

차례

1. 개요
2. 관련 툴
3. 분석 절차
 1. 분석 대상 모듈(exe) 빌드
 2. Instrumentation
 3. 모니터링 실행
 4. 커버리지 확인하기
4. 개선 아이디어
 1. 배치파일을 이용한 작업 단순화
 2. 도구 개선 아이디어

개요

2008 팀 시스템 에디션에는 vsinstr 이란 커맨드라인 도구가 포함되어 C++ 네이티브 모듈도 분석이 가능해졌다.



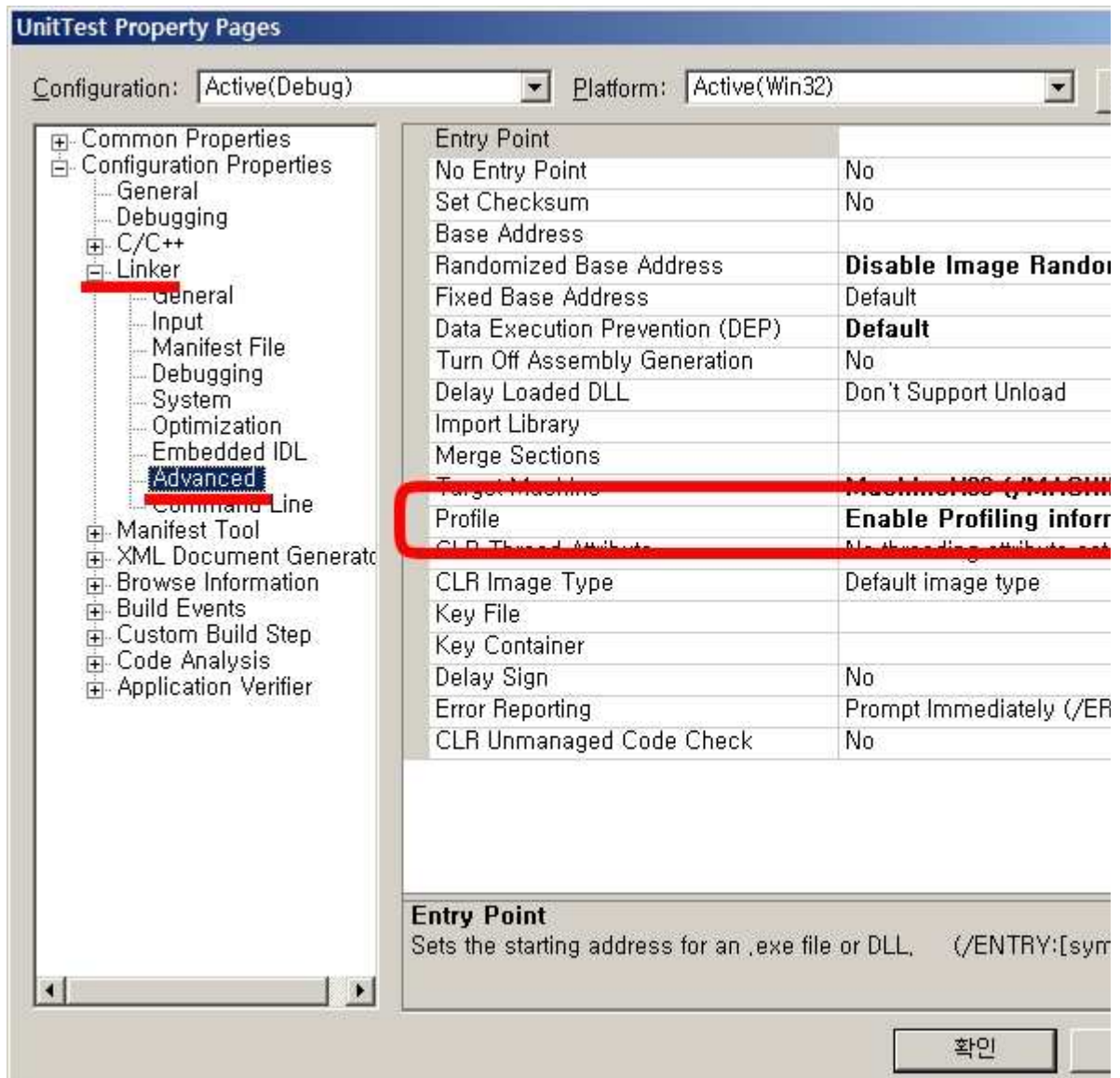
관련 툴

- VSInstr.exe (위치: <VS>\Team Tools\Performance Tools) - 실행 분석을 위한 instrumentation. 실행파일(exe, dll)을 입력받아 instrumented version을 만든다.
- vsperfcmd.exe = 실행 분석 결과를 모니터링하여 로그를 생성한다.

