ClearCase Tool Suite Implementation Reference Manual

Version 1.17-r350

Generated by Doxygen 1.5.4-20071203

Thu Jan 3 21:46:16 2008

Contents

1 ClearCase Tool Suite Implementation Main Page

1.1 Content

This documentation describes the implementation of the ClearCase Tool Suite.

It depends on the

- **SysToMath Base C Library**
- **SysToMath Aids C++ Library**
- **SysToMath Tool Aids C++ Library**

belonging to the SysToMath C and C++ Libraries packages.

2 ClearCase Tool Suite Implementation Module Index

2.1 ClearCase Tool Suite Implementation Modules

Here is a list of all modules:

3 ClearCase Tool Suite Implementation Directory Hierarchy

3.1 ClearCase Tool Suite Implementation Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

4 ClearCase Tool Suite Implementation Class Index

4.1 ClearCase Tool Suite Implementation Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

5 ClearCase Tool Suite Implementation Class Index

5.1 ClearCase Tool Suite Implementation Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

6 ClearCase Tool Suite Implementation File Index

6.1 ClearCase Tool Suite Implementation File List

Here is a list of all files with brief descriptions:

7 ClearCase Tool Suite Implementation Module Documentation

7.1 ctmktree Main Program

7.1.1 Detailed Description

Manual page and main function of the console program ctmktree.

Files

- file ctmktree.cpp

  Main program of the command line based ctmktree application.

Classes

- struct CtData

  An instance of POD type CtData is assigned to each directory to be handled.

Defines

- #define PATHLEN 1024
- #define PATHSEP '/'

Functions

- static bool isVobElement (const std::string &path, bool isDirectory=false)

  The local function isVobElement returns true, if the file (default) or directory designated by path is a VOB element, else false.

- static bool isCheckedOut (const std::string &path, bool isDirectory=false)

  The local function isCheckedOut returns true, if the file (default) or directory designated by path is checked out, else false.

- static bool isWin32Executable (const std::string &path)

  The local function isWin32Executable returns true, if path designates a WIN32 executable file (case unsensitively the extension '.exe' or '.com').

- static int execCommand (const std::string &command, bool verbose=false)

  The local function execCommand returns 0, if the console command was executed successfully.

- static int elemhandler (const char *dirname, const char *entryname, const struct stat *direntry, int status, void *data, size_t len)

The local function elemhandler serves as callback function of recurs.

- static void error (const char *msg,...)

  Print error message and terminate the program.

- int main (int argc, char *argv[ ])

  Main function implementing the command.

Variables

- const char * CopyRight = "(C) 2004-2008 Tom Michaelis, SysToMath"
- const char * Version = "1.17-r350"
- static const char * man [ ]

  Man page of the ClearCase Tool Suite command ctmktree.

7.1.2 Define Documentation

7.1.2.1 #define PATHLEN 1024

Definition at line 197 of file ctmktree.cpp.

Referenced by isCheckedOut(), isLabel(), and isVobElement().

7.1.2.2 #define PATHSEP '/'

Definition at line 204 of file ctmktree.cpp.

Referenced by elemhandler(), and main().

7.1.3 Function Documentation

7.1.3.1 static bool isVobElement (const std::string & *path*, bool *isDirectory* = `false`) [static]

The local function isVobElement returns true, if the file (default) or directory designated by path is a VOB element, else false.

Throws on error.

Definition at line 775 of file ctmktree.cpp.

References PATHLEN, and stmGetoptsProg.

Referenced by elemhandler(), CtOperate::init(), CtData::init(), and main().

7.1.3.2 static bool isCheckedOut (const std::string & *path*, bool *isDirectory* = `false`) [static]

The local function isCheckedOut returns true, if the file (default) or directory designated by path is checked out, else false.

Throws on error.

Definition at line 816 of file ctmktree.cpp.

References PATHLEN, and stmGetoptsProg.

Referenced by elemhandler(), CtOperate::init(), and CtData::init().

7.1.3.3 static bool isWin32Executable (const std::string & *path*) [static]

The local function isWin32Executable returns true, if path designates a WIN32 executable file (case unsensitively the extension '.exe' or '.com').

Definition at line 857 of file ctmktree.cpp.

Referenced by elemhandler(), and DirData::handler().

7.1.3.4 static int execCommand (const std::string & *command*, bool *verbose* = false) [static]

The local function execCommand returns 0, if the console command was executed successfully.

If verbose is true (default is false), additional diagnostics are output to stdout.

Definition at line 868 of file ctmktree.cpp.

Referenced by CtOperate::deinit(), elemhandler(), ArgList::exec(), CtOperate::init(), and main().

7.1.3.5 static int elemhandler (const char * *dirname*, const char * *entryname*, const struct stat * *direntry*, int *status*, void * *data*, size_t *len*) [static]

The local function elemhandler serves as callback function of recurs.

Definition at line 606 of file ctmktree.cpp.

References CtData::checkedOut, CtData::cicommentopt, CtData::deinit(), error(), execCommand(), CtData::init(), isVobElement(), isWin32Executable(), CtData::originallyCheckedOut, CtData::parentCheckedOut, CtData::parentpath, CtData::parentVobElement, PATHSEP, CtData::pElements, CtData::pWin32Execs, stmGetoptsProg, StmRecursDbeg, StmRecursDend, StmRecursLeaf, CtData::verbose, and CtData::vobElement.

Referenced by main().

7.1.3.6 static void error (const char * *msg*, ...) [static]

Print error message and terminate the program.

The local function error prints the printf like formatted message msg on stderr followed by a newline character and terminates the program with exit code 1.

Definition at line 893 of file ctmktree.cpp.

Referenced by elemhandler(), and main().

7.1.3.7 int main (int *argc*, char * *argv*[ ])

Main function implementing the command.

Definition at line 352 of file ctmktree.cpp.

References CopyRight, elemhandler(), error(), execCommand(), stm::pstring::isContainedIn(), man, PATHSEP, StmElements, stmGetopts(), StmGetoptsEnd, StmGetoptsOk, stmGetoptsOptarg, stmGetoptsOpterr, stmGetoptsOptfp, stmGetoptsOptind, StmGetoptsOptionArgError, stmGetoptsProg, stmGetoptsStrict, stmPrintManual(), stmPrintSynopsis(), stmRecurs, StmRecursDoNotRecurse, StmTrue, and

Version.

7.1.4 Variable Documentation

7.1.4.1 const char* CopyRight = "(C) 2004-2008 Tom Michaelis, SysToMath"

Definition at line 71 of file ctmktree.cpp.

Referenced by main().

7.1.4.2 const char* Version = "1.17-r350"

Definition at line 72 of file ctmktree.cpp.

Referenced by main(), and CtUpdate::openProtocol().

7.1.4.3 const char* man[ ] [static]

Initial value:

```
{
"NAME                                                                    ",
"       ctmktree - make or augment a ClearCase tree                      ",
"                                                                        ",
"SYNOPSIS                                                                ",
"       ctmktree [-vf] {-c <comment> | -C <comment-file> | -n} [<directory>]  ",
"       ctmktree -h                                                      ",
"       ctmktree -V                                                      ",
"                                                                        ",
"DESCRIPTION                                                             ",
"       Recursively creates ClearCase VOB elements for all view private files "
"       and directories in the tree spanned by <directory> (or by the current "
"       working directory, if <directory> is not supplied).  For this to work,"
"       exactly one of the comment options -c, -C or -n has to be given.  If  "
"       option -f is given, the handling is not recursive.               ",
"       For directories existing in the VOB, it is irrelevant, if they are    "
"       already checked out.  If a directory in which elements have to be     "
"       created is not yet checked out, it is checked out, worked on it and   "
"       then checked in again.  Directories already checkd out are left       "
"       checked out.  Directories not existing in the VOB are created using   "
"       the ClearCase command mkdir.  These directories are checked in with   "
"       version /main/1 after all its subelements have been created.  It is an"
"       error, if <directory> has to be created and its parent directory is   "
"       no ClearCase VOB element.                                        ",
"       For each file in the specified tree, ctmktree checks to see whether an"
"       element of that name exists in the VOB.  If not, ctmktree invokes the "
"       ClearCase command mkelem on the file and checks it in with version    "
"       /main/1.  Then, for each file checked in which has case insensitively "
"       the extension '.exe' or '.com', its executable attributes are set for "
"       everyone.                                                        ",
"       Unless option -h or -V is supplied, the command returns with an exit  "
"       code 0 on success and with an exit code 1, if any error did occur.    ",
"                                                                        ",
"OPTIONS                                                                 ",
"       -h     help: print this man page on standard output and exit with exit"
"              code 2.                                                   ",
"                                                                        ",
"       -V     version: print version info on standard output and exit with   "
"              exit code 2.                                              ",
"                                                                        ",
"       -v     verbose: issue additional diagnostic messages on stdout.  ",
"                                                                        ",
"       -f     flat: suppress recursive descent into the tree spanned by     "
```

```
"               <directory>.                                                    ",
"                                                                               ",
"       -c <comment>                                                            ",
"               use the string <comment> as ClearCase check in comment.         ",
"                                                                               ",
"       -C <comment-file>                                                       ",
"               use the content of <comment-file> as ClearCase check in         "
"               comment.  The file shall not be inside the tree spanned by      "
"               <directory>.                                                    ",
"                                                                               ",
"       -n      use no ClearCase check in comment.                              "
}
```

Man page of the ClearCase Tool Suite command ctmktree.

Definition at line 113 of file ctmktree.cpp.

7.2 ctoperate Main Program

7.2.1 Detailed Description

Manual page and main function of the console program ctoperate.

Files

- file ctoperate.cpp

  Main program of the command line based ctoperate application.

Classes

- struct Mode
- class ArgList
- struct CtOperate

  An instance of POD type CtOperate is assigned to each directory to be handled.

- struct RootData

  An instance of RootData is assigned to the destination root directory.

Defines

- #define PATHLEN 1024
- #define PATHSEP '/'

Typedefs

- typedef **stm::logger** Logger

Functions

- static bool isVobElement (const boost::filesystem::path &path)

  The local function isVobElement returns true, if path is a VOB element, else false.

- static bool isCheckedOut (const boost::filesystem::path &path)

  The local function isCheckedOut returns true, if is checked out, else false.

- static bool isLabel (const std::string &vobSelector, const std::string &label)

  The local function isLabel returns true, if label is defined as a label in the ClearCase VOB selected by vobSelector, else false.

- static bool isWin32Executable (const boost::filesystem::path &path)

  The local function isWin32Executable returns true, if path designates a WIN32 executable file (case unsensitively the extension '.exe' or '.com').

- static void execCommand (const std::string &command, const Mode &mode)

  The local function execCommand executes the console command.

- static int elemhandler (const char *dirname, const char *entryname, const struct stat *direntry, int status, void *data, size_t len)

  The local function elemhandler serves as callback function of recurs.

- int main (int argc, char *argv[ ])

  Main function implementing the command.

Variables

- const char * CopyRight = "(C) 2005-2008 Tom Michaelis, SysToMath"
- const char * Version = "1.17-r350"
- static const char * man [ ]

  Man page of the ClearCase Tool Suite command ctoperate.

- static size_t ArgList::limit = 2048

7.2.2 Define Documentation

7.2.2.1 #define PATHLEN 1024

Definition at line 236 of file ctoperate.cpp.

7.2.2.2 #define PATHSEP '/'

Definition at line 243 of file ctoperate.cpp.

7.2.3 Typedef Documentation

7.2.3.1 typedef stm::logger Logger

Definition at line 247 of file ctoperate.cpp.

7.2.4 Function Documentation

7.2.4.1 static bool isVobElement (const boost::filesystem::path & *path*) [static]

The local function isVobElement returns true, if path is a VOB element, else false.

Throws on error.

Definition at line 1405 of file ctoperate.cpp.

References PATHLEN.

7.2.4.2 static bool isCheckedOut (const boost::filesystem::path & *path*) [static]

The local function isCheckedOut returns true, if is checked out, else false.

Throws on error.

Definition at line 1449 of file ctoperate.cpp.

References PATHLEN.

7.2.4.3 static bool isLabel (const std::string & *vobSelector*, const std::string & *label*) [static]

The local function isLabel returns true, if label is defined as a label in the ClearCase VOB selected by vobSelector, else false.

Throws on error.

Definition at line 1493 of file ctoperate.cpp.

References PATHLEN.

Referenced by main().

7.2.4.4 static bool isWin32Executable (const boost::filesystem::path & *path*) [static]

The local function isWin32Executable returns true, if path designates a WIN32 executable file (case unsensitively the extension '.exe' or '.com').

