

G++ 2.91.57, cygnus\cygwin-b20\include\g++\stl_pair.h 完整列表

```
/*
 *
 * Copyright (c) 1994
 * Hewlett-Packard Company
 *
 * Permission to use, copy, modify, distribute and sell this software
 * and its documentation for any purpose is hereby granted without fee,
 * provided that the above copyright notice appear in all copies and
 * that both that copyright notice and this permission notice appear
 * in supporting documentation. Hewlett-Packard Company makes no
 * representations about the suitability of this software for any
 * purpose. It is provided "as is" without express or implied warranty.
 *
 *
 * Copyright (c) 1996,1997
 * Silicon Graphics Computer Systems, Inc.
 *
 * Permission to use, copy, modify, distribute and sell this software
 * and its documentation for any purpose is hereby granted without fee,
 * provided that the above copyright notice appear in all copies and
 * that both that copyright notice and this permission notice appear
 * in supporting documentation. Silicon Graphics makes no
 * representations about the suitability of this software for any
 * purpose. It is provided "as is" without express or implied warranty.
 */

/* NOTE: This is an internal header file, included by other STL headers.
 * You should not attempt to use it directly.
 */

#ifndef __SGI_STL_INTERNAL_PAIR_H
#define __SGI_STL_INTERNAL_PAIR_H

__STL_BEGIN_NAMESPACE

template <class T1, class T2>
struct pair {
    typedef T1 first_type;
    typedef T2 second_type;

    T1 first;          // 注意，它是 public
    T2 second;          // 注意，它是 public
    pair() : first(T1()), second(T2()) {}
    pair(const T1& a, const T2& b) : first(a), second(b) {}
};

#ifdef __STL_MEMBER_TEMPLATES
// 以pair A 為pair B的初值。A, B 的對應元素型別可以不同，只要能轉換就好。
template <class U1, class U2>
```

```
    pair(const pair<U1, U2>& p) : first(p.first), second(p.second) {}
#endif
};

template <class T1, class T2>
inline bool operator==(const pair<T1, T2>& x, const pair<T1, T2>& y) {
    return x.first == y.first && x.second == y.second;
    // 第一元素和第二元素都相等，才視為相等。
}

template <class T1, class T2>
inline bool operator<(const pair<T1, T2>& x, const pair<T1, T2>& y) {
    return x.first < y.first || (!y.first < x.first) && x.second < y.second;
    // x的第一元素小於y的第一元素，或，x的第一元素不大於y的第一元素而
    // x的第二元素小於y的第二元素，才視為x小於y。
}

// 根據兩個數值，製造出一個pair
template <class T1, class T2>
inline pair<T1, T2> make_pair(const T1& x, const T2& y) {
    return pair<T1, T2>(x, y);
}

__STL_END_NAMESPACE

#endif /* __SGI_STL_INTERNAL_PAIR_H */

// Local Variables:
// mode:C++
// End:
```