



NAgile開発 NCoverによるコードカバレッジ分析

NAgile.com
<http://www.nagile.com/>

©2006 NAgile.com All rights reserved.

変更履歴

- 2006/12/27
 - ver. 1.0公開

©2006 NAgile.com All rights reserved.

2



Agenda

- コードカバレッジとは？
- NCover
- NCoverインストール
- NCoverの利用
- 分析結果表示
- NUnitとの連携
- NCoverExplorer
- NCoverExplorerインストール
- ビルドエンジンとの連携
- NCoverExplorer.Extras

コードカバレッジとは？

- 分析対象のコードがどの程度網羅(カバレッジ)されているか分析
- コードカバレッジのメリット
 - 使われていないコードの検出
 - コードカバレッジ分析ツールとアプリケーションを実際に行い、全く呼び出されていないコードを探し出し、コードから削除する
 - コードの可読性/メンテナンス性の向上
 - テストがどの程度の製品コードを網羅しているか分析
 - ソフトウェアの品質を表すひとつの目安となる
 - NUnitなどのテストフレームワークを利用し、テストコードを作成する必要がある

NCover

- C#やVB.NETなどで開発したソフトウェアのコードカバレッジ分析を行うフリーソフトウェア
 - 公式サイト
<http://ncover.org/site/>
 - 最新バージョン(2006年12月27日時点)
 - 1.5.5 Beta
- テスト駆動開発で作成したテストコード(NUnit)と組み合わせで利用することが多い
 - NCoverから任意の実行ファイルを読み出し、そのデバッグ情報ファイル(.pdb)からコードカバレッジを分析する仕組み
 - NCoverからWindowsフォームアプリケーションやWindowsサービスなどを呼び出し、UI操作などを実際に行うことでコードカバレッジを測定することも可能

NCoverインストール

- 下記URLから最新バージョンのZIPファイルをダウンロード
<http://ncover.org/SITE/files/default.aspx>
- ZIPファイルを解凍し、MSIインストーラを実行
 - プログラムメニューには登録されず、デフォルトで下記フォルダへインストールされる
 - C:\Program Files\NCover
 - 上記フォルダをパスに追加しておくとう便利
- NCoverExplorer
 - 下記URLから最新バージョンのZIPファイルをダウンロード
<http://www.kiwidude.com/blog/index.html>
 - ZIPファイルを解凍し、作成したフォルダを任意の場所へコピー

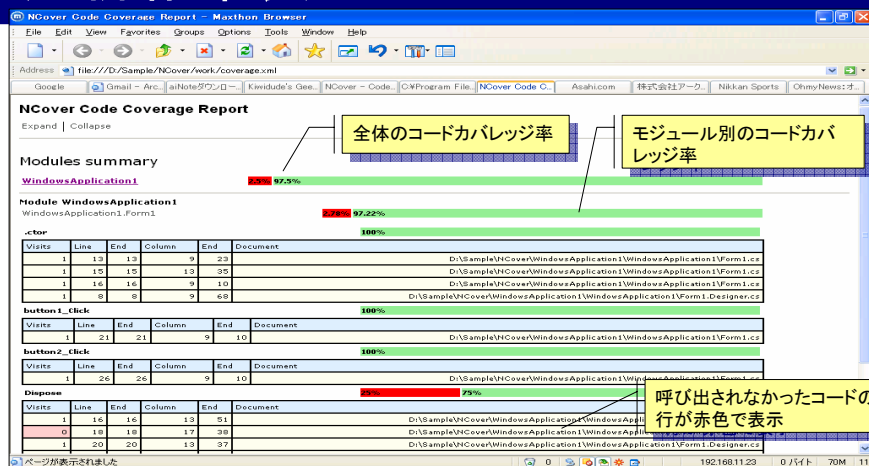
NCoverの利用

■ NCover.Console [コマンド] [引数] [オプション]

