| | [**Overview**](http://docs.google.com/index-overview-summary.html) | [**Project**](http://docs.google.com/project-summary.html) | **Class** | [**Tree**](http://docs.google.com/project-tree.html) | [**Deprecated**](http://docs.google.com/index-deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | | --- | --- | --- | --- | --- | --- | | | ***CarnegieMellonGraphics*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/CarnegieMellonGraphics2/KeyModifiers.html) |
| SUMMARY:  INNER | [FIELD](#30j0zll) | [CONSTR](#1fob9te) | [METHOD](#3znysh7) | DETAIL:  [FIELD](#2et92p0) | [CONSTR](#2jxsxqh) | [METHOD](#4i7ojhp) |  |

## **CarnegieMellonGraphics2**

Class KeyModifiers

   in [CarnegieMellonGraphics.h](http://docs.google.com/CarnegieMellonGraphics.h.html)

class **KeyModifiers**

The KeyModifiers objects act as a "set" whose domain is that of the modifier keys that are available on a given system (usually control, shift, and alt).

Users can only work with the named sets provided, and only operate on them through the overloaded operators : | = Union, & = Intersection, and ~ = Compliment. Equality and stream output support are also provided.

So for example, the set of modifers associated with holding down the control key would simply be KeyModifiers::CONTROL. If you were to hold down both the shift and alt keys, the modifier set would be equal to

KeyModifiers::SHIFT | KeyModifiers::ALT

(union of the two singletons) or

KeyModifiers::SHIFT\_AND\_ALT

or

KeyModifiers::ALT\_AND\_SHIFT

Predefined names have been provided for all subsets (the powerset!) of the entire domain, since it is a small set this is reasonable.

| **Field Summary** | |
| --- | --- |
| static const KeyModifiers | [**ALL\_MODIFIERS**](#tyjcwt)            Matches when all modifiers are active |
| static const KeyModifiers | [**ALT**](#3dy6vkm)            Matches the "alt" key |
| static const KeyModifiers | [**ALT\_AND\_CONTROL**](#1t3h5sf)            Matches if both "control" and "alt" are active |
| static const KeyModifiers | [**ALT\_AND\_SHIFT**](#4d34og8)            Matches if both "shift" and "alt" are active |
| static const KeyModifiers | [**ANY\_MODIFIERS**](#2s8eyo1)            Matches any set of modifiers |
| static const KeyModifiers | [**CONTROL**](#17dp8vu)            Matches "control" key |
| static const KeyModifiers | [**CONTROL\_AND\_ALT**](#3rdcrjn)            Matches if both "control" and "alt" are active |
| static const KeyModifiers | [**CONTROL\_AND\_SHIFT**](#26in1rg)            Matches if both "control" and "shift" are active |
| static const KeyModifiers | [**NO\_MODIFIERS**](#lnxbz9)            Matches if there are no active modifiers |
| static const KeyModifiers | [**SHIFT**](#35nkun2)            Matches the "shift" key |
| static const KeyModifiers | [**SHIFT\_AND\_ALT**](#1ksv4uv)            Matches if both "shift" and "alt" are active |
| static const KeyModifiers | [**SHIFT\_AND\_CONTROL**](#44sinio)            Matches if both "control" and "shift" are active |

| **Constructor Summary** | |
| --- | --- |
|  | [**KeyModifiers**](#z337ya)() |
| private | [**KeyModifiers**](#1y810tw)( int v ) |

| **Method Summary** | |
| --- | --- |
| bool | [**operator!=**](#2xcytpi)( const KeyModifiers& rhs ) const            != Inequality tests on KeyModifiers |
| KeyModifiers& | [**operator&=**](#3whwml4)( const KeyModifiers& rhs )            &= Intersection-Assignment operator on KeyModifiers |
| bool | [**operator==**](#qsh70q)( const KeyModifiers& rhs ) const            == Equality tests on KeyModifiers |
| KeyModifiers& | **operator|=**( const KeyModifiers& rhs )            |= Union-Assignment operator on KeyModifiers |
| KeyModifiers& | operator ~()            ~ Compliment operator on KeyModifiers |

| **Field Detail** |
| --- |

### ALL\_MODIFIERS

public static const KeyModifiers **ALL\_MODIFIERS**;

Matches when all modifiers are active

### ALT

public static const KeyModifiers **ALT**;

Matches the "alt" key

### ALT\_AND\_CONTROL

public static const KeyModifiers **ALT\_AND\_CONTROL**;

Matches if both "control" and "alt" are active

### ALT\_AND\_SHIFT

public static const KeyModifiers **ALT\_AND\_SHIFT**;

Matches if both "shift" and "alt" are active

### ANY\_MODIFIERS

public static const KeyModifiers **ANY\_MODIFIERS**;

Matches any set of modifiers

### CONTROL

public static const KeyModifiers **CONTROL**;

Matches "control" key

### CONTROL\_AND\_ALT

public static const KeyModifiers **CONTROL\_AND\_ALT**;

Matches if both "control" and "alt" are active

### CONTROL\_AND\_SHIFT

public static const KeyModifiers **CONTROL\_AND\_SHIFT**;

Matches if both "control" and "shift" are active

### NO\_MODIFIERS

public static const KeyModifiers **NO\_MODIFIERS**;

Matches if there are no active modifiers

### SHIFT

public static const KeyModifiers **SHIFT**;

Matches the "shift" key

### SHIFT\_AND\_ALT

public static const KeyModifiers **SHIFT\_AND\_ALT**;

Matches if both "shift" and "alt" are active

### SHIFT\_AND\_CONTROL

public static const KeyModifiers **SHIFT\_AND\_CONTROL**;

Matches if both "control" and "shift" are active

| **Constructor Detail** |
| --- |

### KeyModifiers

public **KeyModifiers**();

### KeyModifiers

private **KeyModifiers**( int v );

| **Method Detail** |
| --- |

### operator!=

public bool **operator!=**( const KeyModifiers& rhs ) const;

!= Inequality tests on KeyModifiers

### operator&=

public KeyModifiers& **operator&=**( const KeyModifiers& rhs );

&= Intersection-Assignment operator on KeyModifiers

### operator==

public bool **operator==**( const KeyModifiers& rhs ) const;

== Equality tests on KeyModifiers

### operator|=

public KeyModifiers& **operator|=**( const KeyModifiers& rhs );

|= Union-Assignment operator on KeyModifiers

### operator~

public KeyModifiers& operator ~();

~ Compliment operator on KeyModifiers

| | [**Overview**](http://docs.google.com/index-overview-summary.html) | [**Project**](http://docs.google.com/project-summary.html) | **Class** | [**Tree**](http://docs.google.com/project-tree.html) | [**Deprecated**](http://docs.google.com/index-deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | | --- | --- | --- | --- | --- | --- | | | ***CarnegieMellonGraphics*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**FRAMES**](http://docs.google.com/index.html)    [**NO FRAMES**](http://docs.google.com/CarnegieMellonGraphics2/KeyModifiers.html) |
| SUMMARY:  INNER | [FIELD](#30j0zll) | [CONSTR](#1fob9te) | [METHOD](#3znysh7) | DETAIL:  [FIELD](#2et92p0) | [CONSTR](#2jxsxqh) | [METHOD](#4i7ojhp) |  |