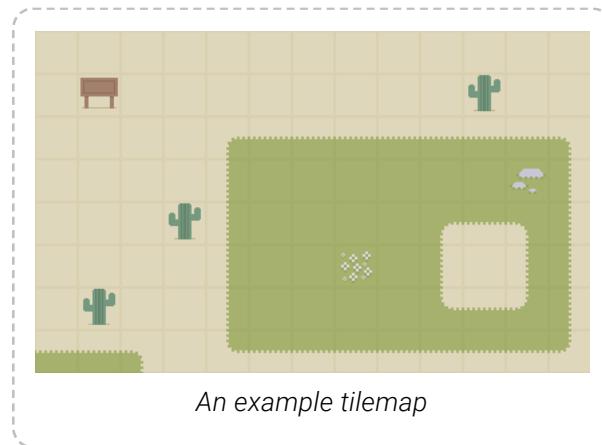


# TILEMAP

**View online:** <https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/tilemap>

The **Tilemap** object allows tile-based projects to be designed more easily. The object's tilemap can also be edited in the [layout view](#) using the [Tilemap Bar](#).



Tilemaps also have significant performance benefits over achieving the same results with other kinds of objects, such as arranging a grid of Sprites. The Tilemap object can optimise collision detection and rendering in a way that scales well even with extremely large Tilemap objects.

For information about editing tilemaps in Construct, see the manual entry for the [Tilemap Bar](#).

A useful behavior to use to move objects around on top of the Tilemap object is the [Tile movement behavior](#).

## Scripting

When using JavaScript or TypeScript coding, the features of this object can be accessed via the [ITilemapInstance script interface](#).

## Tilemap object image

The image used for the Tilemap object is the *tileset*. This is an image that contains every different tile that can be used in the tilemap. The tiles can also be offset and spaced, but this is not normally necessary. The tileset image appears in the Tilemap Bar after selecting the object, allowing you to choose which tiles to draw with.

## Collisions

When testing for collisions with a Tilemap object, empty (erased) tiles count as not colliding, and by default all other tiles count as colliding. A custom collision polygon can be set, or collisions disabled, for individual tiles by double-clicking a tile in the Tilemap bar. The image editor will appear for the tile, where the collision polygon can be modified, or disabled completely by

unticking the *Use collision* property, or right-clicking on the image and selecting *Toggle collision polygon*.

## Tile IDs

Each tile in the tileset has a zero-based index to identify it. This starts with the top-left tile and increments horizontally in rows. The tile ID can easily be seen by hovering the mouse over a tile in the Tilemap Bar. The tile ID is useful for comparing or setting tiles at runtime with the object's conditions, actions and expressions.

## Tile positions

When using tiles in the object's conditions, actions and expressions, positions are generally given in *tiles* instead of layout co-ordinates. You can convert between tile positions and layout co-ordinates using the *PositionToTileX/Y* and *TileToPositionX/Y* expressions.

## Inappropriate uses of Tilemaps

Don't use tilemaps to display large images where every tile in the tilemap is different. This makes it needlessly less efficient to render the image, since it is rendered one tile at a time when you could have just used a Sprite.

## Tilemap Properties

---

### Image

Click the *Edit* link to edit the tileset image from which tiles are drawn.

---

### Initially visible

Choose whether the object is visible or invisible at the start of the layout.

---

### Tile width

### Tile height

The size of tiles in the tilemap, in pixels.

---

### Tile X offset

### Tile Y offset

The offset in pixels of the top-left tile in the tileset image. This is not normally necessary and is provided mainly for compatibility with existing tileset images that have the tiles drawn at an offset.

---

### Tile X spacing

### Tile Y spacing

The spacing in pixels between tiles in the tileset image. This is not normally necessary and is provided mainly for compatibility with existing tileset images that have the tiles drawn apart from each other.

## Tilemap conditions

---

### Compare tile at

Compare the tile ID at a position in the tilemap.

---

### Compare tile state at

Test whether a tile at a position in the tilemap is flipped or rotated from its normal state.

---

### On image URL loaded

### On image URL failed to load

Triggered when *Load image from URL* finishes downloading the image and is ready to display it, or if the load fails.