- [コマンド]
 - NCoverから呼び出す実行ファイル、テストコードのカバレッジを測定する場合はNUnitの実行ファイルを指定
- [引数]
 - コマンドで呼び出す実行ファイルへ渡す引数
- [オプション]
 - //a ... 分析に含めるアセンブリ、セミコロンで区切って複数指定可能
 - //w ... 分析を実行する際の作業フォルダ
 - //l ... ログファイルの保存先を指定、デフォルトは作業フォルダ上
 - //x ... 分析結果のXMLファイルの保存先を指定、デフォルトは作業フォルダ上
 - //ea ... コードカバレッジに含めないメソッドの属性を指定

※ NCoverから別の実行ファイルを呼び出す仕組みのため、[引数]は"/" (スラッシュ) で呼び出す実行ファイルの引数を指定し、NCover自体のオプションは"//" (ダブルスラッシュ) で指定する

分析結果表示



全体のコードカバレッジ率

モジュール別のコードカバレッジ率

呼び出されなかったコードの行が赤色で表示

- NCover付属のXSLを利用し、分析結果のXMLをHTMLとして表示可能
 - Coverage.xmlをcoverage.xmlと同じフォルダへコピー

NUnitとの連携

[コマンド]でnunit-console.exeを指定

```
>NCover.Console "C:\Program Files\NUnit-Net-2.0 2.2.9\bin\nunit-console.exe" "CalcLibrary.Test.dll"
```

[引数]でテストコードを含むアセンブリファイルを指定

- [コマンド]にNUnitコンソールアプリケーション(nunit-console.exe)を指定
 - デフォルトではNUnitはプログラムファイルズの下にインストールされるため、ファイルのパスを"(ダブルクォーテーション)で囲む
 - [引数]にNUnitのテストコードを含むアセンブリファイルを指定
 - NUnitのコマンドライン引数の仕様に従う

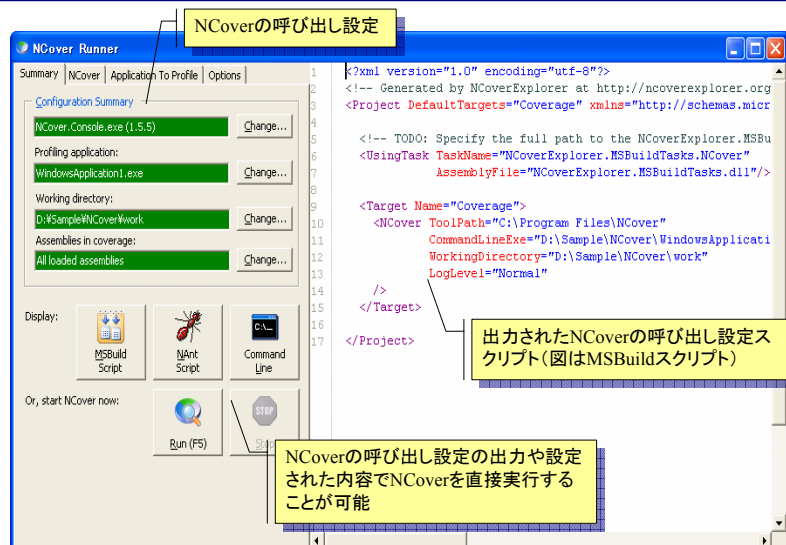
NCoverExplorer

- NCoverExplorer
 - NCoverの分析結果をよりわかりやすく表示したり、NCoverの実行設定を簡単に行うためのアドインツール
 - 公式サイト
<http://www.kiwidude.com/blog/index.html>
 - 最新バージョン(2006年12月27日時点)
 - 1.3.5 (NCover 1.5.xに対応)
- NCover Runner
 - NCoverExplorer上からNCoverを実行するUI
 - NCoverのパラメータ設定をUI上で行うことが可能
 - NCover Runner上で設定したNCoverの呼び出し設定をMSBuild/NAntのスク립トやコマンドラインパラメータとして出力可能
 - ただし、MSBuild/NAntのスク립トはNCoverExplorer.Extrasがインストールされている環境でないと動作しないので注意
 - 起動するには
 - NCoverExplorerのメニューから「File」→「Run NCover」を選択

