| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [FixedModeFlow/include](http://docs.google.com/index.html) - WindowPosition.h (source / [functions](http://docs.google.com/WindowPosition.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 4 | 4 | 100.0 % | | Date: | 2012-03-23 |  | Functions: | 0 | 0 | - | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 17 | 26 | 65.4 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #ifndef NFT\_WINDOWPOSITION\_H  2 : : #define NFT\_WINDOWPOSITION\_H  3 : :   4 : :   5 : : #include <stdint.h>  6 : : #include <boost/function.hpp>  7 : :   8 : : #include "context.h"  9 : : #include "SFGroup.h"  10 : : #include "dec\_types.h"  11 : : #include "SimplePatternBuffer.h"  12 : : #include "Memory.h"  13 : :   14 : : namespace nft {  15 : :   16 : : class WindowPosition : public NftNewDeleteBase {  17 : : public:  18 : : typedef boost::function<void (Cell::Pointer)> CellHandler;  19 : : typedef boost::function<void (SimplePatternBuffer &)> ShotBufferHandler;  20 : : typedef WindowPosition \* Pointer;  21 : : typedef std::vector<Cell::Pointer> CellList;  22 : :   23 : : public:  24 : :   25 : : WindowPosition();  26 : :   27 : : ~WindowPosition();  28 : :   29 : : WindowPosition(const WindowPosition & source);  30 : :   31 : : WindowPosition &  32 : : operator=(const WindowPosition & source);  33 : :   34 : : inline int64\_t  35 : : getPosition() const;  36 : :   37 : : void  38 : : setPosition(int64\_t inValue);  39 : :   40 : : inline size\_t  41 : : getIndex() const;  42 : :   43 : : void  44 : : setIndex( size\_t inValue );  45 : :   46 : : //Applies a given function object to all cells in the window position  47 : : void  48 : : applyForAllCells(const CellHandler & inCellHandler) throw(DPL::Error \*);  49 : :   50 : : void  51 : : addCell(Cell::Pointer inCell) throw(DPL::Error \*);  52 : :   53 : : inline const std::vector<SFGroup::Pointer> &  54 : : getTargetSFGs() const;  55 : :   56 : : void  57 : : addTargetSFG(SFGroup::Pointer inSFG);  58 : :   59 : : //DELAYED MODE::Unused (Note:: The following functions  60 : : //are required for delayed mode only and are not used  61 : : //Applies a given function object to the undistributed shot buffer  62 : : void  63 : : applyToUndistributedShots() throw(DPL::Error \*);  64 : :   65 : : void  66 : : addShotToBuffer(const SimplePattern & inShot,  67 : : uint32\_t inLayerNo) throw(DPL::Error \*);  68 : :   69 : : SimplePatternBuffer  70 : : getBuffer(uint32\_t inLayerNo);  71 : :   72 : : private:  73 : : //X position  74 : : int64\_t mPosition;  75 : : std::vector<SFGroup::Pointer> mTargetSFGs;  76 : : CellList mCells;  77 : : std::vector<SimplePatternBuffer> mShotBuffers; //DELAYED MODE::Unused  78 : : size\_t mIndex;  79 : : };  80 : :   81 : : inline int64\_t  82 [ - + ]: 26000 : WindowPosition::getPosition() const {  83 : : return mPosition;  84 : : }  85 : :   86 : : inline const std::vector<SFGroup::Pointer> &  87 : 8238 : WindowPosition::getTargetSFGs() const {  88 : : return mTargetSFGs;  89 : : }  90 : :   91 : : inline size\_t  92 : 1244 : WindowPosition::getIndex() const {  93 : : return mIndex;  94 : : }  95 : :   96 : : } // namespace nft  97 : :   98 : : #endif |

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