CUNY Academic Commons - Feature #5016

Allow comments to be posted on events

2015-12-11 01:29 PM - Matt Gold

Status:	Assigned		**Start date:**	2015-12-10
Priority name:	Low		**Due date:**	
Assignee:	Raymond Hoh		**% Done:**	0%
Category name:	Events		**Estimated time:**	0.00 hour
Target version:	Future release			

Description
A private group I'm on wants to have conversations around events, so I am curious about whether comments can be added to events.

Related issues:		
Related to CUNY Academic Commons - Feature #5456: Comment on events	**Duplicate**	**2016-04-15**

History

#1 - 2015-12-11 03:46 PM - Boone Gorges

- Target version set to Future release

Yes, it's possible. A couple considerations:

- It'd be nice for Sam to chime in on what the interface might look like. Events have a large "Event Details" section, and some nav/action buttons at the bottom. Where would comments go? Would it be a hidden-by-default section?

- There will be some technical complications arising from the fact that an event can appear in more than one context (user profile, group, top-level). This'll have implications for styling as well as form submission. Adding Ray as a watcher for his thoughts.

We will also need to build the appropriate BP integration for comment activity (appearing in activity streams, etc).

It's a medium-complexity task, probably a day or two of work by me or Ray once we have some UX/design guidance. Let me know how you'd like it prioritized.

#2 - 2016-04-17 10:02 PM - Matt Gold

Another vote for this here - https://redmine.gc.cuny.edu/issues/5456 - with another use case:

"A use case would be if an event has an incorrect location, which a non-event owner spots and wants to notify the owner."

Boone, what do you think of moving this to the 2.0 release?

#3 - 2016-04-21 12:30 AM - Boone Gorges

- Target version changed from Future release to 1.10

OK, let's look at it for 2.0.

There are a couple ways we could go with UX and IA here. I'm going to suggest the following, and ask Sam to chime in with a yea or a nay.

- Comments live with an event, regardless of where the comments were left. That is, if an event appears in five groups, and a member of one of the groups leaves a comment on the event in the context of that group, members of all five groups will see the comment.
- There will be no comment moderation or editing tools (at least as part of the first pass), so that an event owner will not be able to delete or modify unwanted comments on an event.

Ray, I'm thinking it makes sense to do this with CPT comments, but to tell BP to sync them to activity comments on the original "created" activity. This has a couple advantages: privacy controls are inherited automatically in the activity stream, and people will automatically receive email notifications of comments to their posts. But there are probably also some weird complications I'm not thinking of. If it's going to be too hard, maybe we leave activity integration out for a first pass, and rely on manually sending emails to the event author (and comment authors, in the case of replies to a thread).

#4 - 2016-04-21 05:18 PM - Samantha Raddatz

- Comments live with an event, regardless of where the comments were left. That is, if an event appears in five groups, and a member of one of the groups leaves a comment on the event in the context of that group, members of all five groups will see the comment.

Yea -- this should be noted in some way near the 'submit' button for the comments, so that people are aware that anyone with access to the event will

be able to see the note (rather than just the author of the event).

- There will be no comment moderation or editing tools (at least as part of the first pass), so that an event owner will not be able to delete or modify unwanted comments on an event.

Half-yea -- I'm cool with the owner not being able to delete in this first round, but I think the creator of the comment should be able to delete their comment at least, in case of accidental postings, embarrassing typos, etc. Would that be possible, maybe not in the very first iteration, but soon after?

#5 - 2016-04-21 09:32 PM - Boone Gorges

Thanks, Sam!

> Half-yea -- I'm cool with the owner not being able to delete in this first round, but I think the creator of the comment should be able to delete their comment at least, in case of accidental postings, embarrassing typos, etc. Would that be possible, maybe not in the very first iteration, but soon after?

We can do anything at any time :) It's just a question of how much time we want to spend on a given iteration. These moderation tools will need to be built largely from scratch, so they will take a significant amount of time. I gather from your tone that you see them as more or less critical for the first revision, which just makes the scope of that first revision a little bigger.

#6 - 2016-04-21 09:34 PM - Matt Gold

FWIW, I agree with Sam here that there should be some mechanism for deletion in our first iteration.

