**Development of BuddyPress Component: Steps**

Add these lines to the end of this file (to enable plugin installation from dashboard):

**wordpress/wp-config.php**

if(is\_admin()) {

add\_filter('filesystem\_method', create\_function('$a', 'return "direct";' ));

define( 'FS\_CHMOD\_DIR', 0751 );

}

Add these lines to the end of this file (to disable dashboard access to subscribers):

**wordpress/content/themes/<theme>/functions.php**

/\*

add\_action( 'init', 'blockusers\_init' );

function blockusers\_init()

{

if ( is\_admin() && ! current\_user\_can( 'administrator' ) && ! ( defined( 'DOING\_AJAX' ) && DOING\_AJAX ) )

{ wp\_redirect( home\_url() ); exit;

}

}

\*/

function block\_dashboard() {

$file = basename($\_SERVER['PHP\_SELF']);

if (is\_user\_logged\_in() && is\_admin() && !current\_user\_can('edit\_posts') && $file != 'admin-ajax.php'){

wp\_redirect( home\_url() );

exit();

}

}

add\_action('init', 'block\_dashboard');

Or Install **Adminimize** plugin and disable view website feature for subscriber.

Using PhpMyAdmin, create these tables to “wordpress” database:

CREATE TABLE wp\_bp\_experiments (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

creator\_id bigint(20) NOT NULL,

name varchar(100) NOT NULL,

slug varchar(200) NOT NULL,

description longtext NOT NULL,

status varchar(10) NOT NULL DEFAULT 'public',

enable\_forum tinyint(1) NOT NULL DEFAULT '1',

date\_created datetime NOT NULL,

KEY creator\_id (creator\_id),

KEY status (status)

);

CREATE TABLE wp\_bp\_experiments\_members (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

user\_id bigint(20) NOT NULL,

inviter\_id bigint(20) NOT NULL,

is\_admin tinyint(1) NOT NULL DEFAULT '0',

is\_mod tinyint(1) NOT NULL DEFAULT '0',

user\_title varchar(100) NOT NULL,

date\_modified datetime NOT NULL,

comments longtext NOT NULL,

is\_confirmed tinyint(1) NOT NULL DEFAULT '0',

is\_banned tinyint(1) NOT NULL DEFAULT '0',

invite\_sent tinyint(1) NOT NULL DEFAULT '0',

KEY experiment\_id (experiment\_id),

KEY is\_admin (is\_admin),

KEY is\_mod (is\_mod),

KEY user\_id (user\_id),

KEY inviter\_id (inviter\_id),

KEY is\_confirmed (is\_confirmed)

);

CREATE TABLE wp\_bp\_experiments\_experimentmeta (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

meta\_key varchar(255) DEFAULT NULL,

meta\_value longtext DEFAULT NULL,

KEY experiment\_id (experiment\_id),

KEY meta\_key (meta\_key)

);

CREATE TABLE wp\_bp\_experiments\_variables (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

name varchar(100) NOT NULL,

type varchar(20) NOT NULL DEFAULT 'score',

KEY experiment\_id (experiment\_id)

);

CREATE TABLE wp\_bp\_experiments\_report (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

user\_id bigint(20) NOT NULL,

variable\_id bigint(20) NOT NULL,

variable\_value varchar(200) NOT NULL,

comments varchar(200),

date\_modified datetime NOT NULL,

KEY experiment\_id (experiment\_id),

KEY user\_id (user\_id),

KEY variable\_id (variable\_id),

KEY date\_modified (date\_modified)

);

CREATE TABLE wp\_bp\_xprofile\_experiments (

id bigint(20) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name varchar(150) NOT NULL,

description mediumtext NOT NULL,

experiment\_order bigint(20) NOT NULL DEFAULT '0',

can\_delete tinyint(1) NOT NULL,

KEY can\_delete (can\_delete)

);

Drop this table “wp\_bp\_xprofile\_fields” and create a new one.

CREATE TABLE wp\_bp\_xprofile\_fields (

id bigint(20) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,

group\_id bigint(20) unsigned NOT NULL,

experiment\_id bigint(20) unsigned NOT NULL,

parent\_id bigint(20) unsigned NOT NULL,

type varchar(150) NOT NULL,

name varchar(150) NOT NULL,

description longtext NOT NULL,

is\_required tinyint(1) NOT NULL DEFAULT '0',

is\_default\_option tinyint(1) NOT NULL DEFAULT '0',

field\_order bigint(20) NOT NULL DEFAULT '0',

option\_order bigint(20) NOT NULL DEFAULT '0',

order\_by varchar(15) NOT NULL DEFAULT '',

can\_delete tinyint(1) NOT NULL DEFAULT '1',

KEY group\_id (group\_id),

KEY experiment\_id (experiment\_id),

KEY parent\_id (parent\_id),

KEY field\_order (field\_order),

KEY can\_delete (can\_delete),

KEY is\_required (is\_required)

) ;

Switch to folder **wordpress/wp-content/plugins/buddypress**

**wordpress/wp-content/plugins/buddypress/bp-core/admin/bp-core-components.php**

'experiments' => array(

'title' => \_\_( 'User Experiments', 'buddypress' ),

'description' => \_\_( 'Experiments allow your users to organize themselves into specific public, private or hidden sections with separate activity streams and member listings.', 'buddypress' )

),

**wordpress/wp-content/plugins/buddypress/bp-core/admin/bp-core-schema.php**

function bp\_core\_install( $active\_components = false ) {

….

// Extensible Experiments

if ( !empty( $active\_components['experiments'] ) )

bp\_core\_install\_experiments();

}

function bp\_core\_install\_experiments() {

$sql = array();

$charset\_collate = bp\_core\_set\_charset();

$bp\_prefix = bp\_core\_get\_table\_prefix();

$sql[] = "CREATE TABLE {$bp\_prefix}bp\_experiments (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

creator\_id bigint(20) NOT NULL,

name varchar(100) NOT NULL,

slug varchar(200) NOT NULL,

description longtext NOT NULL,

status varchar(10) NOT NULL DEFAULT 'public',

enable\_forum tinyint(1) NOT NULL DEFAULT '1',

date\_created datetime NOT NULL,

KEY creator\_id (creator\_id),

KEY status (status)

) {$charset\_collate};";

$sql[] = "CREATE TABLE {$bp\_prefix}bp\_experiments\_members (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

user\_id bigint(20) NOT NULL,

inviter\_id bigint(20) NOT NULL,

is\_admin tinyint(1) NOT NULL DEFAULT '0',

is\_mod tinyint(1) NOT NULL DEFAULT '0',

user\_title varchar(100) NOT NULL,

date\_modified datetime NOT NULL,

comments longtext NOT NULL,

is\_confirmed tinyint(1) NOT NULL DEFAULT '0',

is\_banned tinyint(1) NOT NULL DEFAULT '0',

invite\_sent tinyint(1) NOT NULL DEFAULT '0',

KEY experiment\_id (experiment\_id),

KEY is\_admin (is\_admin),

KEY is\_mod (is\_mod),

KEY user\_id (user\_id),

KEY inviter\_id (inviter\_id),

KEY is\_confirmed (is\_confirmed)

) {$charset\_collate};";

$sql[] = "CREATE TABLE {$bp\_prefix}bp\_experiments\_experimentmeta (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

meta\_key varchar(255) DEFAULT NULL,

meta\_value longtext DEFAULT NULL,

KEY experiment\_id (experiment\_id),

KEY meta\_key (meta\_key)

) {$charset\_collate};";

$sql[] = "CREATE TABLE {$bp\_prefix}bp\_experiments\_variables (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

name varchar(100) NOT NULL,

type varchar(20) NOT NULL DEFAULT 'score',

KEY experiment\_id (experiment\_id)

) {$charset\_collate};";

$sql[] = " CREATE TABLE {$bp\_prefix}bp\_experiments\_report (

id bigint(20) NOT NULL AUTO\_INCREMENT PRIMARY KEY,

experiment\_id bigint(20) NOT NULL,

user\_id bigint(20) NOT NULL,

variable\_id bigint(20) NOT NULL,

variable\_value varchar(200) NOT NULL,

comments varchar(200),

date\_modified datetime NOT NULL,

KEY experiment\_id (experiment\_id),

KEY user\_id (user\_id),

KEY variable\_id (variable\_id),

KEY date\_modified (date\_modified)

){$charset\_collate};";

dbDelta( $sql );

}

function bp\_core\_install\_extended\_profiles() {

// These values should only be updated if they are not already present

if ( ! bp\_get\_option( 'bp-xprofile-base-experiment-name' ) ) {

bp\_update\_option( 'bp-xprofile-base-experiment-name', \_x( 'General', 'First field-experiment name', 'buddypress' ) );