Definition at line 1536 of file ctoperate.cpp.

7.2.4.5 static void execCommand (const std::string & *command*, const Mode & *mode*) [static]

The local function execCommand executes the console command.

If mode.action is Mode::Verbose, additional diagnostics are output to stdout. If mode.action is Mode::Simulate, actions are not performed but protocolled to stdout instead. On error a std::runtime_error exception is thrown.

Definition at line 1545 of file ctoperate.cpp.

References Mode::action, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::Indent, Mode::logger, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, Mode::Simulate, and Mode::Verbose.

7.2.4.6 static int elemhandler (const char * *dirname*, const char * *entryname*, const struct stat * *direntry*, int *status*, void * *data*, size_t *len*) [static]

The local function elemhandler serves as callback function of recurs.

Definition at line 1090 of file ctoperate.cpp.

References Mode::Operation::CheckIn, CtOperate::destdir, isCheckedOut(), isVobElement(), isWin32Executable(), Mode::Operation::MkElem, CtOperate::mode, CtOperate::pCheckin, CtOperate::pMkelem, CtOperate::pMklabel, CtOperate::pRemove, CtOperate::pUncheckout, CtOperate::pWin32Execs, Mode::Operation::Remove, StmRecursDbeg, StmRecursDend, and StmRecursLeaf.

7.2.4.7 int main (int *argc*, char * *argv*[ ])

Main function implementing the command.

Definition at line 453 of file ctoperate.cpp.

References Mode::action, stm::filespeclist::append(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::AutoInd, Mode::Operation::CheckIn, Mode::cicommentopt, CopyRight, elemhandler(), Mode::exclDirList, Mode::exclFileList, execCommand(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::Indent, stm::filefind< PathContainerT >::insert(), stm::pstring::isContainedIn(), isLabel(), isVobElement(), Mode::label, logger(), man, Mode::Operation::MkElem, Mode::Operation::None, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, stm::filespec::NoPathSeparators, Mode::Normal, Mode::operation, Mode::Operation::Remove, stm::logger< stdIndent, lineLength, errorConsole, lockingPolicy >::setConsole(), Mode::Simulate, stmDstrPrintf(), StmElements, stmGetopts(), StmGetoptsArgumentError, StmGetoptsEnd, stmGetoptsErrbuf, StmGetoptsOk, stmGetoptsOptarg, stmGetoptsOpterr, stmGetoptsOptind, StmGetoptsOptionArgError, StmGetoptsOptionError, stmGetoptsProg, stmGetoptsStrict, stmPrintManual(), stmPrintSynopsis(), StmRectrlTry, stmRecurs, StmRecursDoNotRecurse, StmTrue, Mode::Verbose, and Version.

7.2.5 Variable Documentation

7.2.5.1 const char* CopyRight = "(C) 2005-2008 Tom Michaelis, SysToMath"

Definition at line 71 of file ctoperate.cpp.

7.2.5.2 const char* Version = "1.17-r350"

Definition at line 72 of file ctoperate.cpp.

7.2.5.3 const char* man[ ] [static]

Initial value:

```
{
"NAME                                                                    ",
"       ctoperate - operate on a ClearCase VOB                           ",
"                                                                        ",
"SYNOPSIS                                                                ",
"       ctoperate [-f] [(-v | -t)] (-c <comment> | -C <comment-file> | -n)    ",
"                 [-o (mk | ci | rm)] [(-x <file-spec> | -X <dir-spec>)...]   ",
"                 [(-l | -L) <label>] [(<file-spec> | <dir-spec>)...]         ",
"       ctoperate -h                                                     ",
"       ctoperate -V                                                     ",
"                                                                        ",
"DESCRIPTION                                                             ",
"       The command operates on all files specified by the <file-spec>       "
"       arguments and all elements in the trees spanned by the directories    "
"       specified by the <dir-spec> arguments (or by the current working      "
```

```
"       directory, if no arguments are supplied).  The command operation      "
"       has to be specified by supplying option -o.  Operation -o mk is used  "
"       to create ClearCase VOB elements for all view private files and       "
"       directories leaving them checked out.  Operation -o ci additionally   "
"       causes all checked out file and directory elements to be checked in.  "
"       In contrast to these operations operation -o rm removes view private  "
"       files and directories and does an uncheckout for checked out file and "
"       directory elements. If option -f is given, the handling of directories"
"       is not recursive.                                                     ",
"       The arguments <file-spec> and <dir-spec> may contain pattern matching "
"       operators as explained below.  Be sure to quote them according to the "
"       needs of the shell used.                                              ",
"       For ctoperate to work, the parent directories of the files and of the "
"       directories handled shall be ClearCase VOB elements. Moreover, exactly"
"       one of the comment options -c, -C or n has to be given.               ",
"       Unless option -h or -V is supplied, the command returns with an exit  "
"       code 0 on success and with an exit code 1, if any error did occur.    ",
"                                                                             ",
"OPTIONS                                                                      ",
"       -h     help: print this man page on standard output and exit with exit"
"              code 2.                                                        ",
"                                                                             ",
"       -V     version: print version info on standard output and exit with   "
"              exit code 2.                                                   ",
"                                                                             ",
"       -f     flat: suppress recursive descent into the trees spanned by     "
"              directories.                                                   ",
"                                                                             ",
"       -v     verbose: issue additional diagnostic messages on stdout.       ",
"                                                                             ",
"       -t     tell: do no actions, issue messages on stdout instead.         ",
"                                                                             ",
"       -c <comment>                                                          ",
"              use the string <comment> as ClearCase check in comment.        ",
"                                                                             ",
"       -C <comment-file>                                                     ",
"              use the content of <comment-file> as ClearCase check in        "
"              comment.  The file shall not be inside the tree spanned by one "
"              of the directories <directory>.                                ",
"                                                                             ",
"       -n     use no ClearCase check in comment.                             ",
"                                                                             ",
"       -o <operation>                                                        ",
"              <operation> is one of the following actions:                   ",
"              mk: make ClearCase VOB elements for view private elements      ",
"                  leaving them checked out                                   ",
"              ci: additionally check in checked out VOB elements             ",
"              rm: in contrast remove view private elements and uncheckout    ",
"                  checked out VOB elements                                   ",
"                                                                             ",
"       -x <file-spec>                                                        ",
"              exclude files matched by <file-spec> which may contain pattern "
"              matching operators as explained below but no path separator.   "
"              Be sure to quote <file-spec> according to the needs of the     "
"              shell used.                                                    ",
"                                                                             ",
"       -X <dir-spec>                                                         ",
"              exclude directories and the whole trees spanned by them matched"
"              by <dir-spec> which may contain pattern matching operators as  "
"              explained below but no path separator.  Be sure to quote       "
"              <dir-spec> according to the needs of the shell used.           ",
"                                                                             ",
"       -l <label-list>                                                       ",
"              label all elements handled with the labels specified by the    "
"              non-empty whitespace or comma separated list <label-list>.  The"
"              labels are generated from each list element by brace expansion "
"              as in the bash shell.  Be sure to quote <label-list> according "
```

```
"              to the needs of the shell used.  If a label does not yet exist,"
"              it is created.  This option may be used only in conjunction    "
"              with '-m ci -u ci'.  It is an error, if some other ClearCase   "
"              version of an element to be labeled already bears the label.   ",
"                                                                             ",
"       -L <label-list>                                                       ",
"              label all elements handled with the labels specified by the    "
"              non-empty whitespace or comma separated list <label-list>.  The"
"              labels are generated from each list element by brace expansion "
"              as in the bash shell.  Be sure to quote <label-list> according "
"              to the needs of the shell used.  If a label does not yet exist,"
"              it is created.  This option may be used only in conjunction    "
"              with '-m ci -u ci'.  If some other Clearcase version of an     "
"              element to be labeled already bears the label, this existing   "
"              label is moved to the version just handled.                    ",
"                                                                             ",
"FILENAME EXPANSION                                                           ",
"       The evaluation of the spec arguments <file-spec> and <dir-spec> is    "
"       done using pattern matching based file name expansion.  If the spec   "
"       contains one of the unescaped extended regular expression operators   "
"       '(' or '|', it is interpreted as extended regular expression as       "
"       defined in the Open System Group Base Specification Issue 6, IEEE Std "
"       1003.1-2004 (XBD: 9.4).  Otherwise if the spec contains no unescaped  "
"       '{', it is interpreted as filename expansion pattern as defined in    "
"       IEEE Std 1003.1-2004 (XCU: 2.13.3).  Else the spec is brace expanded  "
"       as in the bash shell and each of the resulting words is interpreted as"
"       filename expansion pattern.                                           ",
"       On windows systems as escape character for brace expansion '%' is used"
"       instead of the usual '\\'.  The bracket expressions used in the       "
"       regular expressions and in the filename expansion patterns conform to "
"       IEEE Std 1003.1-2004 (XBD: 9.3.5) with the exception that equivalence "
"       classes [=x=] of collating elements are not supported.                "
}
```

Man page of the ClearCase Tool Suite command ctoperate.

Definition at line 84 of file ctoperate.cpp.

7.2.5.4 size_t ArgList::limit = 2048 [static, private, inherited]

Definition at line 352 of file ctoperate.cpp.

Referenced by ArgList::operator+=().

7.3 ctupdate Main Program

7.3.1 Detailed Description

Manual page and main function of the console program ctupdate.

Files

- file ctupdate.cpp

  Main program of the command line based ctupdate application.

Classes

- struct CtUpdate

- struct CtUpdate::Operation
- struct CtUpdate::SourceRoot
- struct CtUpdate::SourceRootList
- struct CtUpdate::DestRoot
- struct CtUpdate::DestRootList
- struct DirData

  A DirData object is assigned to each directory to be handled.

Functions

- int main (int argc, char ∗argv[ ])

  Main function implementing the command.

Variables

- const char ∗ CopyRight = "(C) 2005-2008 Tom Michaelis, SysToMath"
- const char ∗ Version = "1.17-r350"
- static const char ∗ man [ ]

  Man page of the ClearCase Tool Suite command ctupdate.

- static const **StmCmdOptSpec** CtUpdate::optSpec [ ]

  Definition of the command line option parameters.

7.3.2 Function Documentation

7.3.2.1 int main (int *argc*, char ∗ *argv*[ ])

Main function implementing the command.

Definition at line 853 of file ctupdate.cpp.

References stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::AutoInd, CtUpdate::ciCommentFile, CtUpdate::cmdLine, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::ColonTable, stm::tools::CtExceptions::Content, CopyRight, CtUpdate::defineDestRoot(), CtUpdate::destRoot, CtUpdate::destRootList, stm::tools::CtExceptions::Directory, CtUpdate::evalCommentOption(), CtUpdate::evalExceptionOption(), CtUpdate::evalLabelOption(), CtUpdate::evalModeOption(), stm::tools::Ct::getCiCommentOption(), stm::tools::Ct::getMode(), DirData::handler(), stm::path< NameT >::head(), stm::tools::CtExceptions::Include, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::Indent, stm::pstring::isContainedIn(), logger(), man, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::mkPath(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, CtUpdate::openProtocol(), stm::basic_logger< StringT, stdIndent, lineLength, lockingPolicy >::paragraph(), stm::tools::CtExceptions::PlainFile, CtUpdate::prog, CtUpdate::protocolFile, stm::logger< stdIndent, lineLength, errorConsole, lockingPolicy >::setConsole(), stm::tools::Ct::setMode(), CtUpdate::sourceRoot, StmCmdLineArgumentError, stmCmdLineGetArg(), stmCmdLineGetArgCount(), stmCmdLineGetCurrentArgIndex(), stmCmdLineGetCurrentOptionArg(), stmCmdLineGetCurrentOptionName(), stmCmdLineGetErrbuf(), stmCmdLineGetFirstOptionId(), stmCmdLineGetNextOptionId(), stmCmdLineGetOpterr, StmCmdLineOptionArgError, stmCmdLinePrintfErrbuf(), stmCmdLineSetOpterr, StmDebugAidsErrmsg, StmDebugAidsVerify,

StmElements, StmId, stmPrintManual(), stmPrintManualFooter(), stmPrintManualHeader(), stmPrintOptionHelp(), stmPrintSynopsis(), StmRectrlTry, StmRecursDoNotRecurse, stmRecurse(), stmTraverse(), stm::path< NameT >::tail_path(), and Version.