NCoverExplorerインストール

- 下記URLから最新バージョンのZIPファイルをダウンロード
<http://www.kiwidude.com/blog/index.html>
- ZIPファイルを解凍し、作成したフォルダを任意の場所へコピー
 - コピー先の例
 - C:\Program Files\NCoverExplorer
 - NCoverExplorer.exeのショートカットをデスクトップに作成しておく
と便利

NCover Runner



The screenshot shows the NCover Runner application interface. On the left, there is a configuration panel with fields for 'NCover Console.exe (1.5.5)', 'Profiling application: WindowsApplication1.exe', 'Working directory: D:\Sample\NCover\work', and 'Assemblies in coverage: All loaded assemblies'. Below these are options for 'Display' (MSBuild Script, NAnt Script, Command Line) and a 'Run (F5)' button. On the right, an MSBuild script is displayed with the following content:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Generated by NCoverExplorer at http://ncoverexplorer.org
3 <Project DefaultTargets="Coverage" xmlns="http://schemas.micr
4
5 <!-- TODO: Specify the full path to the NCoverExplorer.MSBu
6 <UsingTask TaskName="NCoverExplorer.MSBuildTasks.NCover"
7     AssemblyFile="NCoverExplorer.MSBuildTasks.dll"/>
8
9 <Target Name="Coverage">
10 <NCover ToolPath="C:\Program Files\NCover"
11     CommandLineExe="D:\Sample\NCover\Windowsapplicati
12     WorkingDirectory="D:\Sample\NCover\work"
13     LogLevel="Normal"
14 </Target>
15
16 </Project>
```

Two callout boxes provide additional information:

- A yellow box at the top points to the title bar: "NCoverの呼び出し設定"
- A yellow box on the right points to the MSBuild script: "出力されたNCoverの呼び出し設定スクリプト (図はMSBuildスクリプト)"
- A yellow box at the bottom points to the 'Run' button: "NCoverの呼び出し設定の出力や設定された内容でNCoverを直接実行することが可能"

NCoverExplorerによる分析結果表示

コードカバレッジしたアセンブリをツリー表示

Method	Visit Count	Coverage %	Unvisited Points	Visited Points	Line
ctor	1	100%	0	4	8
button1_Click	1	100%	0	1	21
button2_Click	1	100%	0	1	26
Dispose	1	75%	1	3	16
InitializeComponent	1	100%	0	26	31

分析中に呼び出されたコード(緑)と呼び出されなかったコード(赤)をビジュアル表示

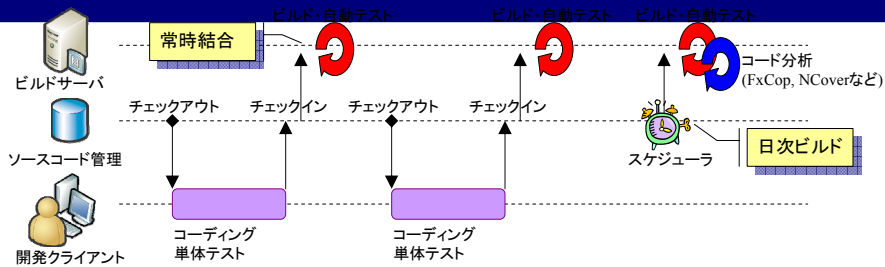
選択されたクラスやメソッドの分析結果詳細

```

10  // <summary>
11  // 使用中のリソースをすべてクリア
12  // </summary>
13  // <param name="disposing">マネージ リソースが破棄される場合 true、破棄されない場合
14  protected override void Dispose(bool disposing)
15  {
16  if (disposing && (components != null))
17  {
18  components.Dispose();
19  }
20  base.Dispose(disposing);
21
22  #region Windows フォーム デザイナーで生成されたコード
23
24  // <summary>
25  // デザイン サポートに必要なメソッドです。このメソッドの内容を
26  // コード エディタで変更しないでください。
27  // </summary>
28  private void InitializeComponent()
29  {
30
31  this.button1 = new System.Windows.Forms.Button();
32  this.button2 = new System.Windows.Forms.Button();
33  this.SuspendLayout();
34  //
35  // button1
36  //
37  }
38  }
39
40  WindowsApplication1.Form1
    