}

$sql[] = "CREATE TABLE {$bp\_prefix}bp\_xprofile\_experiments (

id bigint(20) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name varchar(150) NOT NULL,

description mediumtext NOT NULL,

experiment\_order bigint(20) NOT NULL DEFAULT '0',

can\_delete tinyint(1) NOT NULL,

KEY can\_delete (can\_delete)

) {$charset\_collate};";

$sql[] = "CREATE TABLE {$bp\_prefix}bp\_xprofile\_fields (

id bigint(20) unsigned NOT NULL AUTO\_INCREMENT PRIMARY KEY,

group\_id bigint(20) unsigned NOT NULL,

experiment\_id bigint(20) unsigned NOT NULL,

parent\_id bigint(20) unsigned NOT NULL,

type varchar(150) NOT NULL,

name varchar(150) NOT NULL,

description longtext NOT NULL,

is\_required tinyint(1) NOT NULL DEFAULT '0',

is\_default\_option tinyint(1) NOT NULL DEFAULT '0',

field\_order bigint(20) NOT NULL DEFAULT '0',

option\_order bigint(20) NOT NULL DEFAULT '0',

order\_by varchar(15) NOT NULL DEFAULT '',

can\_delete tinyint(1) NOT NULL DEFAULT '1',

KEY group\_id (group\_id),

KEY experiment\_id (experiment\_id),

KEY parent\_id (parent\_id),

KEY field\_order (field\_order),

KEY can\_delete (can\_delete),

KEY is\_required (is\_required)

) {$charset\_collate};";

// Insert the default experiment and fields

$insert\_sql = array();

if ( !$wpdb->get\_var( "SELECT id FROM {$bp\_prefix}bp\_xprofile\_experiments WHERE id = 1" ) )

$insert\_sql[] = "INSERT INTO {$bp\_prefix}bp\_xprofile\_experiments ( name, description, can\_delete ) VALUES ( " . $wpdb->prepare( '%s', stripslashes( bp\_get\_option( 'bp-xprofile-base-experiment-name' ) ) ) . ", '', 0 );";

if ( !$wpdb->get\_var( "SELECT id FROM {$bp\_prefix}bp\_xprofile\_fields WHERE id = 1" ) )

$insert\_sql[] = "INSERT INTO {$bp\_prefix}bp\_xprofile\_fields ( experiment\_id, parent\_id, type, name, description, is\_required, can\_delete ) VALUES ( 1, 0, 'textbox', " . $wpdb->prepare( '%s', stripslashes( bp\_get\_option( 'bp-xprofile-fullname-field-name' ) ) ) . ", '', 1, 0 );";

dbDelta( $insert\_sql );

}//end extended profile

**wordpress/wp-content/plugins/buddypress/bp-core/admin/bp-core-settings.php**

/\*\* Experiments Section \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

\* Experiments settings section description for the settings page

\*

\* @since BuddyPress (1.6)

\*/

function bp\_admin\_setting\_callback\_experiments\_section() { }

/\*\*

\* Allow all users to create experiment field

\*

\* @since BuddyPress (1.6)

\*

\* @uses checked() To display the checked attribute

\*/

function bp\_admin\_setting\_callback\_experiment\_creation() {

?>

<input id="bp\_restrict\_experiment\_creation" name="bp\_restrict\_experiment\_creation" type="checkbox"value="1" <?php checked( !bp\_restrict\_experiment\_creation( false ) ); ?> />

<label for="bp\_restrict\_experiment\_creation"><?php \_e( 'Enable experiment creation for all users', 'buddypress' ); ?></label>

<p class="description"><?php \_e( 'Administrators can always create experiments, regardless of this setting.', 'buddypress' ); ?></p>

<?php

}

function bp\_core\_admin\_settings\_save() {

...

// Some legacy options are not registered with the Settings API

$legacy\_options = array(

'bp-disable-account-deletion',

'bp-disable-avatar-uploads',

'bp\_disable\_blogforum\_comments',

'bp-disable-profile-sync',

'bp\_restrict\_group\_creation',

'bp\_restrict\_experiment\_creation',

'hide-loggedout-adminbar',

);

}

**wordpress/wp-content/plugins/buddypress/bp-core/admin/bp-core-tools.php**

function bp\_admin\_repair\_list() {

. . .

// Experiments:

// - user experiment count

if ( bp\_is\_active( 'experiments' ) ) {

$repair\_list[10] = array(

'bp-experiment-count',

\_\_( 'Count expriments for each user', 'buddypress' ),

'bp\_admin\_repair\_experiment\_count',

);

}

. . .

}

/\*\*

\* Recalculate experiment counts for each user.

\*

\* @since BuddyPress (2.0.0)

\*

\* @return array

\*/

function bp\_admin\_repair\_experiment\_count() {

global $wpdb, $bp;

if ( ! bp\_is\_active( 'experiments' ) ) {

return;

}

$statement = \_\_( 'Counting the number of experiments for each user&hellip; %s', 'buddypress' );

$result = \_\_( 'Failed!', 'buddypress' );

$sql\_delete = "DELETE FROM {$wpdb->usermeta} WHERE meta\_key IN ( 'total\_experiment\_count' );";

if ( is\_wp\_error( $wpdb->query( $sql\_delete ) ) ) {

return array( 1, sprintf( $statement, $result ) );

}

// Walk through all users on the site

$total\_users = $wpdb->get\_row( "SELECT count(ID) as c FROM {$wpdb->users}" )->c;

if ( $total\_users > 0 ) {

$per\_query = 500;

$offset = 0;

while ( $offset < $total\_users ) {

// But only bother to update counts for users that have experiments

$users = $wpdb->get\_col( $wpdb->prepare( "SELECT user\_id FROM wp\_bp\_experiments\_members WHERE is\_confirmed = 1 AND is\_banned = 0 AND user\_id > %d AND user\_id <= %d", $offset, $offset + $per\_query ) );

foreach ( $users as $user ) {

BP\_Experiments\_Member::refresh\_total\_experiment\_count\_for\_user( $user );

}

$offset += $per\_query;

}

} else {

return array( 2, sprintf( $statement, $result ) );

}

return array( 0, sprintf( $statement, \_\_( 'Complete!', 'buddypress' ) ) );

}

**wordpress/wp-content/plugins/buddypress/bp-core/admin/css/common.css**

(optional – for design purposes)

body.branch-3-6 div#icon-buddypress-groups,

body.branch-3-7 div#icon-buddypress-groups {

background: url('../images/icons32.png') no-repeat -490px -6px;

}

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-admin.php**

public function register\_admin\_settings() {

. . .

/\*\* Experiments Section \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if ( bp\_is\_active( 'experiments' ) ) {

// Add the main section

add\_settings\_section( 'bp\_experiments', \_\_( 'Experiments Settings', 'buddypress' ), 'bp\_admin\_setting\_callback\_experiments\_section', 'buddypress' );

// Allow subscriptions setting

add\_settings\_field( 'bp\_restrict\_experiment\_creation', \_\_( 'Experiment Creation', 'buddypress' ), 'bp\_admin\_setting\_callback\_experiment\_creation', 'buddypress', 'bp\_experiments' );

register\_setting ( 'buddypress', 'bp\_restrict\_experiment\_creation', 'intval' );

}

}//end function

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-adminbar.php**

function bp\_admin\_bar\_my\_account\_root() {

// Only add menu for logged in user

if ( is\_user\_logged\_in() ) {

// Add secondary parent item for all BuddyPress components

$wp\_admin\_bar->add\_menu( array(

'parent' => 'my-account',

'id' => 'my-account-buddypress',

'title' => \_\_( 'My Account' ),

'group' => true,

**'experiment' => true,**

'meta' => array(

'class' => 'ab-sub-secondary'

)

) );

}

}//end function

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-avatars.php**

function bp\_core\_fetch\_avatar( $args = '' ) {

. . .

if ( empty( $item\_id ) ) {

switch ( $object ) {

...

case 'experiment' :

if ( bp\_is\_active( 'experiments' ) ) {

$item\_id = $bp->experiments->current\_experiment->id;

} else {

$item\_id = false;

}

break;

}//end switch

}//end if

if ( empty( $avatar\_dir ) ) {

switch ( $object ) {

...

case 'experiment' :

if ( bp\_is\_active( 'experiments' ) ) {

$avatar\_dir = 'experiment-avatars';

} else {

$avatar\_dir = false;

}

break;

}//end switch

}//end if

/\*\* <img> alt \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if ( false !== strpos( $alt, '%s' ) || false !== strpos( $alt, '%1$s' ) ) {

switch ( $object ) {

. . .

case 'experiment' :

$item\_name = bp\_get\_experiment\_name( experiments\_get\_experiment( array( 'experiment\_id' => $item\_id ) ) );

break;

...

