Events

Events can be triggered in Tiki during execution of functions. A general usage is to trigger events in service controllers.
Events call a callback function
Events can be chained
Activities are triggered event instances that are configured to be recorded. Event instances are not automatically recorded (except for some basic events), you need to set up an activity rule if you want this to happen. Activities can be displayed in activity streams.
Events are not log entries.
Tiki events are defined in the source code, so if you want to setup a new event, you need to modify the definition file.

Definitions

```
lib/setup/events.php
```

An event is definded as below:

event name	a unique name for the event. Example: tiki.comment.update.
priority	Allows events to be forced to execute after others. For example, normal priorities may alter data, but indexing should not happen until all data has been modified. Default priority is 0. Priorities are numeric, false indicates that the event executes at all levels. This is used for chaining and happens transparently when using bind() with an event as the callback.
callback	the name of the callback function to be executed.
arguments	arguments to be passed to the callback function

When defining events it is important to bind and chain them (see function *bind* and *bindPriority* in the Library)
function bind binds an event at normal priority and handles event chaining.
function bindPriority binds the event at a specific priority
Binding can:

- point to another event
- defer to a library with a function within that library
- call a service controller function
- be delegated to another library/controller

```
$events = TikiLib::events(); $defer = function ($lib, $function ) { return Tiki_Event_Lib::defer($lib,
$function); }; //single binding to another event $events->bind('tiki.comment.post', 'tiki.comment.save');
//multiple binding by deferring to a library and a function within that library
$events->bind('tiki.trackeritem.save', $defer('trk', 'update_tracker_summary'));
$events->bind('tiki.trackeritem.save', $defer('trk', 'invalidate_item_cache')); // binding using a service
controller $events->bind('tiki.trackeritem.create', ['Services_MustRead_Controller',
'handleItemCreation']); //delegated binding TikiLib::lib('goalevent')->bindEvents($events);
```

Library

lib/core/Tiki/Event/Manager.php

How to use

In a service controller call the trigger function and supply the necessary inputs

```
TikiLib::events()->trigger('tiki.comment.update', array( 'type' => $comment['objectType'], 'object' => $comment['object'], 'title' => $title, 'comment' => $threadId, 'user' => $GLOBALS['user'], 'content' => $data, 'index_handled' => true, ));
```