7.3.3 Variable Documentation

7.3.3.1 const char* CopyRight = "(C) 2005-2008 Tom Michaelis, SysToMath"

Definition at line 71 of file ctupdate.cpp.

7.3.3.2 const char* Version = "1.17-r350"

Definition at line 72 of file ctupdate.cpp.

7.3.3.3 const char* man[ ] [static]

Initial value:

```
{
"NAME                                                                    ",
"       ctupdate - integrate missing or newer files into a ClearCase VOB ",
"                                                                        ",
"SYNOPSIS                                                                ",
"       ctupdate [(--verbose | --tell-actions)] [--protocol=<file>]      ",
"                [(--args-env=<assignment>)...] [--args-from=<file>]     ",
"                (--comment=<string> | --comment-from=<file> | --no-comment)  ",
"                (                                                       ",
"                    [(--label=<label-list> | --replace-label=<label-list>)]  ",
"                    [--no-dest-root-label]                              ",
"                    [--make=<make-mode>]                                ",
"                    [--update=<update-mode>]                            ",
"                    [--erase=<erase-mode>]                              ",
"                    (                                                   ",
"                        [(--flat | --recursive) [--source-root=<directory>]] ",
"                        [--reset-exceptions]                            ",
"                        (                                               ",
"                            [--exclude-file=<spec>] [--exclude=<spec>]  ",
"                            [--include-file=<spec>] [--include=<spec>]  ",
"                        )...                                            ",
"                        (--source=<spec>)...                            ",
"                    )...                                                ",
"                    --dest-root=<directory>                             ",
"                )...                                                    ",
"       ctupdate --help [<option-name>...]                               ",
"       ctupdate --version                                               ",
"                                                                        ",
"DESCRIPTION                                                             ",
"       The command checks all source file system elements specified by the   "
"       <spec> arguments of option --source, whose corresponding elements in  "
"       the tree spanned by the current destination root specified by the     "
"       <directory> argument of option --dest-root are missing or are out of  "
"       date.  For each such source file system element the program           "
"       determines, if it has to be handled by matching its current source    "
"       root relative path against each <spec> element contained in the       "
"       exception list of the current destination root till a match succeeds. "
"       The first successful match stops the matching process and if the      "
"       matching element was marked for exclusion, the file system element is "
"       not handled, else if the element was marked for inclusion or if none  "
"       of the exception list elements matches, the file system element is    "
"       handled.  Each file system element determined to be handled is copied "
"       to the destination tree according to the options --make and --update. "
"       Option --erase controls if destination file system elements in        "
```

```
"       directories corresponding to a source directory not existing there are"
"       erased.  Unless option --flat is in effect, for source file system    "
"       elements designating a directory the whole tree spanned by this       "
"       directory is handled.                                                  ",
"       The <spec> option arguments may contain pattern matching operators as "
"       explained in section FILENAME EXPANSION.  Be sure to quote them       "
"       according to the needs of the shell used.                              ",
"       For ctupdate to work, the <directory> argument of option --dest-root  "
"       has to designate a ClearCase VOB element, and exactly one of the      "
"       comment options --comment, --comment-from or --no-comment has to be   "
"       given.                                                                 ",
"       For convenience, the option specifiers --source= and --dest-root= of  "
"       trailing options may be omitted.                                       ",
"       Unless option --help or --version is supplied, the command returns    "
"       with an exit code 0 on success and with an exit code 1, if any error  "
"       did occur.                                                             ",
"                                                                              ",
"GLOBAL OPTIONS                                                                ",
"       --help                                                                 ",
"       -h     If no argument <option-name> follows, print the command man    "
"              page on standard output, else the description of the options   "
"              <option-name> and exit with exit code 2.  The arguments        "
"              <option-name> shall be stated without leading '-' or '--'.     ",
"                                                                              ",
"       --version                                                              ",
"       -V     Print version info on standard output and exit with exit code  "
"              2.                                                             ",
"                                                                              ",
"       --verbose                                                              ",
"       -v     Issue additional diagnostic messages on stdout.  This option   "
"              acts globally and is mutually exclusive with the option        "
"              '--tell-actions'.                                              ",
"                                                                              ",
"       --tell-actions                                                         ",
"       -t     Perform no actions, issue messages on stdout instead.  This    "
"              option acts globally and is mutually exclusive with the option "
"              '--verbose'.                                                   ",
"                                                                              ",
"       --protocol=<file>                                                      ",
"       -P <file>                                                              ",
"              Generate the application protocol in <file>.  By default, the  "
"              protocol file 'ctupdate.txt' is generated in the current       "
"              directory, unless the current directory is not writable or the "
"              generation of a protocol is switched off by '--protocol=-'.    "
"              This option acts globally.                                      ",
"                                                                              ",
"       --args-env=<assignment>                                                ",
"       -E <assignment>                                                        ",
"              Defines an argument environment variable. The option argument  "
"              <assignment> shall be of the form <name>=<value>.               ",
"              The argument environment variable <name> with value <value> can"
"              be used in option or command arguments of the command line with"
"              the construct '@{<name>}' which is replaced then by <value>.   "
"              If an argument environment variable <name> is not found, next a"
"              shell environment variable <name> is searched.  If also this   "
"              variable does not exist, the construct is not replaced.        ",
"                                                                              ",
"       --args-from=<file>                                                     ",
"       -A <file>                                                              ",
"              Substitutes the content of <file> into the command line.  The  "
"              file <file> shall contain valid command line options and       "
"              command arguments with the extension that newline or '#' or ';'"
"              characters followed by arbitrary content up to the line end are"
"              regarded as white space.                                        ",
"              Especially <file> may contain other options '--args-from', or  "
"              argument or shell environment variable definitions and usages. ",
"                                                                              ",
```

```
"       --comment=<string>                                                  ",
"       -c <string>                                                         ",
"              Use <string> as ClearCase check in comment.  This option acts  "
"              globally.                                                    ",
"                                                                           ",
"       --comment-from=<file>                                               ",
"       -C <file>                                                           ",
"              Use the content of <file> as ClearCase check in comment.  The  "
"              file shall not be inside the tree spanned by <dest-directory>. "
"              This option acts globally.                                   ",
"                                                                           ",
"       --no-comment                                                        ",
"       -n     Use no ClearCase check in comment.  This option acts globally. ",
"                                                                           ",
"       --dest-root=<directory>                                             ",
"       -D <directory>                                                      ",
"              End the scope of the current destination root, define its path "
"              by <directory> and begin a new destination root scope with     "
"              still undefined path.  The destination root of the just ended  "
"              destination root scope corresponds to all source roots in the  "
"              source tree defined within this scope.                       ",
"                                                                           ",
"DESTINATION ROOT LOCAL OPTIONS                                             ",
"       --source-root=<directory>                                           ",
"       -S <directory>                                                      ",
"              Set <directory> as current source root.  This current source   "
"              root always corresponds to the current destination root in the "
"              destination tree.  This means that all <spec> arguments        "
"              following this option are regarded relative to <directory> in  "
"              the source tree and relative to the current destination root in"
"              the destination tree.  The source root defined by <directory>  "
"              is valid till another option --source-root occurs or till the  "
"              scope of the current destination root is ended by option       "
"              --dest-root.                                                 ",
"              The default value of the current source root is the current    "
"              directory and since this is a redefinable destination root     "
"              local option, this means that for each destination root this   "
"              default can be redefined by a new --source-root option.      ",
"                                                                           ",
"       --label=<label-list>                                                ",
"       -l <label-list>                                                     ",
"              Label all current destination root file system elements handled"
"              which are plain files or directories just checked in or        "
"              containing at least one element which also has been labeled    "
"              with the labels specified by the non-empty whitespace or comma "
"              separated list <label-list>.                                 ",
"              The ClearCase label names are generated from each list element "
"              by brace expansion as in the bash shell.  Be sure to quote     "
"              <label-list> according to the needs of the shell used.       ",
"              If a label name does not yet exist, it is created.  This option"
"              may be used only in conjunction with '--mode=checked-in' and   "
"              '--update=checked-in'.  It is an error, if some other ClearCase"
"              version of an element to be labeled already bears the label.   ",
"              This is a destination root local option, meaning that for each "
"              destination root a new set of labels has to be defined.      ",
"                                                                           ",
"       --replace-label=<label-list>                                        ",
"       -L <label-list>                                                     ",
"              Label all current destination root file system elements handled"
"              which are plain files or directories just checked in or        "
"              containing at least one element which also has been labeled    "
"              with the labels specified by the non-empty whitespace or comma "
"              separated list <label-list>.                                 ",
"              The ClearCase label names are generated from each list element "
"              by brace expansion as in the bash shell.  Be sure to quote     "
"              <label-list> according to the needs of the shell used.       ",
"              If a label name does not yet exist, it is created.  This option"
```

```
"              may be used only in conjunction with '--mode=checked-in' and   "
"              '--update=checked-in'.  If some other Clearcase version of an  "
"              element to be labeled already bears the label, this existing   "
"              label is moved to the version just handled.                    ",
"              This is a destination root local option, meaning that for each "
"              destination root a new set of labels has to be defined.        ",
"                                                                             ",
"       --no-dest-root-label                                                  ",
"       -N     Do not label the current destination root directory.  By       "
"              default the current destination root directory is labeled as   "
"              other destination directories.                                 ",
"              This is a destination root local option, meaning that for each "
"              destination root this option is not in effect by default.      ",
"                                                                             ",
"       --make=<make-mode>                                                    ",
"       -m <make-mode>                                                        ",
"              Make mode: <make-mode> is one of the following modes for making"
"              a new destination element:                                     ",
"              'vp' or 'view-private': make a view private element (default), ",
"              'ci' or 'checked-in': make a checked in VOB element,           ",
"              'co' or 'checked-out': make a checked out VOB element.         ",
"              If the defaults '--make=view-private', '--update=reversible'   "
"              and '--erase=no' have been used, the actions done by ctupdate  "
"              may be revoked by 'ctoperate --operation=remove'.              ",
"              This is a destination root local option, meaning that for each "
"              destination root, initially the default make mode              "
"              'view-private' is in effect, and a new make mode may be        "
"              defined.                                                       ",
"                                                                             ",
"       --update=<update-mode>                                                ",
"       -u <update-mode>                                                      ",
"              Update mode: <mode> is one of the following modes for updating "
"              an existing destination element:                               ",
"              'rv' or 'reversible': update the element reversibly (default), ",
"              'ci' or 'checked-in': update the element and check it in,      ",
"              'co' or 'checked-out': update it leaving it checked out.       ",
"              If the defaults '--make=view-private', '--update=reversible'   "
"              and '--erase=no' have been used, the actions done by ctupdate  "
"              may be revoked by 'ctoperate --operation=remove'.              ",
"              This is a destination root local option, meaning that for each "
"              destination root, initially the default update mode            "
"              'reversible' is in effect, and a new update mode may be        "
"              defined.                                                       ",
"                                                                             ",
"       --erase=<erase-mode>                                                  ",
"       -e <erase-mode>                                                       ",
"              Erase mode: <erase-mode> is one of the following modes for     "
"              erasing a destination element which does not correspond to a   "
"              source element:                                                ",
"              'no': do not erase the element (default),                      ",
"              'vp' or 'view-private': erase it only if it is view private,   ",
"              'all': erase it under all circumstances.                       ",
"              If the defaults '--make=view-private', '--update=reversible'   "
"              and '--erase=no' have been used, the actions done by ctupdate  "
"              may be revoked by 'ctoperate --operation=remove'.              ",
"              This is a destination root local option, meaning that for each "
"              destination root, initially the default erase mode 'no' is in  "
"              effect, and a new erase mode may be defined.                   ",
"                                                                             ",
"       --flat                                                                ",
"       -f     Suppress recursive descent into the trees spanned by source    "
"              directories.  This is a redefinable destination root local     "
"              option, meaning that for each destination root, it has to be   "
"              set anew and, moreover can be reset for each new source root by"
"              option --recursive.                                            ",
"                                                                             ",
"       --recursive                                                           ",
```

```
"       -r     Request recursive descent into the trees spanned by source     "
"              directories.  This option is in effect by default and since it "
"              is a redefinable destination root local option, this means that"
"              for each destination root, it is initially in effect, but can  "
"              be reset for each new source root by option --flat.            ",
"                                                                             ",
"       --exclude-file=<spec>                                                 ",
"       -x <spec>                                                             ",
"              Add all source file system elements matched by <spec> to the   "
"              exception list of the current destination root and mark them   "
"              for exclusion.  This option may be supplied more than once and "
"              <spec> may contain pattern matching operators as explained in  "
"              section FILENAME EXPANSION.  Be sure to quote <spec> according "
"              to the needs of the shell used.                                ",
"              If <spec> ends with a path separator character, the matching   "
"              process is restricted to directories else to plain files.  The "
"              matching occurs recursively for all defined source file system "
"              elements of the current source root.  If <spec> begins with a  "
"              path separator the matching is anchored at this source root.   ",
"              This is a destination root local option, meaning that for each "
"              destination root the exception list is initially empty.        ",
"                                                                             ",
"       --exclude=<spec>                                                      ",
"       -X <spec>                                                             ",
"              Add all source file system elements matched by <spec> to the   "
"              exception list of the current destination root and mark them   "
"              for exclusion.  This option may be supplied more than once and "
"              <spec> may contain pattern matching operators as explained in  "
"              section FILENAME EXPANSION.  Be sure to quote <spec> according "
"              to the needs of the shell used.                                ",
"              If <spec> ends with a path separator character, the matching   "
"              process is restricted to directories.  The matching occurs     "
"              recursively for all defined source file system elements of the "
"              current source root.  If <spec> begins with a path separator   "
"              the matching is anchored at this source root.                  ",
"              This is a destination root local option, meaning that for each "
"              destination root the exception list is initially empty.        ",
"                                                                             ",
"       --include-file=<spec>                                                 ",
"       -i <spec>                                                             ",
"              Add all source file system elements matched by <spec> to the   "
"              exception list of the current destination root and mark them   "
"              for inclusion.  This option may be supplied more than once and "
"              <spec> may contain pattern matching operators as explained in  "
"              section FILENAME EXPANSION.  Be sure to quote <spec> according "
"              to the needs of the shell used.                                ",
"              If <spec> ends with a path separator character, the matching   "
"              process is restricted to directories else to plain files.  The "
"              matching occurs recursively for all defined source file system "
"              elements of the current source root.  If <spec> begins with a  "
"              path separator the matching is anchored at this source root.   ",
"              This is a destination root local option, meaning that for each "
"              destination root the exception list is initially empty.        ",
"                                                                             ",
"       --include=<spec>                                                      ",
"       -I <spec>                                                             ",
"              Add all source file system elements matched by <spec> to the   "
"              exception list of the current destination root and mark them   "
"              for inclusion.  This option may be supplied more than once and "
"              <spec> may contain pattern matching operators as explained in  "
"              section FILENAME EXPANSION.  Be sure to quote <spec> according "
"              to the needs of the shell used.                                ",
"              If <spec> ends with a path separator character, the matching   "
"              process is restricted to directories.  The matching occurs     "
"              recursively for all defined source file system elements of the "
"              current source root.  If <spec> begins with a path separator   "
"              the matching is anchored at this source root.                  ",
```

```
"              This is a destination root local option, meaning that for each "
"              destination root the exception list is initially empty.        ",
"                                                                             ",
"       --reset-exceptions                                                    ",
"       -d     Delete the exception list of the current destination root.     ",
"                                                                             ",
"SOURCE ROOT LOCAL OPTIONS                                                    ",
"       --source=<spec>                                                       ",
"       -s <spec>                                                             ",
"              Add all file system elements in the tree spanned by the current"
"              source root matched by <spec> to the list of source file system"
"              elements to be handled for this source root.  This option may  "
"              be supplied more than once and <spec> may contain pattern      "
"              matching operators as explained in section FILENAME EXPANSION  "
"              with the exception that operator '**' is not supported. Be sure"
"              to quote <spec> according to the needs of the shell used.      ",
"              If <spec> ends with a path separator character, the matching   "
"              process is restricted to directories.  As the matching always  "
"              occurs relative to the current source root, <spec> shall be    "
"              relative and designate file system elements inside the tree    "
"              spanned by the current source root.                            ",
"              This is a source root local option, meaning that for each      "
"              source root the list of file system elements to be handled is  "
"              initially empty.                                               ",
"                                                                             ",
"FILENAME EXPANSION                                                           ",
"       The evaluation of the <spec> arguments is done using pattern matching "
"       based file name expansion.  If <spec> contains one of the unescaped   "
"       extended regular expression operators '(' or '|', it is interpreted as"
"       extended regular expression as defined in the Open System Group Base  "
"       Specification Issue 6, IEEE Std 1003.1-2004 (XBD: 9.4).  Otherwise if "
"       <spec> contains no unescaped '{', it is interpreted as filename       "
"       expansion pattern.  Else <spec> is brace expanded as in the bash shell"
"       and each of the resulting words is interpreted as filename expansion  "
"       pattern.  Filename expansion patterns are defined as in IEEE Std      "
"       1003.1-2004 (XCU: 2.13.3).                                            ",
"       As mandated by IEEE std 1003.1-2004 these pattern matching based file "
"       name expansion operations occur for each path separator separated part"
"       of <spec> independently.  To circumvent this, the operator '**' as    "
"       last characters of such a <spec> part matches a sequence of zero or   "
"       more pairs each consisting of a file system element name component and"
"       a path separator.                                                     ",
"       The bracket expressions used in the regular expressions and in the    "
"       filename expansion patterns conform to IEEE Std 1003.1-2004 (XBD:     "
"       9.3.5) with the exception that equivalence classes [=x=] of collating "
"       elements are not supported.                                           ",
"       On windows systems the escape character for extended regular          "
"       expressions, filename expansion patterns and brace expansion is '%'   "
"       instead of the usual '\\' and as path separators both '/' and '\\' are"
"       allowed.  On other systems only '/' is allowed as path separator and  "
"       both '\\' and '%' may be used as escape characters.  So for           "
"       portability reasons it is recommended on all systems to use '/' as    "
"       path separator and '%' as escape character.                           ",
"                                                                             ",
"EXAMPLE                                                                      ",
"       Let the environment variable 'STMDEVLROOT' be set to the value        "
"       'D:\\Development\\SysToMath\\wcopies', 'PACKAGE' to the value         "
"       'LibStmC', and 'CLEARCASEVOB' to the value 'Z:\\wbit_cr'.             ",
"       Moreover, let the ctupdate argument file 'ctupdate.arg' in the package"
"       directory have the following content:                                 ",
"       --args-env=StmPackage=@{PACKAGE}                                      ",
"       --args-env=StmPkgRoot=@{STMDEVLROOT}/@{StmPackage} # Package directory",
"       --protocol=@{StmPkgRoot}/ctupdate.txt                                 ",
"       --make=checked-in                                                     ",
"       --update=checked-in                                                   ",
"       --erase=all                                                           ",
"       --comment-from=@{StmPkgRoot}/release.txt                              ",
```

```
"       --no-dest-root-label                                                ",
"       --source-root=@{STMDEVLROOT}                                       ",
"       --exclude=.svn/                                                    ",
"       --exclude=\"(DebugMt|DebugMtDll|DebugMtStaticRt)/\"                ",
"       --exclude=\"(ReleaseMt|ReleaseMtDll|ReleaseMtStaticRt)/\"          ",
"       --exclude-file=\"*.{suo,ncb,aps,user,log,bak}\"                    ",
"       --exclude-file=\"/@{StmPackage}/etc/revision*\"                    ",
"       --exclude-file=/@{StmPackage}/release.txt                          ",
"       --exclude-file=/@{StmPackage}/ctupdate.txt                         ",
"       --source=@{StmPackage}                                             ",
"       --dest-root=@{CLEARCASEVOB}/tools/utilities/packets                ",
"       Then with the package directory as the current working directory, the "
"       command                                                            ",
"       'ctupdate --label=LibStmC-1.7.328 --args-from=ctupdate.args'       ",
"       will update and label the package 'LibStmC' in the ClearCase VOB. The "
"       according protocol will be written to the file 'ctupdate.txt'.     "
}
```

