Migrating to Ushahidi 2.5

This page doesn't document all changes in 2.5, but specifically documents breaking changes that may effect developers migrating to v2.5

What's changed

- Mapping:
  The mapping code for the main page has been refactored and is no longer tied to the timeline. We've consolidated the mapping code into a single JavaScript library; ushahidi.js. This library has an event system which is extensible. The stock events can be triggered for actions such as zoom changes, resizing the map viewport, change of base layer etc. As a by the way, decoupling the timeline from the map effectively means that you can plug in your own timeline. The stock timeline is uses jqplot. This change affects those plugins that hook into the main map such the fullscreen map plugin

- Themes:
  - Views for the front-end have been grouped into directories that are named after the controllers that make use of them; For example, all the views for the reports controller are now in the themes/default/views/reports directory.
    This change will most definitely affect any custom themes (besides the stock ones) that you may have on your deployment. To make them compatible with the upcoming release, rename view files to follow the convention outlined above (and used in the default theme). The following is list of views and their new file names:
    - alert_radius_view.php > alerts/radius.php
    - alerts.php > /alerts/main.php
    - alerts_confirm.php > /alerts/confirm.php
    - alerts_unsubscribe.php > /alerts/unsubscribe.php
    - alerts_verify.php > /alerts/verify.php
    - alert_radius_view.php > /alerts/radius.php
    - contact.php >
    - feed.php >
    - feeds.php > /feed/feeds.php
    - footer.php >
    - header.php >
    - help.php > REMOVED
    - help_view.php > REMOVED
    - konana_error_page.php >
    - login.php > /login/main.php
    - main.php > /main/layout.php
    - main_map.php > /main/map.php
    - main_timeline.php > /map/timeline.php
    - page.php >
    - profile.php > /profile/user.php
    - profile_browse.php > /profile/main.php
    - reports.php > /reports/main.php
    - reports_comments.php > /reports/comments.php
    - reports_comments_form.php > /reports/comments_form.php
    - reports_listing.php > /reports/list.php
    - reports_stats.php > /reports/stats.php
    - reports_submit.php > /reports/submit.php
    - reports_submit_custom_forms.php > /reports/submit_custom_forms.php
    - reports_submit_thanks.php > /reports/submit_thanks.php
    - reports_view.php > /reports/detail.php
    - reports_view_custom_forms.php > /reports/detail_custom_forms.php
    - pagination/front-end-reports.php >
    - blocks/main_news.php >
    - blocks/main_reports.php >
  - ALL the views (including the *_js.php) for the front-end are now under the themes directory.
    View moved from application/views to themes/default/views (and into subfolder as appropriate)
    - feeds_atom.php > /feed/atom.php
    - feeds_rss2.php > /feed/rss2.php
    - reports_js.php > reports/reports_js.php
    - reports_view_js.php > reports/view_js.php
    - reports_submit_edit_js.php > reports/submit_edit_js.php
    - login_js.php > login/login_js.php

