| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [SDF12iFormat/src](http://docs.google.com/index.html) - SDF12iDataReader.cpp (source / [functions](http://docs.google.com/SDF12iDataReader.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 36 | 51 | 70.6 % | | Date: | 2012-03-23 |  | Functions: | 14 | 19 | 73.7 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 9 | 17 | 52.9 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include "DPL.h"  2 : : #include "SDF12iDataReader.h"  3 : : #include "MemInfo.h"  4 : :   5 : : namespace nft {  6 : :   7 : :   8 : : SDF12iDataReader::SDF12iDataReader(const MemInfo & inShotData, bool inHasDumpHeader)   9 : 999 : :mMemory(inShotData.getMem()),   10 : : mSize(inShotData.getMemSize()),  11 : : mCurrentPosition( (inHasDumpHeader)?sizeof(SDF12iDumpHeader):0 ),  12 [ # # ][ + + ]: 999 : mHasDumpHeader(inHasDumpHeader) {  13 : : }  14 : :   15 : 999 : SDF12iDataReader::~SDF12iDataReader() {  16 : : }  17 : :   18 : : SDF12iDataReader::SDF12iDataReader(const SDF12iDataReader & source)   19 : 0 : :mMemory(source.mMemory),  20 : : mSize(source.mSize),  21 : : mCurrentPosition( source.mCurrentPosition ),  22 : 0 : mHasDumpHeader(source.mHasDumpHeader){  23 : : }  24 : :   25 : : SDF12iDataReader &  26 : 0 : SDF12iDataReader::operator=(const SDF12iDataReader & source) {  27 : :   28 [ # # ]: 0 : if(this != &source)  29 : : {  30 : 0 : mMemory = source.mMemory;  31 : 0 : mSize = source.mSize;  32 : 0 : mCurrentPosition = source.mCurrentPosition;  33 : 0 : mHasDumpHeader = source.mHasDumpHeader;  34 : : }  35 : 0 : return \*this;  36 : : }  37 : :   38 : : SDF12iDataReader::ItemInfo  39 : 30801 : SDF12iDataReader::getNextItem() {  40 : :   41 : 30801 : char \*currPtr = mMemory + mCurrentPosition;  42 : :   43 : : //std::cout<< mCurrentPosition << std::endl;  44 : 30801 : ItemInfo info;  45 : 30801 : info.setData(currPtr);  46 : :   47 : 30801 : uint32\_t \*ptr = (uint32\_t\*)currPtr;  48 : : SDF12iObjectTypeFlag flag = static\_cast<SDF12iObjectTypeFlag>(  49 : 30801 : (\*ptr & 0xF0000000) >> 28);  50 : 30801 : info.setType(flag);  51 : :   52 [ + - ][ + + : 30801 : switch(flag)  + + + -   - ]  53 : : {  54 : : case eFlagShot:  55 : 16084 : mCurrentPosition += sizeof(SDF12iShot);  56 : 16084 : break;  57 : : case eFlagDPBHeader:  58 : 999 : mCurrentPosition += sizeof(SDF12iDpbHeader);  59 : 999 : break;  60 : : case eFlagTF:  61 : 10542 : mCurrentPosition += sizeof(SDF12iTFHeader);  62 : 10542 : break;  63 : : case eFlagSF:  64 : 2177 : mCurrentPosition += sizeof(SDF12iSFHeader);  65 : 2177 : break;  66 : : case eFlagCheckSum:  67 : 999 : mCurrentPosition += 12;  68 : 999 : break;  69 : : case eFlagDumpHeader:  70 : 0 : mCurrentPosition += sizeof(SDF12iDumpHeader);  71 : : break;  72 : : default :  73 : : // incorrect flag found  74 : : /\* ErrorHandler \*pErr = ErrorHandler::instance();  75 : : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  76 : : ErrorHandler::eErrFunction,  77 : : pErr->getErrorMsg(  78 : : ErrorHandler::eErrFunction,  79 : : "SDF12iDataReader::getNextItem"));  80 : 30801 : \*/ break;  81 : : }  82 : 30801 : return info;  83 : : }  84 : :   85 : :   86 : : SDF12iDataReader::ItemInfo::ItemInfo()   87 : 30801 : :mType(eFlagShot),  88 : 0 : mData(NULL) {  89 : : }  90 : :   91 : :   92 : 30801 : SDF12iDataReader::ItemInfo::~ItemInfo() {  93 : : }  94 : :   95 : :   96 : : SDF12iDataReader::ItemInfo::ItemInfo(const SDF12iDataReader::ItemInfo & source)  97 : 0 : :mType(source.mType),  98 : 0 : mData(source.mData) {  99 : : }  100 : :   101 : : SDF12iDataReader::ItemInfo &  102 : 29802 : SDF12iDataReader::ItemInfo::operator=(const SDF12iDataReader::ItemInfo & source) {  103 : :   104 [ + - ]: 29802 : if(this != &source)  105 : : {  106 : 29802 : mType = source.mType;  107 : 29802 : mData = source.mData;  108 : : }  109 : 29802 : return \*this;  110 : : }  111 : :   112 : :   113 : : SDF12iObjectTypeFlag  114 : 30801 : SDF12iDataReader::ItemInfo::getType() const {  115 : :   116 : 30801 : return mType;  117 : : }  118 : :   119 : : void  120 : 30801 : SDF12iDataReader::ItemInfo::setData(char \*inVal) {  121 : :   122 : 0 : mData = inVal;  123 : : }  124 : :   125 : : void  126 : 30801 : SDF12iDataReader::ItemInfo::setType(SDF12iObjectTypeFlag inVal) {  127 : :   128 : 0 : mType = inVal;  129 : : }  130 : :   131 : :   132 : :   133 : 999 : } // namespace nft |

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