| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [InputParams/src](http://docs.google.com/index.html) - AITable.cpp (source / [functions](http://docs.google.com/AITable.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 28 | 33 | 84.8 % | | Date: | 2012-03-23 |  | Functions: | 8 | 8 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 10 | 30 | 33.3 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include "DPL.h"  2 : : #include "Logger.h"  3 : : #include "ErrorHandler.h"  4 : : #include "AITable.h"  5 : : #include "MemInfo.h"  6 : : namespace nft {  7 : :   8 : : AITable \*  9 : : AITable::create( const MemInfo &inMemory,  10 : 26 : bool inIsSharedMemory, DPL::ExternalMemory \*inExternalMemory) throw (DPL::Error \*) {  11 : 26 : try{  12 : 26 : return new AITable( inMemory, inIsSharedMemory,  13 : : inExternalMemory );  14 : : }  15 [ # # ]: 0 : catch( DPL::Error \*pChild )  16 : : {  17 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  18 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  19 : : ErrorHandler::eErrFunction,  20 : : pErr->getErrorMsg(ErrorHandler::eErrFunction,  21 : : "AITable::create"));  22 [ # # ]: 26 : return NULL;  23 : : }  24 : : }  25 : :   26 : : AITable::AITable( const MemInfo &inMemory, bool inIsSharedMemory,  27 : : DPL::ExternalMemory \*inExternalMemory )  28 : 26 : : mData(inMemory.getMem()),  29 : : mSize(inMemory.getMemSize()),  30 : : mSharedMemory( inIsSharedMemory ),  31 : : mExternalMemory( inExternalMemory ),  32 : 78 : mAiTable( reinterpret\_cast<AI\_param\_t \*>( mData ) ) {  33 : 26 : mNumElements = mSize / sizeof( AI\_param\_t );  34 : : }  35 : :   36 : : AITable::~AITable()  37 : 52 : {  38 [ # # ][ + - ]: 26 : if( !mSharedMemory )  39 : : {  40 [ # # ][ + - ]: 26 : delete [] mData;  41 : : }  42 [ # # ][ - + ]: 26 : delete mExternalMemory;  43 : : }  44 : : bool  45 [ + + ]: 431 : AITable::checkIndex( uint32\_t ai, uint32\_t &position )const {  46 [ # # ][ # # ]: 1501 : for(size\_t i = 0; i < mNumElements; i ++ )  [ + - ][ + + ]  47 : : {  48 [ # # ][ + + ]: 1497 : if( mAiTable[i].AI == ai )  49 : : {  50 : 427 : position = i;  51 : 0 : return true;  52 : : }  53 : 4 : }  54 : 0 : return false;  55 : : }  56 : :   57 : : double  58 : : AITable::getDoseForAI( uint32\_t ai ) const  59 : 431 : {  60 : 431 : uint32\_t position = 0;  61 : : //#ifdef ERROR\_SEED //<<Test\_Id:AITbl\_GetDoseForAI>>  62 : : // bool value = checkIndex( ai, position );  63 : : // value=false;  64 : 431 : if( checkIndex( ai, position ) )  65 : : {  66 : : //#endif  67 : 427 : return getValue( position );  68 : : }  69 : 4 : char buf[BUFSIZ];  70 : 4 : sprintf( buf, "No dose data found for AI %u", ai);  71 : 4 : Logger::instance()->log(LogMessageTable::eDATA\_READER\_ERROR,  72 : : buf);  73 : 4 : ErrorHandler \*pErr = ErrorHandler::instance();  74 : 4 : pErr->errorAdd(NULL, (char\*)\_\_FILE\_\_, (char\*)\_\_FUNCTION\_\_, (int)\_\_LINE\_\_  75 : : ,ErrorHandler::eErrMessageString  76 : : ,pErr->getErrorMsg((ErrorHandler::eErrMessageString)  77 : : ,buf));  78 : 427 : return -1.000000;  79 : : }  80 : :   81 : 1215 : } // namespace nft |

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