Man page of the ClearCase Tool Suite command ctupdate.

Definition at line 84 of file ctupdate.cpp.

7.3.3.4 const StmCmdOptSpec CtUpdate::optSpec [static, private, inherited]

Initial value:

```
{
    {StmId (h), "h\0help\0", StmCmdOptSpecOnlyOption},
    {StmId (V), "V\0version\0", StmCmdOptSpecOnlyOption},
    {StmId (v), "v\0verbose\0"},
    {StmId (t), "t\0tell-actions\0"},
    {StmId (P), "P\0protocol\0", StmCmdOptSpecRequiresArg},
    {StmId (E), "E\0args-env\0", StmCmdOptSpecArgsEnvOption | StmCmdOptSpecRequiresArg},
    {StmId (A), "A\0args-from\0", StmCmdOptSpecArgsFromOption | StmCmdOptSpecRequiresArg},
    {StmId (c), "c\0comment\0", StmCmdOptSpecRequiresArg},
    {StmId (C), "C\0comment-from\0", StmCmdOptSpecRequiresArg},
    {StmId (n), "n\0no-comment\0"},
    {StmId (l), "l\0label\0", StmCmdOptSpecRequiresArg},
    {StmId (L), "L\0replace-label\0", StmCmdOptSpecRequiresArg},
    {StmId (N), "N\0no-dest-root-label\0"},
    {StmId (m), "m\0make\0", StmCmdOptSpecRequiresArg},
    {StmId (u), "u\0update\0", StmCmdOptSpecRequiresArg},
    {StmId (e), "e\0erase\0", StmCmdOptSpecRequiresArg},
    {StmId (f), "f\0flat\0"},
    {StmId (r), "r\0recursive\0"},
    {StmId (S), "S\0source-root\0", StmCmdOptSpecRequiresArg},
    {StmId (x), "x\0exclude-file\0", StmCmdOptSpecRequiresArg},
    {StmId (X), "X\0exclude\0", StmCmdOptSpecRequiresArg},
    {StmId (i), "i\0include-file\0", StmCmdOptSpecRequiresArg},
    {StmId (I), "I\0include\0", StmCmdOptSpecRequiresArg},
    {StmId (d), "d\0reset-exceptions\0"},
    {StmId (s), "s\0source\0", StmCmdOptSpecRequiresArg},
    {StmId (D), "D\0dest-root\0", StmCmdOptSpecRequiresArg},
    {0, NULL}
}
```

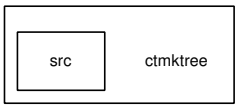
Definition of the command line option parameters.

Definition at line 548 of file ctupdate.cpp.

Referenced by CtUpdate::operator()().

8 ClearCase Tool Suite Implementation Directory Documentation

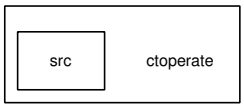
8.1 ctmktree/ Directory Reference



Directories

- directory src

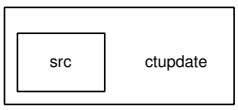
8.2 ctoperate/ Directory Reference



Directories

- directory src

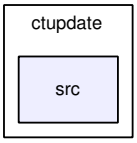
8.3 ctupdate/ Directory Reference



Directories

- directory src

8.4 ctupdate/src/ Directory Reference

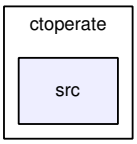


Files

- file ctupdate.cpp

  Main program of the command line based ctupdate application.

8.5 ctoperate/src/ Directory Reference

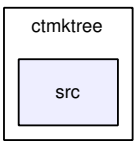


Files

- file ctoperate.cpp

  Main program of the command line based ctoperate application.

8.6 ctmktree/src/ Directory Reference



Files

- file ctmktree.cpp

  Main program of the command line based ctmktree application.

9 ClearCase Tool Suite Implementation Class Documentation

9.1 ArgList Class Reference

9.1.1 Detailed Description

Definition at line 329 of file ctoperate.cpp.

Public Member Functions

- ArgList & operator+= (const std::string &arg)
- void exec (const std::string &command, const Mode &mode)
- bool active () const

Static Private Attributes

- static size_t limit = 2048

9.1.2 Member Function Documentation

9.1.2.1 ArgList& ArgList::operator+= (const std::string & *arg*) [inline]

Definition at line 332 of file ctoperate.cpp.

References limit.

9.1.2.2 void ArgList::exec (const std::string & *command*, const Mode & *mode*) [inline]

Definition at line 340 of file ctoperate.cpp.

References execCommand().

9.1.2.3 bool ArgList::active () const [inline]

Definition at line 349 of file ctoperate.cpp.

The documentation for this class was generated from the following file:

- ctoperate.cpp

9.2 CtData Struct Reference

9.2.1 Detailed Description

An instance of POD type CtData is assigned to each directory to be handled.

Definition at line 250 of file ctmktree.cpp.

