| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [InputParams/src](http://docs.google.com/index.html) - ConverterCommandLineOptions.cpp (source / [functions](http://docs.google.com/ConverterCommandLineOptions.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 23 | 140 | 16.4 % | | Date: | 2012-03-23 |  | Functions: | 6 | 21 | 28.6 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 7 | 35 | 20.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include <sstream>  2 : : #include "ConverterCommandLineOptions.h"  3 : : //#include "ErrorHandler.h"  4 : : #include "print.h"  5 : : #include "SimpleOpt.h"  6 : : #include "UtilParams.h"  7 : :   8 : : namespace nft {  9 : :   10 : :   11 : : ConverterCommandLineOptions::ConverterCommandLineOptions()  12 : 1125 : :mLogLevel(1),  13 : : mOfflineLogFile(),  14 : : mErrorFile(),  15 : : mOutDir(),  16 : : mDumpMode( 0 ),  17 : : mShortHelp( false ),  18 : : mLongHelp( false ),  19 : : mVersion( false ),  20 : : mMngProc(),  21 : : mDistributor(),  22 : : mDose(),  23 : : mLogServer(),  24 : 9000 : mDistProc() {  25 : : }  26 : :   27 : : ConverterCommandLineOptions::ConverterCommandLineOptions(  28 : : const ConverterCommandLineOptions &source )  29 : 0 : :mLogLevel( source.mLogLevel ),  30 : : mOfflineLogFile( source.mOfflineLogFile ),  31 : : mErrorFile( source.mErrorFile ),  32 : : mOutDir( source.mOutDir ),  33 : : mDumpMode( source.mDumpMode ),  34 : : mShortHelp( source.mShortHelp ),  35 : : mLongHelp( source.mLongHelp ),  36 : : mVersion( source.mVersion ),  37 : : mMngProc( source.mMngProc ),  38 : : mDistributor( source.mDistributor ),  39 : : mDose( source.mDose ),  40 : : mLogServer( source.mLogServer ),  41 : 0 : mDistProc( source.mDistProc ) {  42 : : }  43 : :   44 : :   45 : 2250 : ConverterCommandLineOptions::~ConverterCommandLineOptions() {  46 : : }  47 : :   48 : : ConverterCommandLineOptions &  49 : : ConverterCommandLineOptions::operator=(  50 : 0 : const ConverterCommandLineOptions & source) {  51 [ # # ]: 0 : if( this != &source )  52 : : {  53 : 0 : mLogLevel = source.mLogLevel;  54 : 0 : mOfflineLogFile = source.mOfflineLogFile;  55 : 0 : mErrorFile = source.mErrorFile;  56 : 0 : mOutDir = source.mOutDir;  57 : 0 : mShortHelp = source.mShortHelp;  58 : 0 : mLongHelp = source.mLongHelp;  59 : 0 : mVersion = source.mVersion;  60 : 0 : mMngProc = source.mMngProc;  61 : 0 : mDistributor = source.mDistributor;  62 : 0 : mDose = source.mDose;  63 : 0 : mLogServer = source.mLogServer;  64 : 0 : mDistProc = source.mDistProc;  65 : : }  66 : 0 : return \*this;  67 : : }  68 : :   69 : : return\_t  70 : : ConverterCommandLineOptions::parseCommandLine(  71 : 1125 : int argc, char \*argv[] ) {  72 : 1125 : std::ostringstream sout;  73 [ + - ][ + + ]: 22500 : for( int i = 0; i < argc; i++ )  74 : : {  75 : 21375 : sout<<argv[i]<<" ";  76 : : }  77 : : enum {  78 : : ARG\_MNGPROC = 256,  79 : : ARG\_DISTPROC,  80 : : ARG\_DISTRIBUTORPROC,  81 : : ARG\_DOSEPROC,  82 : : ARG\_DUMPDIR,  83 : : ARG\_LOGSERVER,  84 : : ARG\_LOGLEVEL,  85 : : ARG\_LOGFILE,  86 : : ARG\_ERRORFILE,  87 : : ARG\_CONC,  88 : : ARG\_DIAGLEVEL,  89 : : ARG\_SHOTDUMP,  90 : : ARG\_SHORTHELP,  91 : : ARG\_LONGHELP,  92 : : ARG\_SHORT\_VERSION,  93 : : ARG\_VERSION  94 : : };  95 : :   96 : 1125 : CSimpleOpt::SOption options[] = {  97 : : { ARG\_MNGPROC, "-mng\_proc", SO\_REQ\_SEP},  98 : : { ARG\_DISTPROC, "-proc\_name", SO\_REQ\_SEP},  99 : : { ARG\_DISTRIBUTORPROC, "-distributor", SO\_REQ\_SEP},  100 : : { ARG\_DOSEPROC, "-dosereconst", SO\_REQ\_SEP},  101 : : { ARG\_DUMPDIR, "-dump\_dir", SO\_REQ\_SEP},  