| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [Common/include](http://docs.google.com/index.html) - ErrorHandler.h (source / [functions](http://docs.google.com/ErrorHandler.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 2 | 2 | 100.0 % | | Date: | 2012-03-23 |  | Functions: | 0 | 0 | - | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 0 | 0 | - | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #ifndef NFT\_ERRORHANDLER\_H  2 : : #define NFT\_ERRORHANDLER\_H  3 : :   4 : :   5 : : #include <stdint.h>  6 : : #include <string>  7 : :   8 : : #include "Sgs\_Define.h"  9 : : #include "DPL.h"  10 : :   11 : : namespace nft { class ThreadManager; }  12 : :   13 : : namespace nft {  14 : :   15 : : //Contains all the error messages  16 : : class ErrorHandler {  17 : : public:  18 : : enum ErrorID {  19 : : //General : Used for printing messages not present in the   20 : : // table  21 : : eErrMessageString,  22 : : eErrFunction,  23 : : eErrRecordIndex,  24 : :   25 : : //System related  26 : : eErrUnhandled,  27 : : eErrSignalHandler,  28 : : eErrSystem,  29 : : eErrMemoryAllocation,  30 : : eErrMmap,  31 : : eErrMunmap,  32 : : eErrUnload,  33 : : eErrThread,  34 : : eErrFileOpen,  35 : : eErrFileNotFound,  36 : : eErrDataMode,  37 : : eErrTruncate,  38 : :   39 : : //Library level  40 : : eErrIndexOutOfBounds,  41 : : eErrRangeOverflow,  42 : :   43 : : //Application level  44 : : eErrCreateMemoryWindow,  45 : : eErrMessageSend,  46 : : eErrReadData,  47 : : eErrLoadData,  48 : : eErrProcessManagerInit,  49 : : eErrCmlParser,  50 : : eErrParse,  51 : : eErrReadParam,  52 : :   53 : : //Processing engine  54 : : eErrInit,  55 : : eErrExecute,  56 : : eErrGMCValueBeyondLimit,  57 : : eErrCellSizeMoreThanSF,  58 : :   59 : : //Config file parsing  60 : : eErrConfParse,  61 : : eErrConfVal,  62 : :   63 : : //SelfDiagnostics  64 : : eErrSD10,  65 : : eErrSD11,  66 : : eErrSD20,  67 : : eErrSD21,  68 : : eErrSD22,  69 : : eErrSD23,  70 : : eErrSD30\_1,  71 : : eErrSD30\_2,  72 : : eErrSD31,  73 : : eErrSD32\_1,  74 : : eErrSD32\_2,  75 : : eErrSD40,  76 : : eErrSD41\_1,  77 : : eErrSD41\_2,  78 : : eErrSD42,  79 : : eErrSD43,  80 : : eErrSD50,  81 : : eErrSD51\_1,  82 : : eErrSD51\_2,  83 : : eErrSD52,  84 : : eErrSD60,  85 : : eErrSD61,  86 : : eErrSD70,  87 : : eErrSD71,  88 : : eErrSD80,  89 : : eErrSD81,  90 : : eErrSD90,  91 : : eErrSD91,  92 : : eErrSD100,  93 : : eErrSD101,  94 : : eErrSD200,  95 : : eErrPositionOrderSD80,  96 : : eErrShapeOrderSD80,  97 : :   98 : : //END OF LIST  99 : : eErrNumElements  100 : : };  101 : :   102 : : struct ErrorMsg {  103 : : //Error ID for the error message  104 : : ErrorID mID;  105 : :   106 : : //Error message  107 : : std::string mMsg;  108 : :   109 : : // Error type as specified in the DPL  110 : : DPL::ERRTYPE mErrType;  111 : :   112 : : ERROR\_LEVEL mErrLevel;  113 : : };  114 : :   115 : : private:  116 : : DPL::ERRTYPE mErrType;  117 : : ERROR\_LEVEL mErrLevel;  118 : : unsigned int mErrCode;  119 : :   120 : : private:  121 : : //Blocked constructor  122 : : ErrorHandler();  123 : :   124 : : //Blocked copy constructor  125 : : ErrorHandler(const ErrorHandler & source);  126 : :   127 : : //Disallowed assignment operator  128 : : ErrorHandler &  129 : : operator =(const ErrorHandler & source);  130 : :   131 : : public:  132 : :   133 : : ~ErrorHandler();  134 : :   135 : : static ErrorHandler \*  136 : : instance();  137 : :   138 : : static const std::string  139 : : getErrorMsg(const ErrorID eiCode, ...);  140 : :   141 : : //This method will create the error object, store   142 : : //the error level and code, and finally throw it   143 : : //to the function catching it.  144 : : void  145 : : errorAdd(DPL::Error \*inObjSubError, const char \* inFileName,  146 : : const char \* inSourceModuleName, int32\_t inLineNum,  147 : : const ErrorID inErrorCode,  148 : : const std::string &inErrMessage,  149 : : ThreadManager \*inThreadManager = NULL) throw(DPL::Error \*);  150 : :   151 : : inline ERROR\_LEVEL  152 : : getErrorLevel();  153 : :   154 : : inline DPL::ERRTYPE  155 : : getErrorType();  156 : :   157 : : inline unsigned int  158 : : getErrorCode();  159 : :   160 : : protected:  161 : : //Table of error messages  162 : : static const ErrorMsg mMsgTable[eErrNumElements];  163 : :   164 : : public:  165 : : // Returns the error type for a corresponding error message  166 : : inline DPL::ERRTYPE  167 : 535 : getErrorType(const ErrorID eiCode) {  168 : : return mMsgTable[eiCode].mErrType; };  169 : :   170 : : inline ERROR\_LEVEL  171 : 110 : getErrorLevel(const ErrorID eiCode) {  172 : : return mMsgTable[eiCode].mErrLevel; };  173 : :   174 : : void   175 : : setErrorStatus(unsigned int ErrCode);  176 : : };  177 : :   178 : : inline ERROR\_LEVEL  179 : : ErrorHandler::getErrorLevel() {  180 : : return mErrLevel;  181 : : }  182 : :   183 : : inline DPL::ERRTYPE  184 : : ErrorHandler::getErrorType() {  185 : : return mErrType;  186 : : }  187 : :   188 : : inline unsigned int  189 : : ErrorHandler::getErrorCode() {  190 : : return mErrCode;  191 : : }  192 : :   193 : : } // namespace nft  194 : : #endif |

|  |
| --- |
| Generated by: [LCOV version 1.9](http://ltp.sourceforge.net/coverage/lcov.php) |