Public Member Functions

- CtData (const std::string &path, const std::string &cicommopt, bool wordy)

  Constructs the initial CtData object which designates the root's parent data.

- ~CtData ()
- void init (const char *directory)

  Initializes a copy of a parent CtData object to designate its son directory.

- void deinit ()

Public Attributes

- bool verbose

  True for additional diagnostic messages on stdout.

- const char * cicommentopt

  ClearCase check in comment option.

- const char * parentpath

  If not NULL, the path of the parent directory.

- bool parentVobElement

  True, if parent is a VOB element.

- bool parentOriginallyCheckedOut

  True, if parent is a VOB element and was originally checked out.

- bool parentCheckedOut

  True, while parent is checked out.

- const char * dirpath

  Path of directory.

- bool vobElement

  True, if dirpath is a VOB element.

- bool originallyCheckedOut

  True, if dirpath is a VOB element and was originally checked out.

- bool checkedOut

  True, while the dirpath is checked out.

- std::string * pElements

  if not NULL, pointer to a string containing the paths of the file elements to be created by the ClearCase command mkelem.

- std::string * pWin32Execs

  if not NULL, pointer to a string containing the paths of the file elements to be made executable by the ClearCase command protect.

9.2.2 Constructor & Destructor Documentation

9.2.2.1 CtData::CtData (const std::string & *path*, const std::string & *cicommopt*, bool *wordy*) [inline]

Constructs the initial CtData object which designates the root's parent data.

Definition at line 254 of file ctmktree.cpp.

References init().

9.2.2.2 CtData::~CtData () [inline]

Definition at line 272 of file ctmktree.cpp.

References deinit().

9.2.3 Member Function Documentation

9.2.3.1 void CtData::init (const char * *directory*) [inline]

Initializes a copy of a parent CtData object to designate its son directory.

Definition at line 279 of file ctmktree.cpp.

References checkedOut, dirpath, isCheckedOut(), isVobElement(), originallyCheckedOut, parentCheckedOut, parentOriginallyCheckedOut, parentpath, parentVobElement, pElements, pWin32Execs, and vobElement.

Referenced by CtData(), and elemhandler().

9.2.3.2 void CtData::deinit () [inline]

Definition at line 294 of file ctmktree.cpp.

References pElements, and pWin32Execs.

Referenced by elemhandler(), and ~CtData().

9.2.4 Member Data Documentation

9.2.4.1 bool CtData::verbose

True for additional diagnostic messages on stdout.

Definition at line 309 of file ctmktree.cpp.

Referenced by elemhandler().

9.2.4.2 const char* CtData::cicommentopt

ClearCase check in comment option.

Definition at line 311 of file ctmktree.cpp.

Referenced by elemhandler().

9.2.4.3 const char* CtData::parentpath

If not NULL, the path of the parent directory.

Definition at line 312 of file ctmktree.cpp.

Referenced by elemhandler(), and init().

9.2.4.4 bool CtData::parentVobElement

True, if parent is a VOB element.

Definition at line 314 of file ctmktree.cpp.

Referenced by elemhandler(), and init().

9.2.4.5 bool CtData::parentOriginallyCheckedOut

True, if parent is a VOB element and was originally checked out.

Definition at line 315 of file ctmktree.cpp.

Referenced by init().

9.2.4.6 bool CtData::parentCheckedOut

True, while parent is checked out.

Definition at line 317 of file ctmktree.cpp.

Referenced by elemhandler(), and init().

9.2.4.7 const char* CtData::dirpath

Path of directory.

Definition at line 318 of file ctmktree.cpp.

Referenced by init().

9.2.4.8 bool CtData::vobElement

True, if dirpath is a VOB element.

Definition at line 319 of file ctmktree.cpp.

Referenced by elemhandler(), and init().

9.2.4.9 bool CtData::originallyCheckedOut

True, if dirpath is a VOB element and was originally checked out.

Definition at line 320 of file ctmktree.cpp.

Referenced by elemhandler(), and init().

9.2.4.10 bool CtData::checkedOut

True, while the dirpath is checked out.

Definition at line 322 of file ctmktree.cpp.

Referenced by elemhandler(), and init().

9.2.4.11 std::string* CtData::pElements

if not NULL, pointer to a string containing the paths of the file elements to be created by the ClearCase command mkelem.

Definition at line 324 of file ctmktree.cpp.

Referenced by deinit(), elemhandler(), and init().

9.2.4.12 std::string* CtData::pWin32Execs

if not NULL, pointer to a string containing the paths of the file elements to be made executable by the ClearCase command protect.

Definition at line 328 of file ctmktree.cpp.

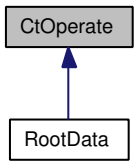
Referenced by deinit(), elemhandler(), and init().

The documentation for this struct was generated from the following file:

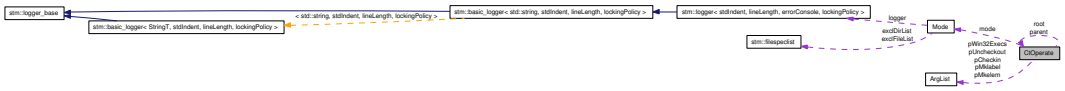
- ctmktree.cpp

9.3 CtOperate Struct Reference

Inheritance diagram for CtOperate:



Collaboration diagram for CtOperate:



9.3.1 Detailed Description

An instance of POD type CtOperate is assigned to each directory to be handled.

Definition at line 362 of file ctoperate.cpp.

Public Member Functions

- CtOperate (const std::string &destPath, const Mode &md)

  Constructs the initial CtOperate object which designates the root destination directory data.

- int init (const std::string &dirname, CtOperate *pParentData=NULL)

  Initializes a CtOperate object for a directory named dirname.

- int deinit ()

Public Attributes

- CtOperate * root

  If not NULL, the root destination directory data.

- CtOperate * parent

  If not NULL, the parent destination directory data.

- const Mode * mode

  Command mode parameters.

- const boost::filesystem::path * destdir

  If not NULL, pointer to the path of the destination directory.

- bool vobElement

  True, if destdir is a VOB element.

- bool checkedOut

  True, while the destdir is checked out.

- bool checkIn

  True, if destdir has been checked out and is to be checked in at the end.

- bool labelDir

  True, if destdir has has to labeled at the end.

- bool uncheckOut

  True, if destdir is to be unchecked out at the end.

- bool exclude

  True, if directory is not to be handled.

- ArgList * pUncheckout

  if not NULL, pointer to a list containing the paths of the file elements to be uncheckedout by the ClearCase command uncheckout.

- std::list< std::string > * pRemove

  if not NULL, pointer to a list of strings containing the paths of the files to be removed from the destination directory.

- ArgList * pMkelem

  if not NULL, pointer to a list containing the paths of the file elements to be created by the ClearCase command mkelem.

- ArgList * pWin32Execs

  if not NULL, pointer to a list containing the paths of the file elements to be made executable by the ClearCase command protect.

- ArgList * pCheckin

  if not NULL, pointer to a list containing the paths of the file elements to be checked in by the ClearCase command checkin.

- ArgList * pMklabel

  if not NULL, pointer to a list containing the paths of the file elements to be labeled by the ClearCase command mklabel.

9.3.2 Constructor & Destructor Documentation

9.3.2.1 CtOperate::CtOperate (const std::string & *destPath*, const Mode & *md*) [inline]

Constructs the initial CtOperate object which designates the root destination directory data.

Definition at line 366 of file ctoperate.cpp.

References init(), and root.

9.3.3 Member Function Documentation

9.3.3.1 int CtOperate::init (const std::string & *dirname*, CtOperate * *pParentData* = NULL)

Initializes a CtOperate object for a directory named dirname.

Definition at line 1181 of file ctoperate.cpp.

References checkedOut, checkIn, Mode::Operation::CheckIn, destdir, exclude, execCommand(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::Indent, isCheckedOut(), isVobElement(), labelDir, logger(), Mode::Operation::MkElem, mode, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, parent, pCheckin, pMkelem, pMklabel, pRemove, pUncheckout, pWin32Execs, Mode::Operation::Remove, root, Mode::Simulate, uncheckOut, and vobElement.

Referenced by CtOperate().

9.3.3.2 int CtOperate::deinit ()

Definition at line 1286 of file ctoperate.cpp.

References checkedOut, checkIn, destdir, exclude, execCommand(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::Indent, labelDir, logger(), mode, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, parent, pCheckin, pMkelem, pMklabel, pRemove, pUncheckout, pWin32Execs, and Mode::Simulate.

Referenced by RootData::~RootData().

9.3.4 Member Data Documentation

9.3.4.1 CtOperate* CtOperate::root

If not NULL, the root destination directory data.

Definition at line 379 of file ctoperate.cpp.

Referenced by CtOperate(), and init().

9.3.4.2 CtOperate* CtOperate::parent

If not NULL, the parent destination directory data.

Definition at line 381 of file ctoperate.cpp.

Referenced by deinit(), and init().

9.3.4.3 const Mode* CtOperate::mode

Command mode parameters.

Definition at line 383 of file ctoperate.cpp.

Referenced by deinit(), elemhandler(), and init().

9.3.4.4 const boost::filesystem::path* CtOperate::destdir

If not NULL, pointer to the path of the destination directory.

Definition at line 384 of file ctoperate.cpp.

Referenced by deinit(), elemhandler(), and init().

9.3.4.5 bool CtOperate::vobElement

True, if destdir is a VOB element.

Definition at line 387 of file ctoperate.cpp.

Referenced by init().

9.3.4.6 bool CtOperate::checkedOut

True, while the destdir is checked out.

Definition at line 389 of file ctoperate.cpp.

Referenced by deinit(), and init().

9.3.4.7 bool CtOperate::checkIn

True, if destdir has been checked out and is to be checked in at the end.

Definition at line 391 of file ctoperate.cpp.

Referenced by deinit(), and init().

9.3.4.8 bool CtOperate::labelDir

True, if destdir has has to labeled at the end.

Definition at line 394 of file ctoperate.cpp.

Referenced by deinit(), and init().

9.3.4.9 bool CtOperate::uncheckOut

True, if destdir is to be unchecked out at the end.

Definition at line 396 of file ctoperate.cpp.

Referenced by init().

9.3.4.10 bool CtOperate::exclude

True, if directory is not to be handled.

Definition at line 398 of file ctoperate.cpp.

Referenced by deinit(), and init().

9.3.4.11 ArgList* CtOperate::pUncheckout

if not NULL, pointer to a list containing the paths of the file elements to be uncheckedout by the ClearCase command uncheckout.

Definition at line 400 of file ctoperate.cpp.

Referenced by deinit(), elemhandler(), and init().

9.3.4.12 std::list<std::string>* CtOperate::pRemove

if not NULL, pointer to a list of strings containing the paths of the files to be removed from the destination directory.

Definition at line 406 of file ctoperate.cpp.

Referenced by deinit(), elemhandler(), and init().

9.3.4.13 ArgList* CtOperate::pMkelem

if not NULL, pointer to a list containing the paths of the file elements to be created by the ClearCase command mkelem.

Definition at line 411 of file ctoperate.cpp.

Referenced by deinit(), elemhandler(), and init().

9.3.4.14 ArgList* CtOperate::pWin32Execs

if not NULL, pointer to a list containing the paths of the file elements to be made executable by the ClearCase command protect.

Definition at line 416 of file ctoperate.cpp.

Referenced by deinit(), elemhandler(), and init().

9.3.4.15 ArgList* CtOperate::pCheckin

if not NULL, pointer to a list containing the paths of the file elements to be checked in by the ClearCase command checkin.

Definition at line 421 of file ctoperate.cpp.

Referenced by deinit(), elemhandler(), and init().

9.3.4.16 ArgList* CtOperate::pMklabel

if not NULL, pointer to a list containing the paths of the file elements to be labeled by the ClearCase command mklabel.

Definition at line 426 of file ctoperate.cpp.

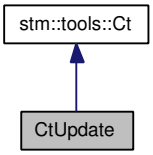
Referenced by deinit(), elemhandler(), and init().

The documentation for this struct was generated from the following file:

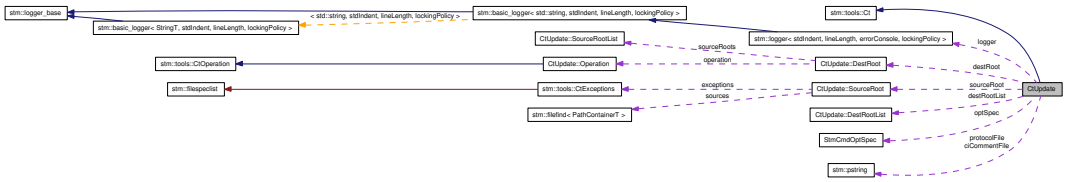
- ctoperate.cpp

9.4 CtUpdate Struct Reference

Inheritance diagram for CtUpdate:



Collaboration diagram for CtUpdate:



9.4.1 Detailed Description

Definition at line 515 of file ctupdate.cpp.