#7 - 2016-04-21 09:41 PM - Boone Gorges

- Assignee changed from Boone Gorges to Daniel Jones

Dan, here's a project for you :)

I'd suggest firing up a feature branch on the bp-event-organiser repo to work on this. Here's a rundown of what I think needs to happen:

- It looks like EO already supports comments (there's a checkbox to enable it in the settings). Figure out how to make it show up on the front end. I think we can probably disable comment posting for non-logged-in users, which should simplify a good deal of stuff.
- Take a look at the way that BuddyPress handles blog comment <-> activity comment syncing, and enable it for this post type. Fool around with it and make sure that it's working as you would expect.
- Take a look at the way that social-paper's comment moderation tools are built, and start building out something similar. The feature will be a bit different for BPEO, but it'll be a start. I don't think there will be a need for comment approval (you should make it so that comments from logged-in users are approved automatically; see the 'pre_comment_approved' filter) but there should, at minimum, be the ability for the **comment author** and the **site admin** to delete comments. We can leave other stuff (deletion by event author, comment editing) until later.

#8 - 2016-10-13 11:47 AM - Boone Gorges

- Target version changed from 1.10 to 1.11

Hi Dan - Have you had a chance to dig into this item? I have a feeling it's too complex to fit into a release in a few weeks, so I'm preemptively moving into the next milestone.

#9 - 2017-04-24 03:52 PM - Boone Gorges

- Target version changed from 1.11 to Future release

Would love to see this happen, but it's too late for 1.11.

#10 - 2017-11-01 02:06 PM - Daniel Jones

Here's another blast from the past.

I think I've got step 1 complete here - allowing front-end control of whether or not comments are open on a group event, and displaying the comments on the single event template.

I still need to work on activity syncing and moderation. Wanted to send this along though for a sanity check:
https://github.com/cuny-academic-commons/bp-event-organiser/commit/4055fd63524091313396a9093e900e59fb906cfc

#11 - 2017-11-01 03:10 PM - Boone Gorges

- Target version changed from Future release to 1.13

Looks good so far. Thanks!

#12 - 2017-11-06 11:13 AM - Daniel Jones

Here's another commit: https://github.com/cuny-academic-commons/bp-event-organiser/commit/35fe835e3072e3c5c7ea9c00a1d3146cc3d32c1a

I've more or less directly lifted code from bp-blogs to set up the syncing. So far I've only worked on the activity comment -> event comments direction. I had disable the initial check bp_activity_type_supports($parent_activity->type, 'post-type-comment-tracking') for now, because I can't get it to return true after a couple attempts. I noticed as well that, at least on my install, comment syncing from activity comment -> group blog post comment doesn't work either, and that this test is returning false in that case as well (see bp_blogs_sync_add_from_activity_comment in bp-blogs-activity.php).

Also - I noticed that bp_blogs has code to allow for edit syncing from activity -> blog post comment, but there's no option to edit activity comments on blog post activities. Is that just a quirk with my settings?

Next I'm going to work on the other direction: event comments -> activity comments (creation, deletion, edit). I think after that I need to make sure that that things like activity filters, etc. are working correctly. After that I'll move on to moderation.

#13 - 2017-11-06 02:47 PM - Boone Gorges

Awesome, thanks for your work on this so far!

Ray - shouldn't there be an easier way to do comment syncing here? I thought that this was built into BP and could be enabled with a couple of post_type_supports flags. https://buddypress.trac.wordpress.org/ticket/6482

#14 - 2017-11-06 03:51 PM - Raymond Hoh

> Ray - shouldn't there be an easier way to do comment syncing here?

Yes:
https://codex.buddypress.org/plugindev/post-types-activities/#tracking-comments-about-a-post-type

However, our use-case might be a little bit different since we also do group events and group activities are recorded a bit differently than regular activity items. This is the part that would require some form of intervention.

#15 - 2017-11-16 09:27 AM - Daniel Jones

Hey all - I've tried to go back and enable this feature just using the post type tracking args, but it isn't working for me. I spent a bunch of time trying to track down why, and realized that at least the first issue is with where to hook in the call to bp_activity_set_post_type_tracking_args(). The documentation seems to recommend bp_init, but that was too early for our case: the 'event' post type hadn't been added to the $wp_post_types global at that point, and so bp_activity_set_post_type_tracking_args('event', $args) returned false. When I moved it to the bp_ready action that seemed to work at least for the front end. However, it looks like bp_ready isn't fired on AJAX requests at all, which means that our tracking args aren't being set when the comment is posted, which means it fails the first test in bp_blogs_sync_add_from_activity_comment(), which I think is what we end up relying on for the synchronization.

Do you all know if there's a good bp-related hook I should use for the AJAX request, or should I just use something like admin_init and test for BuddyPress-ness?

I should also note that, once we pass that first test (bp_activity_type_supports($parent_activity->type, 'post-type-comment-tracking')), we'll still need to pass if (bp_disable_blogforum_comments()) before the synchronization can begin. I don't think it's a great idea to have event comment synching depend on whether or not blog post comments are enabled or not. Do you have thoughts on that? I could try to hook into the bp_disable_blogforum_comments function and filter the result depending on our bpeo settings, but that seems a little messy to me.