}

}//end function

/\*\*

\* Look for uploaded avatar first. Use it if it exists.

\* Set the file names to search for, to select the full size

\* or thumbnail image.

\*/

$legacy\_experiment\_avatar\_name = ( 'full' == $type ) ? '-experimentavatar-full' : '-experimentavatar-thumb';

// Legacy group & experiment avatar check

if ( !isset( $avatar\_url ) ) {

foreach( $avatar\_files as $key => $value )

{

if ( strpos ( $value, $legacy\_group\_avatar\_name )!== false ) {

$avatar\_url = $avatar\_folder\_url . '/' . $avatar\_files[$key];

}//end if

if ( strpos ( $value, $legacy\_experiment\_avatar\_name )!== false ) {

$avatar\_url = $avatar\_folder\_url . '/' . $avatar\_files[$key];

}//end if

}//end for

}//end if Legacy group & experiment avatar check

// Set gravatar object

if ( empty( $email ) ) {

if ( 'user' == $object ) {

$email = bp\_core\_get\_user\_email( $item\_id );

} else if ( 'group' == $object || 'blog' == $object || 'experiment' == $object ) {

$email = "{$item\_id}-{$object}@{bp\_get\_root\_domain()}";

}

}

function bp\_core\_delete\_existing\_avatar( $args = '' ) {

. . .

if ( empty( $item\_id ) ) {

if ( 'user' == $object )

$item\_id = bp\_displayed\_user\_id();

else if ( 'group' == $object )

$item\_id = buddypress()->groups->current\_group->id;

else if ( 'experiment' == $object )

$item\_id = buddypress()->experiments->current\_experiment->id;

if ( !$item\_id ) return false;

}

if ( empty( $avatar\_dir ) ) {

if ( 'user' == $object )

$avatar\_dir = 'avatars';

else if ( 'group' == $object )

$avatar\_dir = 'group-avatars';

else if ( 'experiment' == $object )

$avatar\_dir = 'experiment-avatars';

}

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-buddybar.php**

function bp\_adminbar\_random\_menu() {

. . .

<?php if ( bp\_is\_active( 'experiments' ) ) : ?>

<li class="alt"><a href="<?php echo trailingslashit( bp\_get\_root\_domain() . '/' . bp\_get\_experiments\_root\_slug() ) . '?random-experiment' ?>" rel="nofollow"><?php \_e( 'Random Experiment', 'buddypress' ) ?></a></li>

<?php endif; ?>

}//end function

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-cache.php**

/\*\*

\* Add 'bp' to global experiment of network wide cachable objects.

\*/

function bp\_core\_add\_global\_experiment() {

if ( function\_exists( 'wp\_cache\_add\_global\_experiments' ) ) {

wp\_cache\_add\_global\_experiments( array( 'bp' ) );

}

}

add\_action( 'bp\_loaded', 'bp\_core\_add\_global\_experiment' );

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-classes.php**

/\*\*

\* The total number of experiments the user is a part of.

\*

\* Example: "1 experiment", "2 experiments"

\*

\* @var string

\*/

public $total\_experiments;

/\*\*

\* Populates extra fields such as group and friendship counts.

\*/

public function populate\_extras() {

. . .

if ( bp\_is\_active( 'experiments' ) ) {

$this->total\_experiments = BP\_Experiments\_Member::total\_experiment\_count( $this->id );

$this->total\_experiments = sprintf( \_n( '%d experiment', '%d experiments', $this->total\_experiments, 'buddypress' ), $this->total\_experiments );

}

}

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-filters.php**

// A single experiment

} elseif ( bp\_is\_active( 'experiments' ) && !empty( $bp->experiments->current\_experiment ) && !empty( $bp->bp\_options\_nav[$bp->experiments->current\_experiment->slug] ) ) {

$subnav = isset( $bp->bp\_options\_nav[$bp->experiments->current\_experiment->slug][bp\_current\_action()]['name'] ) ? $bp->bp\_options\_nav[$bp->experiments->current\_experiment->slug][bp\_current\_action()]['name'] : '';

// translators: "experiment name | experiment nav section name"

$title = sprintf( \_\_( '%1$s | %2$s', 'buddypress' ), $bp->bp\_options\_title, $subnav );

// Experiment creation page

} elseif ( bp\_is\_experiment\_create() ) {

$title = \_\_( 'Create an Experiment', 'buddypress' );

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-functions.php**

$page\_titles = array(

. . .

'experiments' => \_x( 'Experiments', 'Page title for the Experiments directory.', 'buddypress' ),

. . .

);

if ( empty( $slug ) ) {

switch ( $search\_which ) {

...

case 'experiments':

$slug = bp\_is\_active( 'experiments' ) ? bp\_get\_experiments\_root\_slug() : '';

break;

…

}

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-loader.php**

private function bootstrap() {

. . .

// Set the included and optional components.

$bp->optional\_components = apply\_filters( 'bp\_optional\_components', array( 'activity', 'blogs', 'forums', 'friends', 'groups', 'experiments','messages', 'notifications', 'settings', 'xprofile' ) );

}//end function

/\*\* Avatars \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Fetches the default Gravatar image to use if the user/group/blog has no avatar or gravatar

. . .

$bp->grav\_default->experiment = apply\_filters( 'bp\_experiment\_gravatar\_default', $bp->grav\_default->user );

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-options.php**

function bp\_get\_default\_options() {

// Default options

$options = array (

/\*\* Components \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Base profile experiments name

'bp-xprofile-base-experiment-name' => 'BaseExperiment',

/\*\* Experiments \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// @todo Move this into the experiments component

// Restrict experiment creation to super admins

'bp\_restrict\_experiment\_creation' => false,

'widget\_bp\_experiments\_widget' => false,

}//end function

/\*\*

\* Is experiment creation turned off?

\*

\* @since BuddyPress (1.6.0)

\*

\* @todo Move into experiments component.

\* @uses bp\_get\_option() To get the experiment creation.

\*

\* @param bool $default Optional. Fallback value if not found in the database.

\* Default: true.

\* @return bool True if experiment creation is restricted, otherwise false.

\*/

function bp\_restrict\_experiment\_creation( $default = true ) {

return (bool) apply\_filters( 'bp\_restrict\_experiment\_creation', (bool) bp\_get\_option( 'bp\_restrict\_experiment\_creation', $default ) );

}

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-template.php**

function bp\_get\_options\_nav() {

. . .

// List type depends on our current component

$list\_type = bp\_is\_experiment() ? 'experiments' : 'personal';

. . .

}

function bp\_search\_form\_type\_select() {

. . .

if ( bp\_is\_active( 'experiments' ) )

$options['experiments'] = \_\_( 'Experiments', 'buddypress' );

}//end function

/\*\*

\* Check whether the current page is part of the Experiments component.

\*

\* @return bool True if the current page is part of the Experiments component.

\*/

function bp\_is\_experiments\_component() {

if ( bp\_is\_current\_component( 'experiments' ) )

return true;

return false;

}

/\*\*

\* Is the current page a user's Experiments activity stream?

\*

\* Eg <http://example.com/members/joe/experiments/>

\*

\* @return True if the current page is a user's Experiments activity stream.

\*/

function bp\_is\_user\_experiments\_activity() {

if ( !bp\_is\_active( 'experiments' ) )

return false;

$slug = bp\_get\_experiments\_slug();

if ( empty( $slug ) )

$slug = 'experiments';

if ( bp\_is\_user\_activity() && bp\_is\_current\_action( $slug ) )

return true;

return false;

}

/\*\*

\* Is the current page part of a user's Experiments page?

\*

\* Eg <http://example.com/members/joe/experiments/> (or a subpage thereof).

\*

\* @return bool True if the current page is a user's Experiments page.

\*/

function bp\_is\_user\_experiments() {

if ( bp\_is\_user() && bp\_is\_experiments\_component() )

return true;

return false;

}

/\*\* Experiments \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

\* Is the current page the experiments directory ?

\*

\* @since BuddyPress (2.0.0)

\*

\* @return True if the current page is the experiments directory.

\*/

function bp\_is\_experiments\_directory() {

if ( bp\_is\_experiments\_component() && ! bp\_current\_action() && ! bp\_current\_item() )

return true;

return false;

}

/\*\*

\* Does the current page belong to a single experiment?

\*

\* Will return true for any subpage of a single experiment.

\*

\* @return bool True if the current page is part of a single experiment.

\*/

function bp\_is\_experiment() {

global $bp;

if ( bp\_is\_experiments\_component() && isset( $bp->experiments->current\_experiment ) && $bp->experiments->current\_experiment )

return true;

return false;

}

/\*\*

\* Is the current page a single experiment's home page?