분석 절차

분석 대상 모듈(exe) 빌드

커버리지 분석을 위해서는 Linker 옵션에서 "/PROFILE" 이 활성화 되어야만 한다. Unit Test 프로젝트의 설정에서 이 옵션을 켜도록 한다. 빌드하면 보통때처럼 .exe가 나온다. (대신 /INCREMENTAL 빌드옵션이 무효화된다)



Instrumentation

vsinstr.exe 가 path에 포함되어 있는지 확인하고 커맨드라인에서 vsinstr 을 실행하여 .exe 파일의 instrumented version을 만든다.

```
prompt>vsinstr /coverage test_module.exe
Microsoft (R) VSInstr Post-Link Instrumentation 9.0.21022 x86
Copyright (C) Microsoft Corp. All rights reserved.

File to Process:
  D:\...\test_module.exe --> D:\...\test_module.exe
Original file backed up to D:\...\test_module.exe.orig

Successfully instrumented file D:\...\test_module.exe.

prompt>
```

모니터링 실행

vsperfcmd 를 이용하여 모니터링 기능을 켜놓고 instrument된 실행파일.exe 를 실행시켜서 실행로그를 얻는다.

```
>vsperfcmd /start:coverage /output:test_module
Microsoft (R) VSPerf Command Version 9.0.21022 x86
Copyright (C) Microsoft Corp. All rights reserved.

>test_module.exe
.....
OK (27 tests, 27 ran, 40 checks, 0 ignored, 0 filtered out, 0
ms)

>vsperfcmd /shutdown
Microsoft (R) VSPerf Command Version 9.0.21022 x86
Copyright (C) Microsoft Corp. All rights reserved.

Shutting down the Profile Monitor
-----
>
```

커버리지 확인하기

위 과정을 마치면 test_module.coverage 라는 파일이 생긴다. 이 파일을 visual c++로 열면 coverage 를 확인할 수 있다.

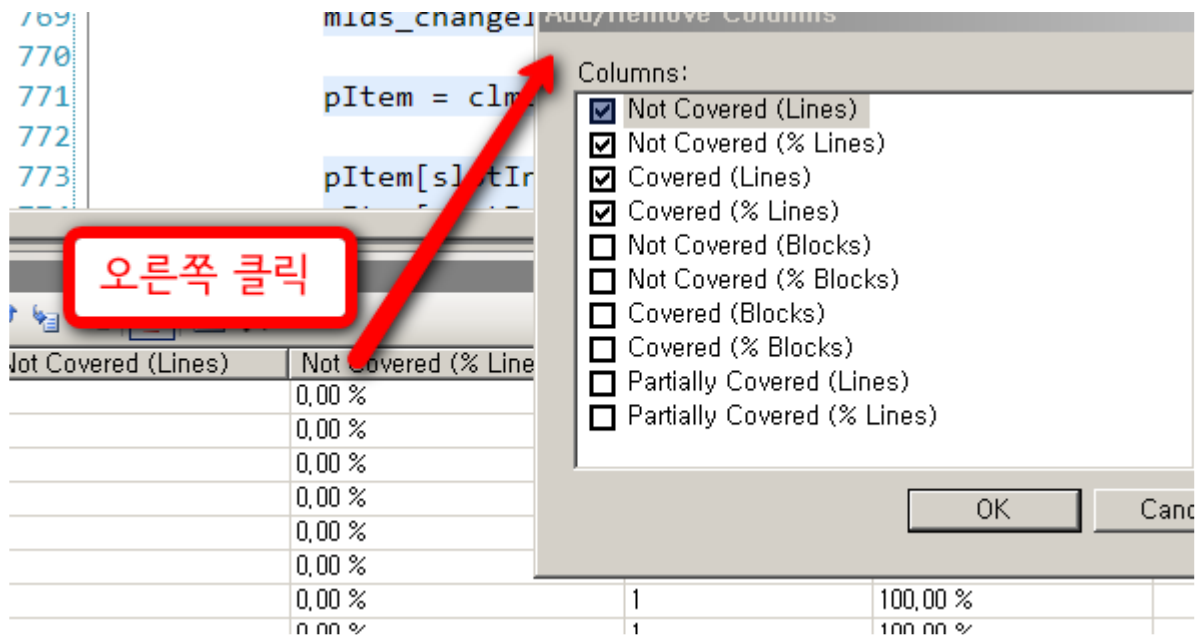
The screenshot shows the Microsoft Visual Studio IDE with the following components:

- Solution Explorer:** Displays the project structure for 'MLDSAgent_IFX', including folders for Header Files, Resource Files, Source Files, and Test, and source files like Database.cpp, Database.h, FileSystem.cpp, FileSystem.h, main.cpp, and MldsContactTest.cpp.
- Code Editor:** Shows the source code for 'MLDSvCard.c'. The code includes declarations for 'bufferStreet', 'i', and 'pBuffer', along with a function that prints debug messages and returns 0.
- Code Coverage Results:** A window titled 'test_module.coverage' showing a table of coverage data for various functions.

Hierarchy	Not Covered (...)	Not Covered (...)	Covered...	Cover...
MLDSGetMsgData	3	100,00 %	0	0,00 %
MLDSGroup2VGroup	43	100,00 %	0	0,00 %
MLDSInitMsgData	3	100,00 %	0	0,00 %
MLDSMemo2VNote	5	13,16 %	33	86,84 %
MLDSPB2VCard	272	61,26 %	172	38,74 %
MLDSParseVBookmark	25	100,00 %	0	0,00 %
MLDSParseVCalendar	27	100,00 %	0	0,00 %
MLDSParseVCard	2	9,52 %	19	90,48 %

The status bar at the bottom indicates 'Ready', 'Ln 1696', and 'Col 20'.

기본 설정은 block 단위 coverage를 보여주는데, 이를 line coverage로 바꾸려면 coverage화면에서 column을 추가/삭제하면 된다.



개선 아이디어

배치파일을 이용한 작업 단순화

- cov.bat 파일 (여러 단계의 coverage 분석 단계를 batch 파일로 처리한다.)

```
set path=<path to vs>\Team Tools\Performance Tools;%PATH%
vsinstr @vsinstr_options.rsp
vsperfcmd /start:coverage /output:<output file name>
<instrumented exe file>
vsperfcmd /shutdown
```

- vsinstr 에서 분석하지 않을 함수명은 아래처럼 /exclude 로 제외시킬 수 있다. 다음 예제는 CppUTest 에 관련된 함수들의 제외목록이다. 함수명을 찾으려면 vsinstr /dumpsfncs instrumented.exe
- vsinstr_options.rsp 파일 (vsinstr.exe 옵션을 별도 파일로 관리할 수 있다)

```
/coverage
/exclude:CommandLineTestRunner::*
/exclude:ConsoleTestOutput::*
/exclude:StringFrom
/exclude:EqualsFailure::*
/exclude:Failure::*
/exclude:JUnitTestOutput::*
/exclude:Utest::*
/exclude:TestResult::*
/exclude:TestRegistry::*
/exclude:TestPlugin::*
/exclude:TestOutput::*
/exclude:TestInstaller::*
/exclude:getParameterField
/exclude:createFileName
/exclude:SimpleString::*
/exclude:SetPointerPlugin::*
/exclude:NullTest::*
/exclude:NullTestPlugin::*
/exclude:MemoryLeakWarning::*
/exclude:MemoryLeakWarningPlugin::*
```

```
/exclude:HexStringFrom
/exclude:GetPlatformSpecificTimeInMillis
/exclude:NtCurrentJob
/exclude:NtCurrentTeb
/exclude:PadStringsToSameLength
/exclude:PlatformSpecificExitCurrentTestImpl
/exclude:PlatformSpecificSprintf
/exclude:GetPlatformSpecificTimeString
/exclude:TimeInMillisImplementation
/exclude:TimeStringImplementation
/exclude:`anonymous namespace'::*
/exclude:`dynamic initializer'*
/exclude:operator*
/exclude:main
/outputpath:<output path>
<output exe name>
```

도구 개선 아이디어

- visual studio plugin 혹은 standalone utility 를 만들어서 복잡한 과정을 단순화 할 수 있다.
- 실행파일을 로드하면 vsinstr 로 instrumentation 하여 분석모니터링 실행하여 결과를 얻는다.
- 결과를 export 하면 xml 로 출력되는데, 이때 각 함수 심볼마다 파일위치 및 커버리지데이터(라인)등이 표시된다. 이를 읽어들여 /exclude 목록을 자동생성하여 다음번 다시 vsinstr 실행하는 식으로 진행할 수 있을 것이다.

CodeCoverageUsingVisualCpp (2009-11-23 04:17:12에 한주영가(이) 마지막으로 수정)