#16 - 2017-11-16 02:37 PM - Boone Gorges

> Do you all know if there's a good bp-related hook I should use for the AJAX request, or should I just use something like admin_init and test for BuddyPress-ness?

You are finding yourself in the hell of the WP hook system :) I suggest hooking to 'init' with a high priority - something like 100, but definitely higher than whatever EO is using to register the 'event' post type - and then checking for BP inside of that hook. function_exists('buddypress') should work. (https://buddypress.trac.wordpress.org/browser/trunk/src/bp-core/bp-core-actions.php is a reference, though it shows that there's nothing reliable after 'init' that fires on wp-admin.)

> I should also note that, once we pass that first test (bp_activity_type_supports($parent_activity->type, 'post-type-comment-tracking')), we'll still need to pass if (bp_disable_blogforum_comments()) before the synchronization can begin. I don't think it's a great idea to have event comment synching depend on whether or not blog post comments are enabled or not. Do you have thoughts on that? I could try to hook into the bp_disable_blogforum_comments function and filter the result depending on our bpeo settings, but that seems a little messy to me.

Groan. It stinks that this is hardcoded in BP. If this function is intended to be used for other post types, then the blogforum_comments check should at least accept a context (like a post type). Ray, you have any suggestions about this? Just-in-time filtering of `bp_disable_blogforum_comments` may be the best we can do.

#17 - 2017-11-16 05:05 PM - Daniel Jones

Alright so we've passed a couple of hurdles here, but there's still more! bp_blogs_sync_add_from_activity_comment() includes a call to switch_to_blog($parent_activity->item_id), which creates unexpected behavior because it adds the comment, but to the wp_comments_### table, where ### is the ID of the event, I think. I could hook into the switch_blog action, fired in switch_to_blog, but I don't know if this is getting to messy and it's better to return to my original approach of copying over a bunch of the comment syncing code from bp-blogs-activtiy.php but modifying it to work for events.

#18 - 2017-11-20 03:34 PM - Boone Gorges

> Alright so we've passed a couple of hurdles here, but there's still more! bp_blogs_sync_add_from_activity_comment() includes a call to switch_to_blog($parent_activity->item_id), which creates unexpected behavior because it adds the comment, but to the wp_comments_### table, where ### is the ID of the event, I think. I could hook into the switch_blog action, fired in switch_to_blog, but I don't know if this is getting to messy and it's better to return to my original approach of copying over a bunch of the comment syncing code from bp-blogs-activtiy.php but modifying it to work for events.

Ugh, right. We are using item_id for something else. The bp-activity comment sync mechanism was not designed for that. Unless Ray has objections, let's go back to what you had (with apologies for the goose chase).

#19 - 2017-11-20 04:10 PM - Raymond Hoh

> The bp-activity comment sync mechanism was not designed for that. Unless Ray has objections, let's go back to what you had (with apologies for the goose chase).

My bad here. I forgot we were using a customized activity recording routine in BPEO.

If we had appropriate filters in BuddyPress, we could leverage some of the bp-activity comment sync code, but we do not so unfortunately, we'll have to nix this idea.

About this commit, https://github.com/cuny-academic-commons/bp-event-organiser/commit/35fe835e3072e3c5c7ea9c00a1d3146cc3d32c1a, ensure that the code is checking if the activity update is a BPEO one before proceeding with the post comment syncing logic. Similarly, check if the WP comment is from the Event Organiser post type before syncing to BP activity.

We need to do this so we do not interfere with the existing BP activity syncing logic.

#20 - 2018-01-15 02:11 PM - Daniel Jones

Hey all - I've taken another run at this and come up against a couple of (I think) obscure issues in syncing comments from the actual event post page to the activity stream:

1. The activity item that's created in bp_activity_post_type_comment (bp-activity/bp-activity-functions.php) when bp_disable_blogforum_comments() returns false should be showing up as an activity comment but it isn't. I tracked down the issue as somehow related to the fact that there are multiple activity items being entered into the wp_bp_activity database for each new event post: one each for the bpeo, events, and groups components. The activity item created from the event post comment gets associated with the bpeo one by default, because I've set its comments_tracking->component_id tracking arg to 'bpeo'. Should I change it to groups instead? I'm not sure if it would create other problems elsewhere.

