| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [FixedModeFlow/src](http://docs.google.com/index.html) - WindowPosition.cpp (source / [functions](http://docs.google.com/WindowPosition.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 22 | 38 | 57.9 % | | Date: | 2012-03-23 |  | Functions: | 10 | 16 | 62.5 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 0 | 6 | 0.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include "WindowPosition.h"  2 : :   3 : : #include "print.h"  4 : :   5 : : namespace nft {  6 : :   7 : :   8 : : WindowPosition::WindowPosition()  9 : 8548 : :mPosition(),  10 : : mTargetSFGs(),  11 : : mCells(),  12 : 34192 : mShotBuffers() {  13 : : }  14 : :   15 : : WindowPosition::WindowPosition(  16 : : const WindowPosition & inSource)  17 : 0 : :mPosition( inSource.mPosition ),  18 : : mTargetSFGs( inSource.mTargetSFGs ),  19 : : mCells( inSource.mCells ),  20 : 0 : mShotBuffers( inSource.mShotBuffers ) {  21 : : }  22 : :   23 : : WindowPosition &  24 : : WindowPosition::operator=(  25 : 0 : const WindowPosition & inSource) {  26 [ # # ]: 0 : if( &inSource != this )  27 : : {  28 : 0 : mPosition = inSource.mPosition;  29 : 0 : mTargetSFGs = inSource.mTargetSFGs;  30 : 0 : mCells = inSource.mCells;  31 : 0 : mShotBuffers = inSource.mShotBuffers;  32 : : }  33 : 0 : return \*this;  34 : : }  35 : :   36 : : void  37 : 8548 : WindowPosition::setPosition(int64\_t value) {  38 : 8548 : mPosition = value;  39 : : }  40 : :   41 : : void  42 : 8548 : WindowPosition::setIndex(size\_t inValue) {  43 : 8548 : mIndex = inValue;  44 : : }  45 : :   46 : : //Applies a given function object to all cells in the window position  47 : : void  48 : : WindowPosition::applyForAllCells(  49 : 8478 : const WindowPosition::CellHandler & inCellHandler) throw(DPL::Error \*) {  50 : 8478 : if( !mCells.empty() )  51 : : {  52 : 1138 : e9k\_debug("Found cells at position %lld (%p)\n",  53 : : getPosition(), this);  54 : :   55 : : }  56 : : else  57 : : {  58 : 1138 : return;  59 : : }  60 : 1138 : for( CellList::iterator cell = mCells.begin();  61 : : cell != mCells.end(); cell++ )  62 : : {  63 : 1244 : try  64 : : {  65 : 1244 : inCellHandler( \*cell );  66 : : }  67 [ # # ]: 0 : catch( DPL::Error \*pChild )  68 : : {  69 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  70 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  71 : : ErrorHandler::eErrFunction,  72 : : pErr->getErrorMsg(  73 : : ErrorHandler::eErrFunction, "inCellHandler"));  74 : : }  75 : : }  76 [ # # ]: 8478 : return;  77 : : }  78 : :   79 : : void  80 : : WindowPosition::addCell(  81 : 1244 : Cell::Pointer inCell) throw(DPL::Error \*) {  82 : 1244 : mCells.push\_back( inCell );  83 : : }  84 : :   85 : :   86 : 17096 : WindowPosition::~WindowPosition() {  87 : 8548 : mCells.clear();  88 : : }  89 : :   90 : :   91 : : void  92 : 8813 : WindowPosition::addTargetSFG(SFGroup::Pointer inSFG) {  93 : 8813 : mTargetSFGs.push\_back( inSFG );  94 : : }  95 : :   96 : : //DELAYED MODE::Unused (Note:: The following functions  97 : : //are required for delayed mode only and are not used  98 : : //Applies a given function object to the undistributed shot buffer  99 : : void  100 : 0 : WindowPosition::applyToUndistributedShots() throw(DPL::Error \*) {  101 : : }  102 : :   103 : : void  104 : 0 : WindowPosition::addShotToBuffer(const SimplePattern & inShot, uint32\_t inLayerNo) throw(DPL::Error \*) {  105 : : }  106 : :   107 : : SimplePatternBuffer  108 : 0 : WindowPosition::getBuffer(uint32\_t inLayerNo) {  109 : 0 : return mShotBuffers[ inLayerNo ];  110 : : }  111 : :   112 : :   113 : 1215 : } // namespace nft |

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