**二进制数据流v0.9**

**研发三部**

模块名：<weedong/core/bstream/bstream.h>

依赖: <weedong/core/os.h>

简介(详细请看<weedong/core/bstream/bstream.h> )：

二进制数据流——提供一套针对二进制数据进行输入输出操作的流对象，并且提供指针接口对数据读写位置进行灵活调整。

结构(类)：

class CStream

{

uint32 available(void);

uint32 length(void);

void clear(void);

uint32 position(void);

void position( uint32 uiPosition );

uint8& operator [] ( int32 position );

void read( void \*pvData, uint32 uiSize );

void write( const void \*pvData, uint32 uiSize );

void read( std::string &strValue, uint32 uiSize );

void write( std::string &strValue, uint32 uiSize );

CStream& operator >> ( std::vector<uint8> &vecBuff );

CStream& operator << ( std::vector<uint8> &vecBuff );

CStream& operator << ( CStream& stream );

CStream& operator << ( const char \*v );

CStream& operator << ( const int8 &v );

CStream& operator >> ( int8 &v );

CStream& operator << ( const uint8 &v );

CStream& operator >> ( uint8 &v );

CStream& operator << ( const int16 &v );

CStream& operator >> ( int16 &v );

CStream& operator << ( const uint16 &v );

CStream& operator >> ( uint16 &v );

CStream& operator << ( const int32 &v );

CStream& operator >> ( int32 &v );

CStream& operator << ( const uint32 &v );

CStream& operator >> ( uint32 &v );

CStream& operator << ( const float &v );

CStream& operator >> ( float &v );

CStream& operator << ( const double &v );

CStream& operator >> (double &v );

};

示例：

#include <weedong/core/bstream/bstream.h>

main()

{

wd::CStream stream; //声明数据流

std::vector<uint8> buff; //声明缓存空间

stream << 3.1415 << 21 << “name”;

stream >> buff;

}