

# CAPE-OPEN

Delivering the power of component software  
and open standard interfaces  
in Computer-Aided Process Engineering

---

## CAPE-OPEN Logging and Testing Tool

### Open Source Development

---



[www.colan.org](http://www.colan.org)

---

## ARCHIVAL INFORMATION

---

Filename	COLTT Development.doc
Authors	Michael Halloran for CO-LaN
Status	For approval
Date	June 2010
Version	version 1.0
Number of pages	
Versioning	0: First Draft for review
	1:Added COLTTViewer and TRAC
Additional material	
Web location	
Implementation specifications version	
Comments	

---

## IMPORTANT NOTICES

---

### **Disclaimer of Warranty**

CO-LaN documents and publications include software in the form of *sample code*. Any such software described or provided by CO-LaN --- in whatever form --- is provided "as-is" without warranty of any kind. CO-LaN and its partners and suppliers disclaim any warranties including without limitation an implied warrant or fitness for a particular purpose. The entire risk arising out of the use or performance of any sample code --- or any other software described by the CAPE-OPEN Laboratories Network --- remains with you.

Copyright © 2010 CO-LaN and/or suppliers. All rights are reserved unless specifically stated otherwise. CO-LaN is a non for profit organization established under French law of 1901.

### **Trademark Usage**

Many of the designations used by manufacturers and seller to distinguish their products are claimed as trademarks. Where those designations appear in CO-LaN publications, and the authors are aware of a trademark claim, the designations have been printed in caps or initial caps.

Microsoft, Microsoft Word, Visual Basic, Visual Basic for Applications, Internet Explorer, Windows and Windows NT are registered trademarks and ActiveX is a trademark of Microsoft Corporation.

Netscape Navigator is a registered trademark of Netscape Corporation.

Adobe Acrobat is a registered trademark of Adobe Corporation.

# Table of Contents

1.	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	References	5
1.4	Glossary	5
1.5	Overview	7
2.	Pre-requisites	8
2.1	Install Visual Studio 2008	8
2.2	Install the Subversion (SVN) client software.	8
2.3	Install AnkhSVN	8
2.4	Install the CAPE-OPEN Type Libraries	9
2.5	Download the COLTT Source code	9
3.	Working with COLTT source code	10
3.1	Building the COLTT projects	10
3.2	Registering the COLoggers DLLs	11
3.3	Testing that your build of COLTT is working	11
4.	Development Support	13

## 1. Introduction

### 1.1 Purpose

CO-LaN has decided that COLTT software should be managed as an open source development project. This document is intended to help developers understand the development environment required to work on the COLTT project.

### 1.2 Scope

CAPE-OPEN is a set of standards that define interfaces and protocols to allow the integration of process modeling software components from diverse vendors. Achieving interoperability between CAPE-OPEN compliant software may not been easy for a variety of reasons: there is no reference implementation for the standards so they are open to interpretation in a number of areas; vendors can misunderstand the specifications or make errors and are most often not able to test their implementations with other vendor's implementations; and, integration is often performed by industrial users who do not have the tools, or knowledge, necessary to track down problems.

COLTT is a tool that can be used to log the interaction between a process modeling components that comply with the CAPE-OPEN standards.

### 1.3 References

CAPE-OPEN standards documentation can be found here

<http://www.co-lan.org/index-3.html>

### 1.4 Glossary

Term	Definition	Stakeholder
CO-LaN	CAPE-OPEN Laboratories Network – the organization that administers the CAPE-OPEN standard	Operating companies, software vendors, academic institutions, government agencies, individuals
COLTT	CAPE-OPEN Logging and Testing Tool	CO-LaN
DLL	Dynamic Link Library – a file containing executable code that can be shared between programs	
PMC	Process Modeling Component – a software component, typically implemented as a DLL that implements CAPE-OPEN interfaces.	Software vendors
PME	Process Modeling Environment – a program within which PMCs can be used	Software vendors



## 1.5 Overview

This document assumes familiarity with:

- standard software development processes;
- the use of source code management systems to manage changes to source code files;
- the use of Microsoft Visual Studio for compiling and linking MS Windows executables;
- the use of Microsoft Visual Studio for debugging executables;
- the use of development tools such as regsvr32, "regedit" and "depends"
- Microsoft COM component development using ATL
- CAPE-OPEN component development

Section 2 describes the pre-requisites needed to develop the COLTT components. Section 3 describes the process for developing the components using Microsoft Visual Studio and the Subversion source code management system.

## 2. Pre-requisites

### 2.1 Install Visual Studio 2008

You need Microsoft Visual Studio 2008 Standard Edition to build the COLTT source code. The Visual Studio Express version cannot be used since COLTT relies on the ATL libraries which are not included in the Express distribution.

