username' );
        $USERINFO['mail'] = current_user_get_field( 'email' );

        $t_project_name = getNS( getID() );
        $t_project_id = project_get_id_by_name( $t_project_name );
        $t_access_level = access_get_project_level( $t_project_id );
        $t_access_level_string = strtoupper( get_enum_to_string( config_get( 'access_levels_enum_string' ), $t_a

        $USERINFO['grps'] = array( $t_access_level_string );

        $_SERVER['REMOTE_USER'] = $USERINFO['name'];
        $_SESSION[$conf['title']]['auth']['user'] = $USERINFO['name'];
        $_SESSION[$conf['title']]['auth']['info'] = $USERINFO;

        return true;
    }

    // to be sure
    auth_logoff();

    return false;
}

/**
 * Logout from Mantis
 */
function logOff(){
    auth_logout();
}
}
```

Authorisation Configuration

Create dokuwiki\conf\acl.auth.php with the code below.

```
# acl.auth.php
# <?php exit()?>
# Don't modify the lines above
#
# Access Control Lists
#
# Editing this file by hand shouldn't be necessary. Use the ACL
# Manager interface instead.
#
# If your auth backend allows special char like spaces in groups
# or user names you need to urlencode them (only chars <128, leave
# UTF-8 multibyte chars as is)
#
# none      0
# read      1
# edit      2
# create    4
# upload    8
#
*          @VIEWER          1
*          @REPORTER        2
*          @DEVELOPER        8
*          @MANAGER          8
*          @ADMINISTRATOR    8
*          @ALL              0
```

Mantis Syntax Plug-in

This plugin is to be placed in dokuwiki\lib\plugins\mantis\syntax.php.

```
<?php
/**
 * Mantis Plugin: Hyperlinks references to Mantis Issues
 *
 * @license    GPL 2 (http://www.gnu.org/licenses/gpl.html)
 * @author     Victor Bocktor (http://www.futureware.biz)
 */

if(!defined('DOKU_INC')) define('DOKU_INC',realpath(dirname(__FILE__).'../../').'/');
if(!defined('DOKU_PLUGIN')) define('DOKU_PLUGIN',DOKU_INC.'lib/plugins/');
require_once(DOKU_PLUGIN.'syntax.php');

/**
 * A plug-in that hyper links references to Mantis issues.  References
 * to Mantis issues should use the following format: ~~Mantis:123~~.
 *
 * All DokuWiki plugins to extend the parser/rendering mechanism
 * need to inherit from this class
 */
class syntax_plugin_mantis extends DokuWiki_Syntax_Plugin {

    /**
     * return some info
     */
    function getInfo(){
        return array(
            'author' => 'Victor Bocktor',
            'email'  => 'vbctor at users . sourceforge . net',
            'date'   => '2006-05-18',
            'name'   => 'Mantis Issues Plugin',
            'desc'   => 'Support References to Mantis Issues',
            'url'    => 'http://www.futureware.biz',
        );
    }

    /**
     * What kind of syntax are we?
     */
    function getType(){
        return 'substitution'; # typo is intentional
    }

    /**
     * What about paragraphs?
     */
}
```

```

function getPType(){
    return 'normal';
}

/**
 * Where to sort in?
 */
function getSort(){
    return 156;
}

/**
 * Connect pattern to lexer
 */
function connectTo($mode) {
    $this->Lexer->addSpecialPattern('~Mantis:[0-9]+~~', $mode, 'plugin_mantis');
}

/**
 * Handle the match
 */
function handle($match, $state, $pos, &$handler){
    $match = substr( $match, 9, -2 ); // strip "~Mantis:" from start and "~~" from end
    return array( strtolower( $match ) );
}

/**
 * Create output
 */
function render($format, &$renderer, $data) {
    if ( $format == 'xhtml' ) {
        $renderer->externallink( MANTIS_URL . 'view.php?id=' . $data[0], $data[0] );
        return true;
    }
    return false;
}
}
?>

```

Mantis Steps

Configuration

Add the following configuration items to `config_inc.php`. In the proper integration this will be part of `config_defaults_inc.php`.

```

#####
# Wiki Integration
#####

# Wiki Integration Enabled?
$g_wiki_enable = OFF;

# Wiki Engine
$g_wiki_engine = 'dokuwiki';

# Wiki namespace to be used as root for all pages relating to this mantis installation.
$g_wiki_root_namespace = 'mantis';

# URL under which the wiki engine is hosted. Must be on the same server.
$g_wiki_engine_url = $t_protocol . '://' . $t_host . '/%wiki_engine%';

```

wiki_api.php

Create `mantis/core/wiki_api.php` with the following content:

```

<?php
# Mantis - a php based bugtracking system
# Copyright (C) 2000 - 2002 Kenzaburo Ito - kenito@300baud.org
# Copyright (C) 2002 - 2006 Mantis Team - mantisbt-dev@lists.sourceforge.net

```

```

# This program is distributed under the terms and conditions of the GPL
# See the README and LICENSE files for details

# -----
# $Id: $
# -----

require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'helper_api.php' );
require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'utility_api.php' );
require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'database_api.php' );
require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'authentication_api.php' );
require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'gpc_api.php' );
require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'access_api.php' );
require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'project_api.php' );
require_once( dirname( __FILE__ ) . DIRECTORY_SEPARATOR . 'wiki_' . config_get( 'wiki_engine' ) . '_api' );

function wiki_call( $p_function, $p_args_array ) {
    $t_function = 'wiki_' . config_get_global( 'wiki_engine' ) . '_' . $p_function;
    return call_user_func_array( $t_function, $p_args_array );
}

function wiki_is_enabled() {
    return config_get( 'wiki_enable' ) == ON;
}

function wiki_ensure_enabled() {
    if ( !wiki_is_enabled() ) {
        access_denied();
    }
}

function wiki_get_url_for_issue( $p_issue_id ) {
    return wiki_call( 'get_url_for_issue', array( $p_issue_id ) );
}

function wiki_get_url_for_project( $p_project_id ) {
    return wiki_call( 'get_url_for_project', array( $p_project_id ) );
}

function wiki_string_display_links( $p_string ) {
    if ( !wiki_is_enabled() ) {
        return $p_string;
    }

    return wiki_call( 'string_display_links', array( $p_string ) );
}

?>

```

wiki_dokuwiki_api.php

Create mantis/core/wiki_dokuwiki_api.php with the following content:

