

ost::IPv4Mask(3)

ost::IPv4Mask(3)

**NAME**

ost::IPv4Mask –

Internet addresses used specifically as masking addresses (such as '255.255.255.0') are held in the **IPv4Mask** derived object.

**SYNOPSIS**

#include &lt;address.h&gt;

Inherits **ost::IPv4Address**.**Public Member Functions****IPv4Mask** (const char \*mask)*Create the mask from a null terminated ASCII string such as '255.255.255.128'.***IPv4Address & operator=** (unsigned long addr)*Allows assignment from the return of functions like inet\_addr() or htonl().***Friends****\_\_EXPORT IPv4Host operator&** (const **IPv4Host** &addr, const **IPv4Mask** &mask)*Masks are usually used to coerce host addresses into a specific router or class domain.***Detailed Description**

Internet addresses used specifically as masking addresses (such as '255.255.255.0') are held in the **IPv4Mask** derived object.

The separate class is used so that C++ type casting can automatically determine when an **IPv4Address** object is really a mask address object rather than simply using the base class. This also allows manipulative operators for address masking to operate only when presented with a Masked address as well as providing cleaner and safer source.

**Author:**

David Sugar &lt;dyfet AT ostel DOT com&gt; Internet Address Mask such as subnet masks.

**Constructor & Destructor Documentation****ost::IPv4Mask::IPv4Mask** (const char \* mask)

Create the mask from a null terminated ASCII string such as '255.255.255.128'. **Parameters:**  
*mask* null terminated ASCII mask string.

**Member Function Documentation****IPv4Address& ost::IPv4Mask::operator=** (unsigned long addr) [inline]*Allows assignment from the return of functions like inet\_addr() or htonl().*Reimplemented from **ost::IPv4Address**.References **ost::IPv4Address::operator=()**.**Friends And Related Function Documentation****\_\_EXPORT IPv4Host operator&** (const **IPv4Host** & addr, const **IPv4Mask** & mask) [friend]

Masks are usually used to coerce host addresses into a specific router or class domain. This can be done by taking the Inet Host Address object and 'and'ing it with an address mask. This operation can be directly expressed in C++ through the & operator.

**Returns:**

a internet host address that has been masked.

**Parameters:***addr* host address to be masked by subnet.*mask* inetnet mask address object to mask by.**Author**

Generated automatically by Doxygen for GNU CommonC++ from the source code.

