SFML

* [Main Page](http://docs.google.com/index.htm)
* [Modules](http://docs.google.com/modules.htm)
* [Classes](http://docs.google.com/annotated.htm)
* [Files](http://docs.google.com/files.htm)
* [Class List](http://docs.google.com/annotated.htm)
* [Class Index](http://docs.google.com/classes.htm)
* [Class Hierarchy](http://docs.google.com/hierarchy.htm)
* [Class Members](http://docs.google.com/functions.htm)
* **sf**
* [VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm)

[Public Member Functions](#_gjdgxs) | [List of all members](http://docs.google.com/classsf_1_1VertexArray-members.htm)

sf::VertexArray Class Reference

[Graphics module](http://docs.google.com/group__graphics.htm)

Define a set of one or more 2D primitives. [More...](http://docs.google.com/classsf_1_1VertexArray.htm#details)

#include <[VertexArray.hpp](http://docs.google.com/VertexArray_8hpp_source.htm)>

Inheritance diagram for sf::VertexArray:



| Public Member Functions | |
| --- | --- |
|  | [VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm#a15729e01df8fc0021f9774dfb56295c1) () |
|  | Default constructor. |
|  | |
|  | [VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm#abf85f4baff4c282e2d07ea97b5913aad) ([PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) type, unsigned int vertexCount=0) |
|  | Construct the vertex array with a type and an initial number of vertices. |
|  | |
| unsigned int | [getVertexCount](http://docs.google.com/classsf_1_1VertexArray.htm#a683fa176683f3b0343bab164608378f9) () const |
|  | Return the vertex count. |
|  | |
| [Vertex](http://docs.google.com/classsf_1_1Vertex.htm) & | [operator[]](http://docs.google.com/classsf_1_1VertexArray.htm#a5db1da21b170ecf4c52d49030db385fd) (unsigned int index) |
|  | Get a read-write access to a vertex by its index. |
|  | |
| const [Vertex](http://docs.google.com/classsf_1_1Vertex.htm) & | [operator[]](http://docs.google.com/classsf_1_1VertexArray.htm#a0e95bde955c929651b0d9f8f00a354be) (unsigned int index) const |
|  | Get a read-only access to a vertex by its index. |
|  | |
| void | [clear](http://docs.google.com/classsf_1_1VertexArray.htm#a3654c424aca1f9e468f369bc777c839c) () |
|  | Clear the vertex array. |
|  | |
| void | [resize](http://docs.google.com/classsf_1_1VertexArray.htm#a9884c43c4f5ba152046ab3a5c91efb3b) (unsigned int vertexCount) |
|  | Resize the vertex array. |
|  | |
| void | [append](http://docs.google.com/classsf_1_1VertexArray.htm#a80c8f6865e53bd21fc6cb10fffa10035) (const [Vertex](http://docs.google.com/classsf_1_1Vertex.htm) &vertex) |
|  | Add a vertex to the array. |
|  | |
| void | [setPrimitiveType](http://docs.google.com/classsf_1_1VertexArray.htm#aa38c10707c28a97f4627ae8b2f3ad969) ([PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) type) |
|  | Set the type of primitives to draw. |
|  | |
| [PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) | [getPrimitiveType](http://docs.google.com/classsf_1_1VertexArray.htm#af2205f76fe98fb3cf1f303f25d43c045) () const |
|  | Get the type of primitives drawn by the vertex array. |
|  | |
| [FloatRect](http://docs.google.com/classsf_1_1Rect.htm) | [getBounds](http://docs.google.com/classsf_1_1VertexArray.htm#a741d1b1acbb175289eab37bbf49cbb24) () const |
|  | Compute the bounding rectangle of the vertex array. |
|  | |

## Detailed Description

Define a set of one or more 2D primitives.

[sf::VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm) is a very simple wrapper around a dynamic array of vertices and a primitives type.

It inherits [sf::Drawable](http://docs.google.com/classsf_1_1Drawable.htm), but unlike other drawables it is not transformable.

Example:

[sf::VertexArray](http://docs.google.com/classsf_1_1VertexArray.htm) lines([sf::LinesStrip](http://docs.google.com/group__graphics.htm#gga5ee56ac1339984909610713096283b1ba5b09910f5d0f39641342184ccd0d1de3), 4);

lines[0].position = [sf::Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)(10, 0);

lines[1].position = [sf::Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)(20, 0);

lines[2].position = [sf::Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)(30, 5);

lines[3].position = [sf::Vector2f](http://docs.google.com/classsf_1_1Vector2.htm)(40, 2);

window.draw(lines);

See Also[sf::Vertex](http://docs.google.com/classsf_1_1Vertex.htm)

Definition at line [45](http://docs.google.com/VertexArray_8hpp_source.htm#l00045) of file [VertexArray.hpp](http://docs.google.com/VertexArray_8hpp_source.htm).

