| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [FixedModeFlow/src](http://docs.google.com/index.html) - TFDistributor.cpp (source / [functions](http://docs.google.com/TFDistributor.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 93 | 110 | 84.5 % | | Date: | 2012-03-23 |  | Functions: | 6 | 6 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 27 | 40 | 67.5 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include "Sgs\_Define.h"  2 : :   3 : : #include "ConverterParams.h"  4 : : #include "DXPCalculator.h"  5 : : #include "Segment.h"  6 : : #include "SFTableAccessor.h"  7 : : #include "TFDistributor.h"  8 : : #include "TFInfo.h"  9 : : #include "TFOrderingPriority.h"  10 : : #include "UnitConversion.h"  11 : : #include "SDDB.h"  12 : : #include "print.h"  13 : :   14 : : namespace nft {  15 : :   16 : : TFDistributor::TFDistributor(SFTable::Pointer inSFT)  17 : 2116 : :mSFT( inSFT ) {  18 : : }  19 : :   20 : :   21 : 2116 : TFDistributor::~TFDistributor() {  22 : : }  23 : :   24 : : void  25 : 10307 : TFDistributor::operator()(const TFInfo &inTFInfo) const throw(DPL::Error \*) {  26 [ + - ]: 20614 : TFInfo info = inTFInfo;  27 : 10307 : ConverterParams \*params = ConverterParams::instance();  28 : : const DPMANAGER\_MESSAGE\_CONVERTER \*pStartMsg  29 : 10307 : = params->getStartupMessage();  30 : 10307 : const sgs\_column\_t \*pSecondParams = params->getColumnParams();  31 : : #ifdef SWINGBACK\_BEFORE\_GMC  32 : : info.mTFPosition = mSwingBackApplier( info );  33 : : #endif  34 : 10307 : const GMCMap \*pGMCMap = params->getGMCMap();  35 : : //IF GMC is enabled for TF  36 [ + + ]: 10307 : if( pSecondParams->gmc.mode & 0x00000004 )  37 : : {  38 : 128 : DoublePosition corrPos;  39 : 128 : bool gmcDone = false;  40 [ + + ]: 128 : if( pSecondParams->gmc.mode & 0x00000001 )  41 : : {  42 : 64 : corrPos += mGMCCorrector.calculate( info.mTFPosition,  43 : : pGMCMap, pSecondParams->gmc,  44 : : GMCorrector::eGMCPolynomial );  45 : 64 : gmcDone = true;  46 : : }  47 [ + + ]: 128 : if( pSecondParams->gmc.mode & 0x00000002 )  48 : : {  49 : 80 : corrPos += mGMCCorrector.calculate( info.mTFPosition,  50 : : pGMCMap, pSecondParams->gmc,  51 : : GMCorrector::eGMCMap );  52 : 80 : gmcDone = true;  53 : : }  54 : 128 : info.mTFPosition += TF::AbsolutePosition(   55 : : SgsRoundINT8(corrPos.getX()),  56 : : SgsRoundINT8(corrPos.getY()) );  57 : 128 : try  58 : : {  59 [ + + ]: 128 : if( gmcDone )  60 : : {  61 : 112 : Logger::instance()->log(LogMessageTable::eSHOT\_INFO,  62 : : "GMC is enabled for TF");  63 : 112 : mGMCCorrector.check( pSecondParams->gmc, info.mTFPosition );  64 : : }  65 : : else  66 : : {  67 : 16 : Logger::instance()->log(LogMessageTable::eSHOT\_INFO,  68 : : "GMC is disabled for TF");  69 : : }  70 : : }  71 [ # # ]: 0 : catch(DPL::Error \*pChild)  72 : : {  73 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  74 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  75 : : ErrorHandler::eErrFunction,  76 : : pErr->getErrorMsg( ErrorHandler::eErrFunction,  77 : : "GMCorrector::check"));  78 : : }  79 : : }  80 : : else  81 : : {  82 : 10179 : Logger::instance()->log(LogMessageTable::eSHOT\_INFO,  83 : : "GMC is disabled for TF");  84 : : }  85 : : #ifndef SWINGBACK\_BEFORE\_GMC  86 : 10307 : info.mTFPosition = mSwingBackApplier( info );  87 : : #endif  88 : 10307 : SFTableAccessor