\*

\* URL will vary depending on which experiment tab is set to be the "home". By

\* default, it's the experiment's recent activity.

\*

\* @return bool True if the current page is a single experiment's home page.

\*/

function bp\_is\_experiment\_home() {

if ( bp\_is\_single\_item() && bp\_is\_experiments\_component() && ( !bp\_current\_action() || bp\_is\_current\_action( 'home' ) ) )

return true;

return false;

}

/\*\*

\* Is the current page part of the experiment creation process?

\*

\* @return bool True if the current page is part of the experiment creation process.

\*/

function bp\_is\_experiment\_create() {

if ( bp\_is\_experiments\_component() && bp\_is\_current\_action( 'create' ) )

return true;

return false;

}

/\*\*

\* Is the current page part of a single experiment's admin screens?

\*

\* Eg [http://example.com/experiments/myexperiment/admin/settings/.](http://example.com/groups/mygroup/admin/settings/.)

\*

\* @return bool True if the current page is part of a single experiment's admin.

\*/

function bp\_is\_experiment\_admin\_page() {

if ( bp\_is\_single\_item() && bp\_is\_experiments\_component() && bp\_is\_current\_action( 'admin' ) )

return true;

return false;

}

/\*\*

\* Is the current page a experiment's forum page?

\*

\* Only applies to legacy bbPress forums.

\*

\* @return bool True if the current page is a experiment forum page.

\*/

function bp\_is\_experiment\_forum() {

$retval = false;

// At a forum URL

if ( bp\_is\_single\_item() && bp\_is\_experiments\_component() && bp\_is\_current\_action( 'forum' ) ) {

$retval = true;

// If at a forum URL, set back to false if forums are inactive, or not

// installed correctly.

if ( ! bp\_is\_active( 'forums' ) || ! bp\_forums\_is\_installed\_correctly() ) {

$retval = false;

}

}

return $retval;

}

/\*\*

\* Is the current page a experiment's activity page?

\*

\* @return True if the current page is a experiment's activity page.

\*/

function bp\_is\_experiment\_activity() {

if ( bp\_is\_single\_item() && bp\_is\_experiments\_component() && bp\_is\_current\_action( 'activity' ) )

return true;

return false;

}

/\*\*

\* Is the current page a experiment forum topic?

\*

\* Only applies to legacy bbPress (1.x) forums.

\*

\* @return bool True if the current page is part of a experiment forum topic.

\*/

function bp\_is\_experiment\_forum\_topic() {

if ( bp\_is\_single\_item() && bp\_is\_experiments\_component() && bp\_is\_current\_action( 'forum' ) && bp\_is\_action\_variable( 'topic', 0 ) )

return true;

return false;

}

/\*\*

\* Is the current page a experiment forum topic edit page?

\*

\* Only applies to legacy bbPress (1.x) forums.

\*

\* @return bool True if the current page is part of a experiment forum topic edit page.

\*/

function bp\_is\_experiment\_forum\_topic\_edit() {

if ( bp\_is\_single\_item() && bp\_is\_experiments\_component() && bp\_is\_current\_action( 'forum' ) && bp\_is\_action\_variable( 'topic', 0 ) && bp\_is\_action\_variable( 'edit', 2 ) )

return true;

return false;

}

/\*\*

\* Is the current page a experiment's Members page?

\*

\* Eg [http://example.com/experiments/myexperiment/members/.](http://example.com/groups/mygroup/members/.)

\*

\* @return bool True if the current page is part of a experiment's Members page.

\*/

function bp\_is\_experiment\_members() {

if ( bp\_is\_single\_item() && bp\_is\_experiments\_component() && bp\_is\_current\_action( 'members' ) )

return true;

return false;

}

/\*\*

\* Is the current page a experiment's Invites page?

\*

\* Eg [http://example.com/experiments/myexperiment/send-invites/.](http://example.com/groups/mygroup/send-invites/.)

\*

\* @return bool True if the current page is a experiment's Send Invites page.

\*/

function bp\_is\_experiment\_invites() {

if ( bp\_is\_experiments\_component() && bp\_is\_current\_action( 'send-invites' ) )

return true;

return false;

}

/\*\*

\* Is the current page a experiment's Request Membership page?

\*

\* Eg [http://example.com/experiments/myexperiment/request-membership/.](http://example.com/groups/mygroup/request-membership/.)

\*

\* @return bool True if the current page is a experiment's Request Membership page.

\*/

function bp\_is\_experiment\_membership\_request() {

if ( bp\_is\_experiments\_component() && bp\_is\_current\_action( 'request-membership' ) )

return true;

return false;

}

/\*\*

\* Is the current page a leave experiment attempt?

\*

\* @return bool True if the current page is a Leave Experiment attempt.

\*/

function bp\_is\_experiment\_leave() {

if ( bp\_is\_experiments\_component() && bp\_is\_single\_item() && bp\_is\_current\_action( 'leave-experiment' ) )

return true;

return false;

}

/\*\*

\* Is the current page part of a single experiment?

\*

\* Not currently used by BuddyPress.

\*

\* @todo How is this functionally different from bp\_is\_experiment()?

\*

\* @return bool True if the current page is part of a single experiment.

\*/

function bp\_is\_experiment\_single() {

if ( bp\_is\_experiments\_component() && bp\_is\_single\_item() )

return true;

return false;

}

function bp\_get\_the\_body\_class( $wp\_classes = array(), $custom\_classes = false ) {

. . .

/\*\* Components \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if ( !bp\_is\_blog\_page() ) :

. . .

if ( bp\_is\_experiments\_component() )

$bp\_classes[] = 'experiments';

. . .

endif;

/\*\* User \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

. . .

if ( bp\_is\_user\_experiments() )

$bp\_classes[] = 'my-experiments';

. . .

endif;

if ( bp\_is\_user\_experiments\_activity() )

$bp\_classes[] = 'experiments-activity';

. . .

/\*\* Experiments \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if ( bp\_is\_experiment\_leave() )

$bp\_classes[] = 'leave-experiment';

if ( bp\_is\_experiment\_invites() )

$bp\_classes[] = 'experiment-invites';

if ( bp\_is\_experiment\_members() )

$bp\_classes[] = 'experiment-members';

if ( bp\_is\_experiment\_forum\_topic() )

$bp\_classes[] = 'experiment-forum-topic';

if ( bp\_is\_experiment\_forum\_topic\_edit() )

$bp\_classes[] = 'experiment-forum-topic-edit';

if ( bp\_is\_experiment\_forum() )

$bp\_classes[] = 'experiment-forum';

if ( bp\_is\_experiment\_admin\_page() ) {

$bp\_classes[] = 'experiment-admin';

$bp\_classes[] = bp\_get\_experiment\_current\_admin\_tab();

}

if ( bp\_is\_experiment\_create() ) {

$bp\_classes[] = 'experiment-create';

$bp\_classes[] = bp\_get\_experiments\_current\_create\_step();

}

if ( bp\_is\_experiment\_home() )

$bp\_classes[] = 'experiment-home';

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-update.php**

function bp\_update\_to\_1\_5() {

. . .

delete\_site\_option( 'bp-experiments-db-version' );

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-core/bp-core-widgets.php**

<p>

<label for="bp-core-widget-experiments-default"><?php \_e('Default members to show:', 'buddypress'); ?>

<select name="<?php echo $this->get\_field\_name( 'member\_default' ) ?>">

<option value="newest" <?php if ( $member\_default == 'newest' ) : ?>selected="selected"<?php endif; ?>><?php \_e( 'Newest', 'buddypress' ) ?></option>

<option value="active" <?php if ( $member\_default == 'active' ) : ?>selected="selected"<?php endif; ?>><?php \_e( 'Active', 'buddypress' ) ?></option>

<option value="popular" <?php if ( $member\_default == 'popular' ) : ?>selected="selected"<?php endif; ?>><?php \_e( 'Popular', 'buddypress' ) ?></option>

</select>

</label>

</p>

**wordpress/wp-content/plugins/buddypress/bp-core/css/admin-bar-rtl.css**

#wpadminbar .quicklinks li#wp-admin-bar-experiment-admin-with-avatar > a img {

margin: -2px -10px 0 23px;

}

#wpadminbar .quicklinks li#wp-admin-bar-experiment-admin-with-avatar ul {

left: auto;

right: 40px;

}

#wpadminbar .quicklinks li#wp-admin-bar-experiment-admin-with-avatar ul ul {

left: auto;

right: 0;

}

#wpadminbar .quicklinks li#wp-admin-bar-experiment-admin-with-avatar ul {

left: auto;

right: 0;

