

Vec2_tpl< F >

```
#include <Cry_Vector2.h>
```

Inherits INumberVector< F, 2, Vec2_tpl< F > >.

Public Types

operatorVector< F, 2, **Vec2_tpl**< F > > **NV**

Public Member Functions

INLINE **Vec2_tpl** (type_zero)

INLINE **Vec2_tpl** (F s)

template<typename F2 >

INLINE **Vec2_tpl** (const **Vec2_tpl**< F2 > &in)

INLINE **Vec2_tpl** (F vx, F vy)

E **Vec2_tpl** & **set** (F nx, F ny)

template<class F1 >

INLINE **Vec2_tpl** (const F1 *psrc)

Vec2_tpl & **NormalizeSafe** (const struct **Vec2_tpl**< F > &safe=Vec2Constants< F >::fVec2_Zero)
Normalize if non-0, otherwise set to specified "safe" value.

Vec2_tpl **GetNormalizedSafe** (const struct **Vec2_tpl**< F > &safe=Vec2Constants< F >::fVec2_OneX)
const

INLINE F **GetLength2** () const

void **SetLength** (F fLen)

INLINE F **area** () const

INLINE **operator F*** ()

E **Vec2_tpl** & **flip** ()

E **Vec2_tpl** & **zero** ()

INE **Vec2_tpl** **rot90ccw** () const

INE **Vec2_tpl** **rot90cw** () const

INLINE F **atan2** () const

INLINE bool **IsZeroFast** (F e=(F) 0.0003) const

INE **Vec2_tpl** **Perp** () const

INLINE F **Cross** (const **Vec2_tpl** &v) const
The size of the "parallel-trapets" area spanned by the two vectors.

INLINE F **operator^** (const **Vec2_tpl** &v) const

INLINE F **operator*** (const **Vec2_tpl** &v) const

INLINE bool	operator== (const Vec2_tpl < F > &o) const
INLINE bool	operator!= (const Vec2_tpl < F > &o) const
INLINE void	SetLerp (const Vec2_tpl &p, const Vec2_tpl &q, F t)
void	SetSlerp (const Vec2_tpl &p, const Vec2_tpl &q, F t)

Static Public Member Functions

LINE **Vec2_tpl** **CreateLerp** (const **Vec2_tpl** &p, const **Vec2_tpl** &q, F t)

LINE **Vec2_tpl** **CreateSlerp** (const **Vec2_tpl** &p, const **Vec2_tpl** &q, F t)

Public Attributes

F	x
F	y
	AUTO_STRUCT_INFO

Detailed Description

```
template<class F>
struct Vec2_tpl< F >
```

General-purpose 2D vector implementation

See also

Vec2, Vec2i

Member Function Documentation

```
template<class F>
inline Vec2_tpl Vec2_tpl< F >::
Perp() const
```

Returns

A vector perpendicular to this one (this->Cross(newVec) points "up").

SetLerp()

```

template<class F>
ILINE void Vec2_tpl< F >::
SetLerp( const Vec2_tpl< F > & p,
         const Vec2_tpl< F > & q,
         F t
        )

```

inline

Linear-Interpolation between Vec3 (lerp). Example: Vec3 r=Vec3::CreateLerp(p, q, 0.345f);

```

template<class F>
void Vec2_tpl< F >::
SetSlerp( const Vec2_tpl< F > & p,
          const Vec2_tpl< F > & q,
          F t
        )

```

inline

Spherical-Interpolation between 3d-vectors (geometrical slerp). both vectors are assumed to be normalized. Example: Vec3 r=Vec3::CreateSlerp(p, q, 0.5674f);