Color_tpl< T >

`#include <Cry_Color.h>`

Inherited by SerializableColor_tpl< T >.

Public Member Functions

ILINE	**Color_tpl** (T _x, T _y, T _z, T _w)
ILINE	**Color_tpl** (T _x, T _y, T _z)
ILINE	**Color_tpl** (const unsigned int abgr)
ILINE	**Color_tpl** (const f32 c)
ILINE	**Color_tpl** (const **ColorF** &c)
ILINE	**Color_tpl** (const **ColorB** &c)
ILINE	**Color_tpl** (const **ColorF** &c, float fAlpha)
ILINE	**Color_tpl** (const **Vec3** &c, float fAlpha)
void	**Clamp** (float Min=0, float Max=1.0f)
void	**ScaleCol** (float f)
float	**Luminance** () const
ILINE float	**Max** () const
float	**NormalizeCol** (**ColorF** &out) const
ILINE	**Color_tpl** (const **Vec3** &vVec)
ILINE **Color_tpl** &	**operator=** (const **Vec3** &v)
ILINE **Color_tpl** &	**operator=** (const **Color_tpl** &c)
ILINE T &	**operator[]** (int index)
ILINE T	**operator[]** (int index) const
ILINE void	**set** (T _x, T _y, T _z, T _w)
ILINE **Color_tpl**	**operator+** () const
ILINE **Color_tpl**	**operator-** () const
ILINE **Color_tpl** &	**operator+=** (const **Color_tpl** &v)
ILINE **Color_tpl** &	**operator-=** (const **Color_tpl** &v)
ILINE **Color_tpl** &	**operator*=** (const **Color_tpl** &v)
ILINE **Color_tpl** &	**operator/=** (const **Color_tpl** &v)
ILINE **Color_tpl** &	**operator*=** (T s)
ILINE **Color_tpl** &	**operator/=** (T s)
ILINE **Color_tpl**	**operator+** (const **Color_tpl** &v) const
ILINE **Color_tpl**	**operator-** (const **Color_tpl** &v) const

ILINE Color_tpl	**operator*** (const Color_tpl &v) const
ILINE Color_tpl	**operator/** (const Color_tpl &v) const
ILINE Color_tpl	**operator*** (T s) const
ILINE Color_tpl	**operator/** (T s) const
ILINE bool	**operator==** (const Color_tpl &v) const
ILINE bool	**operator!=** (const Color_tpl &v) const
ILINE unsigned char	**pack_rgb332** () const
ILINE unsigned short	**pack_argb4444** () const
ILINE unsigned short	**pack_rgb555** () const
ILINE unsigned short	**pack_rgb565** () const
ILINE unsigned int	**pack_bgr888** () const
ILINE unsigned int	**pack_rgb888** () const
ILINE unsigned int	**pack_abgr8888** () const
ILINE unsigned int	**pack_argb8888** () const
Vec4	**toVec4** () const
Vec3	**toVec3** () const
void	**toHSV** (f32 &h, f32 &s, f32 &v) const
void	**fromHSV** (f32 h, f32 s, f32 v)
void	**clamp** (T bottom=0.0f, T top=1.0f)
void	**maximum** (const Color_tpl< T > &ca, const Color_tpl< T > &cb)
void	**minimum** (const Color_tpl< T > &ca, const Color_tpl< T > &cb)
void	**abs** ()
void	**adjust_contrast** (T c)
void	**adjust_saturation** (T s)
void	**adjust_luminance** (float newLum)
void	**lerpFloat** (const Color_tpl< T > &ca, const Color_tpl< T > &cb, float s)
void	**negative** (const Color_tpl< T > &c)
void	**grey** (const Color_tpl< T > &c)
Color_tpl< T >	**RGB2mCIE** () const mCIE: adjusted to compensate problems of DXT compression (extra bit in green channel causes green/purple artifacts).
Color_tpl< T >	**mCIE2RGB** () const mCIE: adjusted to compensate problems of DXT compression (extra bit in green channel causes green/purple artefacts).
void	**rgb2srgb** ()
void	**srgb2rgb** ()
void	**GetMemoryUsage** (class ICrySizer *pSizer) const

template<>

ILINE **Color_tpl** (f32 _x, f32 _y, f32 _z, f32 _w)

template<>

ILINE **Color_tpl** (f32 _x, f32 _y, f32 _z)

template<>

ILINE **Color_tpl** (const unsigned int abgr)

template<>

ILINE **Color_tpl** (const float c)

template<>

ILINE **Color_tpl** (const **ColorF** &c)

template<>

ILINE **Color_tpl** (const **ColorB** &c)

template<>

ILINE **Color_tpl** (const **Vec3** &c, float fAlpha)

template<>

ILINE **Color_tpl** (const **ColorF** &c, float fAlpha)

Static Public Member Functions

static uint32 **ComputeAvgCol_Fast** (const uint32 dwCol0, const uint32 dwCol1)

Helper function - maybe we can improve the integration.

Public Attributes

T **r**

T **g**

T **b**

T **a**

AUTO_STRUCT_INFO

Detailed Description

template<class T>

struct Color_tpl< T >

RGBA Color structure

See also

ColorB and ColorF

Member Function Documentation

template<class T >

void Color_tpl< T >::
adjust_luminance (float newLum) inline

Optimized yet equivalent version of replacing luminance in XYZ space. Color and luminance are expected to be linear.

template<class T>

void Color_tpl< T >::
adjust_saturation (T s) inline

Approximate values for each component's contribution to luminance. Based upon the NTSC standard described in ITU-R Recommendation BT.709.