| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [LibVsb/src](http://docs.google.com/index.html) - dec\_types.cpp (source / [functions](http://docs.google.com/dec_types.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 8 | 30 | 26.7 % | | Date: | 2012-03-23 |  | Functions: | 4 | 5 | 80.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 0 | 14 | 0.0 % | |  | |
|  |

|  |
| --- |
| 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 15:52:50 IST 2007  6 : : \* File Name: dec\_types.cpp   7 : : \* Purpose:   8 : : \* Author:   9 : : \*  10 : : \* Revision History:   11 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/   12 : :   13 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  14 : : \* Header Include section   15 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  16 : : #include "dec\_types.h"  17 : : #include "Logger.h"  18 : : #include "ErrorHandler.h"  19 : :   20 : :   21 : : namespace nft {  22 : : /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*r  23 : : \* Function Name: getPatternSize()   24 : : \* Purpose: Returns the pattern size for an Elementary Pattern.  25 : : \* Parameters: int32\_t& Sx,   26 : : \* int32\_t& Sy  27 : : \* Return Values: void  28 : : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  29 : : void EP::getPatternSize(int32\_t& Sx, int32\_t& Sy) const  30 : 0 : {  31 [ # # ][ # # : 0 : switch(PatCode)  # # # # ]  32 : : {  33 : : case 0x01 :  34 : : case 0x02 :  35 : : case 0x03 :  36 : : case 0x04 :  37 : : case 0x0d :  38 : : case 0x0e :  39 : 0 : Sx = L1;  40 : 0 : Sy = L1 + L2;  41 : 0 : break ;  42 : : case 0x09 :  43 : : case 0x0a :  44 : : case 0x0b :  45 : : case 0x0c :  46 : : case 0x0f :  47 : : case 0x10 :  48 : 0 : Sx = L1 + L2;  49 : 0 : Sy = L2;  50 : 0 : break ;  51 : : case 0x05 :  52 : : case 0x06 :  53 : 0 : Sx = L1;  54 : 0 : Sy = L1 + L2 + L1;  55 : 0 : break ;  56 : : case 0x07 :  57 : : case 0x08 :  58 : 0 : Sx = L1 + L2 + L2;  59 : 0 : Sy = L2;  60 : 0 : break ;  61 : : case 0x11 :  62 : : case 0x31 :  63 : : case 0x32 :  64 : : case 0x33 :  65 : : case 0x34 :  66 : : case 0x35 :  67 : 0 : Sx = L1;  68 : 0 : Sy = L2;  69 : : break ;  70 : : default :  71 : 0 : break ;  72 : : /\*  73 : : logAdd(glog,LOG\_LV2,  74 : : "ERR:[getPatternSize()]Invalid Pattern code 0x%x\n",PatCode);\*/  75 : : //Log::logger::instance()->logAdd(18,"[ERROR] Invalid Pattern code"); //TODO  76 : : //errMsgs\* pInstance = errMsgs :: instance();  77 : : //std::string sTemp = pInstance->LookUpErrorTable(25, PatCode);  78 : : //throw new util::Error(sTemp, (const char\*)\_\_FILE\_\_,   79 : : //(uint32\_t)\_\_LINE\_\_, NULL);  80 : : }  81 [ # # ][ # # ]: 0 : if( ( Sx == Sy ) &&   82 : : (PatCode == 0x0d || PatCode == 0x0e ||   83 : : PatCode == 0x0f || PatCode == 0x10))  84 : : {  85 : : /\*  86 : : logAdd(glog,LOG\_LV4, "WRN:[getPatternSize()]Pattern Bounding Size is not valid\n");  87 : : logAdd(glog,LOG\_LV4, "\tSx = %d, Sy = %d for Pattern Code =0x%x\n",  88 : : \*Sx, \*Sy, ep->PatCode);  89 : : return 502;  90 : : \*/  91 : 0 : return;   92 : : }  93 : :   94 [ # # ]: 0 : if( 0 >= Sx || 0 >= Sy)  95 : : {  96 : : /\*  97 : : logAdd(glog,LOG\_LV4, "WRN:[getPatternSize()]Pattern Bounding Size is not valid\n");  98 : : logAdd(glog,LOG\_LV4,  99 : : "\tSx = %d, Sy = %d for Pattern Code =0x%x\n", \*Sx, \*Sy, ep->PatCode);  100 : : return 501;  101 : : \*/  102 : 0 : return;  103 : : }  104 : 0 : return ;  105 : : }  106 : :   107 : 108 : void FlattenedContext::setIsMoreCell(bool value){  108 : 108 : isMoreCell = value;  109 : : }  110 : :   111 : 162 : void FlattenedContext::setNumCellX(unsigned int value){  112 : 162 : numCellX = value;  113 : : }  114 : :   115 : 323 : void FlattenedContext::setNumCellY(unsigned int value){  116 : 323 : numCellY = value;  117 : : }  118 : 1215 : } //namespace nft |

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