

Использование покрытия кода для определения объема протестированного кода

ХАФИЗОВ КАМИЛЬ

7-21

Требования

Функция проверки объема протестированного кода доступна только в выпуске Visual Studio Enterprise.

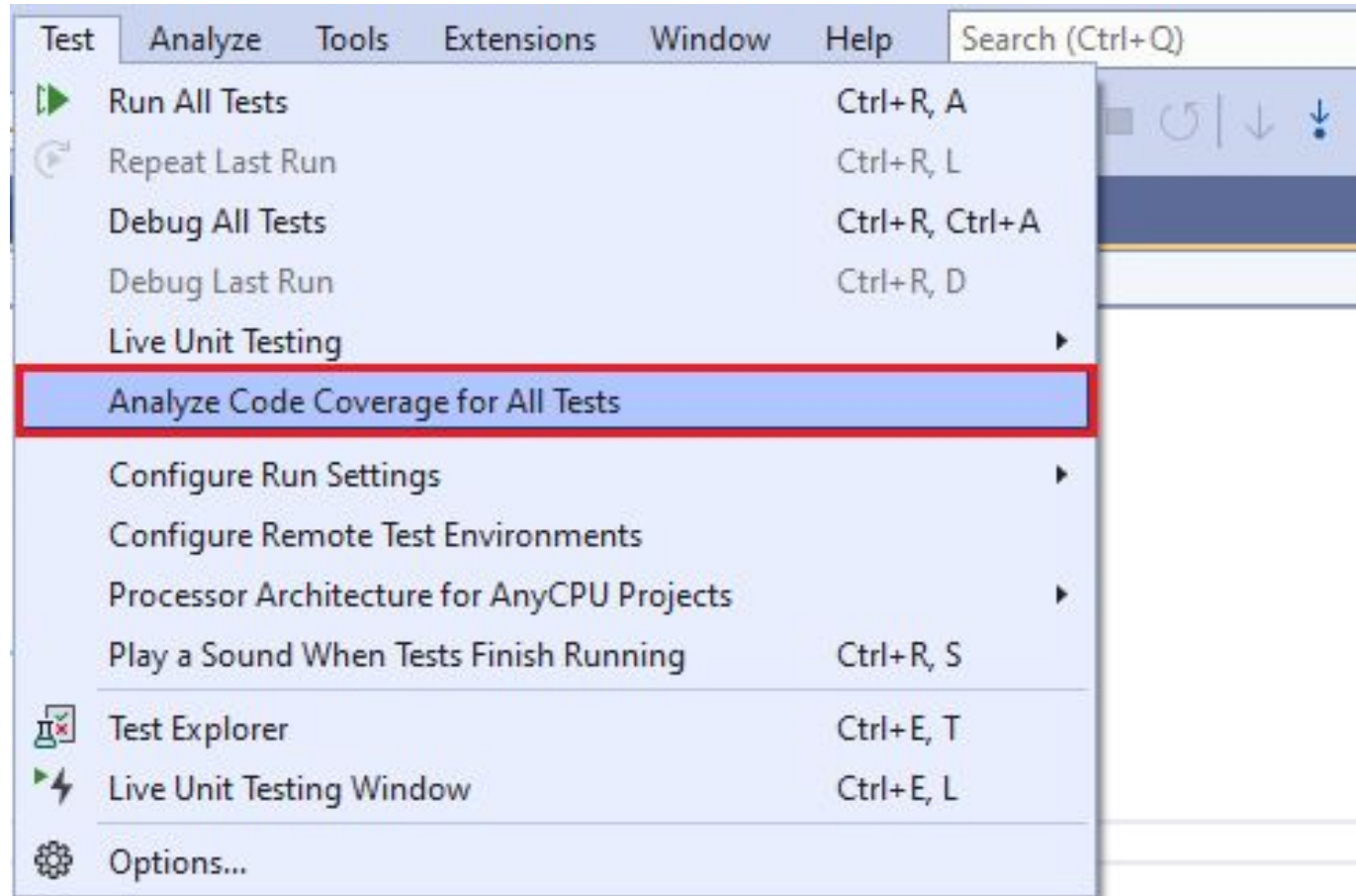
Функция объема протестированного кода в Visual Studio позволяет определить, какая доля кода в вашем проекте охватывается закодированными тестами, например модульными тестами. Чем больше эта доля, тем эффективнее защита от ошибок.


Анализ объема протестированного кода возможен как для управляемого (CLR), так и для неуправляемого (машинного) кода.