2. I'm having an issue with mptt_left and mptt_right values: it looks like the values created for the activity items associated with the event comments are falling outside the bounds allowed for in the SQL that's generated by get_activity_comments (bp-activity/classes/class-bp-activity-activity.php), which decides which activity items are shown as comments on a given parent activity item. I'm not 100% sure how those values work, or why they aren't working in this case.

#21 - 2018-01-18 11:22 AM - Boone Gorges

> I tracked down the issue as somehow related to the fact that there are multiple activity items being entered into the wp_bp_activity database for each new event post:

Oh dear, what a huge pain.

Simply changing to 'groups' probably is not enough. An event can be associated with more than one group, which means that it'll have more than one activity item in the 'groups' component.

Any technical solution we come up with is going to be technically awful. So maybe we should take a step back and ask what we're trying to accomplish with the activity syncing. It seems to me that the most important thing is to trigger email notifications (or digest notifications), which will require 'groups' activity items to be generated. This suggests that we ought at least to be creating one activity_comment for each 'groups' activity item. But this will mean that activity comments won't be visible in non-group-linked activity streams, so I think the most logical thing to do is to sync the activity_comment to **each** activity item associated with an event. This is terrible in its own ways, but I feel like it's the only way to stay internally

consistent. Dan and Ray, what do you think?

(The best solution is probably to go back in time and not create separate activity items for each event "context", which is how we did it in Social Paper, but this is a huge issue that can't easily be reversed.)

A side note that comments on an event will be visible to all members who have the ability to view the event. We may want to have something in the interface that makes this clear to users, who might assume that comments will be visible only to members of the current group.

> I'm having an issue with mptt_left and mptt_right values: it looks like the values created for the activity items associated with the event comments are falling outside the bounds allowed for in the SQL that's generated by get_activity_comments (bp-activity/classes/class-bp-activity-activity.php), which decides which activity items are shown as comments on a given parent activity item. I'm not 100% sure how those values work, or why they aren't working in this case.

Can you give more details about how the event comments are being created, which is leading up to the mptt values? bp_activity_post_type_comment() or something like that?

#22 - 2018-01-18 08:09 PM - Raymond Hoh

> Any technical solution we come up with is going to be technically awful. So maybe we should take a step back and ask what we're trying to accomplish with the activity syncing.

Agreed. I needed a few days to think about this.

Right now, we are creating multiple activity items if an event is connected to a group (via bpeo_create_group_activity_items()) and then removing duplicate activity items from the sitewide activity stream (via bpeo_remove_duplicates_from_activity_stream().

Let's imagine we haven't touched the activity creation code for events before.

What if we started recording the event activity ID into group meta if an event is connected to a group? Then, in the group activity loop, we do a lookup for event activity IDs connected to a group and add them to the activity template loop if it matches.

This way, we get to inherit the post comment <-> activity comment syncing code from BuddyPress for free and we don't have to record multiple activity items.

This technique doesn't address group email notifications though, but perhaps we can manually trigger that ourselves when an activity item is saved into the database. If we decide to go with this technique, we'll need to write a script to calculate what activity items belong to which group and to remove all our duplicate activity items.

