| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [SFLayers/src](http://docs.google.com/index.html) - SFTableCreator.cpp (source / [functions](http://docs.google.com/SFTableCreator.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 84 | 110 | 76.4 % | | Date: | 2012-03-23 |  | Functions: | 5 | 5 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 36 | 70 | 51.4 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include <iostream>   2 : : #include "Logger.h"  3 : : #include "SFTableCreator.h"  4 : : #include "SDDB.h"  5 : : #include "print.h"  6 : :   7 : : namespace nft {  8 : :   9 : : SFTable::Pointer   10 : : SFTableCreator::constructSFTable(  11 : : ConverterParams \*inParams,  12 : 1113 : const IntRect &inCellArea ) throw(DPL::Error \*) {  13 : 1113 : SFTable::Pointer sfTable = new(std::nothrow) SFTable();  14 : : //EUPA >>  15 : : //Code injected to test for data dump of erroneous case of converter execution  16 : : //Test Case ID : Neg\_Dump\_Data  17 : : #ifdef NEG\_DUMP\_DATA  18 : : //if( 0 == sfTable )  19 : : //{  20 : : #else  21 : : //#ifdef ERROR\_SEED //<<Test\_Id:SFTableCreator\_sfTable\_Set\_To\_0>>  22 : : // sfTable = 0;  23 : : //#endif  24 [ - + ]: 1113 : if ( 0 == sfTable )  25 : : {  26 : : #endif  27 : : //<< EUPA  28 : 0 : int iErrNo = errno; //System error no  29 : 0 : char systemErrorMsg[BUFSIZ];  30 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  31 : 0 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  32 : : ErrorHandler::eErrMemoryAllocation,  33 : : pErr->getErrorMsg(  34 : : ErrorHandler::eErrMemoryAllocation,  35 : : strerror\_r( iErrNo, systemErrorMsg, BUFSIZ )));  36 : : #ifdef NEG\_DUMP\_DATA  37 : : //}  38 : : #else  39 : : }  40 : : #endif  41 : 1113 : try  42 : : {  43 : 1113 : createLayers( inParams, sfTable, inCellArea );  44 : : }  45 [ + - ]: 2 : catch( DPL::Error \*pChild )  46 : : {  47 : 2 : ErrorHandler \*pErr = ErrorHandler::instance();  48 : 2 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  49 : : ErrorHandler::eErrFunction,  50 : : pErr->getErrorMsg(  51 : : ErrorHandler::eErrFunction, "SFTableCreator::createLayers"));  52 : : }  53 : 1111 : return sfTable;  54 : : }  55 : :   56 : : void  57 : : SFTableCreator::createLayers( ConverterParams \*inParams,  58 : : SFTable::Pointer &outTable,  59 : 1113 : const IntRect &inCellArea ) throw(DPL::Error \*) {  60 : 1113 : uint32\_t numLayers = inParams->getStartupMessage()->uiSFLCountInDpb;  61 : : ProcessingMode mode = static\_cast<ProcessingMode>(  62 : : inParams->getStartupMessage()->usProcessingMode  63 : 1113 : & 0x00000003); //First 2 bits  64 [ + + ][ # # ]: 2302 : SFLayer::Pointer layers = new(std::nothrow) SFLayer[ numLayers ];  [ # # ]  65 : : //#ifdef ERROR\_SEED //<<Test\_Id:SFTableCreator\_layers\_Set\_To\_NULL>>  66 : : // layers = NULL;  67 : : //#endif  68 [ - + ]: 1113 : if( NULL == layers )  69 : : {  70 : 0 : int iErrNo = errno; //System error no  71 : 0 : char systemErrorMsg[BUFSIZ];  72 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  73 : 0 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  74 : : ErrorHandler::eErrMemoryAllocation,  75 : : pErr->getErrorMsg(  76 : : ErrorHandler::eErrMemoryAllocation,  77 : : strerror\_r( iErrNo, systemErrorMsg, BUFSIZ )));  78 : : }  79 [ + - ][ + + ]: 2300 : for( uint32\_t i = 0; i < numLayers; i++ )  80 : : {  81 : 1189 : SFLayer::Pointer layer = layers + i;  82 : : const DPMANAGER\_MESSAGE\_SFL\_INFO \*pLayerInfo  83 : 1189 : = inParams->getLayerParams() + i;  84 : 1189 : e9k\_debug("SFCX: %u SFCY: %u SFO: (%u, %u) TFCX: %u TFCY: %u TFO: (%u, %u)\n",  85 : : pLayerInfo->uiSFCountX, pLayerInfo->uiSFCountY,  86 : : pLayerInfo->llSFLOriginX, pLayerInfo->llSFLOriginY,  87 : : pLayerInfo->usTFCountX, pLayerInfo->usTFCountY,  88 : : pLayerInfo->llTFOriginX, pLayerInfo->llTFOriginY );  89 : : //#ifdef ERROR\_SEED  90 : : // mode = eProcessArbitraryTFLess; //<<Test\_Id:SFTableCreator\_cols\_Set\_To\_2>>  91 : : // mode = eProcessNormal; //<<Test\_Id:SFTableCreator\_tfl\_Set\_To\_NULL>>  92 : : //#endif  93 [ + + ][ + + ]: 1189 : switch( mode )  [ - + ][ + + ]  [ - + ]  94 : : {  95 : : case eProcessNormal:  96 : : case eProcessTFLess:  97 : : {  98 : 1104 : layer->setNoOfCols( pLayerInfo->uiSFCountX );  99 : 1104 : layer->setNoOfRows( pLayerInfo->uiSFCountY );  100 : 1104 : break;  101 : : }  102 : : case eProcessArbitrary:  103 : : case eProcessArbitraryTFLess:  104 : : {  105 : 85 : IntPosition cellSize = inCellArea.getSize();  106 : 85 : int32\_t spanX = cellSize.getX() - pLayerInfo->llSFLOriginX;  107 : 85 : int32\_t spanY = cellSize.getY() - pLayerInfo->llSFLOriginY;  108 : : uint32\_t cols = static\_cast<uint32\_t>(  109 : 85 : spanX/ inParams->getStartupMessage()->iSFSizeX);  110 : 85 : cols = ( spanX % inParams->getStartupMessage()->iSFSizeX )  111 : : ? cols + 1 : cols;  112 : : uint32\_t rows = static\_cast<uint32\_t>(  113 : 85 : spanY/ inParams->getStartupMessage()->iSFSizeY );  114 : 85 : rows = ( spanY % inParams->getStartupMessage()->iSFSizeY )  115 : : ? rows + 1 : rows;  116 [ - + ]: 85 : cols = (cols == 0) ? 1 : cols;  117 [ - + ]: 85 : rows = (rows == 0) ? 1 : rows;  118 : : #ifndef MULTIPLE\_SF\_MODE  119 : : //#ifdef ERROR\_SEED //<<Test\_Id:SFTableCreator\_cols\_Set\_To\_2>>  120 : : // cols = 2;  121 : : //#endif  122 [ + + ]: 85 : if( cols > 1 || rows > 1 )  123 : : {  124 : 2 : Logger::instance()->log(  125 : : LogMessageTable::eSHOT\_ERROR\_CELL\_MORE\_THAN\_SF,  126 : : spanX, spanY,  127 : : inParams->getStartupMessage()->iSFSizeX,  128 : : inParams->getStartupMessage()->iSFSizeY);  129 : 2 : ErrorHandler \*pErr = ErrorHandler::instance();  130 : 2 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  131 : : ErrorHandler::eErrCellSizeMoreThanSF,  132 : : pErr->getErrorMsg(  133 : : ErrorHandler::eErrCellSizeMoreThanSF,  134 : : spanX, spanY,  135 : : inParams->getStartupMessage()->iSFSizeX,  136 : : inParams->getStartupMessage()->iSFSizeY));  137 : : }  138 : : #endif  139 : 83 : layer->setNoOfCols( cols );  140 : 83 : layer->setNoOfRows( rows );  141 : 85 : break;  142 : : }  143 : : }  144 : 1187 : layer->setLayerNo( pLayerInfo->uiSFLID );  145 : 1187 : layer->setLLx( pLayerInfo->llSFLOriginX );  146 : 1187 : layer->setLLy( pLayerInfo->llSFLOriginY );  147 [ + + ]: 1187 : if( eProcessNormal == mode )  148 : : {  149 : 1076 : TFLayers::Pointer tfl = new(std::nothrow) TFLayers();  150 : : //#ifdef ERROR\_SEED //<<Test\_Id:SFTableCreator\_tfl\_Set\_To\_NULL>>  151 : : // tfl = NULL;  152 : : //#endif  153 [ - + ]: 1076 : if( NULL == tfl )  154 : : {  155 : 0 : int iErrNo = errno; //System error no  156 : 0 : char systemErrorMsg[BUFSIZ];  157 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  158 : 0 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  159 : : ErrorHandler::eErrMemoryAllocation,  160 : : pErr->getErrorMsg(  161 : : ErrorHandler::eErrMemoryAllocation,  162 : : strerror\_r( iErrNo, systemErrorMsg, BUFSIZ )));  163 : : }  164 : 1076 : tfl->setNumberOfTFX( pLayerInfo->usTFCountX );  165 : 1076 : tfl->setNumberOfTFY( pLayerInfo->usTFCountY );  166 : 1076 : tfl->setLayerLocationX( pLayerInfo->llTFOriginX );  167 : 1076 : tfl->setLayerLocationY( pLayerInfo->llTFOriginY );  168 : 1076 : tfl->setTFSizeX( inParams->getStartupMessage()->iTFSizeX );  169 : 1076 : tfl->setTFSizeY( inParams->getStartupMessage()->iTFSizeY );  170 : 1076 : layer->setTFL( tfl );  171 : : }  172 : 1187 : try  173 : : {  174 : 1187 : createSFGroups( inParams, layer, i, inCellArea );  175 : : }  176 [ # # ]: 0 : catch( DPL::Error \*pChild )  177 : : {  178 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  179 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  180 : : ErrorHandler::eErrFunction,  181 : : pErr->getErrorMsg(  182 : : ErrorHandler::eErrFunction,  183 : : "SFTableCreator::createSFGroups"));  184 : : }  185 : 1187 : try  186 : : {  187 : 1187 : SDDB::instance()->sflayerCreated( layer );  188 : : }  189 [ # # ]: 0 : catch(DPL::Error \*pChild)  190 : : {  191 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  192 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  193 : : ErrorHandler::eErrFunction,  194 : : pErr->getErrorMsg( ErrorHandler::eErrFunction,  195 : : "SDDB::sflayerCreated"));  196 : : }  197 : : }  198 [ + - ]: 1113 : outTable->setSFLayers( layers );  199 : : }  200 : :   201 : : void  202 : : SFTableCreator::createSFGroups( ConverterParams \*inParams,  203 : : SFLayer::Pointer &outLayer,  204 : : uint32\_t inSFLNo,   205 : 1187 : const IntRect &inCellArea ) throw(DPL::Error \*) {  206 : 1187 : uint32\_t numCols = outLayer->getNoOfCols();  207 : : SFGroup::Pointer sfgs  208 [ + + ][ # # ]: 10083 : = new(std::nothrow) SFGroup[ numCols ];  [ # # ]  209 : : //#ifdef ERROR\_SEED //<<Test\_Id:SFTableCreator\_sfgs\_Set\_To\_NULL>>  210 : : // sfgs = NULL;  211 : : //#endif  212 [ - + ]: 1187 : if( NULL == sfgs )  213 : : {  214 : 0 : int iErrNo = errno; //System error no  215 : 0 : char systemErrorMsg[BUFSIZ];  216 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  217 : 0 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  218 : : ErrorHandler::eErrMemoryAllocation,  219 : : pErr->getErrorMsg(  220 : : ErrorHandler::eErrMemoryAllocation,  221 : : strerror\_r( iErrNo, systemErrorMsg, BUFSIZ )));  222 : : }  223 : : const DPMANAGER\_MESSAGE\_CONVERTER \*pStartMsg  224 : 1187 : = inParams->getStartupMessage();  225 : 1187 : uint32\_t numRows = outLayer->getNoOfRows();  226 : 1187 : int64\_t xPosition = outLayer->getLLx();  227 : 1187 : int64\_t yPosition = outLayer->getLLy();  228 : 1187 : int32\_t sfWidth = pStartMsg->iSFSizeX;  229 : 1187 : int32\_t sfHeight = pStartMsg->iSFSizeY;  230 : 1187 : int64\_t sfPositionX = xPosition;  231 [ + - ][ + + ]: 10083 : for( SFGroup::Pointer sfg = sfgs; sfg != sfgs + numCols;  232 : : sfg++, sfPositionX += sfWidth )  233 : : {  234 : 8896 : sfg->setNoOfRows( numRows );  235 : 8896 : createSubFields( inParams, sfg, inCellArea );  236 : 8896 : SubField::Pointer sfs = sfg->getSubFields();  237 : 8896 : int64\_t sfPositionY = yPosition;  238 [ + - ][ + + ]: 614516 : for( SubField::Pointer sf = sfs; sf != sfs + numRows; sf++,  239 : : sfPositionY += sfHeight )  240 : : {  241 : 605620 : sf->setSFLNo( inSFLNo );  242 : 605620 : sf->setXCoordinate( sfPositionX );  243 : 605620 : sf->setYCoordinate( sfPositionY );  244 : : }  245 : : }  246 [ # # ]: 1187 : outLayer->setSFGroups( sfgs );  247 : : }  248 : :   249 : : void  250 : : SFTableCreator::createSubFields( ConverterParams \*inParams,  251 : : SFGroup::Pointer &outSFGroup,  252 : 8896 : const IntRect &inCellArea ) throw(DPL::Error \*) {  253 : 8896 : uint32\_t numRows = outSFGroup->getNoOfRows();  254 [ + + ][ # # ]: 614516 : SubField::Pointer sfs = new(std::nothrow) SubField[ numRows ];  [ # # ]  255 : : //#ifdef ERROR\_SEED //<<Test\_Id:SFTableCreator\_sfs\_Set\_To\_NULL>>  256 : : // sfs = NULL;  257 : : //#endif  258 [ - + ]: 8896 : if( NULL == sfs )  259 : : {  260 : 0 : int iErrNo = errno; //System error no  261 : 0 : char systemErrorMsg[BUFSIZ];  262 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  263 : 0 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  264 : : ErrorHandler::eErrMemoryAllocation,  265 : : pErr->getErrorMsg(  266 : : ErrorHandler::eErrMemoryAllocation,  267 : : strerror\_r( iErrNo, systemErrorMsg, BUFSIZ )));  268 : : }  269 [ # # ]: 8896 : outSFGroup->setSubFields( sfs );  270 : : //Population of coordinates to be done during GMC and distribution  271 : : //Segments are created when required  272 : : }  273 : :   274 : 1215 : } //namesapce nft |

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