Public Types

- typedef **stm::logger** Logger

Public Member Functions

- CtUpdate (**Logger** &logger)
- ~CtUpdate ()
- void operator() (int argc, const char **argv)
- void evalCommentOption ()
- void evalLabelOption ()
- void evalModeOption ()
- void evalExceptionOption ()
- void defineDestRoot ()
- void openProtocol ()

Public Attributes

- **stm::pstring** ciCommentFile
- **stm::pstring** protocolFile
- boost::filesystem::ofstream protocol
- **StmCmdLine** cmdLine
- SourceRoot * sourceRoot
- DestRoot * destRoot
- DestRootList * destRootList
- **Logger** & logger
- const char * prog

Static Private Attributes

- static const **StmCmdOptSpec** optSpec [ ]

  Definition of the command line option parameters.

Classes

- struct DestRoot
- struct DestRootList
- struct Operation
- struct SourceRoot
- struct SourceRootList

9.4.2 Member Typedef Documentation

9.4.2.1 typedef stm::logger CtUpdate::Logger

Reimplemented from **stm::tools::Ct**.

Definition at line 517 of file ctupdate.cpp.

9.4.3 Constructor & Destructor Documentation

9.4.3.1 CtUpdate::CtUpdate (Logger & *logger*) `[explicit]`

Definition at line 1553 of file ctupdate.cpp.

9.4.3.2 CtUpdate::~CtUpdate ()

Definition at line 1564 of file ctupdate.cpp.

References cmdLine, destRoot, destRootList, stm::tools::Ct::logger(), sourceRoot, stmCmdLineDestroy(), and stm::basic_logger< StringT, stdIndent, lineLength, lockingPolicy >::unsetStream().

9.4.4 Member Function Documentation

9.4.4.1 void CtUpdate::operator() (int *argc*, const char ** *argv*)

Definition at line 1574 of file ctupdate.cpp.

References cmdLine, destRoot, destRootList, optSpec, prog, protocolFile, sourceRoot, stmCmdLineCreate(), stmCmdLineGetProg(), and StmDebugAidsErrmsg.

9.4.4.2 void CtUpdate::evalCommentOption ()

Definition at line 1592 of file ctupdate.cpp.

References ciCommentFile, cmdLine, stm::tools::Ct::getCiCommentOption(), stm::tools::Ct::setCiCommentOption(), stmCmdLineGetCurrentOptionArg(), stmCmdLineGetCurrentOptionId(), StmDebugAidsVerify, and StmId.

Referenced by main().

9.4.4.3 void CtUpdate::evalLabelOption ()

Definition at line 1646 of file ctupdate.cpp.

References cmdLine, destRoot, stmCmdLineGetCurrentOptionArg(), stmCmdLineGetCurrentOptionId(), StmDebugAidsVerify, and StmId.

Referenced by main().

9.4.4.4 void CtUpdate::evalModeOption ()

Definition at line 1673 of file ctupdate.cpp.

References CtUpdate::Operation::Erase::All, CtUpdate::Operation::Update::CheckedIn, CtUpdate::Operation::Make::CheckedIn, CtUpdate::Operation::Update::CheckedOut, CtUpdate::Operation::Make::CheckedOut, cmdLine, destRoot, CtUpdate::Operation::Erase::No, CtUpdate::Operation::Update::Revertible, stmCmdLineGetCurrentOptionArg(), stmCmdLineGetCurrentOptionId(), StmDebugAidsVerify, StmId, CtUpdate::Operation::Erase::ViewPrivate, and CtUpdate::Operation::Make::ViewPrivate.

Referenced by main().

9.4.4.5 void CtUpdate::evalExceptionOption ()

Definition at line 1758 of file ctupdate.cpp.

References stm::pstring::allowedSeparators, cmdLine, stm::tools::CtExceptions::Directory, stm::tools::CtExceptions::Exclude, stm::tools::CtExceptions::Include, stm::tools::CtExceptions::NoFlags, stm::tools::CtExceptions::PlainFile, sourceRoot, stmCmdLineGetCurrentOptionArg(), stmCmdLineGetCurrentOptionId(), StmDebugAidsVerify, and StmId.

Referenced by main().

9.4.4.6 void CtUpdate::defineDestRoot ()

Definition at line 1822 of file ctupdate.cpp.

References destRoot, destRootList, stm::tools::Ct::isVobElement(), and sourceRoot.

Referenced by main().

9.4.4.7 void CtUpdate::openProtocol ()

Definition at line 1864 of file ctupdate.cpp.

References stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::AutoInd, cmdLine, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::ColonTable, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::KeepWs, stm::tools::Ct::logger(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::mkPath(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, prog, protocol, protocolFile, stm::basic_logger< StringT, stdIndent, lineLength, lockingPolicy >::setStream(), stmCmdLineGetArg(), stmCmdLineGetArgCount(), stm::basic_logger< StringT, stdIndent, lineLength, lockingPolicy >::underline(), and Version.

Referenced by main().

9.4.5 Member Data Documentation

9.4.5.1 stm::pstring CtUpdate::ciCommentFile

Definition at line 537 of file ctupdate.cpp.

Referenced by evalCommentOption(), and main().

9.4.5.2 stm::pstring CtUpdate::protocolFile

Definition at line 538 of file ctupdate.cpp.

Referenced by main(), openProtocol(), and operator()().

9.4.5.3 boost::filesystem::ofstream CtUpdate::protocol

Definition at line 539 of file ctupdate.cpp.

Referenced by openProtocol().

9.4.5.4 StmCmdLine CtUpdate::cmdLine

Definition at line 540 of file ctupdate.cpp.

Referenced by evalCommentOption(), evalExceptionOption(), evalLabelOption(), evalModeOption(), main(), openProtocol(), operator()(), and ~CtUpdate().

9.4.5.5 SourceRoot* CtUpdate::sourceRoot

Definition at line 541 of file ctupdate.cpp.

Referenced by defineDestRoot(), evalExceptionOption(), main(), operator()(), and ~CtUpdate().

9.4.5.6 DestRoot* CtUpdate::destRoot

Definition at line 542 of file ctupdate.cpp.

Referenced by defineDestRoot(), evalLabelOption(), evalModeOption(), main(), operator()(), and ~CtUpdate().

9.4.5.7 DestRootList* CtUpdate::destRootList

Definition at line 543 of file ctupdate.cpp.

Referenced by defineDestRoot(), main(), operator()(), and ~CtUpdate().

9.4.5.8 Logger& CtUpdate::logger

Definition at line 544 of file ctupdate.cpp.

9.4.5.9 const char* CtUpdate::prog

Definition at line 545 of file ctupdate.cpp.

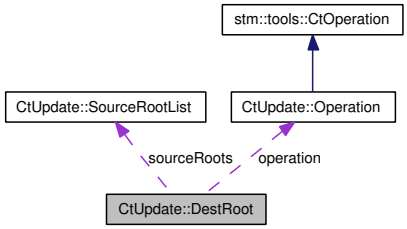
Referenced by main(), openProtocol(), and operator()().

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.5 CtUpdate::DestRoot Struct Reference

Collaboration diagram for CtUpdate::DestRoot:



9.5.1 Detailed Description

Definition at line 675 of file ctupdate.cpp.

Public Member Functions

- DestRoot (**stm::tools::Ct** &ct)
- ~DestRoot ()
- bool appendSourceRoot (CtUpdate::SourceRoot *sourceRoot, const std::string &newDirectory=std::string())

Public Attributes

- boost::filesystem::path path
- CtUpdate::SourceRootList * sourceRoots
- CtUpdate::Operation * operation

9.5.2 Constructor & Destructor Documentation

9.5.2.1 CtUpdate::DestRoot::DestRoot (stm::tools::Ct & *ct*) [inline]

Definition at line 677 of file ctupdate.cpp.

9.5.2.2 CtUpdate::DestRoot::~DestRoot () [inline]

Definition at line 682 of file ctupdate.cpp.

References operation, and sourceRoots.

9.5.3 Member Function Documentation

9.5.3.1 bool CtUpdate::DestRoot::appendSourceRoot (CtUpdate::SourceRoot * *sourceRoot*, const std::string & *newDirectory* = std::string ()) [inline]

Definition at line 688 of file ctupdate.cpp.

References CtUpdate::Operation::Update::CheckedIn, CtUpdate::Operation::Make::CheckedIn, operation, and sourceRoots.

9.5.4 Member Data Documentation

9.5.4.1 boost::filesystem::path CtUpdate::DestRoot::path

Definition at line 704 of file ctupdate.cpp.

9.5.4.2 CtUpdate::SourceRootList* CtUpdate::DestRoot::sourceRoots

Definition at line 705 of file ctupdate.cpp.

Referenced by appendSourceRoot(), and ~DestRoot().

9.5.4.3 CtUpdate::Operation* CtUpdate::DestRoot::operation

Definition at line 706 of file ctupdate.cpp.

Referenced by appendSourceRoot(), and ~DestRoot().

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.6 CtUpdate::DestRootList Struct Reference

9.6.1 Detailed Description

Definition at line 710 of file ctupdate.cpp.

Public Member Functions

- bool append (CtUpdate::DestRoot *destRoot, CtUpdate::SourceRoot *sourceRoot)

9.6.2 Member Function Documentation

9.6.2.1 bool CtUpdate::DestRootList::append (CtUpdate::DestRoot * *destRoot*, CtUpdate::SourceRoot * *sourceRoot*) [inline]

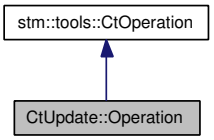
Definition at line 716 of file ctupdate.cpp.

The documentation for this struct was generated from the following file:

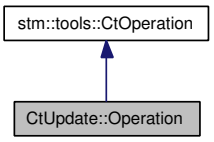
- ctupdate.cpp

9.7 CtUpdate::Operation Struct Reference

Inheritance diagram for CtUpdate::Operation:



Collaboration diagram for CtUpdate::Operation:



9.7.1 Detailed Description

Definition at line 552 of file ctupdate.cpp.

Public Member Functions

- Operation (**stm::tools::Ct** &**ct**)

Public Attributes

- int make
- int update
- int erase
- bool noDestRootLabel

Classes

- struct Erase
- struct Make
- struct Update

9.7.2 Constructor & Destructor Documentation

9.7.2.1 CtUpdate::Operation::Operation (stm::tools::Ct & *ct*) [inline]

Definition at line 584 of file ctupdate.cpp.

9.7.3 Member Data Documentation

9.7.3.1 int CtUpdate::Operation::make

Definition at line 592 of file ctupdate.cpp.

9.7.3.2 int CtUpdate::Operation::update

Definition at line 593 of file ctupdate.cpp.

9.7.3.3 int CtUpdate::Operation::erase

Definition at line 594 of file ctupdate.cpp.

9.7.3.4 bool CtUpdate::Operation::noDestRootLabel

Definition at line 595 of file ctupdate.cpp.

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.8 CtUpdate::Operation::Erase Struct Reference

9.8.1 Detailed Description

Definition at line 574 of file ctupdate.cpp.

Public Types

- enum {
  No,
  ViewPrivate,
  All }

9.8.2 Member Enumeration Documentation

9.8.2.1 anonymous enum

Enumerator:

No

ViewPrivate

All

Definition at line 576 of file ctupdate.cpp.

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.9 CtUpdate::Operation::Make Struct Reference

9.9.1 Detailed Description

Definition at line 554 of file ctupdate.cpp.

Public Types

- enum {
  ViewPrivate,
  CheckedIn,
  CheckedOut }

9.9.2 Member Enumeration Documentation

9.9.2.1 anonymous enum

Enumerator:

ViewPrivate
CheckedIn
CheckedOut

Definition at line 556 of file ctupdate.cpp.

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.10 CtUpdate::Operation::Update Struct Reference

9.10.1 Detailed Description

Definition at line 564 of file ctupdate.cpp.