Follow the installation instructions provided.

### 2.2 Install the Subversion (SVN) client software.

Subversion (SVN) is used to manage the COLTT source code files which held in the SourceForge repository.

To download the Subversion client software (you don't need the server because we're using the server running at Sourceforge) you will need to register at [www.collab.net](http://www.collab.net). Go there and select the Join Now link to register. Then go to this page:

<http://www.open.collab.net/downloads/subversion/>

And use the Download link for this version:

CollabNet Subversion Command-Line Client v1.6.2 (for Windows)

Install this software on your development machine.

This is a command line client for Subversion (SVN). To test it you can open the Command Line Window for Visual Studio 2008, which you will find in the Visual Studio Tools Folder under Microsoft Visual Studio 2008 in your start menu.

In this window you should be able to execute commands such as:

```
svn help
```

to get basic help on all svn commands.

You will also get a CollabNet Subversion Client entry under All Programs in the Start Menu where you will find a PDF copy of the SVN book which will give you an introduction to how svn is used.

### 2.3 Install AnkhSVN

AnkhSVN is Source Code Control Provider for Visual Studio which provides access to a Subversion repository from within the Visual Studio environment. Working on COLTT source code does not require that AnKHSVN is installed since the SVN client can also be

used for any operation on the COLTT repository. However, using AnKHSVN is recommended because it makes access to the repository much more convenient; in particular it makes it very easy to see which files have been changed locally compared to the repository.

Download the latest stable build of AnkhSVN from

<http://ankhsvn.open.collab.net/>

There's a link to version 2.1.7141 on that page which will download it. We need to use the 2.1 version for compatibility with the 1.6 version of SubVersion that we are using.

Before installing it, make sure you don't have any older versions of AnkhSVN on your machine; if you do uninstall them.

Install AnkhSVN according to the instructions provided.

## 2.4 Install the CAPE-OPEN Type Libraries

Download the CAPE-OPEN 1.1 Type library from <http://www.colan.org/index-38.html>

Unzip the files and install CAPE-OPEN.msi.

## 2.5 Download the COLTT Source code

To get the source code:

Open a Visual Studio 2008 Command Prompt window

Cd to (or mkdir if necessary and then cd to) the directory under which COLTT source code should be placed

To get the source code from SourceForge execute the following command:

```
svn co https://cape-openloggin.svn.sourceforge.net/svnroot/cape-openloggin/COLTT  
COLTT
```

This will create a subdirectory called COLTT and place all the source necessary to build and link COLTT.

The source tree will look like this:

COLTT

    CAPE-OPEN - *contains fixed include files describing the CAPE-OPEN  
    interfaces; these should not be modified*

    1-1-0

COLoggersST - *contains source code to build the COLTT logging component*  
debug  
release

COLoggingController - *contains source code to build COLogginController.exe, which is used to configure the COLTT logging components*

debug  
release

COLTTViewer – *contains source code to build COLTTViewer.exe, which is used to view COLTT logfiles*

debug  
release

log4cxx-0.9.7 - *contains the headfiles and .lib files necessary to link in the Open Source Log4cxx component which is used to generate the COLTT log files*

include  
msvc

SVN will also create .svn directories within this tree, and the build process will create others.

### **3. Working with COLTT source code**

#### **3.1 Building the COLTT projects**

1. Start Visual Studio and Open the Solution for building COLoggingController.exe:

COLTT/COLoggingController/COLoggingController.sln

Note that COLoggingController must be built before any of the COLoggers DLLs the first time the build is done.

2. Within Visual Studio select the “Debug” configuration and build the project.

This should produce COLoggingController.exe in COLTT/COLoggingController/debug.

3. Select the Release configuration and build again to produce COLTT/COLoggingController/release/CoLoggingController.exe

4. Start Visual Studio and Open the Solution for building COLTTViewer.exe:

COLTT/COLTTViewer/COLTTViewer.sln

5. Within Visual Studio select the “Debug” configuration and build the project.

This should produce COLTTViewer.exe in COLTT/COLTTViewer /debug.

6. Select the Release configuration and build again to produce COLTT/COLTTViewer/release/COLTTViewer.exe

7. Open COLTT/COLoggersST/COLoggersST.sln and build the following four configurations:

"COLoggers Debug" produces COLTT/COLoggersST/debug/COLoggers.dll - the multithreaded debug version of the logging component

"COLoggersST Debug" produces COLTT/COLoggersST/debug/COLoggersST.dll - the single-threaded debug version of the logging component

"COLoggers Release" produces COLTT/COLoggersST/Release/COLoggers.dll - the multithreaded release build version of the logging component

"COLoggersST Release" produces COLTT/COLoggersST/Release/COLoggersST.dll - the single-threaded release build version of the logging component

Depending on privileges, these builds may report the following error:

```
c:\COLTT\COLoggersST>regsvr32 /s /c
"c:\COLTT\COLoggersST\Debug\COLoggersST.dll"
c:\COLTT\COLoggersST>if errorlevel 1 goto VCReportError
c:\COLTT\COLoggersST>echo Project : error PRJ0019: A tool returned an error code
from "Performing registration"
Project : error PRJ0019: A tool returned an error code from "Performing registration"
```

On the Microsoft Vista operating system this is caused if the user does not have the necessary privilege to register a DLL. To get the DLLs registered follow the instructions in the next section.