}

#wpadminbar .quicklinks li#wp-admin-bar-experiment-admin-with-avatar ul ul {

left: auto;

right: 0;

}

**wordpress/wp-content/plugins/buddypress/bp-themes/bp-default/functions.php**

// Experiment buttons

if ( bp\_is\_active( 'experiments' ) ) {

add\_action( 'bp\_experiment\_header\_actions', 'bp\_experiment\_join\_button', 5 );

add\_action( 'bp\_experiment\_header\_actions', 'bp\_experiment\_new\_topic\_button', 20 );

add\_action( 'bp\_directory\_experiments\_actions', 'bp\_experiment\_join\_button' );

}

function bp\_dtheme\_activity\_secondary\_avatars( $action, $activity ) {

switch ( $activity->component ) {

case 'groups' :

case 'experiments' :

case 'friends' :

// Only insert avatar if one exists

if ( $secondary\_avatar = bp\_get\_activity\_secondary\_avatar() ) {

$reverse\_content = strrev( $action );

$position = strpos( $reverse\_content, 'a<' );

$action = substr\_replace( $action, $secondary\_avatar, -$position - 2, 0 );

}

break;

}

return $action;

}

**wordpress/wp-content/plugins/buddypress/bp-themes/bp-default/experiments (Optional)**

-Create a new folder and sub-folders – similar to groups.

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress/experiments**

-Create a new folder and sub-folders – similar to groups.

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress-functions.php**

protected function setup\_actions() {

. . .

// Experiment buttons

if ( bp\_is\_active( 'experiments' ) ) {

add\_action( 'bp\_experiment\_header\_actions', 'bp\_experiment\_join\_button', 5 );

add\_action( 'bp\_experiment\_header\_actions', 'bp\_experiment\_new\_topic\_button', 20 );

add\_action( 'bp\_directory\_experiments\_actions', 'bp\_experiment\_join\_button' );

}

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-activity/bp-activity-actions.php**

function bp\_activity\_action\_permalink\_router() {

. . .

// Redirect based on the type of activity

} else if (bp\_is\_active( 'experiments' ) && $activity->component == buddypress()->experiments->id ) {

// Activity is a user update

if ( ! empty( $activity->user\_id ) ) {

$redirect = bp\_core\_get\_user\_domain( $activity->user\_id, $activity->user\_nicename, $activity->user\_login ) . bp\_get\_activity\_slug() . '/' . $activity->id . '/';

// Activity is something else

} else {

// Set redirect to experiment activity stream

if ( $experiment = experiments\_get\_experiment( array( 'experiment\_id' => $activity->item\_id ) ) ) {

$redirect = bp\_get\_experiment\_permalink( $experiment ) . bp\_get\_activity\_slug() . '/' . $activity->id . '/';

}

}

// Set redirect to users' activity stream

}

. . .

}

function bp\_activity\_action\_post\_update() {

. . .

// Post to experiments object

} else if ( 'experiments' == $object && bp\_is\_active( 'experiments' ) ) {

if ( (int) $item\_id ) {

$activity\_id = experiments\_post\_update( array( 'content' => $content, 'experiment\_id' => $item\_id ) );

}

. . .

}

function bp\_activity\_action\_sitewide\_feed() {

. . .

if ( ! bp\_is\_activity\_component() || ! bp\_is\_current\_action( 'feed' ) || bp\_is\_user() || ! empty( $bp->groups->current\_group ) || ! empty( $bp->groups->current\_experiment ))

return false;

. . .

}

/\*\*

\* Load the activity feed for a user's experiments.

\*

\* @since BuddyPress (1.2)

\*

\* @uses bp\_is\_active()

\* @uses bp\_is\_user\_activity()

\* @uses bp\_is\_current\_action()

\* @uses bp\_get\_experiments\_slug()

\* @uses bp\_is\_action\_variable()

\* @uses status\_header()

\*

\* @return bool False on failure.

\*/

function bp\_activity\_action\_my\_experiments\_feed() {

if ( ! bp\_is\_active( 'experiments' ) || ! bp\_is\_user\_activity() || ! bp\_is\_current\_action( bp\_get\_experiments\_slug() ) || ! bp\_is\_action\_variable( 'feed', 0 ) ) {

return false;

}

// get displayed user's experiment IDs

$experiments = experiments\_get\_user\_experiments();

$experiment\_ids = implode( ',', $experiments['experiments'] );

// setup the feed

buddypress()->activity->feed = new BP\_Activity\_Feed( array(

'id' => 'myexperiments',

/\* translators: Member experiments activity RSS title - "[Site Name] | [User Display Name] | Experiments Activity" \*/

'title' => sprintf( \_\_( '%1$s | %2$s | Experiment Activity', 'buddypress' ), bp\_get\_site\_name(), bp\_get\_displayed\_user\_fullname() ),

'link' => trailingslashit( bp\_displayed\_user\_domain() . bp\_get\_activity\_slug() . '/' . bp\_get\_experiments\_slug() ),

'description' => sprintf( \_\_( "Public experiment activity feed of which %s is a member of.", 'buddypress' ), bp\_get\_displayed\_user\_fullname() ),

'activity\_args' => array(

'object' => buddypress()->experiments->id,

'primary\_id' => $experiment\_ids,

'display\_comments' => 'threaded'

)

) );

}

add\_action( 'bp\_actions', 'bp\_activity\_action\_my\_experiments\_feed' );

**wordpress/wp-content/plugins/buddypress/bp-activity/bp-activity-template.php**

function bp\_has\_activities( $args = '' ) {

**. . .**

// Experiment filtering

if ( !empty( $bp->experiments->current\_experiment ) ) {

$object = $bp->experiments->id;

$primary\_id = $bp->experiments->current\_experiment->id;

if ( ( 'public' != $bp->experiments->current\_experiment->status ) && ( experiments\_is\_user\_member( bp\_loggedin\_user\_id(), $bp->experiments->current\_experiment->id ) || bp\_current\_user\_can( 'bp\_moderate' ) ) )

$show\_hidden = true;

}

**. . .**

// If you have passed a "scope" then this will override any filters you have passed.

if ( 'just-me' == $scope || 'friends' == $scope || 'groups' == $scope || 'experiments' == $scope || 'favorites' == $scope || 'mentions' == $scope ) {

…

case 'experiments':

if ( bp\_is\_active( 'experiments' ) ) {

$experiments = experiments\_get\_user\_experiments( $user\_id );

if ( empty( $experiments['experiments'] ) )

return false;

$object = $bp->experiments->id;

$primary\_id = implode( ',', (array) $experiments['experiments'] );

$user\_id = 0;

}

break;

…

}

**}**

switch ( $activities\_template->activity->component ) {

. . .

case 'experiments' :

$object = 'experiment';

$item\_id = $activities\_template->activity->item\_id;

$link = '';

$name = '';

// Only if experiments is active

if ( bp\_is\_active( 'experiments' ) ) {

$experiment = experiments\_get\_experiment( array( 'experiment\_id' => $item\_id ) );

$link = bp\_get\_experiment\_permalink( $experiment );

$name = $experiment->name;

}

if ( empty( $alt ) ) {

$alt = \_\_( 'Experiment logo', 'buddypress' );

if ( ! empty( $name ) ) {

$alt = sprintf( \_\_( 'Experiment logo of %s', 'buddypress' ), $name );

}

}

break;

. . .

}

function bp\_get\_activity\_css\_class() {

global $activities\_template;

$mini\_activity\_actions = apply\_filters( 'bp\_activity\_mini\_activity\_types', array(

'friendship\_accepted',

'friendship\_created',

'new\_blog',

'joined\_group',

'created\_group',

'joined\_experiment',

'created\_experiment',

'new\_member'

) );

. . .

}

function bp\_get\_member\_activity\_feed\_link() {

…

elseif ( bp\_is\_active( 'experiments' ) && bp\_is\_current\_action( bp\_get\_experiments\_slug() ) )

$link = bp\_displayed\_user\_domain() . bp\_get\_activity\_slug() . '/' . bp\_get\_experiments\_slug() . '/feed/';

…

}

**wordpress/wp-content/plugins/buddypress/bp-activity/bp-activity-screens.php**

/\*\*

\* Load the 'My Experiments' activity page.

