The Super-Grid

When you initialize a ZM with a chunk size and world origin you're defining a higher-dimension or super-grid (sGrid) virtually layered on top of the normal Tilemap Grid.

In the illustration below, one Chunk is any number of Tilemap locations, from 4x4 to 128x128 or any reasonable value. If the chunk size is 4x4 then this 5x4 super-grid below actually comprises 20x16 tile positions. The position of a chunk is its origin, the lower-left corner.

SuperGrid.png

The sGrid coordinate system's origin is the same as the world origin provided when the ZM instance is created. It's also important to understand that each sGrid only applies to one ZM, even those created with the same chunk size and world origin. ZMs are completely unaware of each other, by design.

The array of Chunks is a sGrid virtually layered on top of the Tilemap's grid. Array elements are called Zones and are addressable using Locators. Each Zone comprises one Chunk.

Locators are RectInts, which are a 2D version of a BoundsInt: a RectInt is also a struct and its position is also in the lower-left corner. They're used throughout ZMs to define Zones, and to access and/or query data from a ZM instance.

Each Locator must be aligned to the sGrid of the ZM. This means that its position must be divisible by the chunk size, e.g., a chunk size of 4 means the sGrid addresses must be divisible by 4. The size of each locator is usually the size of a chunk, but doesn't have to be, as we'll see later.

Here's a detail of one Zone, outlined in red. Each Zone is aligned with the sGrid. The size of each Zone is the size of a Chunk. In this example, the size of a Chunk and of Zones is 4 x 4 Grid locations. Hence, the sGrid is 4x the Tilemap Grid.

SuperGridandLocator.png

The Locator of a Chunk can be used to locate a particular chunk on the sGrid just like a Tilemap position addresses a specific location on a Tilemap.

There are several ZM instance methods to convert between Tilemap Grid address space and sGrid address space.

- GetLocatorForGridPosition takes a Tilemap Grid position and returns a Locator.
- GetLocatorForWorldPosition takes a world coordinate position and returns a Locator. There are also instance methods to check for and convert to sGrid alignment.
- IsAlignedToGrid takes a Tilemap Grid position and returns true if it's aligned to the sGrid.

- AlignToGrid takes a Tilemap Grid position and returns one that’s aligned to the sGrid.
- GetLocatorForSgridPosition takes a sGrid position and returns a Locator.
- GetGridPositionForSgridPosition takes a sGrid position and returns a Tilemap Grid Position.

You don’t need to use the sGrid, just be aware of it. But if you do need the conversions offered by the last two methods above, they’re available for your use. Note that the conversions will produce different results on ZMs with different chunk sizes and/or world origins. In a way, a chunk is like a giant tile and the sGrid is like the Tilemap grid with bigger element sizes.

The GetLocator for GridPosition and WorldPosition methods have an align parameter which if true (the default value) will automatically use the AlignToGrid method on the Tilemap Grid position.

Referring to the above illustration, any Tilemap grid coordinate within the Zone will return the Locator for the Zone when AlignToGrid is used. IsAlignedToGrid will only return true when the Tilemap grid coordinate matches the exact position of the Locator, shown here as X.

Revision #3
Created 2025-07-08 14:12:05 UTC by Vonchor
Updated 2025-07-08 14:24:25 UTC by Vonchor