102 : : { ARG\_LOGSERVER, "-log\_server", SO\_REQ\_SEP},  103 : : { ARG\_LOGLEVEL, "-log\_level", SO\_REQ\_SEP},  104 : : { ARG\_LOGFILE, "-log\_file", SO\_REQ\_SEP},  105 : : { ARG\_ERRORFILE, "-error\_file", SO\_REQ\_SEP},  106 : : { ARG\_SHORTHELP, "-h", SO\_NONE},  107 : : { ARG\_LONGHELP, "-help", SO\_NONE},  108 : : { ARG\_SHORT\_VERSION, "-v", SO\_NONE},  109 : : { ARG\_VERSION, "--version", SO\_NONE},  110 : : SO\_END\_OF\_OPTIONS  111 : 1125 : };  112 : :   113 : 1125 : CSimpleOpt args (argc, argv, options);  114 : :   115 : : // Processes all the command line options  116 : 1125 : bool bErrorFound = false;  117 : :   118 : 1125 : std::string errMsg;  119 : 1125 : while (args.Next())   120 : : {  121 : 1125 : ESOError error = args.LastError ();  122 [ + - ]: 1125 : if (error == SO\_SUCCESS)   123 : : {  124 : 1125 : switch (args.OptionId ())   125 : : {  126 : : case ARG\_MNGPROC :  127 : : {  128 : 0 : setMngProc( args.OptionArg() );  129 : 0 : break;  130 : : }  131 : : case ARG\_DISTPROC :  132 : : {  133 : 0 : setDistProc( args.OptionArg() );  134 : 0 : break;  135 : : }  136 : : case ARG\_DISTRIBUTORPROC :  137 : : {  138 : 0 : setDistributor( args.OptionArg() );  139 : 0 : break;  140 : : }  141 : : case ARG\_DOSEPROC :  142 : : {  143 : 0 : setDose( args.OptionArg() );  144 : 0 : break;  145 : : }  146 : : case ARG\_DUMPDIR :  147 : : {  148 : 0 : setOutDir( args.OptionArg() );  149 : 1125 : break;  150 : : }  151 : : case ARG\_LOGSERVER :  152 : : {  153 : 1125 : setLogServer( args.OptionArg() );  154 : 0 : break;  155 : : }  156 : : case ARG\_LOGLEVEL :  157 : : {  158 : : uint64\_t logLevel = getUIntValue(  159 : 0 : args.OptionArg() );  160 : 0 : if( std::numeric\_limits<uint64\_t>::max()  161 : : == logLevel )  162 : : {  163 : 0 : bErrorFound = true;  164 : 0 : errMsg = "Invalid option for -log\_level";  165 : : }  166 : 0 : setLogLevel( static\_cast<uint16\_t>(logLevel) );  167 : 0 : break;  168 : : }  169 : : case ARG\_LOGFILE :  170 : : {  171 : 0 : setOfflineLogFile( args.OptionArg() );  172 : 0 : break;  173 : : }  174 : : case ARG\_SHORTHELP :  175 : : case ARG\_LONGHELP :  176 : : {  177 : 0 : setHelp( true );  178 : 0 : break;  179 : : }  180 : : case ARG\_SHORT\_VERSION:  181 : : case ARG\_VERSION :  182 : : {  183 : 0 : setVersion( true );  184 : 0 : break;  185 : : }  186 : : case ARG\_ERRORFILE :  187 : : {  188 : 0 : setErrorFile( args.OptionArg() );  189 : 0 : break;  190 : : }  191 : : }  192 : :   193 : : }  194 : : else   195 : : {  196 [ # # ][ # # : 0 : switch (error)   # # # ]  197 : : {  198 : : case SO\_OPT\_INVALID:  199 : : {  200 : 0 : bErrorFound = true;  201 : 0 : errMsg = "Invalid Option";  202 : 0 : break;  203 : : }  204 : : case SO\_OPT\_MULTIPLE:  205 : : {  206 : 0 : bErrorFound = true;  207 : 0 : errMsg = "Multiple options matched the supplied option text";  208 : 0 : break;  209 : : }  210 : : case SO\_ARG\_INVALID:  211 : : {  212 : 0 : bErrorFound = true;  213 : 0 : errMsg = "Argument was supplied but is not valid for this option";  214 : 0 : break;  215 : : }  216 : : case SO\_ARG\_INVALID\_TYPE:  217 : : {  218 : 0 : bErrorFound = true;  219 : 0 : errMsg = "Argument was supplied in wrong format for this option";  220 : 0 : break;  221 : : }  222 : : case SO\_ARG\_MISSING:  223 : : {  224 : 0 : bErrorFound = true;  225 : 0 : errMsg = "Required argument was not supplied";  226 : : break;  227 : : }  228 : : default:  229 [ + + ]: 2250 : break;  230 : : }  231 : : }  232 : : }  233 : :   234 [ - + ]: 1125 : if( bErrorFound )  235 : : {  