

```

#ifndef _rt_sts
#define _rt_sts
#include "rt_msk.h"
#define RT_S_SUCCESS __S(1)
#define RT_I_INFO __I(1)
#define RT_I_NODELETE __I(2)
#define RT_I_FID_PROC_END __I(3)
#define RT_I_SHM_NOMATCH __I(4)
#define RT_I_FID_NOTFOUND __I(5)
#define RT_I_NOMATCH __I(6)
#define RT_I_GET_ENV __I(7)
#define RT_I_RECEIVE __I(8)
#define RT_I_EXIT __I(9)
#define RT_I_ERRNO __I(10)
#define RT_I_START __I(11)
#define RT_I_SIGNAL __I(12)
#define RT_I_LOGIN __I(13)
#define RT_I_REST_DATA __I(14)
#define RT_I_SSL_OPEN __I(15)
#define RT_W_WARNING __W(1)
#define RT_W_TIMEOUT __W(2)
#define RT_W_BREAK_QUEUE __W(3)
#ifndef R2000SVR_RSP0014
#define RT_W_TCPL_INVALID_SIZE __W(4)
#endif
#define RT_E_ERROR __E(1)
#define RT_E_ELEM_RANGE __E(2)
#define RT_E_SSL_ERROR __E(3)
#define RT_E_THREAD_ALREADY_ACTIVE __E(4)
#define RT_E_THREAD_DEADLOCK __E(5)
#define RT_E_THREAD_NOT_FOUND __E(6)
#define RT_E_NO_VALUE __E(7)
#define RT_E_EXIST __E(8)
#define RT_E_NOENT __E(9)
#define RT_E_ORA_ERROR __E(10)
#define RT_E_ORA_CONNECT __E(11)
#define RT_E_DBI_NOTFOUND __E(12)
#define RT_E_SHM_GET_ERROR __E(13)
#define RT_E_SHM_CREATE_ERROR __E(14)
#define RT_E_INITIALIZE __E(15)
#define RT_E_CONNECT_ERROR __E(16)
#define RT_E_GET_ENV __E(17)
#define RT_E_UNEXPECTED_VALUE __E(18)
#define RT_E_LOCK __E(19)
#define RT_E_UNLOCK __E(20)
#define RT_E_QUEUE_PUSH_ERROR __E(21)
#define RT_E_QUEUE_POP_ERROR __E(22)
#define RT_E_INVALID __E(23)
#define RT_E_NOT_FOUND __E(24)
#define RT_E_ACCESS __E(25)
#define RT_E_AGAIN __E(26)
#define RT_E_FAULT __E(27)
#define RT_E_FBIG __E(28)
#define RT_E_IDRM __E(29)
#define RT_E_NOMEM __E(30)
#define RT_E_LISTEN __E(31)
#define RT_E_SELECT __E(32)
#define RT_E_ACCEPT __E(33)
#define RT_E_CONNECT_CLOSED __E(34)
#define RT_E_ALREADY_CONNECT __E(35)
#define RT_E_FILE_READ_ERROR __E(36)
#define RT_E_SHM_NOMATCH __E(37)
#define RT_E_NO_INITIALIZE __E(38)
#define RT_E_ALREADY_INITIALIZE __E(39)
#define RT_E_SOCKET_BLOCK __E(40)
#define RT_E_BAD_REAL_TYPE __E(41)
#define RT_E_MAX_ENTRY __E(42)
#define RT_E_FORMAT __E(43)
#define RT_E_SMTP_PROTOCOL __E(44)
#define RT_E_THREAD_FREEZE __E(45)
#define RT_E_TIMEOUT __E(46)
#define RT_E_SELECT_TIMEOUT __E(47)
#define RT_E_SELECT_INVALID __E(48)
#define RT_E_FD_NOSET __E(49)
#define RT_E_SEND_EAGAIN __E(50)
#ifndef RTS00394
#define RT_E_USE_EXT_MEM __E(51)
#define RT_E_NO_SPACE __E(52)
#define RT_E_NOWAIT __E(53)
#endif
#define RT_F_FATAL __F(1)
#define RT_F_ALLOCATE __F(2)
#define RT_F_NULL_POINTER __F(3)
#define ERR_CHECK(_sts_) (~_sts_ & MSK_M_NORMAL)
#define FATAL_CHECK(_sts_) (~_sts_ & MSK_M_FATAL)
#define ERR_RETURN(_sts_) if(ERR_CHECK(_sts_)){ return (_sts_); }
#define NULL_RETURN(_p_) if((_p_) == NULL){ return RT_F_NULL_POINTER; }
#define ALLOC_RETURN(_p_) if((_p_) == NULL){ return RT_F_ALLOCATE; }
#endif
#ifndef _rt_msk
#define _rt_msk
#define MSK_M_UNKNOWN 0x00000000
#define MSK_M_NORMAL 0x00000001
#define MSK_M_SUCCESS 0x00000002
#define MSK_M_INFORMATION 0x00000004
#define MSK_M_WARNING 0x00000010
#define MSK_M_ERROR 0x00000020
#define MSK_M_FATAL 0x00000040
#define __S(_p_) ( (_p_ << 8) + (MSK_M_SUCCESS | MSK_M_NORMAL) )
#define __I(_p_) ( (_p_ << 8) + (MSK_M_INFORMATION | MSK_M_NORMAL) )
#define __W(_p_) ( (_p_ << 8) + MSK_M_WARNING )
#define __E(_p_) ( (_p_ << 8) + MSK_M_ERROR )
#define __F(_p_) ( (_p_ << 8) + MSK_M_FATAL )
#endif

```