

Namespaces and Interfaces

Namespaces

The GUI formatter for the Brush and Selection inspectors displays information in class-hierarchical order. But it needs to know what not to display, otherwise it will breeze through the class hierarchy all the way to `UnityEngine.Object`.

Therefore, by default it ignores anything outside of specific namespaces. The `TilePlus` namespace is hard-coded in.

The configuration editor has a `Namespaces` text field where you can provide a comma-delimited list of namespaces to use. The default for that text field includes `TilePlusDemo`.

When creating your own tile classes, place the namespace that you're using in this list. Don't forget commas! Note that if you add a namespace, attributes are still required to display information.

For example, if you were to add the `UnityEngine.Tilemaps.Tile` namespace then the `TilePlusBase`' base class of `Tile` would not appear in a foldout.

- **Note: while you can certainly add this namespace and add `TilePlus` attributes to your own subclasses of `Tile`, the results might not be what you expect, and any editing changes would affect the project asset; i.e., ALL placed Tiles.**

Please click the `Reload` button in the configuration editor when you change this. Also, be aware that if you click "Reset To Defaults" that you'll need to re-add the namespaces!

Interfaces

`ITilePlus` specifies several properties and a few methods that are common to all tiles subclassed from `TilePlusBase` since that class implements everything in the interface.

Please note that any subclasses of `TilePlusBase` using `ITilePlus` properties with default members need to specify the interface to 'override' the defaults. This can be seen in the tiles which support simulation (`TpSlideShow`, `TpAnimatedTile`, and `TpFlexAnimatedTile`).

`ITpPersistence` specifies properties required for tiles using `TpLib`'s save/restore framework.

ITpMessaging specifies properties required for tiles using TpLib's messaging framework.

ITpSpawnUtilClient specifies properties required for tiles which spawn prefabs or paint tiles when using the SpawningUtil library methods.

ITpMessaging and ITpPersistence are the interfaces you'll most likely implement if creating your own tiles and you want to use TpLib's messaging and save/restore frameworks. If you don't want to use those then you can ignore those interfaces.

Revision #3

Created 2025-07-14 11:38:39 UTC by Vonchor

Updated 2025-08-13 17:43:26 UTC by Vonchor