\*

\* @since BuddyPress (1.2)

\*

\* @uses bp\_is\_active()

\* @uses bp\_update\_is\_item\_admin()

\* @uses bp\_current\_user\_can()

\* @uses do\_action() To call the 'bp\_activity\_screen\_experiments' hook

\* @uses bp\_core\_load\_template()

\* @uses apply\_filters() To call the 'bp\_activity\_template\_experiments\_activity' hook

\*/

function bp\_activity\_screen\_experiments() {

if ( !bp\_is\_active( 'experiments' ) )

return false;

bp\_update\_is\_item\_admin( bp\_current\_user\_can( 'bp\_moderate' ), 'activity' );

do\_action( 'bp\_activity\_screen\_experiments' );

bp\_core\_load\_template( apply\_filters( 'bp\_activity\_template\_experiments\_activity', 'members/single/home' ) );

}

function bp\_activity\_screen\_single\_activity\_permalink() {

. . .

// If activity is from an experiment, do an extra cap check

if ( isset( $bp->experiments->id ) && $activity->component == $bp->experiments->id ) {

// Activity is from an experiment, but experiments is currently disabled

if ( !bp\_is\_active( 'experiments') ) {

bp\_do\_404();

return;

}

// Check to see if the experiment is not public, if so, check the

// user has access to see this activity

if ( $experiment = experiments\_get\_experiment( array( 'experiment\_id' => $activity->item\_id ) ) ) {

// Experiment is not public

if ( 'public' != $experiment->status ) {

// User is not a member of experiment

if ( !experiments\_is\_user\_member( bp\_loggedin\_user\_id(), $experiment->id ) ) {

$has\_access = false;

}

}

}

}

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-activity/bp-activity-loader.php**

// Additional menu if experiments is active

if ( bp\_is\_active( 'experiments' ) ) {

$sub\_nav[] = array(

'name' => \_\_( 'Experiments', 'buddypress' ),

'slug' => bp\_get\_experiments\_slug(),

'parent\_url' => $activity\_link,

'parent\_slug' => $this->slug,

'screen\_function' => 'bp\_activity\_screen\_experiments',

'position' => 50,

'item\_css\_id' => 'activity-experiments'

);

}

public function setup\_admin\_bar( $wp\_admin\_nav = array() ) {

. . .

// Experiments?

if ( bp\_is\_active( 'experiments' ) ) {

$wp\_admin\_nav[] = array(

'parent' => 'my-account-' . $this->id,

'id' => 'my-account-' . $this->id . '-experiments',

'title' => \_\_( 'Experiments', 'buddypress' ),

'href' => trailingslashit( $activity\_link . bp\_get\_experiments\_slug() )

);

}

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress/activity/index.php**

<?php do\_action( 'bp\_before\_activity\_type\_tab\_experiments' ); ?>

<?php if ( bp\_is\_active( 'experiments' ) ) : ?>

<?php if ( bp\_get\_total\_experiment\_count\_for\_user( bp\_loggedin\_user\_id() ) ) : ?>

<li id="activity-experiments"><a href="<?php echo bp\_loggedin\_user\_domain() . bp\_get\_activity\_slug() . '/' . bp\_get\_experiments\_slug() . '/'; ?>" title="<?php \_e( 'The activity of experiments I am a member of.', 'buddypress' ); ?>"><?php printf( \_\_( 'My Experiments <span>%s</span>', 'buddypress' ), bp\_get\_total\_experiment\_count\_for\_user( bp\_loggedin\_user\_id() ) ); ?></a></li>

<?php endif; ?>

<?php endif; ?>

<?php if ( bp\_is\_active( 'experiments' ) ) : ?>

<option value="created\_experiment"><?php \_e( 'New Experiments', 'buddypress' ); ?></option>

<option value="joined\_experiment"><?php \_e( 'Experiment Memberships', 'buddypress' ); ?></option>

<?php endif; ?>

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress/activity/post-form.php**

<p class="activity-greeting"><?php if ( bp\_is\_group() )

printf( \_\_( "What's new in %s, %s?", 'buddypress' ), bp\_get\_group\_name(), bp\_get\_user\_firstname() );

else if ( bp\_is\_experiment() )

printf( \_\_( "What's new in %s, %s?", 'buddypress' ), bp\_get\_experiment\_name(), bp\_get\_user\_firstname() );

else

printf( \_\_( "What's new, %s?", 'buddypress' ), bp\_get\_user\_firstname() );

?></p>

<?php if ( bp\_is\_active( 'experiments' ) && !bp\_is\_my\_profile() && !bp\_is\_experiment() ) : ?>

<div id="whats-new-post-in-box">

<?php \_e( 'Post in', 'buddypress' ); ?>:

<select id="whats-new-post-in" name="whats-new-post-in">

<option selected="selected" value="0"><?php \_e( 'My Profile', 'buddypress' ); ?></option>

<?php if ( bp\_has\_experiments( 'user\_id=' . bp\_loggedin\_user\_id() . '&type=alphabetical&max=100&per\_page=100&populate\_extras=0' ) ) :

while ( bp\_experiments() ) : bp\_the\_experiment(); ?>

<option value="<?php bp\_experiment\_id(); ?>"><?php bp\_experiment\_name(); ?></option>

<?php endwhile;

endif; ?>

</select>

</div>

<input type="hidden" id="whats-new-post-object" name="whats-new-post-object" value="experiments" />

<?php elseif ( bp\_is\_experiment\_home() ) : ?>

<input type="hidden" id="whats-new-post-object" name="whats-new-post-object" value="experiments" />

<input type="hidden" id="whats-new-post-in" name="whats-new-post-in" value="<?php bp\_experiment\_id(); ?>" />

<?php endif; ?>

**Modify steps for experiments’ variables creation**

Modify this file:

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments-classes.php**

/\*\*

\* Total count of experiment variables.

\*

\* @access public

\* @var int

\*/

public $total\_variable\_count;

/\*\*

\* Is the current factor a variable of this experiment?

\*

\* @since BuddyPress (1.2.0)

\* @var bool

\*/

public $is\_variable;

/\*\*

\* BuddyPress experiment Variable objects.

\*/

class BP\_Experiments\_Variable {

/\*\*

\* ID of the variable.

\*

\* @access public

\* @var string

\*/

var $id;

/\*\*

\* Name of the variable.

\*

\* @access public

\* @var string

\*/

var $name;

/\*\*

\* Type of the variable.

\*

\* @access public

\* @var string

\*/

var $type;

/\*\*

\* ID of the experiment associated with the variable.

\*

\* @access public

\* @var int

\*/

var $experiment\_id;

/\*\*

\* Constructor method.

\*

\* @param int $experiment\_id Optional. Can be used to

\* look up a variable.

\* @param int $id Optional. The unique ID of the variable object.

\* @param bool $populate Whether to populate the properties of the

\* located variable. Default: true.

\*/

public function \_\_construct($experiment\_id = 0, $id = false, $populate = true ) {

// Experiment ID is not empty, and ID is

if ( !empty( $experiment\_id ) && empty( $id ) ) {

$this->experiment\_id = $experiment\_id;

if ( !empty( $populate ) ) {

$this->populate();

}

}

// ID is not empty

if ( !empty( $id ) ) {

$this->id = $id;

if ( !empty( $populate ) ) {

$this->populate();

}

}

}

/\*\*

\* Populate the object's properties.

\*/

public function populate() {

global $wpdb, $bp;

if ( $this->experiment\_id && !$this->id )

$sql = $wpdb->prepare( "SELECT \* FROM wp\_bp\_experiments\_variables WHERE id = %d AND experiment\_id = %d", $this->id, $this->experiment\_id );

if ( !empty( $this->id ) )

$sql = $wpdb->prepare( "SELECT \* FROM wp\_bp\_experiments\_variables WHERE id = %d", $this->id );

$variable = $wpdb->get\_row($sql);

if ( !empty( $variable ) ) {

$this->id = $variable->id;

$this->experiment\_id = $variable->experiment\_id;

$this->name = $variable->name;

$this->type = $variable->type;

/\*

$this->user = new BP\_Core\_User( $this->user\_id );

\*/

}

}

/\*\*

\* Save the variable data to the database.

\*

\* @return bool True on success, false on failure.

