| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [LibVsb/src](http://docs.google.com/index.html) - link.cpp (source / [functions](http://docs.google.com/link.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 43 | 49 | 87.8 % | | Date: | 2012-03-23 |  | Functions: | 7 | 7 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 6 | 12 | 50.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*   2 : : \* Copyrighted 2007 by Interra Systems, USA  3 : : \* All rights reserved   4 : : \*  5 : : \* Date of Creation: Fri Jul 20 15:53:18 IST 2007  6 : : \* File Name: link.cpp   7 : : \* Purpose:   8 : : \* Author:   9 : : \*  10 : : \* Revision History:   11 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/   12 : :   13 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  14 : : \* Header Include section   15 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  16 : : #include <cstdio>  17 : : using namespace std;   18 : : #ifdef HAVE\_INTTYPES\_H  19 : : #include <inttypes.h>  20 : : #endif  21 : : #include <cmath>  22 : : #include "link.h"  23 : : #include "dec\_types.h"  24 : : #include "vsb\_types.h"  25 : : #include "ErrorHandler.h"  26 : : #include "Logger.h"  27 : : #include "StrictAliasUnion.h"  28 : :   29 : : using namespace std;  30 : :   31 : : namespace nft {  32 : : const uint32\_t VSBLinkReader::RELATION\_RECORD\_SIZE  33 : : = sizeof(vsbRelationRecord);  34 : :   35 : : const uint32\_t VSBLinkReader::CELL\_OPERATION\_SIZE  36 : : = sizeof(CellOperation);  37 : :   38 : : const uint32\_t VSBLinkReader::CELL\_OP\_HEADER\_SIZE  39 : : = sizeof(CellOperationTableHeader);  40 : :   41 : :   42 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  43 : : \* Function Name: VSBLinkReader()   44 : : \* Purpose: Initializes to read the link file  45 : : \* Parameters: char\* lfile  46 : : \* Return Values: None  47 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  48 : : VSBLinkReader::VSBLinkReader(char\* lfile)  49 : 2188 : {  50 : 1094 : linkReInit(lfile);  51 : : }  52 : :   53 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  54 : : \* Function Name: linkReInit()  55 : : \* Purpose: Initialize the link file for reading from the buffer   56 : : \* Parameters: char\* lfile  57 : : \* Return Values: void  58 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  59 : :   60 : : void VSBLinkReader::linkReInit(char\* lfile)  61 : 1094 : {  62 : : //Point to the byte buffer of the link file  63 : 1094 : mLinkFile = lfile;  64 : :   65 : : //reads the link.n file header   66 : :   67 : 1094 : uint32\_t l\_length[4] = {0, 0, 0, 0};  68 : 1094 : uint32\_t\* length = NULL;  69 : : //#ifdef ERROR\_SEED //<<Test\_Id:link\_mLinkFile\_Set\_To\_NULL>>  70 : : // mLinkFile = NULL;  71 : : //#endif  72 [ - + ]: 1094 : if (mLinkFile == NULL)  73 : : {  74 : 0 : Logger::instance()->log(LogMessageTable::eUTIL\_ERROR\_GENERAL\_MESSAGE,  75 : : "VSB Link Data is NULL");  76 : :   77 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  78 : 0 : pErr->errorAdd(NULL, (char\*)\_\_FILE\_\_, (char\*)\_\_FUNCTION\_\_, (int)\_\_LINE\_\_  79 : : ,ErrorHandler::eErrMessageString  80 : : ,pErr->getErrorMsg((ErrorHandler::eErrMessageString)  81 : : ,"VSB Link data is NULL"));  82 : : }  83 : 1094 : linkSegment\_[VSB\_LINK\_HEADER] = mLinkFile;  84 : 1094 : length = (uint32\_t\*) linkSegment\_[VSB\_LINK\_HEADER];  85 [ + + ]: 4376 : for (uint32\_t i=0; i<NUM\_LINK\_SEGMENTS; i++)  86 : : {  87 : 4376 : l\_length[i] = BE2LE\_32(length[i]);   88 : : }  89 : 1094 : length = l\_length;  90 : : //What is stored in the linkSegment array?  91 : : //Ans: It is the offset of the 4 regions of the  92 : : // link.n file  93 : : // length[0] = magic\_number  94 : : // length[1] = header\_segment\_length (in words)  95 : : // length[2] = length of relation table segment (in words)  96 : : // length[3] = length of cell operation table (in words)  97 : : // length[4] = length of file mapping table segment (in words)   98 : :   99 : 1094 : linkSegment\_[VSB\_LINK\_REL\_TBL]= linkSegment\_[VSB\_LINK\_HEADER] +   100 : : (length[1] \* WordSize);  101 : 1094 : linkSegment\_[VSB\_LINK\_CELL\_OP\_TBL] = linkSegment\_[VSB\_LINK\_REL\_TBL] +  102 : : (length[2] \* WordSize);  103 : 1094 : linkSegment\_[VSB\_LINK\_FILE\_MAP\_TBL] = linkSegment\_[VSB\_LINK\_CELL\_OP\_TBL] +  104 : : ( length[3] \* WordSize);  105 : :   106 : : //Relation record table occupies "length[2]" words.  