| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [SFLayers/src](http://docs.google.com/index.html) - SubField.cpp (source / [functions](http://docs.google.com/SubField.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 66 | 79 | 83.5 % | | Date: | 2012-03-23 |  | Functions: | 16 | 16 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 32 | 56 | 57.1 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include <strings.h>  2 : : #include "DPL.h"  3 : :   4 : : #include "ConverterParams.h"  5 : : #include "SubField.h"  6 : : #include "Segment.h"  7 : : #include "TFOrderingPriority.h"  8 : :   9 : : namespace nft {  10 : :   11 : : SubField::SubField()  12 : 605620 : :mXCoordinate(),  13 : : mYCoordinate(),  14 : : mXwrtVChip(),  15 : : mYwrtVChip(),  16 : : mIsGmcDone(),  17 : : mDxpValue(),  18 : : mIsEmpty( true ),  19 : 605620 : mIsSegmentCreated() {  20 [ # # ][ + + ]: 3633720 : for( size\_t i = 0; i < 6; i++ )  21 : : {  22 : 3633720 : mSegments[i] = NULL;  23 : : }  24 : : }  25 : :   26 : 605620 : SubField::~SubField() {  27 : : //Do not call release segments here  28 : : //It may throw exception  29 : : }  30 : :   31 : : void  32 : 605620 : SubField::setXCoordinate(int32\_t inValue) {  33 : 605620 : mXCoordinate = inValue;  34 : : }  35 : :   36 : : void  37 : 605620 : SubField::setYCoordinate(int32\_t inValue) {  38 : 605620 : mYCoordinate = inValue;  39 : : }  40 : :   41 : : void  42 : 2179 : SubField::setXwrtVChip( int32\_t inValue) {  43 : 2179 : mXwrtVChip = inValue;  44 : : }  45 : :   46 : : void  47 : 2167 : SubField::setIsGmcDone( bool inValue ) {  48 : 2167 : mIsGmcDone = inValue;  49 : : }  50 : :   51 : : void  52 : 2156 : SubField::setDxpValue( int16\_t inValue ) {  53 : 2156 : mDxpValue = inValue;  54 : : }  55 : :   56 : : void  57 : 2179 : SubField::setYwrtVChip( int32\_t inValue) {  58 : 2179 : mYwrtVChip = inValue;  59 : : }  60 : :   61 : : void  62 : : SubField::setSFLNo( uint32\_t inValue )  63 : 605620 : {  64 : 605620 : mSFLNo = inValue;  65 : : }  66 : :   67 : : void  68 : : SubField::setIsEmpty( bool inValue )  69 : 2167 : {  70 : 2167 : mIsEmpty = inValue;  71 : : }  72 : :   73 : : void  74 : 2167 : SubField::createSegments( ConverterParams \*inParams ) throw(DPL::Error \*) {  75 : : const DPMANAGER\_MESSAGE\_CONVERTER \*pStartMsg  76 : 2167 : = inParams->getStartupMessage();  77 : 2167 : bool isArbitraryMode = pStartMsg->usProcessingMode & 0x01;  78 : : TFOrderingPriority tfPriority = isArbitraryMode ? eLocationBeforeShape  79 : : : static\_cast<TFOrderingPriority>(  80 [ + + ]: 2167 : pStartMsg->usTFPriority);  81 [ - + ]: 2167 : uint16\_t segmentCountX = (isArbitraryMode) ? 1 : pStartMsg->usSegmentCountX;  82 [ + - ]: 2167 : uint16\_t segmentCountY = (isArbitraryMode) ? 1 : pStartMsg->usSegmentCountY;  83 [ + + ]: 2167 : if( eLocationBeforeShape == tfPriority   84 : : || eLocationWithBucketSort == tfPriority )  85 : : {  86 : 2103 : Segment::Pointer segment = NULL;  87 : 2103 : try  88 : : {  89 : 2103 : segment = createSingleSegmentLayer(  90 : : segmentCountX, segmentCountY );  91 : : }  92 [ # # ]: 0 : catch( DPL::Error \*pChild )  93 : : {  94 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  95 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  96 : : ErrorHandler::eErrFunction,  97 : : pErr->getErrorMsg(  98 : : ErrorHandler::eErrFunction,  99 : : "SubField::createSingleSegmentLayer"));  100 : : }  101 : 2103 : mSegments[ 0 ] = segment;  102 : : }  103 : : else  104 : : {  105 [ + + ]: 384 : for( int i = 1; i <= 5; i++ )  106 : : {  107 : 320 : Segment::Pointer segment = NULL;  108 : 320 : try  109 : : {  110 : 320 : segment = createSingleSegmentLayer(  111 : : segmentCountX, segmentCountY );  112 : : }  113 [ # # ]: 0 : catch( DPL::Error \*pChild )  114 : : {  115 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  116 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  117 : : ErrorHandler::eErrFunction,  118 : : pErr->getErrorMsg(  119 : : ErrorHandler::eErrFunction,  120 : : "SubField::createSingleSegmentLayer"));  121 : : }  122 : 320 : mSegments[ i ] = segment;  123 : : }  124 : : }  125 [ # # ]: 2167 : mIsSegmentCreated = true;  126 : : }  127 : :   128 : : //void  129 : : //SubField::setSegments(Segment::Pointer inSegment, size\_t inIndex) {  130 : : // mSegments[ inIndex ] = inSegment;  131 : : //}  132 : :   133 : : Segment::Pointer  134 : : SubField::createSingleSegmentLayer( uint16\_t inSegmentCountX,  135 : 2423 : uint16\_t inSegmentCountY ) throw(DPL::Error \* ) {  136 : : Segment::Pointer segs  137 [ + + ][ # # ]: 35962 : = new(std::nothrow) Segment[ inSegmentCountX \* inSegmentCountY ];  [ # # ]  138 : : //#ifdef ERROR\_SEED //<<Test\_Id:SubField\_segs\_Set\_To\_NULL>>  139 : : // segs = NULL;  140 : : //#endif  141 [ - + ]: 2423 : if( NULL == segs )  142 : : {  143 : 0 : int iErrNo = errno; //System error no  144 : 0 : char systemErrorMsg[BUFSIZ];  145 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  146 : 0 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  147 : : ErrorHandler::eErrMemoryAllocation,  148 : : pErr->getErrorMsg(  149 : : ErrorHandler::eErrMemoryAllocation,  150 : : strerror\_r( iErrNo, systemErrorMsg, BUFSIZ )));  151 : : }  152 : 2423 : return segs;  153 : : }  154 : :   155 : : void  156 : 605620 : SubField::releaseSegments() throw(DPL::Error \*) {  157 : 605620 : ConverterParams \*params = ConverterParams::instance();  158 : 605620 : bool arbitraryMode = params->getStartupMessage()->usProcessingMode & 0x01;  159 [ + + ]: 605620 : if( arbitraryMode )  160 : : {  161 [ + - ]: 83 : if( mSegments[0] )  162 : : {  163 : 83 : mSegments[0]->releaseBuffer();  164 [ + - ][ + + ]: 166 : delete [] mSegments[0];  165 : 83 : mSegments[0] = 0;  166 : : }  167 : 83 : return;  168 : : }  169 : 605537 : uint16\_t segX = params->getStartupMessage()->usSegmentCountX;  170 : 605537 : uint16\_t segY = params->getStartupMessage()->usSegmentCountY;  171 [ + + ][ # # ]: 4238842 : for( size\_t i = 0; i < 6; i++ )  172 : : {  173 : 3633222 : Segment::Pointer layer = mSegments[ i ];  174 [ + + ]: 3633222 : if( !layer )  175 : : {  176 : 2340 : continue;  177 : : }  178 [ + - ][ + + ]: 11124 : for( uint16\_t col = 0; col < segX; col++ )  179 : : {  180 [ + - ][ + + ]: 42240 : for( uint16\_t row = 0; row < segY; row++ )  181 : : {  182 : : Segment::Pointer pSegment  183 : 33456 : = layer + ( col \* segY + row ); //Column major  184 : 33456 : try  185 : : {  186 : 33456 : pSegment->releaseBuffer();  187 : : }  188 [ # # ]: 0 : catch( DPL::Error \*pChild )  189 : : {  190 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  191 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  192 : : ErrorHandler::eErrFunction,  193 : : pErr->getErrorMsg(  194 : : ErrorHandler::eErrFunction,  195 : : "Segment::releaseBuffer"));  196 : : }  197 : : }  198 : : }  199 [ + - ][ + + ]: 35796 : delete [] mSegments[i];  200 : 2340 : mSegments[i] = 0;  201 : : }  202 : : }  203 : :   204 : 1215 : } // namespace nft |

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