236 : 0 : e9k\_error("Invalid Parameter: %s \n",errMsg.c\_str());  237 : 0 : return eRet\_failure;  238 : : }  239 : 1125 : return eRet\_success;  240 : : }  241 : :   242 : : return\_t   243 : : ConverterCommandLineOptions::validate()  244 : 0 : {  245 : 0 : std::string errMsg;  246 : 0 : bool bValidate = false;  247 : :   248 [ # # ]: 0 : if( !mLogServerFlag)  249 : : {  250 : 0 : errMsg = "-log\_server";  251 : 0 : bValidate = true;  252 : : }  253 : :   254 [ # # ]: 0 : if( !mLogFileFlag )  255 : : {  256 : 0 : errMsg = "-log\_file";  257 : 0 : bValidate = true;  258 : : }  259 : :   260 [ # # ]: 0 : if( !mErrorFileFlag )  261 : : {  262 : 0 : errMsg = "-error\_file";  263 : 0 : bValidate = true;  264 : : }  265 : :   266 [ # # ]: 0 : if( !mMngProcFlag )  267 : : {  268 : 0 : errMsg = "-mng\_proc";  269 : 0 : bValidate = true;  270 : : }  271 : :   272 [ # # ]: 0 : if( !mDistProcFlag )  273 : : {  274 : 0 : errMsg = "-proc\_name";  275 : 0 : bValidate = true;  276 : : }  277 : :   278 [ # # ]: 0 : if( !mDistributorFlag )  279 : : {  280 : 0 : errMsg = "-distributor";  281 : 0 : bValidate = true;  282 : : }  283 : :   284 [ # # ]: 0 : if( !mDoseFlag )  285 : : {  286 : 0 : errMsg = "-dose";  287 : 0 : bValidate = true;  288 : : }  289 : :   290 [ # # ]: 0 : if( bValidate )   291 : : {  292 : 0 : e9k\_error("Missing Option %s \n",errMsg.c\_str());  293 : 0 : return eRet\_failure;  294 : : }  295 : :   296 : 0 : return eRet\_success;  297 : : }  298 : :   299 : : void  300 : 0 : ConverterCommandLineOptions::setLogLevel(uint16\_t value) {  301 : 0 : mLogLevel = value;  302 : : }  303 : :   304 : : void  305 : : ConverterCommandLineOptions::setOfflineLogFile(  306 : 0 : const std::string &value) {  307 : 0 : mOfflineLogFile = value;  308 : 0 : mLogFileFlag = true;  309 : : }  310 : :   311 : : void  312 : : ConverterCommandLineOptions::setErrorFile(  313 : 0 : const std::string &value) {  314 : 0 : mErrorFile = value;  315 : 0 : mErrorFileFlag = true;  316 : : }  317 : : void  318 : : ConverterCommandLineOptions::setOutDir(  319 : 0 : const std::string &value) {  320 : 0 : mOutDir = value;  321 : 0 : mOutDirFlag = true;  322 : : }  323 : :   324 : : void  325 : 0 : ConverterCommandLineOptions::setDumpMode(int32\_t value) {  326 : 0 : mDumpMode = value;  327 : : }  328 : :   329 : : void  330 : 0 : ConverterCommandLineOptions::setHelp(bool value) {  331 : 0 : mLongHelp = value;  332 : 0 : mShortHelp = value;  333 : : }  334 : :   335 : : void  336 : 0 : ConverterCommandLineOptions::setVersion(bool value) {  337 : 0 : mVersion = value;  338 : : }  339 : :   340 : : void  341 : 0 : ConverterCommandLineOptions::setMngProc(const std::string &value) {  342 : 0 : mMngProc = value;  343 : 0 : mMngProcFlag = true;  344 : : }  345 : :   346 : : void  347 : 0 : ConverterCommandLineOptions::setDistributor(const std::string &value) {  348 : 0 : mDistributor = value;  349 : 0 : mDistributorFlag = true;  350 : : }  351 : :   352 : : void  353 : 1125 : ConverterCommandLineOptions::setLogServer(const std::string &value) {  354 : 1125 : mLogServer = value;  355 : 0 : mLogServerFlag = true;  356 : : }  357 : :   358 : : void  359 : 0 : ConverterCommandLineOptions::setDose(const std::string &value) {  360 : 0 : mDose = value;  361 : 0 : mDoseFlag = true;  362 : : }  363 : :   364 : : void  365 : 0 : ConverterCommandLineOptions::setDistProc(const std::string &value) {  366 : 0 : mDistProc = value;  367 : 0 : mDistProcFlag = true;  368 : : }  369 : :   370 : : } // namespace nft |

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