To illustrate what I mean, here's some sample code I started writing that injects event activity items into a particular group activity stream:

```
add_filter( 'bp_has_activities', function( $retval, $template, $r ) {
    // Bail if not on a group page.
    if ( ! bp_is_group() ) {
        return $retval;
    }

    /*
     * @todo We'll want to inject event activity items to empty group streams...
     *
     * @todo Also, if the first page of a group has a low activity count, we'll also
     *       need to grab more events to fill up the activity page...
     */
    if ( empty( $template->activity_count ) ) {
        return $retval;
    }

    // Activity ID pointers in the activity loop.
    $end   = reset( $template->activities );
    $end   = $end->id;
    $first = end( $template->activities );
    $first = $first->id;

    /*
     * Grab event activity IDs connected to group.
     *
     * My current idea is to record a new group meta entry when we record an
     * activity entry for BPEO.  We would need a backfill script for this.
     *
     * Activity IDs would be a separate group meta entry. Group meta could become
     * bloated over the long haul though... if that's the case, maybe a single
     * comma-delimited meta entry?
```

```php
     */
    //$event_ids_connected_to_group = array_map( 'intval', (array) groups_get_groupmeta( bp_get_current_group_
id(), 'bpeo_event_activity_ids', false ) );
    // $event_ids_connected_to_group = rsort( $event_ids_connected_to_group );

    // Add some test activity IDs in the meantime.
    $event_ids_connected_to_group = array( 1527, 1526, 804 );

    if ( empty( $event_ids_connected_to_group ) ) {
        return $retval;
    }

    $current_ids = wp_list_pluck( $template->activities, 'id' );
    $add_ids = array();

    // Check what the current activity page is.
    if ( 1 === (int) $r['page'] ) {
        // @todo We'll have to do some backfilling if there is a low activity count.
        // Check against $template->activity_count.
        foreach ( $event_ids_connected_to_group as $id ) {
            if ( $id > $first && ! in_array( $id, $current_ids, true ) ) {
                $add_ids[] = $id;
            }
        }

    } else {
        foreach ( $event_ids_connected_to_group as $id ) {
            // UPDATE: This will probably not work well... we'd need to reference
            // the activity ID pointer from the previous lookup...
            if ( $id > $first && $id < $end ) {
                $add_ids[] = $id;
            }
        }
    }

    }

    if ( empty( $add_ids ) ) {
        return $retval;
    }

    // Fetch event activity items.
    $a = bp_activity_get( array(
        'display_comments'  => true,
        'show_hidden'       => true,
        'in'                => $add_ids,
    ) );

    $add_ids  = wp_list_pluck( $a['activities'], 'id' );
    $to_add   = reset( $add_ids );
    $modified = $GLOBALS['activities_template']->activities;

    // Inject our event activity items into the activity template.
    foreach ( $template->activities as $i => $activity ) {
        if ( $to_add > $activity->id ) {
            array_splice( $modified, $i, 0, array( $a['activities'][ key( $add_ids ) ] ) );

            $to_add = next( $add_ids );
            if ( false === $to_add ) {
                break;
            }
        }
    }

    // Fix up the $activities_template global.
    $GLOBALS['activities_template']->activities      = $modified;
    $GLOBALS['activities_template']->activity_count = $GLOBALS['activities_template']->activity_count + count(
 $a );

    return $retval;
}, 999, 3 );
```

Update: I didn't think of what would happen on user activity pages, particularly on a user's "Activity > Groups" page or for something like My Commons. It might be possible to grab a list of a user's group IDs and following a similar approach above, but that would probably be intensive when looping through all of a user's groups to grab the relevant event activity IDs for injection.

#23 - 2018-01-19 10:25 AM - Boone Gorges

Thanks for taking the time to think this over, Ray. The strategy you describe is similar to what we use in Social Paper, minus the groupmeta part (which acts like a caching mechanism). In SP, we record all activity items with component = cacsp, and then we include them dynamically in group/user activity streams based on permissions. See https://github.com/cuny-academic-commons/social-paper/blob/master/includes/hooks-buddypress-group.php#L148. This is a bit less hacky than Ray's bp_has_activities callback, but similar in spirit. SP then manually triggers BPGES when necessary: https://github.com/cuny-academic-commons/social-paper/blob/master/includes/hooks-buddypress-group.php#L309

BPEO and SP were, for me, an experiment in doing this kind of activity linking in both ways. On reflection, the SP way might be slightly less insane :)

Switching to the other system would be OK by me. Much of the skeleton logic can be borrowed from SP. I am somewhat concerned that others are using BPEO other than the Commons (I know for a fact that the OpenLab is), which means that we might need to build a more general migration script.

I'll spend some time in the upcoming days thinking more about this. It's a big change and I don't want to undertake it rashly. Maybe we can bracket a few minutes during our Tuesday call to talk through the ramifications.

#24 - 2018-03-22 10:22 AM - Boone Gorges

Hi Dan - Are you still around to wrap up work on this ticket? If so, could you let me know where you think things stand? I realize that a huge amount of work has gone into it, more than is probably worthwhile for a feature with impact as limited as this one, but I don't want to scrap it altogether if you think that we're in the ballpark.

#25 - 2018-04-23 10:42 AM - Boone Gorges

- Target version changed from 1.13 to 1.14

#26 - 2018-09-27 05:21 PM - Boone Gorges

- Assignee changed from Daniel Jones to Raymond Hoh

Ray, sometime in the next month or so, could I ask you to take a look at the work that Dan previously did toward this ticket? Maybe spend 30 minutes trying to understand the state of things, and then make an assessment about what you think it'd take to close the loop. It may turn out that we don't want to devote the time to it for this release, but it's hard to gauge without a sense of how much work it'll be.

#27 - 2018-10-22 06:05 PM - Raymond Hoh

- Target version changed from 1.14 to 1.15

I meant to bump this last week, so bumping now.

If we want this in 1.15, let's discuss how important this feature is vis-a-vis other features when we start planning for that milestone.

#28 - 2019-03-01 02:23 PM - Boone Gorges

- Priority name changed from Normal to Low

- Target version changed from 1.15 to Future release

This ticket is the gift that keeps on giving. Let's shelve.