\*/

public function save() {

global $wpdb, $bp;

$this->experiment\_id = apply\_filters( 'experiments\_variable\_experiment\_id\_before\_save', $this->experiment\_id, $this->id );

$this->name = apply\_filters( 'experiments\_variable\_name\_before\_save', $this->name, $this->id );

$this->type = apply\_filters( 'experiments\_variable\_type\_before\_save', $this->type, $this->id );

do\_action\_ref\_array( 'experiments\_variable\_before\_save', array( &$this ) );

if ( !empty( $this->id ) ) {

$sql = $wpdb->prepare( "UPDATE wp\_bp\_experiments\_variables SET name = %s, type = %s WHERE id = %d", $this->name, $this->type, $this->id );

} else {

/\*

// Ensure that variable is not already a variable of the experiment before inserting

if ( $wpdb->get\_var( $wpdb->prepare( "SELECT id FROM {$bp->experiments->table\_name\_variables} WHERE id = %d AND experiment\_id = %d LIMIT 1", $this->id, $this->experiment\_id ) ) ) {

return false;

}

\*/

$sql = $wpdb->prepare( "INSERT INTO wp\_bp\_experiments\_variables (experiment\_id, name, type) VALUES (%d, %s, %s)", $this->experiment\_id, $this->name, $this->type );

//$sql = $wpdb->prepare( "INSERT INTO wp\_bp\_experiments\_variables (experiment\_id, name, type) VALUES (%d, %s, %s)", 23, 'happiness', 'count' );

}

if ( !$wpdb->query( $sql ) )

return false;

if ( empty( $this->id ) )

$this->id = $wpdb->insert\_id;

// Update the experiment’s variable count

//self::refresh\_total\_variable\_count\_for\_experiment( $this->id );

do\_action\_ref\_array( 'experiments\_variable\_after\_save', array( &$this ) );

return true;

}

/\*\*

\* Remove the current variable of an experiment.

\*

\* @return bool True on success, false on failure.

\*/

public function remove() {

global $wpdb, $bp;

$sql = $wpdb->prepare( "DELETE FROM {$bp->experiments->table\_name\_variables} WHERE id = %d AND experiment\_id = %d", $this->id, $this->experiment\_id );

if ( !$result = $wpdb->query( $sql ) )

return false;

// Update the experiment's variable count

self::refresh\_total\_variable\_count\_for\_experiment( $this->experiment\_id );

return $result;

}

/\*\* Static Methods \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

\* Refresh the total\_variable\_count for a experiment.

\*

\* @since BuddyPress (1.8.0)

\*

\* @param int $experiment\_id ID of the experiment.

\* @return bool True on success, false on failure.

\*/

public static function refresh\_total\_variable\_count\_for\_experiment( $experiment\_id ) {

return experiments\_update\_experimentmeta( $experiment\_id, 'total\_variable\_count', (int) BP\_Experiments\_Experiment::get\_total\_variable\_count( $experiment\_id ) );

}

/\*\*

\* Get the IDs of all a given experiment's variables.

\*

\* @param int $experiment\_id ID of the experiment.

\* @return array IDs of all experiment variables.

\*/

public static function get\_experiment\_variable\_ids( $experiment\_id ) {

global $bp, $wpdb;

return $wpdb->get\_col( $wpdb->prepare( "SELECT id FROM {$bp->experiments->table\_name\_variables} WHERE experiment\_id = %d", $experiment\_id ) );

}

/\*\*

\* Get variables of an experiment.

\*

\* @deprecated BuddyPress (1.8.0)

\*/

public static function get\_all\_for\_experiment( $experiment\_id, $limit = false, $page = false ) {

global $bp, $wpdb;

\_deprecated\_function( \_\_METHOD\_\_, '1.8', 'BP\_Experiment\_Variable\_Query' );

$pag\_sql = '';

if ( !empty( $limit ) && !empty( $page ) )

$pag\_sql = $wpdb->prepare( "LIMIT %d, %d", intval( ( $page - 1 ) \* $limit), intval( $limit ) );

if ( empty( $variables ) ) {

return false;

}

if ( empty( $pag\_sql ) ) {

$total\_variable\_count = count( $variables );

} else {

$total\_variable\_count = $wpdb->get\_var( apply\_filters( 'bp\_experiment\_variables\_count', $wpdb->prepare( "SELECT COUNT(id) FROM {$bp->experiments->table\_name\_variables} m WHERE experiment\_id = %d", $experiment\_id ) ) );

}

return array( 'variables' => $variables, 'count' => $total\_variable\_count );

}

/\*\*

\* Delete all variables for a given experiment.

\*

\* @param int $experiment\_id ID of the experiment.

\* @return int Number of records deleted.

\*/

public static function delete\_all( $experiment\_id ) {

global $wpdb, $bp;

return $wpdb->query( $wpdb->prepare( "DELETE FROM {$bp->experiments->table\_name\_variables} WHERE experiment\_id = %d", $experiment\_id ) );

}

}//end class BP\_Experiments\_Variable

Add these functions here:

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments-functions.php**

function experiments\_create\_variables( $args = '' ) {

extract( $args );

/\*\*

\* Possible parameters (pass as assoc array):

\* 'experiment\_id'

\* 'creator\_id'

\* 'name'

\* 'description'

\* 'slug'

\* 'status'

\* 'enable\_forum'

\* 'date\_created'

\*/

if (empty( $experiment\_id ) )

return false;

//$experiment = experiments\_get\_experiment( array( 'experiment\_id' => $experiment\_id ) );

$variable1 = new BP\_Experiments\_Variable;

$variable1->experiment\_id = $experiment\_id;

$variable1->name = $variable1\_name;

$variable1->type= $variable1\_type;

if ( !$variable1->save() )

return false;

else{

$variable2 = new BP\_Experiments\_Variable;

$variable2->experiment\_id = $experiment\_id;

$variable2->name = $variable2\_name;

$variable2->type= $variable2\_type;

if ( !$variable2->save() )

return false;

}

do\_action( 'experiments\_created\_variables', $variable1->id, $variable1 );

do\_action( 'experiments\_created\_variables', $variable2->id, $variable2 );

return true;

}

Add these functions here:

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments -template.php**

function bp\_new\_experiment\_variable() {

echo bp\_get\_new\_experiment\_variable();

}

function bp\_get\_new\_experiment\_variable() {

global $bp;

if ( isset( $bp->experiments->current\_experiment->variable ) )

$description = $bp->experiments->current\_experiment->variable;

else

$variable = '';

return apply\_filters( 'bp\_get\_new\_experiment\_variable', $variable );

}

function bp\_experiment\_total\_variables( $experiment = false ) {

echo bp\_get\_experiment\_total\_variables( $experiment );

}

function bp\_get\_experiment\_total\_variables( $experiment = false )

{

global $experiments\_template;

if ( empty( $experiment ) )

$experiment =& $experiments\_template->experiment;

return apply\_filters( 'bp\_get\_experiment\_total\_variables', $experiment->total\_variable\_count );

}

function bp\_experiment\_variable\_count() {

echo bp\_get\_experiment\_variable\_count();

}

function bp\_get\_experiment\_variable\_count() {

global $experiments\_template;

if ( 1 == (int) $experiments\_template->experiment->total\_variable\_count )

return apply\_filters( 'bp\_get\_experiment\_variable\_count', sprintf( \_\_( '%s variable', 'buddypress' ), bp\_core\_number\_format( $experiments\_template->experiment->total\_variable\_count ) ) );

else

return apply\_filters( 'bp\_get\_experiment\_variable\_count', sprintf( \_\_( '%s variables', 'buddypress' ), bp\_core\_number\_format( $experiments\_template->experiment->total\_variable\_count ) ) );

}

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments-loader.php**

// Preconfigured experiment creation steps

$this->experiment\_creation\_steps = apply\_filters( 'experiments\_create\_experiment\_steps', array(

'experiment-details' => array(

'name' => \_\_( 'Details', 'buddypress' ),

'position' => 0

),

'experiment-variables' => array(

'name' => \_\_( 'Measurements', 'buddypress' ),

'position' => 10

),

'experiment-settings' => array(

'name' => \_\_( 'Settings', 'buddypress' ),

'position' => 20

)

) );

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress/experiments/create.php**

<?php /\* experiment creation step 2: experiment variables \*/ ?>

<?php if ( bp\_is\_experiment\_creation\_step( 'experiment-variables' ) ) : ?>

<?php do\_action( 'bp\_before\_experiment\_variables\_creation\_step' ); ?>

<label for="experiment-variable1"><?php \_e( 'Measurement 1 (required)', 'buddypress' ); ?></label>

<input type="text" name="variable1\_name" id="variable1\_name" aria-required="true" value="<?php bp\_new\_experiment\_variable(); ?>" />

<select id="varaible1-type" name="variable1\_type">

<option value="score"><?php \_e( 'Score', 'buddypress' ); ?></option>

<option value="binary"><?php \_e( 'Binary', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Count', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Time', 'buddypress' ); ?></option>

<?php do\_action( 'bp\_experiment\_variable1\_type\_options' ); ?>

</select>

<label for="experiment-variable2"><?php \_e( 'Measurement 2 (required)', 'buddypress' ); ?></label>

<input type="text" name="variable2\_name" id="variable2\_name" aria-required="true" value="<?php bp\_new\_experiment\_variable(); ?>" />

<select id="varaible2-type" name="variable2\_type">

<option value="score"><?php \_e( 'Score', 'buddypress' ); ?></option>

<option value="binary"><?php \_e( 'Binary', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Count', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Time', 'buddypress' ); ?></option>

<?php do\_action( 'bp\_experiment\_variable2\_type\_options' ); ?>

</select>

<?php

do\_action( 'bp\_after\_experiment\_variables\_creation\_step' );

do\_action( 'experiments\_custom\_experiment\_fields\_editable' ); // @Deprecated

wp\_nonce\_field( 'experiments\_create\_save\_experiment-variables' ); ?>

<?php endif; ?>

This is not needed.