```

<?php
# Mantis - a php based bugtracking system
# Copyright (C) 2000 - 2002 Kenzaburo Ito - kenito@300baud.org
# Copyright (C) 2002 - 2006 Mantis Team - mantisbt-dev@lists.sourceforge.net
# This program is distributed under the terms and conditions of the GPL
# See the README and LICENSE files for details

# -----
# $Id: $
# -----

function wiki_dokuwiki_get_url_for_page_id( $p_page_id ) {
    $t_root_url = config_get_global( 'wiki_engine_url' );

    $t_root_namespace = config_get( 'wiki_root_namespace' );

    if ( is_blank( $t_root_namespace ) ) {
        $t_page_id = $p_page_id;
    } else {
        $t_page_id = $t_root_namespace . ':' . $p_page_id;
    }

    return $t_root_url . 'doku.php?id=' . urlencode( $t_page_id );
}

function wiki_dokuwiki_get_page_id_for_issue( $p_issue_id ) {
    $c_issue_id = db_prepare_int( $p_issue_id );

```

```

        $t_project_id = bug_get_field( $p_issue_id, 'project_id' );
        $t_project_name = project_get_name( $t_project_id );

        # create a namespace for the project to contain all project documentation.
        # create within it a namespace for issues. This is to allow the creation of a _template.txt
        # file to act as the template for issues belonging to this project.
        return $t_project_name . ':issue:' . $c_issue_id;
    }

    function wiki_dokuwiki_get_url_for_issue( $p_issue_id ) {
        return wiki_dokuwiki_get_url_for_page_id( wiki_dokuwiki_get_page_id_for_issue( $p_issue_id ) );
    }

    function wiki_dokuwiki_get_page_id_for_project( $p_project_id ) {
        $t_home = 'start';
        if ( $p_project_id == ALL_PROJECTS ) {
            return $t_home;
        } else {
            $t_project_name = project_get_name( $p_project_id );
            return $t_project_name . ':' . $t_home;
        }
    }

    function wiki_dokuwiki_get_url_for_project( $p_project_id ) {
        return wiki_dokuwiki_get_url_for_page_id( wiki_dokuwiki_get_page_id_for_project( $p_project_id ) );
    }

    function wiki_dokuwiki_string_display_links( $p_string ) {
        $t_string = $p_string;
        $t_string = str_replace( '[', '{', $p_string );

        $t_wiki_web = config_get_global( 'wiki_engine_url' );

        preg_match_all( '/(^|.+)atewiki(?!<=^|W)' . '\\[\\[ ' . '([a-zA-Z0-9_:]+)\\]\\]/s',
            $p_string, $t_matches, PREG_SET_ORDER );

        $t_result = '';

        foreach ( $t_matches as $t_match ) {
            $t_result .= $t_match[1];

            if ( isset( $t_match[2] ) ) {
                $t_result .= '<a href="' . wiki_dokuwiki_get_url_for_page_id( $t_match[2] ) . ' '
            }
        }

        return $t_result;
    }
}
?>

```

html_api.php

Add the following with the rest of the includes:

```
require_once( $t_core_dir . 'wiki_api.php' );
```

In the `print_menu()` function, add the following code after the code which prints the Docs button (the block with the "Project Documentation Page" comment).

```

# Project Wiki
if ( wiki_is_enabled() ) {
    $t_current_project = helper_get_current_project();
    $t_menu_options[] = '<a href="wiki.php?type=project&id=' . $t_current_project . '">' . lang_get_default
}

```

Add the following function:

```

# -----
# Print a button to create a wiki page
function html_button_wiki( $p_bug_id ) {
    if ( ON == config_get( 'wiki_enable' ) ) {
        if ( access_has_bug_level( config_get( 'update_bug_threshold' ), $p_bug_id ) ) {
            html_button( 'wiki.php',
                lang_get_defaulted( 'Wiki' ),
                array( 'id' => $p_bug_id, 'type' => 'issue' ),

```

```
        'get' );  
    }  
}
```

Update `html_button()` to the following implementation:

```
# -----  
# Print an html button inside a form  
function html_button ( $p_action, $p_button_text, $p_fields = null, $p_method = 'post' ) {  
    $p_action      = urlencode( $p_action );  
    $p_button_text = string_attribute( $p_button_text );  
    if ( null === $p_fields ) {  
        $p_fields = array();  
    }  
  
    if ( strtolower( $p_method ) == 'get' ) {  
        $t_method = 'get';  
    } else {  
        $t_method = 'post';  
    }  
  
    PRINT "<form method=\"{$t_method}\" action=\"{$p_action}\">\n";  
  
    foreach ( $p_fields as $key => $val ) {  
        $key = string_attribute( $key );  
        $val = string_attribute( $val );  
  
        PRINT " <input type=\"hidden\" name=\"{$key}\" value=\"{$val}\" />\n";  
    }  
  
    PRINT " <input type=\"submit\" class=\"button\" value=\"{$p_button_text}\" />\n";  
    PRINT "</form>\n";  
}
```

mantisbt/7075/integration_with_dokuwiki.txt · Last modified: 2006/05/17 11:01 by vbactor