Subject: Re: MSC8 compile error "Button Example"
Posted by yoco on Fri, 14 Jul 2006 12:44:39 GMT

View Forum Message <> Reply to Message

I solved this problem.

The problem caused by the default charset in Taiwan is Big-5 (similar with iso-8895-1).

In far-east country, their words are more than 255,
far more than a byte can represent,
so sometimes they hava to combine two bytes to represent one words.

In this charset,
If the ASCII of a char is smaller than 127, it will treate as a normal ANSI char.
If the ASCII of a char is bigger than 128, it will be combined with the next char to represent one far-east word.

A segment of the Ctrl.iml file
In standerd ANSI, it looks like this

but in Big-5 charset, it is...

The second qoute which to close the string literal,
was been combined with the previous byte to present one fareast word,
cause of the prevouis byte is bigger than 128.
Since there are no qoute to close the string literal,
so it raised a C2001 compile error in MS VC++.

I had wrote a simple program to translate the original .iml file to a new one,
by replace byte bigger than 128 to OCT string literal form.
Recompile, and everything is fine.

I hope this may be helpful to you.

PS. the simple program.

```
int main( int args, char* argv[] )
{
 string fn = argv[1] ;
 ifstream fin ( fn.c_str() ,ios_base::binary ) ;
 if ( !fin ) { cout << fn << " not exist!" << endl ; return 0 ;}

 string cmd( "copy" ) ;
 cmd = cmd + " " + fn + " " + fn + ".bak" ;
 system ( cmd.c_str() ) ;

 fin.close() ;
 fin.open( (fn+".bak").c_str() ) ;

 ofstream fout(fn.c_str()) ;

 char buf [5] = "\\" ;
 unsigned char ch ;
 while ( fin.get((char&)ch) ) {
  if ( ch<128 ) { fout.put(ch) ; }
  else {
   buf[1] = '0' + ch/64 ;
   buf[2] = '0' + (ch%64)/8 ;
   buf[3] = '0' + ch%8 ;
   fout << buf ;
  }
 }

}
```