        - main_js.php > main/main_js.php
        - alerts_js.php > alerts/alerts_js.php
- Settings Table: The structure for the settings table has been modified so that data are stored as key/value pairs. Previously, this table only had one row and each setting was a column. In the new structure, there are only 3 columns: id, key, value. This now allows plugins to add their own settings. The settings model class (application/models/settings) has the necessary utility methods for retrieving and saving data from/to the restructured table. There is an upgrade script that shall effect this change on your schema.
    - Old schema:

```
CREATE TABLE IF NOT EXISTS `settings` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `site_name` varchar(255) DEFAULT NULL,
  `site_tagline` varchar(255) DEFAULT NULL,
  `site_banner_id` int(11) DEFAULT NULL,
  `site_email` varchar(120) DEFAULT NULL,
  `site_key` varchar(100) DEFAULT NULL,
  `site_language` varchar(10) NOT NULL DEFAULT 'en_US',
  `site_style` varchar(50) NOT NULL DEFAULT 'default',
  `site_timezone` varchar(80) DEFAULT NULL,
  `site_contact_page` tinyint(4) NOT NULL DEFAULT '1',
  `site_help_page` tinyint(4) NOT NULL DEFAULT '1',
  `site_message` text NOT NULL,
  `site_copyright_statement` text,
  `site_submit_report_message` text NOT NULL,
  `allow_reports` tinyint(4) NOT NULL DEFAULT '1',
  `allow_comments` tinyint(4) NOT NULL DEFAULT '1',
  `allow_feed` tinyint(4) NOT NULL DEFAULT '1',
  `allow_stat_sharing` tinyint(4) NOT NULL DEFAULT '1',
  `allow_clustering` tinyint(4) NOT NULL DEFAULT '0',
  `cache_pages` tinyint(4) NOT NULL DEFAULT '0',
  `cache_pages_lifetime` int(4) NOT NULL DEFAULT '1800',
  `private_deployment` tinyint(4) NOT NULL DEFAULT '0',
  `default_map` varchar(100) NOT NULL DEFAULT 'osm_mapnik',
  `default_map_all` varchar(20) NOT NULL DEFAULT 'CC0000',
  `default_map_all_icon_id` int(11) DEFAULT NULL,
  `api_google` varchar(200) DEFAULT NULL,
  `api_live` varchar(200) DEFAULT NULL,
  `api_akismet` varchar(200) DEFAULT NULL,
  `default_country` int(11) DEFAULT NULL,
  `multi_country` tinyint(4) NOT NULL DEFAULT '0',
  `default_city` varchar(150) DEFAULT NULL,
  `default_lat` varchar(100) DEFAULT NULL,
  `default_lon` varchar(100) DEFAULT NULL,
  `default_zoom` tinyint(4) NOT NULL DEFAULT '10',
  `items_per_page` smallint(6) NOT NULL DEFAULT '5',
  `items_per_page_admin` smallint(6) NOT NULL DEFAULT '20',
  `sms_provider` varchar(100) DEFAULT NULL,
  `sms_no1` varchar(100) DEFAULT NULL,
  `sms_no2` varchar(100) DEFAULT NULL,
  `sms_no3` varchar(100) DEFAULT NULL,
  `google_analytics` text,
  `twitter_hashtags` text,
  `blocks` text,
  `blocks_per_row` tinyint(4) NOT NULL DEFAULT '2',
  `date_modify` datetime DEFAULT NULL,
  `stat_id` bigint(20) DEFAULT NULL COMMENT 'comes from centralized stats',
  `stat_key` varchar(30) NOT NULL,
  `email_username` varchar(100) NOT NULL,
  `email_password` varchar(100) NOT NULL,
```

```
`email_port` int(11) NOT NULL,
`email_host` varchar(100) NOT NULL,
`email_servertype` varchar(100) NOT NULL,
`email_ssl` int(5) NOT NULL,
`ftp_server` varchar(100) DEFAULT NULL,
`ftp_user_name` varchar(100) DEFAULT NULL,
`alerts_email` varchar(120) NOT NULL,
`checkins` tinyint(4) NOT NULL DEFAULT '0',
`facebook_appid` varchar(150) DEFAULT NULL,
`facebook_appsecret` varchar(150) DEFAULT NULL,
`allow_alerts` tinyint(4) NOT NULL DEFAULT '0',
`manually_approve_users` tinyint(4) NOT NULL DEFAULT '0',
`require_email_confirmation` tinyint(4) NOT NULL DEFAULT '0',
`db_version` varchar(20) DEFAULT NULL,
`ushahidi_version` varchar(20) DEFAULT NULL,
PRIMARY KEY (`id`)
```

```
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COMMENT='Stores a deployment’s general settings'
AUTO_INCREMENT=2 ;
```