Эта функция доступна в меню "Тестирование" при запуске методов тестов в Обозревателе тестов. Таблица результатов отображает процент кода, выполняемого в каждой сборке, каждом классе и каждой процедуре. В редакторе исходного кода выделяется протестированный код.

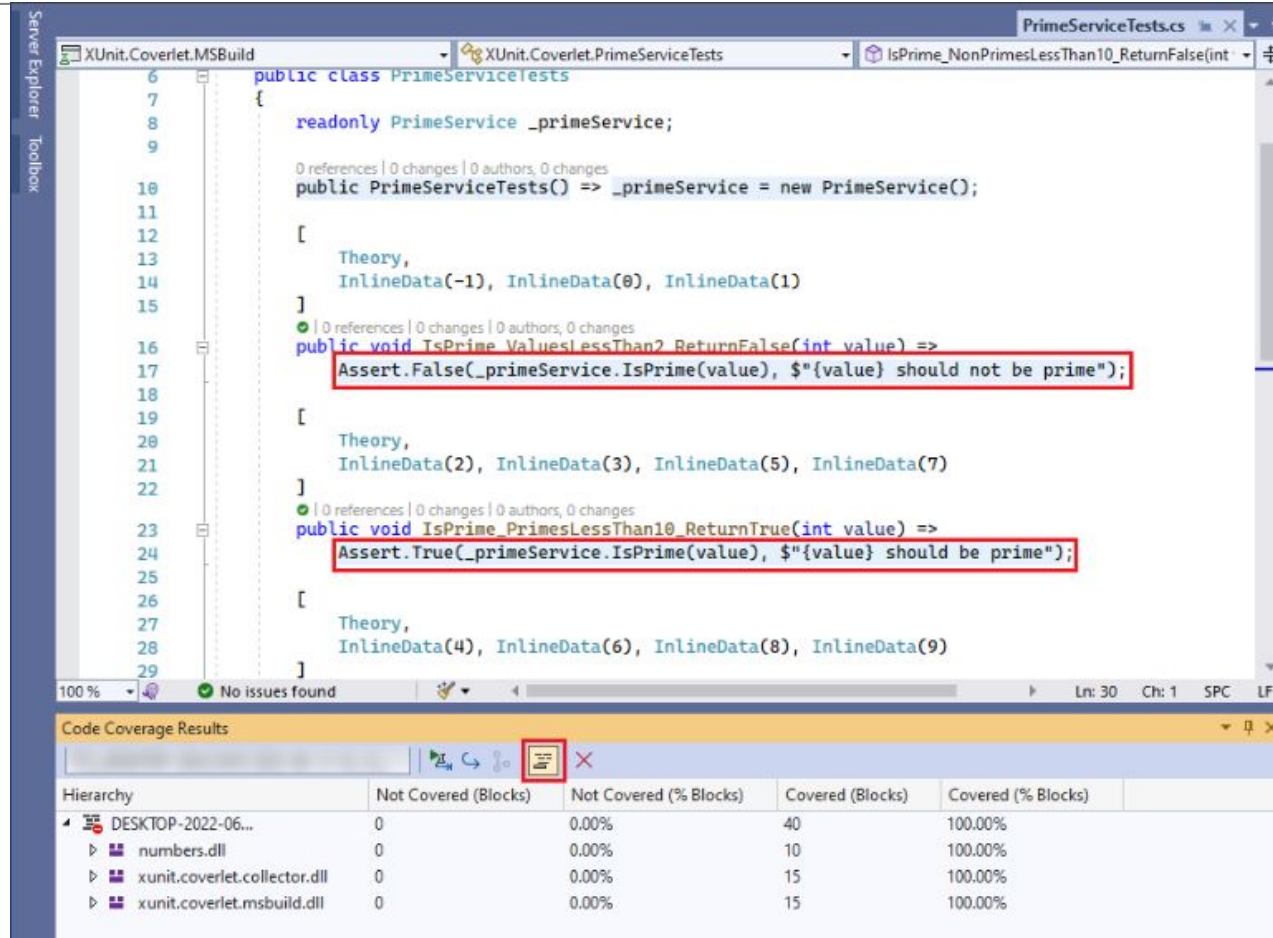
Анализ покрытия кода

1) В меню **Тестирование** выберите **Анализ покрытия кода для всех тестов**.



2) Чтобы после выполнения тестов увидеть, какие строки были обработаны, щелкните 

Цвета отображения покрытия кода в окне Результаты покрытия кода. По умолчанию код, охватываемый тестами, выделяется голубым цветом.