Public Types

- enum {
  Revertible,
  CheckedIn,
  CheckedOut }

9.10.2 Member Enumeration Documentation

9.10.2.1 anonymous enum

Enumerator:

Revertible

CheckedIn

CheckedOut

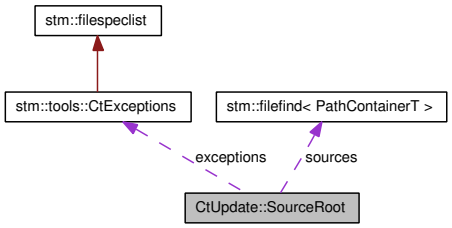
Definition at line 566 of file ctupdate.cpp.

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.11 CtUpdate::SourceRoot Struct Reference

Collaboration diagram for CtUpdate::SourceRoot:



9.11.1 Detailed Description

Definition at line 599 of file ctupdate.cpp.

Public Types

- typedef **stm::filefind**< std::set< boost::filesystem::path > > SourceFileFind

Public Member Functions

- SourceRoot ()
- ~SourceRoot ()
- int insertSources (const std::string &spec)

Public Attributes

- bool flat
- **stm::tools::CtExceptions** exceptions
- **SourceFileFind** * sources

9.11.2 Member Typedef Documentation

9.11.2.1 typedef stm::filefind<std::set<boost::filesystem::path> > CtUpdate::SourceRoot::SourceFileFind

Definition at line 601 of file ctupdate.cpp.

9.11.3 Constructor & Destructor Documentation

9.11.3.1 CtUpdate::SourceRoot::SourceRoot () [inline]

Definition at line 603 of file ctupdate.cpp.

9.11.3.2 CtUpdate::SourceRoot::~SourceRoot () [inline]

Definition at line 615 of file ctupdate.cpp.

References sources.

9.11.4 Member Function Documentation

9.11.4.1 int CtUpdate::SourceRoot::insertSources (const std::string & *spec*) [inline]

Definition at line 620 of file ctupdate.cpp.

References stm::filespec::beginsWithRootDirectory(), stm::filespec::beginsWithRootName(), stm::filefind< PathContainerT >::NoPlainFiles, stm::filespec::normalize(), sources, and stm::filefind< PathContainerT >::StoreRelative.

9.11.5 Member Data Documentation

9.11.5.1 bool CtUpdate::SourceRoot::flat

Definition at line 645 of file ctupdate.cpp.

9.11.5.2 stm::tools::CtExceptions CtUpdate::SourceRoot::exceptions

Definition at line 646 of file ctupdate.cpp.

9.11.5.3 SourceFileFind* CtUpdate::SourceRoot::sources

Definition at line 647 of file ctupdate.cpp.

Referenced by insertSources(), and ~SourceRoot().

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.12 CtUpdate::SourceRootList Struct Reference

9.12.1 Detailed Description

Definition at line 651 of file ctupdate.cpp.

Public Member Functions

- bool append (CtUpdate::SourceRoot *sourceRoot, const std::string &newDirectory=std::string())

9.12.2 Member Function Documentation

9.12.2.1 bool CtUpdate::SourceRootList::append (CtUpdate::SourceRoot * *sourceRoot*, const std::string & *newDirectory* = `std::string ()`) [inline]

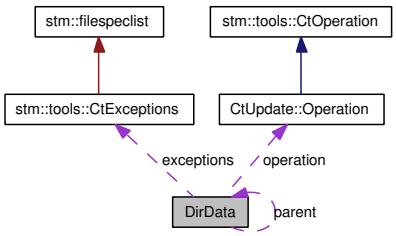
Definition at line 657 of file ctupdate.cpp.

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.13 DirData Struct Reference

Collaboration diagram for DirData:



9.13.1 Detailed Description

A DirData object is assigned to each directory to be handled.

Definition at line 735 of file ctupdate.cpp.

Public Types

- typedef **CtUpdate::Logger** Logger

Public Member Functions

- DirData (const CtUpdate::DestRoot *destRoot, const CtUpdate::SourceRoot *sourceRoot=NULL, const boost::filesystem::path &relPath=boost::filesystem::path())

  Constructs the initial DirData object which designates the root directory data indicated by parent == NULL.

- ~DirData ()
- int init (const std::string &dirleaf, DirData *pParentData=NULL)

  Initializes a DirData object for a directory named dirname.

- int deinit ()

Static Public Member Functions

- static int handler (const char *dirname, const char *entryname, const struct stat *direntry, int status, void *data, size_t len)

  The static method handler serves as callback function of stmRecurse and stmTraverse.

Public Attributes

- DirData * parent

  If not NULL, the parent destination directory data.

- const CtUpdate::Operation * operation

  ClearTool Command parameters.

- const **stm::tools::CtExceptions** * exceptions

  If not NULL, pointer to the current exception list.

- const boost::filesystem::path * reldir

  Pointer to the currently handled directory relative to the current destination root.

- const boost::filesystem::path * sourcedir

  If not NULL, pointer to the path of the source directory.

- const boost::filesystem::path * destdir

  If not NULL, pointer to the path of the destination directory.

- bool created

  True, if destdir has been created.

- bool vobElement

  True, if destdir is a VOB element.

- bool checkedOut

  True, while the destdir is checked out.

- bool checkIn

  True, if destdir has been checked out and is to be checked in at the end.

- bool labelDir

  True, if destdir has has to be labeled at the end.

- bool exclude

  True, if directory is not to be handled.

- **CtUpdate::ArgList** * pCheckout

  if not NULL, pointer to a list containing the paths of the file elements to be checked out by the ClearCase command checkout.

- std::list< boost::filesystem::path > * pCopy

  if not NULL, pointer to a list of paths of the files to be copied to the destination directory.

- **CtUpdate::ArgList** * pMkelem

  if not NULL, pointer to a list containing the paths of the file elements to be created by the ClearCase command mkelem.

- **CtUpdate::ArgList** * pWin32Execs

if not NULL, pointer to a list containing the paths of the file elements to be made executable by the ClearCase command protect.

- **CtUpdate::ArgList** * pCheckin

  if not NULL, pointer to a list containing the paths of the file elements to be checked in by the ClearCase command checkin.

- **CtUpdate::ArgList** * pMklabel

  if not NULL, pointer to a list containing the paths of the file elements to be labeled by the ClearCase command mklabel.

9.13.2 Member Typedef Documentation

9.13.2.1 typedef CtUpdate::Logger DirData::Logger

Definition at line 737 of file ctupdate.cpp.

9.13.3 Constructor & Destructor Documentation

9.13.3.1 DirData::DirData (const CtUpdate::DestRoot * *destRoot*, const CtUpdate::SourceRoot * *sourceRoot* = `NULL`, const boost::filesystem::path & *relPath* = `boost::filesystem::path ()`)

Constructs the initial DirData object which designates the root directory data indicated by parent == NULL.

Definition at line 1928 of file ctupdate.cpp.

References destdir, init(), stm::tools::Ct::isVobElement(), operation, reldir, and sourcedir.

9.13.3.2 DirData::~DirData ()

Definition at line 1971 of file ctupdate.cpp.

References deinit().

9.13.4 Member Function Documentation

9.13.4.1 int DirData::init (const std::string & *dirleaf*, DirData * *pParentData* = `NULL`)

Initializes a DirData object for a directory named dirname.

Definition at line 1977 of file ctupdate.cpp.

References stm::tools::Ct::ArgList::active(), CtUpdate::Operation::Erase::All, CtUpdate::Operation::Update::CheckedIn, CtUpdate::Operation::Make::CheckedIn, checkedOut, checkIn, created, destdir, exceptions, exclude, stm::tools::Ct::ArgList::exec(), stm::tools::Ct::execCommand(), stm::tools::Ct::getCiCommentOption(), stm::tools::Ct::getMode(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::Indent, stm::tools::Ct::isCheckedOut(), stm::tools::Ct::isVobElement(), labelDir, stm::tools::Ct::logger(), CtUpdate::Operation::Erase::No, stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, operation, parent, pCheckin, pCheckout, pCopy, pMkelem, pMklabel, pWin32Execs, reldir, sourcedir, StmDebugAidsVerify, CtUpdate::Operation::Make::ViewPrivate, and vobElement.

Referenced by DirData(), and handler().

9.13.4.2 int DirData::deinit ()

Definition at line 2148 of file ctupdate.cpp.

References CtUpdate::Operation::Update::CheckedIn, CtUpdate::Operation::Make::CheckedIn, checkedOut, checkIn, created, destdir, stm::tools::Ct::execCommand(), stm::tools::Ct::getCiCommentOption(), stm::tools::Ct::getMode(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::Indent, labelDir, stm::tools::Ct::logger(), stm::basic_logger< std::string, stdIndent, lineLength, lockingPolicy >::NoPara, operation, parent, pCheckin, pCheckout, pCopy, pMkelem, pMklabel, pWin32Execs, reldir, sourcedir, and StmDebugAidsVerify.

Referenced by handler(), and ~DirData().

9.13.4.3 int DirData::handler (const char * *dirname*, const char * *entryname*, const struct stat * *direntry*, int *status*, void * *data*, size_t *len*) [static]

The static method handler serves as callback function of stmRecurse and stmTraverse.

Definition at line 2272 of file ctupdate.cpp.

References CtUpdate::Operation::Make::CheckedIn, CtUpdate::Operation::Update::CheckedIn, checkIn, deinit(), destdir, exceptions, exclude, init(), stm::tools::Ct::isCheckedOut(), stm::tools::Ct::isVobElement(), stm::isWin32Executable(), operation, pCheckin, pCheckout, pCopy, pMkelem, pMklabel, pWin32Execs, reldir, CtUpdate::Operation::Update::Revertible, StmDebugAidsVerify, StmRecursDbeg, StmRecursDend, StmRecursLeaf, and CtUpdate::Operation::Make::ViewPrivate.

Referenced by main().

9.13.5 Member Data Documentation

9.13.5.1 DirData* DirData::parent

If not NULL, the parent destination directory data.

Definition at line 758 of file ctupdate.cpp.

Referenced by deinit(), and init().

9.13.5.2 const CtUpdate::Operation* DirData::operation

ClearTool Command parameters.

Definition at line 760 of file ctupdate.cpp.

Referenced by deinit(), DirData(), handler(), and init().

9.13.5.3 const stm::tools::CtExceptions* DirData::exceptions

If not NULL, pointer to the current exception list.

Definition at line 761 of file ctupdate.cpp.

Referenced by handler(), and init().

9.13.5.4 const boost::filesystem::path* DirData::reldir

Pointer to the currently handled directory relative to the current destination root.

Definition at line 763 of file ctupdate.cpp.

Referenced by deinit(), DirData(), handler(), and init().

9.13.5.5 const boost::filesystem::path* DirData::sourcedir

If not NULL, pointer to the path of the source directory.

Definition at line 767 of file ctupdate.cpp.

Referenced by deinit(), DirData(), and init().

9.13.5.6 const boost::filesystem::path* DirData::destdir

If not NULL, pointer to the path of the destination directory.

Definition at line 770 of file ctupdate.cpp.

Referenced by deinit(), DirData(), handler(), and init().

9.13.5.7 bool DirData::created

True, if destdir has been created.

Definition at line 773 of file ctupdate.cpp.

Referenced by deinit(), and init().

9.13.5.8 bool DirData::vobElement

True, if destdir is a VOB element.

Definition at line 775 of file ctupdate.cpp.

Referenced by init().

9.13.5.9 bool DirData::checkedOut

True, while the destdir is checked out.

Definition at line 777 of file ctupdate.cpp.

Referenced by deinit(), and init().

9.13.5.10 bool DirData::checkIn

True, if destdir has been checked out and is to be checked in at the end.

Definition at line 779 of file ctupdate.cpp.

Referenced by deinit(), handler(), and init().

9.13.5.11 bool DirData::labelDir

True, if destdir has has to be labeled at the end.

Definition at line 782 of file ctupdate.cpp.

Referenced by deinit(), and init().

9.13.5.12 bool DirData::exclude

True, if directory is not to be handled.

Definition at line 784 of file ctupdate.cpp.

Referenced by handler(), and init().

9.13.5.13 CtUpdate::ArgList* DirData::pCheckout

if not NULL, pointer to a list containing the paths of the file elements to be checked out by the ClearCase command checkout.

Definition at line 786 of file ctupdate.cpp.

Referenced by deinit(), handler(), and init().

9.13.5.14 std::list<boost::filesystem::path>* DirData::pCopy

if not NULL, pointer to a list of paths of the files to be copied to the destination directory.

Definition at line 791 of file ctupdate.cpp.

Referenced by deinit(), handler(), and init().