- New schema:

```
CREATE TABLE IF NOT EXISTS `settings` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `key` varchar(100) NOT NULL DEFAULT '' COMMENT 'Unique identifier for the configuration parameter',
  `value` text COMMENT 'Value for the settings parameter',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uq_settings_key` (`key`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

- Installer:
  The installer has been updated to work with the new settings table structure and we now perform the installation check in index.php. Previously, the installation check was being done via a hook (application/hooks/0_init.php)
- Configuration Files:
  The following config files are no longer in the repository: config.php, auth.php and encryption.php. Instead, we are only maintaining their templates (.template.php) which are the ones that the installer uses to create the config files to be used by the application. This change will affect users who automatically update/upgrade their deployment straight from GitHub - a git pull shall DELETE the mentioned config files (the ones we're no longer tracking) so please BACKUP these config files before engaging in acts of git fu.
- Permission Tables:
  User permissions have been refactored into a separate table form roles. This now allows plugins to add their own permissions. There is an upgrade script that shall affect this change on your schema.
  - Old Schema

```
CREATE TABLE IF NOT EXISTS `roles` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(32) NOT NULL,
  `description` varchar(255) NOT NULL,
  `reports_view` tinyint(4) NOT NULL DEFAULT '0',
  `reports_edit` tinyint(4) NOT NULL DEFAULT '0',
  `reports_evaluation` tinyint(4) NOT NULL DEFAULT '0',
  `reports_comments` tinyint(4) NOT NULL DEFAULT '0',
  `reports_download` tinyint(4) NOT NULL DEFAULT '0',
  `reports_upload` tinyint(4) NOT NULL DEFAULT '0',
  `messages` tinyint(4) NOT NULL DEFAULT '0',
  `messages_reporters` tinyint(4) NOT NULL DEFAULT '0',
  `stats` tinyint(4) NOT NULL DEFAULT '0',
  `settings` tinyint(4) NOT NULL DEFAULT '0',
  `manage` tinyint(4) NOT NULL DEFAULT '0',
  `users` tinyint(4) NOT NULL DEFAULT '0',
  `manage_roles` tinyint(4) NOT NULL DEFAULT '0',
  `checkin` tinyint(4) NOT NULL DEFAULT '1',
  `checkin_admin` tinyint(4) NOT NULL DEFAULT '0',
  `access_level` tinyint(4) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uniq_name` (`name`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COMMENT='Defines user access levels and privileges on
a deployment' AUTO_INCREMENT=5 ;
```

  - New Schema

```
CREATE TABLE IF NOT EXISTS `roles` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(32) NOT NULL,
  `description` varchar(255) NOT NULL,
  `access_level` tinyint(4) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uniq_name` (`name`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 COMMENT='Defines user access levels and privileges on a deployment' AUTO_INCREMENT=5 ;

CREATE TABLE IF NOT EXISTS `permissions_roles` (
  `role_id` int(11) NOT NULL,
  `permission_id` int(11) NOT NULL,
  PRIMARY KEY (`role_id`,`permission_id`)
) ENGINE=MyISAM COMMENT='Stores permissions assigned to roles';

CREATE TABLE IF NOT EXISTS `permissions` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(32) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name_UNIQUE` (`name`)
) ENGINE=MyISAM AUTO_INCREMENT=16 COMMENT='Stores permissions used for access control';
```

- Previous methods for checking permissions

```
admin::permissions()

admin::admin_access()
```

  are now deprecated in favor of new extensions to the Auth library:

```
Auth::instance()->has_permission($permission, $user);

Auth::instance()->admin_access($user);
// or
Auth::instance()->has_permission('admin_ui', $user);
```

  The $user parameter is optional and defaults to the logged in user if not passed.
  If you already have a user object you can check permissions directly on the objection like so:

```
$user->has_permission($permission);
```

- Access to member and admin dashboard are now governed by permissions: 'admin_ui' and 'member_ui'
  Existing custom roles will be automatically granted the admin_ui permission but this can now be added or removed as needed.