accessor( mSFT, params );  89 : : SFGroup::Pointer sfg = accessor.getSFGroup(  90 : 10307 : info.mTFPosition, info.mLayerNo );  91 : : SubField::Pointer sf = accessor.getSubField(  92 : 10307 : info.mTFPosition, info.mLayerNo );  93 [ + + ]: 10307 : if( NULL == sfg )  94 : : {  95 : 51 : Logger::instance()->log(LogMessageTable::eTF\_OUSIDE\_SF\_TABLE,  96 : : info.mTFPosition.getX(), info.mTFPosition.getY(),  97 : : info.mLayerNo);  98 : 51 : e9k\_debug("TF at %lld, %lld is outside SF Table for layer %d\n",  99 : : info.mTFPosition.getX(), info.mTFPosition.getY(),  100 : : info.mLayerNo);  101 : 51 : info.mTF->setDiscarded( true );  102 : 51 : info.mTF->setDistributed( true );  103 : 51 : return;  104 : : }  105 : 10256 : if( !sf->getIsGmcDone() )  106 : : {  107 : 2096 : e9k\_debug("TF (%p) is being added to SFG at %p\n",  108 : : info.mTF, sfg );  109 : 2096 : if( sfg->getIsEmpty() )  110 : : {  111 : 1332 : sfg->setIsEmpty( false );  112 : : }  113 : : LongPosition sfAbsPosition(  114 : : sf->getXCoordinate() + pStartMsg->llVchipOriginX  115 : : + pStartMsg->llDpbX  116 : : + (pStartMsg->iSFSizeX >> 1),  117 : : sf->getYCoordinate() + pStartMsg->llVchipOriginY  118 : : + pStartMsg->llDpbY  119 : 2096 : + (pStartMsg->iSFSizeY >> 1) );  120 : : //1. Calculate DXP value and store it for future reference  121 [ + + ]: 2096 : if( pSecondParams->kbr.mode )  122 : : {  123 : : int16\_t dxpValue = DXPCalculator::calculateDxpValue(  124 : : sfAbsPosition,  125 : : ConverterParams::instance()->getKBRMap(),  126 : 2073 : pSecondParams->kbr );  127 : 2073 : size\_t tmp = dxpValue;  128 : 2073 : Logger::instance()->log(LogMessageTable::eSHOT\_KBR\_VALUE,  129 : : dxpValue);  130 : 2073 : sf->setDxpValue( dxpValue );  131 : : }  132 : : //2. IF GMC is enabled for SF, perform it  133 [ + + ]: 2096 : if( !( pSecondParams->gmc.mode & 0x00000004 ) )  134 : : {  135 : 2089 : DoublePosition corrPos;  136 : 2089 : bool gmcDone = false;  137 [ + + ]: 2089 : if( pSecondParams->gmc.mode & 0x00000001 )  138 : : {  139 : 25 : corrPos += mGMCCorrector.calculate( sfAbsPosition,  140 : : pGMCMap, pSecondParams->gmc,  141 : : GMCorrector::eGMCPolynomial );  142 : 25 : gmcDone = true;  143 : : }  144 [ + + ]: 2089 : if( pSecondParams->gmc.mode & 0x00000002 )  145 : : {  146 : 28 : corrPos += mGMCCorrector.calculate( sfAbsPosition,  147 : : pGMCMap, pSecondParams->gmc,  148 : : GMCorrector::eGMCMap );  149 : 28 : gmcDone = true;  150 : : }  151 : 2089 : sf->setXwrtVChip(  152 : : sfAbsPosition.getX() + SgsRoundINT(corrPos.getX()) );  153 : 2089 : sf->setYwrtVChip(  154 : : sfAbsPosition.getY() + SgsRoundINT(corrPos.getY()) );  155 : 2089 : try  156 : : {  157 [ + + ]: 2089 : if( gmcDone )  158 : : {  159 : : LongPosition sfPos(  160 : 40 : sf->getXwrtVChip(), sf->getYwrtVChip());  161 : 40 : Logger::instance()->log(LogMessageTable::eSHOT\_INFO,  162 : : "GMC is enabled for SF");  163 : 52 : mGMCCorrector.check( pSecondParams->gmc, sfPos );  164 : : }  165 : : else  166 : : {  167 : 2049 : Logger::instance()->log(LogMessageTable::eSHOT\_INFO,  168 : : "GMC is disabled for SF");  169 : : }  170 : : }  171 [ + - ]: 12 : catch(DPL::Error \*pChild)  172 : : {  173 : 12 : ErrorHandler \*pErr = ErrorHandler::instance();  174 : 12 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  175 : : ErrorHandler::eErrFunction,  176 : : pErr->getErrorMsg( ErrorHandler::eErrFunction,  177 : : "GMCorrector::check"));  178 : : }  179 : : }  180 : : else  181 : : {  182 : 7 : Logger::instance()->log(LogMessageTable::eSHOT\_INFO,  183 : : "GMC is disabled for SF");  184 : 7 : sf->setXwrtVChip( sfAbsPosition.getX() );  185 : 7 : sf->setYwrtVChip( sfAbsPosition.getY() );  186 : : }  187 : 2084 : sf->setIsGmcDone( true );  188 : 2084 : if( !sf->getIsSegmentCreated() )  189 : : {  190 : : //TODO::ERROR  191 : 2084 : sf->createSegments(ConverterParams::instance());  192 : : }  193 : 2096 : sf->setIsEmpty( false );  194 : : }  195 : : TFOrderingPriority tfPriority  196 : 10244 : = static\_cast<TFOrderingPriority>(pStartMsg->usTFPriority);  197 : 10244 : Segment::Pointer targetSegment = 0;  198 [ + - ][ + + ]: 10244 : switch( tfPriority )  [ + - ]  199 : : {  200 : : case eLocationBeforeShape:  201 : : case eLocationWithBucketSort:  202 : : {  203 : 8708 : try{   204 : 8708 : targetSegment = accessor.getSegmentForLocationPriority(  205 : : info.mTFPosition, info.mLayerNo );  206 : : }  207 [ # # ]: 0 : catch( DPL::Error \*pChild )  208 : : {  209 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  210 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  211 : : ErrorHandler::eErrFunction,  212 : : pErr->getErrorMsg(  213 : : ErrorHandler::eErrFunction,  214 : : "TFDistributor::operator()"));  215 : : }  216 : 0 : break;  217 : : }  218 : : case eShapeBeforeLocation:  219 : : case eShapeWithBucketSort:  220 : : {  221 : 1536 : try{  222 : 1536 : targetSegment = accessor.getSegmentForPatternCode(  223 : : info.mTFPosition, info.mLayerNo,  224 : : info.mShotCode );  225 : : }  226 [ # # ]: 0 : catch( DPL::Error \*pChild )  227 : : {  228 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  229 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  230 : : ErrorHandler::eErrFunction,  231 : : pErr->getErrorMsg(  232 : : ErrorHandler::eErrFunction,  233 : : "TFDistributor::operator()"));  234 : : }  235 : 10244 : break;  236 : : }  237 : : }  238 [ - + ]: 10244 : if( NULL == targetSegment )  239 : : {  240 : 0 : info.mTF->setDiscarded( true );  241 : 0 : info.mTF->setDistributed( true );  242 : : //TODO::LOG  243 : 0 : Logger::instance()->log(LogMessageTable::eSHOT\_INFO,  244 : : "TF falls outside the given subfield layer");  245 : 0 : return;  246 : : }  247 : 10244 : TF::Pointer tf = info.mTF;  248 : 10244 : TFBuffer::Pointer tfBuffer = targetSegment->getTFs();  249 : 10244 : tfBuffer->addTF( tf );  250 : 10244 : tf->setParentSF( sf );  251 : 10244 : tf->setLocX( info.mTFPosition.getX() - sf->getXCoordinate() );  252 : 10244 : tf->setLocY( info.mTFPosition.getY() - sf->getYCoordinate() );  253 : 10244 : info.mTF->setDiscarded( false );  254 : 10244 : info.mTF->setDistributed( true );  255 : 10244 : SDDB \*pSDDB = SDDB::instance();   256 : 10244 : try  257 : : {  258 : 10244 : pSDDB->TFDistributed(\*tf, info, sf);  259 : : }  260 [ # # ]: 0 : catch(DPL::Error \*pChild)  261 : : {  262 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  263 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_,  264 : : \_\_LINE\_\_, ErrorHandler::eErrFunction,  265 : : pErr->getErrorMsg(ErrorHandler::eErrFunction,  266 : : "SDDB::TFDistributed"));  267 : : }  268 : : }  269 : :   270 : 1215 : } // namespace nft |

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