

```

doc_GThreadManager.cpp

GThreadManager.h, GThreadGroup.h, GThreadGroup.cpp,
#ifdef _GTHREAD_MANAGER_
#define _GTHREAD_MANAGER_
#include "GThreadGroup.h"
#include "GThread.h"
class GThreadManager
    : public GThreadGroup
{
public:
    static GThreadManager* GetInstance();
    GThreadGroup* CreateGroup( const char *_sKey );
    static void DeleteInstance();
private:
    static GThreadManager *_pManager;
    static const char *_sGroupName;
    GThreadManager( const char* _sKey ) : GThreadGroup( _sKey ){}
    virtual ~GThreadManager() {}
};
#endif
#ifdef _GTHREADGROUP_
#define _GTHREADGROUP_
#include "GThreadBase.h"
#include "Runnable.h"
#include "GThread.h"
#include "String.h"
#include "Mutex.h"
#include "rt_sts.h"
#include "Map.h"
class GThreadGroup
    : public GThreadBase
{
public:
    virtual int AddThread(
        const char *_sKey
        , Runnable *_pObj
        , int _nSetPriority = THREAD_PRIORITY_DEF
        , int _nSetPolicy = SCHED_OTHER
        , int _nSetScope = THREAD_SCOPE_SYSTEM
    );
    virtual int AddGroup( GThreadGroup *_pObject );
    virtual int StartThread( int _nInterval )
    {
        return this->ExecAllObject( EXEC_START_THREAD, _nInterval );
    }
    virtual int StopThread( int _nMode )
    {
        return this->ExecAllObject( EXEC_STOP_THREAD, _nMode );
    }
    virtual int ActivCount( int &nCount )
    {
        return this->ExecAllObject( EXEC_ACTIV_COUNT, _nCount );
    }
    virtual int ThreadCount( int &nCount )
    {
        return this->ExecAllObject( EXEC_THREAD_COUNT, _nCount );
    }
    virtual int CheckThread()
    {
        int nDummy = 0;
        return this->ExecAllObject( EXEC_CHECK_THREAD, nDummy );
    }
    virtual int GetMaxPriority( int &nPriority )
    {
        return this->ExecAllObject( EXEC_MAX_PRIORITY, _nPriority );
    }
    virtual int Join()
    {
        int nDummy = 0;
        return this->ExecAllObject( EXEC_THREAD_JOIN, nDummy );
    }
protected:
    static const int EXEC_START_THREAD = 0;
    static const int EXEC_STOP_THREAD = 1;
    static const int EXEC_ACTIV_COUNT = 2;
    static const int EXEC_THREAD_COUNT = 3;
    static const int EXEC_CHECK_THREAD = 4;
    static const int EXEC_MAX_PRIORITY = 5;
    static const int EXEC_THREAD_JOIN = 6;
    Map< String, GThreadBase* > oThreadMap;
    Mutex oMutex;
    const String sGroupName;
    friend class GThreadManager;

```