**wordpress/wp-content/plugins/buddypress/bp-themes/bp-default/experiments/create.php**

<?php /\* experiment creation step 2: experiment variables \*/ ?>

<?php if ( bp\_is\_experiment\_creation\_step( 'experiment-variables' ) ) : ?>

<?php do\_action( 'bp\_before\_experiment\_variables\_creation\_step' ); ?>

<label for="experiment-variable1"><?php \_e( 'Measurement 1 (required)', 'buddypress' ); ?></label>

<input type="text" name="variable1\_name" id="variable1\_name" aria-required="true" value="<?php bp\_new\_experiment\_variable(); ?>" />

<select id="varaible1-type" name="variable1\_type">

<option value="score"><?php \_e( 'Score', 'buddypress' ); ?></option>

<option value="binary"><?php \_e( 'Binary', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Count', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Time', 'buddypress' ); ?></option>

<?php do\_action( 'bp\_experiment\_variable1\_type\_options' ); ?>

</select>

<label for="experiment-variable2"><?php \_e( 'Measurement 2 (required)', 'buddypress' ); ?></label>

<input type="text" name="variable2\_name" id="variable2\_name" aria-required="true" value="<?php bp\_new\_experiment\_variable(); ?>" />

<select id="varaible2-type" name="variable2\_type">

<option value="score"><?php \_e( 'Score', 'buddypress' ); ?></option>

<option value="binary"><?php \_e( 'Binary', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Count', 'buddypress' ); ?></option>

<option value="count"><?php \_e( 'Time', 'buddypress' ); ?></option>

<?php do\_action( 'bp\_experiment\_variable2\_type\_options' ); ?>

</select>

<?php

do\_action( 'bp\_after\_experiment\_variables\_creation\_step' );

do\_action( 'experiments\_custom\_experiment\_fields\_editable' ); // @Deprecated

wp\_nonce\_field( 'experiments\_create\_save\_experiment-variables' ); ?>

<?php endif; ?>

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments-actions.php**

function experiments\_action\_create\_experiment() {

. . .

if ( 'experiment-variables' == bp\_get\_experiments\_current\_create\_step() ) {

$name = $\_POST['name'];

$type = $\_POST['type'];

foreach( $name as $key => $variable\_name ) {

if ( empty( $variable\_name) || !strlen( trim( $variable\_name ) )

) {

bp\_core\_add\_message( \_\_( 'Please fill in all of the required fields', 'buddypress' ), 'error' );

bp\_core\_redirect( bp\_get\_root\_domain() . '/' . bp\_get\_experiments\_root\_slug() . '/create/step/' . bp\_get\_experiments\_current\_create\_step() . '/' );

}

}//end for

foreach( $type as $key => $variable\_type ) {

if ( empty( $variable\_type) || !strlen( trim( $variable\_type ) )

) {

bp\_core\_add\_message( \_\_( 'Please fill in all of the required fields', 'buddypress' ), 'error' );

bp\_core\_redirect( bp\_get\_root\_domain() . '/' . bp\_get\_experiments\_root\_slug() . '/create/step/' . bp\_get\_experiments\_current\_create\_step() . '/' );

}

}//end for

if(count($name) != count($type))

{

bp\_core\_add\_message( \_\_( 'Please fill in all of the required fields', 'buddypress' ), 'error' );

bp\_core\_redirect( bp\_get\_root\_domain() . '/' . bp\_get\_experiments\_root\_slug() . '/create/step/' . bp\_get\_experiments\_current\_create\_step() . '/' );

}//end if

if ( false == experiments\_create\_variables( array( 'experiment\_id' => $bp->experiments->new\_experiment\_id, 'name' => $name, 'type' => $type ) ) ) {

bp\_core\_add\_message( \_\_( 'There was an error saving experiment variables, please try again.', 'buddypress' ), 'error' );

bp\_core\_redirect( bp\_get\_root\_domain() . '/' . bp\_get\_experiments\_root\_slug() . '/create/step/' . bp\_get\_experiments\_current\_create\_step() . '/' );

}

}//end if ( 'experiment-variables' == bp\_get\_experiments\_current\_create\_step() )

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress/experiments/single/home.php**

// Default to activity

elseif ( bp\_is\_active( 'activity' ) ) : bp\_get\_template\_part( 'experiments/single/report' );

Add this file:

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress/experiments/single/report.php**

**wordpress/wp-content/plugins/buddypress/bp-xprofiles/bp-xprofile-admin.php**

**Experiment Creation In One Step**

Modify the files:

**wordpress/wp-content/plugins/buddypress/bp-experiments/ bp-experiments-template.php**

Change button title from “Create a Experiment” to “Create an Experiment”.

function bp\_get\_experiment\_create\_button() {

. . .

$button\_args = array(

'id' => 'create\_experiment',

'component' => 'experiments',

'link\_text' => \_\_( 'Create an Experiment', 'buddypress' ),

'link\_title' => \_\_( 'Create an Experiment', 'buddypress' ),

'link\_class' => 'button experiment-create bp-title-button',

'link\_href' => trailingslashit( bp\_get\_root\_domain() ) . trailingslashit( bp\_get\_experiments\_root\_slug() ) . trailingslashit( 'create' ),

'wrapper' => false,

);

. . .

}

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments-screens.php**

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments-functions.php**

function experiments\_create\_experiment( $args = '' ) {

}

**wordpress/wp-content/plugins/buddypress/bp-experiments/bp-experiments-loader.php**

// Preconfigured experiment creation steps

$this->experiment\_creation\_steps = apply\_filters( 'experiments\_create\_experiment\_steps', array(

'experiment-details' => array(

'name' => \_\_( 'Details', 'buddypress' ),

'position' => 0

/\*

),

'experiment-variables' => array(

'name' => \_\_( 'Measurements', 'buddypress' ),

'position' => 10

),

'experiment-settings' => array(

'name' => \_\_( 'Settings', 'buddypress' ),

'position' => 20

\*/

)

) );

**wordpress/wp-content/plugins/buddypress/bp-experiments/ bp-experiments-template.php**

function bp\_experiment\_creation\_single\_form\_action() {

echo bp\_get\_experiment\_creation\_single\_form\_action();

}

function bp\_get\_experiment\_creation\_single\_form\_action() {

global $bp;

if ( !bp\_action\_variable( 1 ) ) {

$keys = array\_keys( $bp->experiments->experiment\_creation\_steps );

$bp->action\_variables[1] = array\_shift( $keys );

}

return apply\_filters( 'bp\_get\_experiment\_creation\_single\_form\_action', trailingslashit( bp\_get\_root\_domain() . '/' . bp\_get\_experiments\_root\_slug() . '/create/step/' . bp\_action\_variable( 1 ) ) );

}

**wordpress/wp-content/plugins/buddypress/bp-experiments/ bp-experiments-actions.php**

function experiments\_action\_create\_experiment\_all\_steps(){

global $bp;

// If we're not at domain.org/experiments/create/ then return false

if ( !bp\_is\_experiments\_component() || !bp\_is\_current\_action( 'create' ) )

return false;

if ( !is\_user\_logged\_in() )

return false;

if ( !bp\_user\_can\_create\_experiments() ) {

bp\_core\_add\_message( \_\_( 'Sorry, you are not allowed to create experiments.', 'buddypress' ), 'error' );

bp\_core\_redirect( trailingslashit( bp\_get\_root\_domain() . '/' . bp\_get\_experiments\_root\_slug() ) );

}

}

function experiments\_action\_create\_experiment() {

…

…

}

**wordpress/wp-content/plugins/buddypress/bp-templates/bp-legacy/buddypress/experiments/single/create.php**

<form action="<?php bp\_experiment\_creation\_single\_form\_action(); ?>" method="post" id="create-experiment-form" class="standard-form" enctype="multipart/form-data">

**Setting Experiments as the home page**

such that “Create an Experiment” button is visible:

In <theme>/index.php, delete everything and write this:

<?php

/\* Redirect browser \*/

header("Location: http://localhost/wordpress/experiments");

/\* Make sure that code below does not get executed when we redirect. \*/

exit;

?>

On Wordpress dashboard, Go to Settings->Reading. Set Front-page displays to “your latest post”.