

Historical overview of Maps integration in Tiki

Maps have been supported in Tiki since 2003 (which is why some call it a GeoCMS). There is geo-related info in various places (users, trackers, image galleries, articles, blog posts, etc.). This was originally done using [MapServer](#), an active and powerful FLOSS mapping solution. However, it requires a dedicated server and more importantly, access to map data (which is not easy).

Later on, [Google Maps](#) arrived, providing an easy to use map integration to regular web sites, even without having to manage mapping data. Thus, Google Maps specific code was added to Tiki, which was convenient for a lot of people.

Then, after [a community discussion](#), starting in [Tiki7](#), [OpenLayers](#) (another option was [Mapstraction](#)) was added as a native way to handle maps, which permits the use of tiles from Google Maps, Bing Maps, OpenStreetMap (which is like Wikipedia but for maps), MapQuest (which serves OpenStreetMap maps), etc.

The [Cartograf](#) project further improved maps in [Tiki8](#), [Tiki9](#), [Tiki10](#) and [Tiki11](#). Many features were added, including Street View support.

In [Tiki12](#), all Google Maps specific code was [removed](#) in favor of using OpenLayers, so Google Maps is accessible via the [OpenLayers Google Layer](#). In addition the Natural Access project (no longer active) added further new capabilities to upload any existing line and polygon data as files and to be able to further customise how data objects were shown on the underlying map layer.

In [Tiki15](#) all MapServer-specific code has been removed. Also OpenLayers 2.x continues to be used for the map layer and integration with Tiki to allow editable map objects to be overlaid on the map layer but experimentation has started with the integration of OpenLayers 3.x.

In [Tiki20](#) integration with OpenLayers 3.x and higher was improved, and more features were exposed through the corresponding [PluginMap](#) parameters, as well as adding new tilesets, some of them using vector tiles instead of just the usual raster tiles.

Map-related documentation, as of summer of 2019, still makes reference to the different historical approaches but as it continues to be improved the older methods that are no longer used will be deprecated/removed. Volunteers to help with documentation improvement : please contact [marclaporte](#) at [tiki dot org](#)

Module map_edit_features

Introduced in [Tiki9](#), this module allows to edit features on maps and adapt tracker input/markers to avoid internal conflicts in OpenLayers.

- Related Topics
- [Maps](#)

Usage

Module map_edit_features

```
{MODULE(module=map_edit_features, parameter=value)}text {MODULE}
```

Parameters

Allows shapes to be drawn over the map.
*Required parameters are in **bold**.*
[Go to the source code](#)

Parameters	Accepted Values	Description	Default	Since
<code>trackerId</code>	digits	Tracker to store the feature in.		
<code>hiddeninput</code>	text	Hidden values to be sent over to the dialog. fieldName(value)		
<code>editdetail</code>	int	Edit the tracker item's details through a dialog after the initial item creation.	0	
<code>insertmode</code>	text	Target mode to enter after successfully inserting an item		
<code>standard</code>	int	Display the standard draw controls	1	

Related Pages

See these pages for information that applies for all modules:

- [Module](#)--Overall introduction and overview
- [Module Admin](#)--Introduction to the module admin pages for site-wide and individual module settings
 - [Assigning Modules](#)--How to select a module to configure it
 - [Module Settings Interface](#)--Interface for configuring individual module settings
 - [Module Settings Parameters](#)--Explanation of standard parameter settings for modules
 - [Creating Custom Modules](#)--How to create user modules
- [Index of Modules](#)--Links to the documentation of individual modules
- [User Modules](#)--how (if the feature is active) users can choose and place their modules
- [Mods Type Modules](#)--links to modules that are installed as components