9.13.5.15 CtUpdate::ArgList* DirData::pMkelem

if not NULL, pointer to a list containing the paths of the file elements to be created by the ClearCase command mkelem.

Definition at line 795 of file ctupdate.cpp.

Referenced by deinit(), handler(), and init().

9.13.5.16 CtUpdate::ArgList* DirData::pWin32Execs

if not NULL, pointer to a list containing the paths of the file elements to be made executable by the ClearCase command protect.

Definition at line 800 of file ctupdate.cpp.

Referenced by deinit(), handler(), and init().

9.13.5.17 CtUpdate::ArgList* DirData::pCheckin

if not NULL, pointer to a list containing the paths of the file elements to be checked in by the ClearCase command checkin.

Definition at line 805 of file ctupdate.cpp.

Referenced by deinit(), handler(), and init().

9.13.5.18 CtUpdate::ArgList* DirData::pMklabel

if not NULL, pointer to a list containing the paths of the file elements to be labeled by the ClearCase command mklabel.

Definition at line 810 of file ctupdate.cpp.

Referenced by deinit(), handler(), and init().

The documentation for this struct was generated from the following file:

- ctupdate.cpp

9.14 Mode Struct Reference

Collaboration diagram for Mode:

9.14.1 Detailed Description

Definition at line 250 of file ctoperate.cpp.

Public Types

- enum {
  Normal,
  Verbose,
  Simulate }

Public Member Functions

- Mode (**Logger** &log)

Public Attributes

- std::string label
- std::string cicommentopt
- **stm::filespeclist** exclFileList
- **stm::filespeclist** exclDirList
- **Logger** & logger
- int action
- int operation

Classes

- struct Operation

9.14.2 Member Enumeration Documentation

9.14.2.1 anonymous enum

Enumerator:

Normal

Verbose

Simulate

Definition at line 252 of file ctoperate.cpp.

9.14.3 Constructor & Destructor Documentation

9.14.3.1 Mode::Mode (Logger & *log*) [inline]

Definition at line 270 of file ctoperate.cpp.

9.14.4 Member Data Documentation

9.14.4.1 std::string Mode::label

Definition at line 276 of file ctoperate.cpp.

Referenced by main().

9.14.4.2 std::string Mode::cicommentopt

Definition at line 277 of file ctoperate.cpp.

Referenced by main().

9.14.4.3 stm::filespeclist Mode::exclFileList

Definition at line 278 of file ctoperate.cpp.

Referenced by main().

9.14.4.4 stm::filespeclist Mode::exclDirList

Definition at line 279 of file ctoperate.cpp.

Referenced by main().

9.14.4.5 Logger& Mode::logger

Definition at line 280 of file ctoperate.cpp.

Referenced by execCommand().

9.14.4.6 int Mode::action

Definition at line 281 of file ctoperate.cpp.

Referenced by execCommand(), and main().

9.14.4.7 int Mode::operation

Definition at line 282 of file ctoperate.cpp.

Referenced by main().

The documentation for this struct was generated from the following file:

- ctoperate.cpp

9.15 Mode::Operation Struct Reference

9.15.1 Detailed Description

Definition at line 259 of file ctoperate.cpp.

Public Types

- enum {
  None,
  MkElem,
  CheckIn,
  Remove }

9.15.2 Member Enumeration Documentation

9.15.2.1 anonymous enum

Enumerator:

None
MkElem
CheckIn
Remove

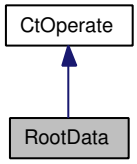
Definition at line 261 of file ctoperate.cpp.

The documentation for this struct was generated from the following file:

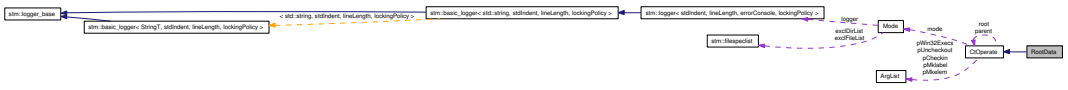
- ctoperate.cpp

9.16 RootData Struct Reference

Inheritance diagram for RootData:



Collaboration diagram for RootData:



9.16.1 Detailed Description

An instance of RootData is assigned to the destination root directory.

Definition at line 436 of file ctoperate.cpp.

Public Member Functions

- RootData (const std::string &destPath, const Mode &md)

  Constructs the initial CtOperate object which designates the root destination directory data.

- ~RootData ()

9.16.2 Constructor & Destructor Documentation

9.16.2.1 RootData::RootData (const std::string & *destPath*, const Mode & *md*) [inline]

Constructs the initial CtOperate object which designates the root destination directory data.

Definition at line 440 of file ctoperate.cpp.

9.16.2.2 RootData::~RootData () [inline]

Definition at line 444 of file ctoperate.cpp.

References CtOperate::deinit().

The documentation for this struct was generated from the following file:

- ctoperate.cpp

10 ClearCase Tool Suite Implementation File Documentation

10.1 ctmktree.cpp File Reference

10.1.1 Detailed Description

Main program of the command line based ctmktree application.

Version:

1.17-r350

Date:

2008-01-03 21:41:50 (Tom)

Author:

Tom Michaelis
SysToMath
Wittelsbacherstr. 7
D-80469 Munich

Contact:

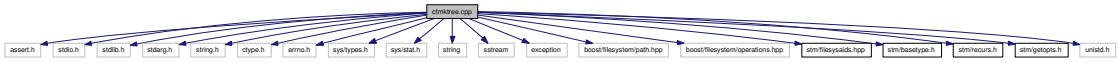
http://www.SysToMath.com
mailto:Tom.Michaelis@SysToMath.com

This C++ program file contains the definition of the man page and of the function **main()** implementing the command line interface of the ctmktree application which makes or augments a ClearCase tree.

Definition in file ctmktree.cpp.

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <ctype.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <string>
#include <sstream>
#include <exception>
#include <boost/filesystem/path.hpp>
#include <boost/filesystem/operations.hpp>
#include <stm/filesysaids.hpp>
#include <stm/basetype.h>
#include <stm/recurs.h>
#include <stm/getopts.h>
#include <unistd.h>
```

Include dependency graph for ctmktree.cpp:



Classes

- struct CtData

  An instance of POD type CtData is assigned to each directory to be handled.

Defines

- #define PATHLEN 1024
- #define PATHSEP '/'

Functions

- static bool isVobElement (const std::string &path, bool isDirectory=false)

  The local function isVobElement returns true, if the file (default) or directory designated by path is a VOB element, else false.

- static bool isCheckedOut (const std::string &path, bool isDirectory=false)

  The local function isCheckedOut returns true, if the file (default) or directory designated by path is checked out, else false.

- static bool isWin32Executable (const std::string &path)

  The local function isWin32Executable returns true, if path designates a WIN32 executable file (case unsensitively the extension '.exe' or '.com').

- static int execCommand (const std::string &command, bool verbose=false)

  The local function execCommand returns 0, if the console command was executed successfully.

- static int elemhandler (const char *dirname, const char *entryname, const struct stat *direntry, int status, void *data, size_t len)

  The local function elemhandler serves as callback function of recurs.

- static void error (const char *msg,...)

  Print error message and terminate the program.

- int main (int argc, char *argv[ ])

  Main function implementing the command.

Variables

- const char * CopyRight = "(C) 2004-2008 Tom Michaelis, SysToMath"
- const char * Version = "1.17-r350"
- static const char * man [ ]

  Man page of the ClearCase Tool Suite command ctmktree.

10.2 ctoperate.cpp File Reference

10.2.1 Detailed Description

Main program of the command line based ctoperate application.

Version:

1.17-r350

Date:

2008-01-03 21:41:50 (Tom)

Author:

Tom Michaelis
SysToMath
Wittelsbacherstr. 7
D-80469 Munich

Contact:

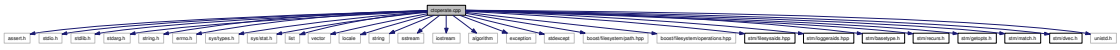
http://www.SysToMath.com
mailto:Tom.Michaelis@SysToMath.com

This C++ program file contains the definition of the man page and of the function **main()** implementing the command line interface of the ctoperate application which operates on a ClearCase VOB.

Definition in file ctoperate.cpp.

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <list>
#include <vector>
#include <locale>
#include <string>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <exception>
#include <stdexcept>
#include <boost/filesystem/path.hpp>
#include <boost/filesystem/operations.hpp>
#include <stm/filesysaids.hpp>
#include <stm/loggeraids.hpp>
#include <stm/basetype.h>
#include <stm/recurs.h>
#include <stm/getopts.h>
#include <stm/match.h>
#include <stm/dvec.h>
#include <unistd.h>
```

Include dependency graph for ctoperate.cpp:



Classes

- struct Mode
- struct Mode::Operation
- class ArgList
- struct CtOperate

  An instance of POD type CtOperate is assigned to each directory to be handled.

- struct RootData

  An instance of RootData is assigned to the destination root directory.

Defines

- #define PATHLEN 1024
- #define PATHSEP '/'

Typedefs

- typedef **stm::logger** Logger

Functions

- static bool isVobElement (const boost::filesystem::path &path)

  The local function isVobElement returns true, if path is a VOB element, else false.

- static bool isCheckedOut (const boost::filesystem::path &path)

  The local function isCheckedOut returns true, if is checked out, else false.

- static bool isLabel (const std::string &vobSelector, const std::string &label)

  The local function isLabel returns true, if label is defined as a label in the ClearCase VOB selected by vobSelector, else false.

- static bool isWin32Executable (const boost::filesystem::path &path)

  The local function isWin32Executable returns true, if path designates a WIN32 executable file (case unsensitively the extension '.exe' or '.com').

- static void execCommand (const std::string &command, const Mode &mode)

  The local function execCommand executes the console command.

- static int elemhandler (const char *dirname, const char *entryname, const struct stat *direntry, int status, void *data, size_t len)

  The local function elemhandler serves as callback function of recurs.

- int main (int argc, char *argv[ ])

  Main function implementing the command.

Variables

- const char * CopyRight = "(C) 2005-2008 Tom Michaelis, SysToMath"
- const char * Version = "1.17-r350"
- static const char * man [ ]

  Man page of the ClearCase Tool Suite command ctoperate.

10.3 ctupdate.cpp File Reference

10.3.1 Detailed Description

Main program of the command line based ctupdate application.

Version:

1.17-r350

Date:

2008-01-03 21:41:50 (Tom)

Author:

Tom Michaelis
SysToMath
Wittelsbacherstr. 7
D-80469 Munich

Contact:

http://www.SysToMath.com
mailto:Tom.Michaelis@SysToMath.com

This C++ program file contains the definition of the man page and of the function **main()** implementing the command line interface of the ctupdate application which integrates missing or newer files into a ClearCase VOB.

Definition in file ctupdate.cpp.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <list>
```

```
#include <vector>
#include <locale>
#include <string>
#include <sstream>
#include <iostream>
#include <algorithm>
#include <exception>
#include <stdexcept>
#include <boost/filesystem/path.hpp>
#include <boost/filesystem/fstream.hpp>
#include <boost/filesystem/operations.hpp>
#include <stm/filesysaids.hpp>
#include <stm/loggeraids.hpp>
#include <stm/debugaids.hpp>
#include <stm/divaids.hpp>
#include <stm/toolaids/ctaids.hpp>
#include <stm/basetype.h>
#include <stm/recurs.h>
#include <stm/getopts.h>
#include <stm/match.h>
```

Include dependency graph for ctupdate.cpp:

Classes

- struct CtUpdate
- struct CtUpdate::Operation
- struct CtUpdate::Operation::Make
- struct CtUpdate::Operation::Update
- struct CtUpdate::Operation::Erase
- struct CtUpdate::SourceRoot
- struct CtUpdate::SourceRootList
- struct CtUpdate::DestRoot
- struct CtUpdate::DestRootList
- struct DirData

  A DirData object is assigned to each directory to be handled.

Functions

- int main (int argc, char *argv[ ])

Main function implementing the command.

Variables

- const char * CopyRight = "(C) 2005-2008 Tom Michaelis, SysToMath"
- const char * Version = "1.17-r350"
- static const char * man [ ]

  Man page of the ClearCase Tool Suite command ctupdate.

11 ClearCase Tool Suite Implementation Page Documentation

11.1 ClearCase Tool Suite

The ClearCase Tool Suite consists of the following console programs providing various utilities facilitating some IBM Rational ClearCase related tasks:

- ctmktree Main Program
- ctoperate Main Program
- ctupdate Main Program

Index