## Constructor & Destructor Documentation

| sf::VertexArray::VertexArray | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Default constructor.

Creates an empty vertex array.

| | sf::VertexArray::VertexArray | ( | [PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) | *type*, | | --- | --- | --- | --- | |  |  | unsigned int | *vertexCount* = 0 | |  | ) |  |  | | explicit |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Construct the vertex array with a type and an initial number of vertices.

Parameters

| type | Type of primitives |
| --- | --- |
| vertexCount | Initial number of vertices in the array |

## Member Function Documentation

| void sf::VertexArray::append | ( | const [Vertex](http://docs.google.com/classsf_1_1Vertex.htm) & | *vertex* | ) |  |
| --- | --- | --- | --- | --- | --- |

Add a vertex to the array.

Parameters

| vertex | [Vertex](http://docs.google.com/classsf_1_1Vertex.htm) to add |
| --- | --- |

| void sf::VertexArray::clear | ( |  | ) |  |
| --- | --- | --- | --- | --- |

Clear the vertex array.

This function removes all the vertices from the array. It doesn't deallocate the corresponding memory, so that adding new vertices after clearing doesn't involve reallocating all the memory.

| [FloatRect](http://docs.google.com/classsf_1_1Rect.htm) sf::VertexArray::getBounds | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Compute the bounding rectangle of the vertex array.

This function returns the axis-aligned rectangle that contains all the vertices of the array.

ReturnsBounding rectangle of the vertex array

| [PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) sf::VertexArray::getPrimitiveType | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Get the type of primitives drawn by the vertex array.

ReturnsPrimitive type

| unsigned int sf::VertexArray::getVertexCount | ( |  | ) | const |
| --- | --- | --- | --- | --- |

Return the vertex count.

ReturnsNumber of vertices in the array

| [Vertex](http://docs.google.com/classsf_1_1Vertex.htm)& sf::VertexArray::operator[] | ( | unsigned int | *index* | ) |  |
| --- | --- | --- | --- | --- | --- |

Get a read-write access to a vertex by its index.

This function doesn't check *index*, it must be in range [0, [getVertexCount()](http://docs.google.com/classsf_1_1VertexArray.htm#a683fa176683f3b0343bab164608378f9) - 1]. The behaviour is undefined otherwise.

Parameters

| index | Index of the vertex to get |
| --- | --- |

ReturnsReference to the index-th vertex See Also[getVertexCount](http://docs.google.com/classsf_1_1VertexArray.htm#a683fa176683f3b0343bab164608378f9)

| const [Vertex](http://docs.google.com/classsf_1_1Vertex.htm)& sf::VertexArray::operator[] | ( | unsigned int | *index* | ) | const |
| --- | --- | --- | --- | --- | --- |

Get a read-only access to a vertex by its index.

This function doesn't check *index*, it must be in range [0, [getVertexCount()](http://docs.google.com/classsf_1_1VertexArray.htm#a683fa176683f3b0343bab164608378f9) - 1]. The behaviour is undefined otherwise.

Parameters

| index | Index of the vertex to get |
| --- | --- |

ReturnsConst reference to the index-th vertex See Also[getVertexCount](http://docs.google.com/classsf_1_1VertexArray.htm#a683fa176683f3b0343bab164608378f9)

| void sf::VertexArray::resize | ( | unsigned int | *vertexCount* | ) |  |
| --- | --- | --- | --- | --- | --- |

Resize the vertex array.

If *vertexCount* is greater than the current size, the previous vertices are kept and new (default-constructed) vertices are added. If *vertexCount* is less than the current size, existing vertices are removed from the array.

Parameters

| vertexCount | New size of the array (number of vertices) |
| --- | --- |

| void sf::VertexArray::setPrimitiveType | ( | [PrimitiveType](http://docs.google.com/group__graphics.htm#ga5ee56ac1339984909610713096283b1b) | *type* | ) |  |
| --- | --- | --- | --- | --- | --- |

Set the type of primitives to draw.

This function defines how the vertices must be interpreted when it's time to draw them:

* As points
* As lines
* As triangles
* As quads The default primitive type is sf::Points.

Parameters

| type | Type of primitive |
| --- | --- |

The documentation for this class was generated from the following file:

* [VertexArray.hpp](http://docs.google.com/VertexArray_8hpp_source.htm)

Copyright � Laurent Gomila  ::  Documentation generated by [doxygen](http://www.doxygen.org/)  ::