

Exposition

A TileFab can encompass all the tiles and Prefabs contained within a Tilemap hierarchy (Grid with child Tilemaps). The simplest use for this sort of a TileFab is to just load the whole thing at once to populate an area of a scene with tiles in one operation.

However, when you create a TileFab from a Grid Selection they're more like rectangular pieces of layer cake, as seen in this crude illustration:

[ChunkLayers.png](#)

Rather than repeat "TileFab from a Grid Selection" let's just call this a Chunk. This section is mostly about ways to use Chunks.

A Chunk is a TileFab, but there are several important differences regarding how position data is stored.

TileFab

- The CellBounds (a BoundsInt) of each Tilemap are used to limit what is stored in each Bundle.
- The stored BoundsInt for each Bundle is the CellBounds of each Tilemap.
- The stored positions of tiles and Prefabs in the Bundles are the raw positions from the Tilemaps.
- The `m_FromGridSelection` field is false.

Chunk

- The Grid Selection defines the limiting area.
- The stored BoundsInt for the Chunk always has a zero-based origin. The size is that of the Grid Selection.
- The stored positions of tiles and Prefabs are adjusted to have a zero-based origin relative to the lower-left corner of the Chunk.
- The `m_FromGridSelection` field is true.

This means that a Chunk's size is always defined by its BoundsInt's size. All stored positions in a Chunk are relative to the position of its BoundsInt, i.e., all positions are relative to the lower-left corner (the Position property) of the Chunk.

Or you can think of it as "relative addressing" since you can paint a Chunk anywhere but the tiles within the chunk retain the same relative positions.

Conversely, a TileFab's stored positions are the raw positions directly from the Tilemap. If you paint a Tilefab with the Painter, all objects' (tiles and Prefabs) positions are added to the mouse pointer

position.

TileFabs are good for saving and loading entire Tilemaps. In real life, Chunks are more useful.

If you paint a Chunk or one of its Bundles using Tile+Painter, you'll notice that the mouse pointer position is always at the bottom-left of the group of tiles in the Chunk.

[Chunk.png](#)

This is because Chunks are organized in the scene using RectInts, and a RectInt's position is the lower-left corner. This is important to remember, as you'll see later.

Ways To Use Them

- Load a Chunk (or any TileFab) anywhere you want.
- Paint TileFabs, Chunks, or Bundles using Tile+Painter.
- Load automatically as the camera moves.

Loading

The easiest way is to use a TpAnimZoneLoader Tile. One can paint one of those Tiles on a Tilemap, view its fields using Tile+Brush or Tile+Painter, and set up a trigger zone. Your program periodically sends the position of your Player (or something else) to this Tile using the TpMessaging library. The Tile posts a Trigger event when that position is within the trigger zone. Your code observes this and calls a method in TileFabLib to load some TileFab.

Please note that this is a 'legacy' tile from earlier releases: it may not work properly with the layout system.

You can also call the LoadTileFab method in TileFabLib. This provides fine-grained control over the process, including optional filtering during the load process.

Painting: See the Tile+Painter documentation.

Automatically: Monitor the position of your camera and add/delete chunks as they move in and out of the camera view. There's an example of this in the Chunking demo.

Revision #7

Created 8 July 2025 11:18:12 by Vonchor

Updated 22 July 2025 17:31:18 by Vonchor