# Basic Architecture

# Libraries

TPT libraries are divided into Editor support and Runtime support, with separate assembly definitions.

## The Editor Library

This includes support for the custom Selection Inspector, the Tile+Brush, several custom editor windows, diagnostic tools, and Tile+Painter. Painter is a Painting / Editing tool with separate documentation for you to read. It’s UI-Elements based and does away with the concept of brushes completely.

## The Runtime library

The most important part of the Runtime library is generically called TpLib and comprises several different subsystems. Much of what you use it for can be loosely thought of as a “Tilemap Database” or TMDB.

By using the TMDB you generally don’t have to keep track of what Tilemap a tile is placed on nor its position. For example, you can query TpLib for all TPT tiles with a particular tag and send them all a message to activate animation with ONE method call.

Keeping track of tiles in this fashion is much easier and becomes very important when chunks of tiles are dynamically added and removed from Tilemaps as part of TPT’s Layout system. For more details see the online documentation.

**Example: a Waypoint. Your player reaches the waypoint. You preserve its GUID in a save file so you can easily locate it again without knowing its position or even which Tilemap it’s on.**

# Organization

TPT is divided into two types of code: Static libraries and Scriptable Runtime Services (SRS).

# Static Libraries

**TpLib:** The main enabling library for this system.

**TileFabLib:** This library handles loading Tilemap archives. It's also the underlying basis of the Layout subsystem.

**TpEvents:** Tiles can post events. A MonoBehaviour or other code can interpret these, or scriptable objects called "Actions" can be used to automatically handle the events.

**TpTileUtils:** A library of utilities related to Tilemaps and tiles.

**TpServiceManager:** Managers Scriptable Runtime Services (see below).

# SRS

SRS is short for "Scriptable Runtime Service." These are Scriptable Objects which can be dynamically loaded at runtime (only) and have special features such as being able to receive Update events.

TPT uses SRS instead of static classes when possible.

1. Faster Domain reloading.
2. If you don't need certain SRS then they aren't in memory.
3. SRS can be unloaded from memory when not needed.

One loads and destroys Services with the TpServiceManager.

These basic SRS are provided with TilePlus Toolkit:

- TpMessaging
- TpSpawner
- TpTweener
- TpTilePositionDb

You can easily create SRS for your own use, see the online documentation. The Layout demo uses custom SRS for game state, file access, and the customized layout controller for the Layout system.

Using a SRS for game state is something that one shouldn't normally do as it's using the service like a Singleton which is generally frowned upon. Although a SRS makes a great singleton, there's no 'Instance' property, the only access is via the service locator.

---

Revision #6
Created 2025-07-15 13:42:35 UTC by Vonchor

Updated 2025-08-26 15:06:30 UTC by Vonchor