

E_Lib Quick Reference Card

Use in C++

Use	Meaning
Eerr<<info<<"OK"<<eom;	Prints to error message stream: <my_prog> Information: OK
Eerr<<warning<<"A warning"<<eom;	Prints to error message stream: <my_prog> Warning: A warning
Eerr<<error<<"An error"<<eom;	Prints to error message stream: <my_prog> Error: An error
Eerr<<internal<<"My fault"<<eom;	Prints to error message stream: <my_prog> Internal error: My fault
Eerr<<fatal<<"No chance"<<eom;	Prints to error message stream: <my_prog> Fatal error: No chance
Eerr.SetFrom("<my_prog> ");	Sets the source of the messages to "<my_prog> "
void func() { ErrSource_c err_src("func - "); ... }	Sets the source of the messages to "func - " until destructor of err_src is called
ErrReport_c::SetFileErrReport("myerrors.txt");	Messages will go to file myerrors.txt
ErrReport_c::SetStreamErrReport(cout);	Messages will go to standard output
WindowErrReport_c* wEerr = ...; ErrReport_c::SetErrReport(wEerr);	Messages will go to standard message box (on MS Windows only)
ErrReport_c::UnsetErrReport();	Rolls back to previous error message stream
void func() { ... if(...smth...) throw(Exception_c ("I can't ")); ... }	Class Exception_c should be a base for any exceptions to be thrown
throw EXIT_EXC();	Throws ExitException_c - to exit from program (not to use exit() , etc.)
E_ASSERT(smth);	Generates an internal error message and throws AssertException_c - to use instead of assert()
try{ my_func(); } E_CATCH_ALL;	To catch all exceptions and to handle them by E_Lib default rules

Integration with M_Lib

Formats in M_Lib dictionary file	
<pre># E_Lib prefixes - must be present # (the prefix contents are for example) ELIB_INFO_TYPE "I<\$1> (info): " ELIB_WARNING_TYPE "W<\$1> (warning): " ELIB_ERROR_TYPE "E<\$1> (error): " ELIB_FATAL_TYPE "E<\$1> (fatal error): " ELIB_INTERNAL_TYPE "E<\$1> (internal error): "</pre> <pre># Sources specified by M_Lib dictionary ID MY_PROGRAM_SRC "my_program "</pre> <pre># Not required - just for the example below COMMON_FL_OPEN_FAIL "Can not open file [<\$1>]"</pre>	
Use in C++	Meaning
<pre>Eerr.SetFrom("MY_PROGRAM_SRC"); Eerr << error << MSG1(COMMON_FL_OPEN_FAIL, char*, fname) << eom;</pre>	Prints to error message stream: my_program E7 (error): Can not open file [aaa.txt]

C++ Compiler

Pre-processor define	Meaning
E_NO_MLIB_USE	To use E_Lib independently, with no integration with M_Lib
_UNICODE	To build and use E_Lib for UNICODE messages (default is ASCII messages)
CPP_WITH_STD_EXCEPTION	To use C++ standard exception class exception (Exception_c will be inherited from it)