107 : : //  108 : 1094 : mNRR = (length[2] \* WordSize) / sizeof(vsbRelationRecord);  109 : : //relationRecordSize\_ = sizeof(vsbRelationRecord);  110 : : }  111 : :   112 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  113 : : \* Function Name: linkGetRelationRecord()  114 : : \* Purpose: Returns the Relation Record  115 : : \* Parameters: record index  116 : : \* relation record  117 : : \* Return Values: void  118 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  119 : : void VSBLinkReader::linkGetRelationRecord   120 : : (  121 : : const uint32\_t &ri, //record index  122 : : vsbRelationRecord \*\*rr //relation record  123 : : )const throw (DPL::Error \*)  124 : 1261 : {  125 : : //precondition  126 [ - + ]: 1261 : if (ri >= mNRR )   127 : : {  128 : 0 : Logger::instance()->log(LogMessageTable::eUTIL\_ERROR\_GENERAL\_MESSAGE,  129 : : "Invalid Record Index");  130 : :   131 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  132 : 0 : pErr->errorAdd(NULL, (char\*)\_\_FILE\_\_, (char\*)\_\_FUNCTION\_\_, (int)\_\_LINE\_\_  133 : : ,ErrorHandler::eErrRecordIndex  134 : : ,pErr->getErrorMsg((ErrorHandler::eErrRecordIndex)  135 : : ,ri));  136 : : }  137 : : //process  138 [ # # ]: 1261 : \*rr = (vsbRelationRecord\*) (linkSegment\_[VSB\_LINK\_REL\_TBL] + RELATION\_RECORD\_SIZE \* ri);  139 : : }  140 : :   141 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  142 : : \* Function Name: linkGetScaleFactor()  143 : : \* Purpose: Returns the Scale Factor  144 : : \* Parameters: vsbRelationRecord \*relationRecord  145 : : \* Return Values: double  146 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  147 : : double VSBLinkReader::linkGetScaleFactor(vsbRelationRecord \*relationRecord) const  148 : 1261 : {  149 : 1261 : int64\_t F;  150 : 1261 : uint32\_t B;  151 : 1261 : uint32\_t E;  152 : 1261 : char\* temp;  153 : : // alias\_int alias\_intVal;  154 : :   155 : : #ifndef SOLARIS //default LINUX  156 : 1261 : int l\_temp[4] = {0, 0, 0, 0};  157 : 1261 : short int si\_mirrorOp = 0; //only 1 bit needed but we need to do endian conversion  158 : 1261 : vsbMirrorNdCellOp \*mirrorOp = (vsbMirrorNdCellOp \*) &si\_mirrorOp;  159 : :   160 : : // Jump to the [Mirror\_flag(1 bit), Cell\_Operation\_Index(15 bits)]  161 : : // part of the relation record  162 : 1261 : si\_mirrorOp = \*((short int\*) ((char\*) relationRecord + l\_MIRROR\_ND\_CELL\_OP\_OFFSET) );  163 : 1261 : si\_mirrorOp = BE2LE\_16 (si\_mirrorOp);  164 : :   165 : : // Now access the appropriate Cell\_Operation[i] in the   166 : : // Cell Operation Table Segment  167 : : // Note that "i" is obtained from the relation record above.   168 : 1261 : temp = (char\*) (linkSegment\_[VSB\_LINK\_CELL\_OP\_TBL] +   169 : : CELL\_OP\_HEADER\_SIZE +  170 : : CELL\_OPERATION\_SIZE \* mirrorOp->CellOperationIndex  171 : : );  172 : : //Cell\_Operation[i] is of the form:  173 : : // +--------------+  174 : : // | B (u\_int4) |  175 : : // +--------------+  176 : : // | E (u\_int4) |  177 : : // +--------------+  178 : : // | F (int8) |  179 : : // +--------------+   180 : 1261 : memcpy(&l\_temp, temp, SIZEOF\_INT \* 4);  181 : 1261 : B = BE2LE\_32((uint32\_t) l\_temp[0]);  182 : 1261 : E = BE2LE\_32((uint32\_t) l\_temp[1]);  183 : 1261 : F = BE2LE\_64( \*reinterpret\_cast<int64\_t \*>(l\_temp + 2) ); //VSB12 Format says  184 : : //int8 but E5K has  185 : : //uint64\_t or uint8   186 : : // alias\_intVal.value1 = BE2LE\_64(\* ((int64\_t\*) &l\_temp[2]) );  187 : : // F = alias\_intVal.value2;  188 : : #else //SOLARIS  189 : : temp = (char\*) (linkSegment\_[VSB\_LINK\_CELL\_OP\_TBL] +   190 : : CELL\_OP\_HEADER\_SIZE +   191 : : CELL\_OPERATION\_SIZE \*  192 : : relationRecord->CellOperationIndex  193 : : );  194 : : B = \*((uint32\_t \*)temp);  195 : : E = \*((uint32\_t \*)(temp + 4));  196 : : F = \*((int64\_t \*) (temp + 8));  197 : : F = F << 32;  198 : : F += \*((int64\_t\*)(temp + 12));  199 : : #endif //OS  200 : :   201 [ + - ][ + - ]: 1261 : return (double)F/pow((double)B, (double)E);  202 : : }  203 : :   204 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  205 : : \* Function Name: linkGetMirrorState()   206 : : \* Purpose: Returns the mirroring state.  207 : : \* Parameters: vsbRelationRecord \*rr  208 : : \* Return Values: int  209 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  210 : : int VSBLinkReader::linkGetMirrorState (vsbRelationRecord \*rr) const  211 : 1261 : {  212 : 1261 : short int si\_mirrorOp = 0;  213 : 1261 : vsbMirrorNdCellOp \*mirrorOp = (vsbMirrorNdCellOp\*) &si\_mirrorOp;  214 : 1261 : si\_mirrorOp = \*((short int\*) ((char\*) rr + l\_MIRROR\_ND\_CELL\_OP\_OFFSET) );  215 : 1261 : si\_mirrorOp = BE2LE\_16 (si\_mirrorOp);  216 : 1261 : return mirrorOp->Mirror;  217 : : }  218 : :   219 : 1215 : } //namespace nft |

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