| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [InputParams/include](http://docs.google.com/index.html) - DOCTable.h (source / [functions](http://docs.google.com/DOCTable.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 1 | 1 | 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\_DOCTABLE\_H  2 : : #define NFT\_DOCTABLE\_H  3 : :   4 : : #include <cstddef>  5 : : #include <stdint.h>  6 : : #include "Memory.h"  7 : : #include "ErrorHandler.h"  8 : :   9 : : namespace DPL { class ExternalMemory; }  10 : : namespace nft { struct MemInfo; }   11 : :   12 : : namespace nft {  13 : :   14 : : class DOCTable : public NftNewDeleteBase {  15 : :   16 : : public:  17 : : static DOCTable \*  18 : : create( const MemInfo &inMemory, bool inIsSharedMemory = false,  19 : : DPL::ExternalMemory \*inExternalMemory = NULL )throw (DPL::Error \*);  20 : :   21 : : ~DOCTable();  22 : :   23 : : private:  24 : : DOCTable( const MemInfo &inMemory, bool inIsSharedMemory,  25 : : DPL::ExternalMemory \*inExternalMemory );  26 : :   27 : : DOCTable(const DOCTable & source);  28 : :   29 : : DOCTable &  30 : : operator =(const DOCTable & source);  31 : :   32 : : public:  33 : : inline const int16\_t \*  34 : : getDOCData() const;  35 : :   36 : : private:  37 : : char \* mData;  38 : : size\_t mSize;  39 : : bool mSharedMemory;  40 : : DPL::ExternalMemory \* mExternalMemory;  41 : : };  42 : :   43 : : inline const int16\_t \*  44 : 15978 : DOCTable::getDOCData() const {  45 : : return reinterpret\_cast<int16\_t \*>( mData );  46 : : }  47 : :   48 : : } // namespace nft  49 : : #endif |

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