---

### Brush exists

Check if a tilemap brush exists

## Tilemap actions

---

### Download

Invoke a download of the current tilemap data (from the *TilesJSON* expression) as a JSON file. This can be useful for in-game level editors.

---

### Load

Load the current tiles from a string of JSON data from a previous use of the *TilesJSON* expression.

---

### Erase tile

Erase the tile at a position.

---

### Erase tile range

Erase a rectangular area of tiles in the tilemap.

---

### Erase tile with brush

Erase tiles using a brush created in the [Tilemap Brush Editor](#). When using an auto tiling brush, the specified position will be modified, along with the surrounding eight positions. If a patch brush is used the whole patch will be erased.

---

### **Erase tile with brush (by name)**

Like **Erase tile with brush**, but allows you to specify the name of the brush using a string.

---

### **Erase tiles with patch brush**

Erase tiles using a patch brush. This is similar to **Erase tile with brush**, with the additional options to specify to *mirror*, *flip* and *rotate* the brush.

---

### **Erase tiles with patch brush (by name)**

Like **Erase tiles with patch brush**, but allows you to specify the name of the brush using a string.

---

### **Set tile**

Set the tile at a position in the tilemap by its tile ID. The tile that is set can also optionally be flipped or rotated.

---

### **Set tile range**

As with *Set tile*, but sets a rectangular area of tiles in the tilemap.

---

### **Set tile state**

Set the tile flipped or rotated state at a position in the tilemap. The tile ID is not changed.

---

### **Set tile state range**

Set the flipped or rotated state for a rectangular area of tiles in the tilemap. None of the tile IDs in the rectangular area are changed.

---

### **Set tile with brush**

Set tiles using a brush created in the [Tilemap Brush Editor](#). When using an auto tiling brush, the specified position will be modified, along with the surrounding eight positions. Using a patch brush modifies all the tiles covered by the patch, the selected position corresponds to the top left of the patch.

---

### **Set tile with brush (by name)**

Like **Set tile with brush**, but allows you to specify the name of the brush using a string.

## Set tile with patch brush

Set tiles using a patch brush. This is similar to **Set tile with brush**, with the additional options to specify to *mirror*, *flip* and *rotate* the brush. The **Invalid indexes** parameter allows you to choose what should happen when a patch brush doesn't have a valid tile set at a given position, **Erase** deletes the corresponding tile, while **Ignore** leaves the tile unchanged.

## Set tile with patch brush (by name)

Like **Set tile with patch brush**, but allows you to specify the name of the brush using a string.

## Load image from URL

Load a new tilemap image from a given URL. It is not used until the image has finished downloading, and *On image URL loaded* triggers. Images loaded from different domains are subject to the same cross-domain restrictions as AJAX requests - for more information see the section on cross-domain in the **AJAX** object. Data URIs can also be passed as an image, e.g. from a canvas snapshot or camera image.

# Tilemap expressions

## TilesJSON

Retrieve the tile data in JSON format, which can be loaded in again later using the *Load* action. Note this differs from the built-in **AsJSON** expression, which returns the entire object state (including position, size, behaviors etc), whereas **TilesJSON** returns only the tile data.

## MapDisplayWidth

## MapDisplayHeight

The size of the displayed tilemap in tiles. For example if a Tilemap is 320px wide with tiles 32px wide, the display width is 10 as at this size it can fit 10 tiles in to the width.

## PositionToTileX(x)

## PositionToTileY(y)

Convert an X or Y layout co-ordinate in to the corresponding tile number in the tilemap. For example, this can be used to get the tile position under the mouse.

## SnapX(x)

## SnapY(y)

Snap an X or Y layout co-ordinate to the nearest tile. This also returns a layout co-ordinate, but aligned to the nearest tile in the tilemap.

## TileAt(x, y)

Return the tile ID at a position in the tilemap. Note the position is given in tiles, not layout co-ordinates. If the tile at the given position is empty (has been erased), the expression returns -1.

---

**TileWidth****TileHeight**

The width and height of each tile, as specified in the Tilemap properties.

---

**TileToPositionX(x)****TileToPositionY(y)**

Convert a tile position to layout co-ordinates. For example, this can be used to position a Sprite object on top of a given tile.

---

**BrushWidth****BrushHeight**

The width and height (in tiles) of a patch brush. Auto tiling brushes always return 0.