```

©2006 NAgle.com All rights reserved. 13

ビルドエンジンとの連携



- CruiseControl.NETとビルドエンジンを利用し、コードカバレッジ分析を実施
- 通常、分析処理は長い処理時間を必要とするため日次ビルドと組み合わせることを推奨
 - 自動テスト
 - 常時結合で、ソースコードがチェックインされるたびにNUnitのテストコードを実行
 - 分析系ツール(NCover, FxCopなど)
 - 日次ビルドとして分析系ツールを実行し、コードカバレッジを毎日記録

MSBuildでの呼び出し例

MSBuild

```
<ItemGroup>
  <NCoverArguments Include="&quot;$(NUnitConsolePath)¥nunit-console.exe&quot;" />
  <NCoverArguments Include="&quot;$(BinFolderPath)¥$(ProjectName).Test.dll&quot;" />
  <NCoverArguments Include="//w" />
  <NCoverArguments Include="&quot;$(BinFolderPath)&quot;" />
  <NCoverArguments Include="//x" />
  <NCoverArguments Include="&quot;$(CCNetArtifactDirectory)¥Coverage.Xml&quot;" />
  <NCoverArguments Include="//l" />
  <NCoverArguments Include="&quot;$(CCNetArtifactDirectory)¥Coverage.Log&quot;" />
</ItemGroup>

<Exec Command="NCover.Console.exe @(NCoverArguments, ' ')"
  ContinueOnError="true"
  WorkingDirectory="$(BinFolderPath)" >
</Exec>
```

NCover.Console.exeに渡す引数は、非常に多くなるのでリスト項目の中に定義することで可読性が高まる
XML属性の中にリスト項目の値を記述するため、ダブルクォーテーションを"として記述

<Exec>タスクを利用し、NCover.Console.exeを呼び出す
NCover.Console.exeに渡す引数は、リスト項目をスペース区切りに変換して動的に生成

※ MSBuildスクリプトの不要な要素を省略しているため、そのままでは動作しません

©2006 NAgle.com All rights reserved.

15



NAntでの呼び出し例

NAnt

```
<exec program="NCover.Console.exe"
  basedir="C:¥Program Files¥NCover"
  workingdir="D:¥Sample¥NCover¥work" >
  <arg value="&quot;C:¥Program Files¥NUnit-Net-2.0 2.2.9¥bin¥nunit-console.exe&quot;" />
  <arg value="&quot;Debug¥CalcLibrary.Test.dll&quot;" />
  <arg value="//w" />
  <arg value="&quot;D:¥Sample¥NCover¥work&quot;" />
  <arg value="//x" />
  <arg value="&quot;D:¥Sample¥NCover¥work¥Coverage.Xml&quot;" />
  <arg value="//l" />
  <arg value="&quot;D:¥Sample¥NCover¥work¥Coverage.Log&quot;" />
</exec>
```

<Exec>タスクを利用し、NCover.Console.exeを呼び出す

NCover.Console.exeに渡す引数を<arg>要素に記述
複数定義することも可能

※ NAntスクリプトの不要な要素を省略しているため、そのままでは動作しません

©2006 NAgle.com All rights reserved.

16



NCoverExplorer.Extras

- MSBuild/NAntからNCoverを呼び出すことができる拡張タスク
 - <Exec>タスクを使わずにNCoverを簡単に呼び出すことが可能
 - 詳細は下記URLを参照
 - <http://www.kiwidude.com/blog/2006/10/ncoverexplorer-v135.html>
- MSBuild用タスクのヘルプ
 - <http://www.kiwidude.com/dotnet/doc/NCoverExplorer.MSBuildTasks/index.html>
- NAnt用タスクのヘルプ
 - <http://www.kiwidude.com/dotnet/doc/NCoverExplorer.NAntTasks/index.html>

参考URL

- NCover公式サイト
 - <http://ncover.org/site/>
- NCoverExplorer公式サイト
 - <http://www.kiwidude.com/blog/index.html>