### 3.2 Registering the COLoggers DLLs

1. On the Start Menu navigate to the entry for "Microsoft Visual Studio 2008 Command Window"
2. Right click on it and select Run as Administrator - Click Continue if Windows asks to permission
3. In the command window, cd to COLTT\COLoggersST\debug and execute the commands

```
regsvr32 COLoggersST.dll
regsvr32 COLoggers.dll
```

If regsvr32 reports an error loading either DLL, or if it doesn't report that the DLL was successfully registered, you may have a dependency problem. In this case run the ndepends tool to load the DLL and check whether all references are resolved correctly.

4. If you want to register the release version execute the same two commands in the COLTT\COLoggersST\release directory

### 3.3 Testing that your build of COLTT is working

To test that your build of COLTT is working, you will need to be able to run a CAPE-OPEN

compliant simulator. In these instructions COFE ([www.cocosimulator.org](http://www.cocosimulator.org)) will be used as an example.

1. Open COLTT/COLoggersST/COLoggersST.sln
2. Open the file COLoggersST.cpp and go to line 217
3. Copy the following 3 lines:

```
sprintf(MsgBuffer, "COLTT Version is 1.08.5");  
ModuleLogger->info(MsgBuffer);  
GUIPtr->SendMessage2GUI(MsgBuffer, "");
```

4. Paste the copied lines below the originals and change them to:

```
sprintf(MsgBuffer, "Being run by %s", getenv("USERNAME"));  
ModuleLogger->info(MsgBuffer);  
GUIPtr->SendMessage2GUI(MsgBuffer, "");
```

5. Build the COLoggers Debug and COLoggersST Debug configurations.
6. If the builds report errors registering the DLLs but you have already registered your debug builds using the Visual Studio command window as described above you do not need to register the DLLs again. If they are not registered then follow the instructions given in section 3.2
7. Run COLTT/COLoggingController/Debug/COLoggingController.exe and configure the COFE Mixer unit to be logged. Substitute a different Unit Operation for Mixer if you are using a different simulator or have your own COFE example.
  - a. In the Search box, open the “All System” node
  - b. Open “Unit Operations”
  - c. Select “Mixer”
  - d. If the Unit Operation already has logging enabled, disable it by unchecking the “Enable Logging” checkbox – this clears the reference to any previous version of the COLTT logger.
  - e. Check the “Enable Logging” checkbox to switch logging on
8. Using MS Visual Studio open COLTT/COLoggersST/debug/CAPE-OPENLogs.ini
9. Edit the line shown below to define the directory where you want the COLTT log written to.

```
log4j.appender.A1.File=C:\\Users\\Michael\\Documents\\
```

(Note that in the build configuration the Log Files tab in the COLoggingController application does not correctly update the log file location so the location is set manually by updating the .ini file. This only needs to be done once; each log will subsequently be written to the specified directory with a different name. If you want to change the directory edit the .ini file again.)

10. Run COFE (or the simulator you are using for the test)
11. Load a file that uses the logged Unit Operation
12. Run the simulation.
13. In the Real Time Logging tab in the COLoggingController window scroll to the top and check that the modified output is displayed.
14. Using MS Visual Studio open the log file for the run which will be in the directory defined in the .ini file. It will be named using the pattern <simulator name>\_<mmddyy>\_<hhmmss>.log and check that the modified output is shown at the top of the file.

## 4. Development Support

For help with any development issues, search the CAPE-OPEN project forums on SourceForge at:

<https://sourceforge.net/projects/cape-openlogin/forums/index/page/1>

These forums are monitored by the Project Admin group who will respond to any issues.

If you want to become a member of the Project contact a member of the Project Admin group via SourceForge, here:

[https://sourceforge.net/project/memberlist.php?group\\_id=244852](https://sourceforge.net/project/memberlist.php?group_id=244852)

If you want to work with reported defects for COLTT, they are held in the TRAC system hosted on SourceForge at:

<https://sourceforge.net/apps/trac/cape-openlogin/>

Use the “View Tickets” menu item and then select the “Active Tickets” report to see current outstanding defects.