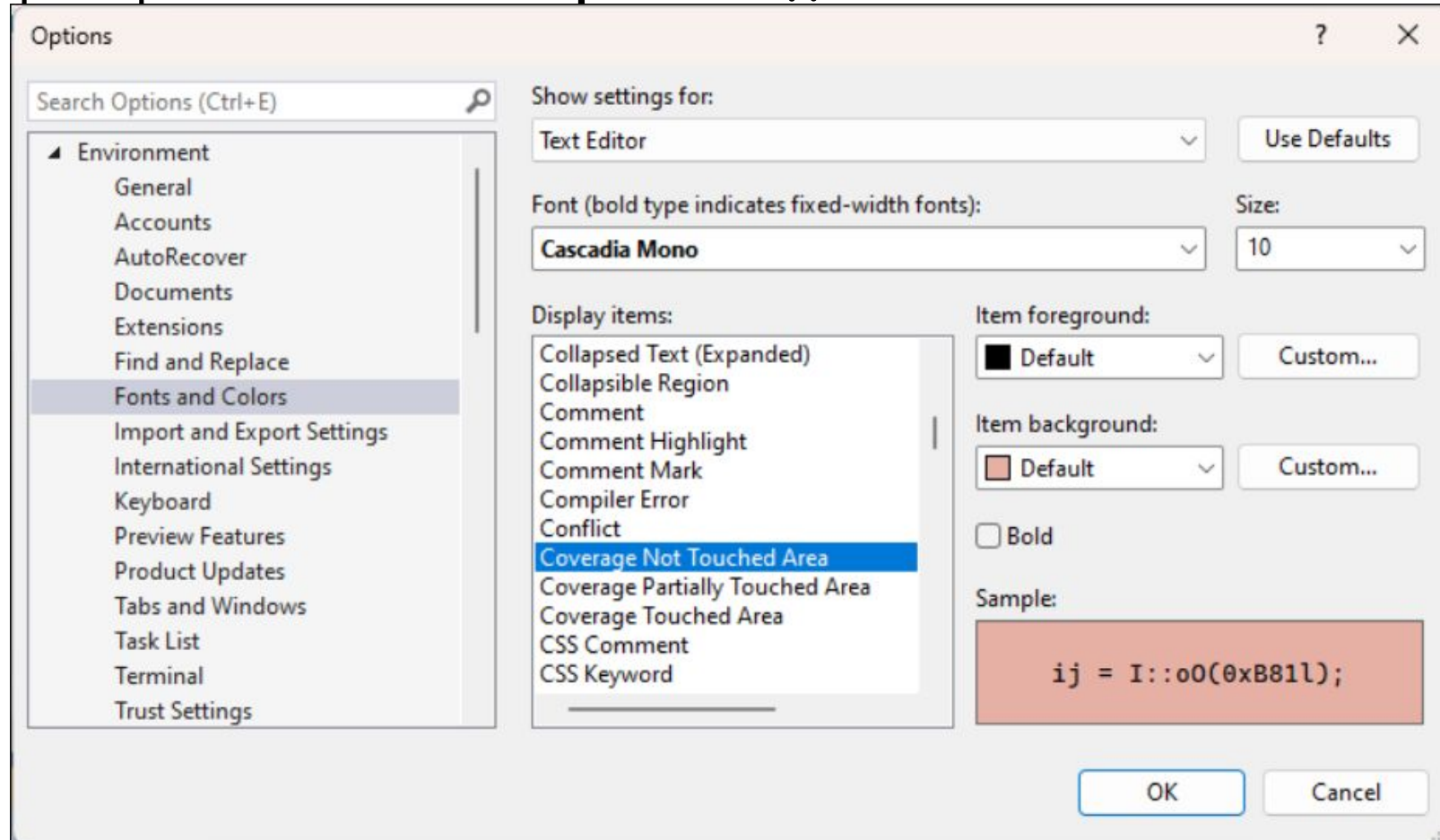


```
6 public class PrimeServiceTests
7 {
8     readonly PrimeService _primeService;
9
10    0 references | 0 changes | 0 authors, 0 changes
11    public PrimeServiceTests() => _primeService = new PrimeService();
12
13    [
14        Theory,
15        InlineData(-1), InlineData(0), InlineData(1)
16    ]
17    0 references | 0 changes | 0 authors, 0 changes
18    public void IsPrime_ValuesLessThan2_ReturnFalse(int value) =>
19        Assert.False(_primeService.IsPrime(value), $"{value} should not be prime");
20
21    [
22        Theory,
23        InlineData(2), InlineData(3), InlineData(5), InlineData(7)
24    ]
25    0 references | 0 changes | 0 authors, 0 changes
26    public void IsPrime_PrimesLessThan10_ReturnTrue(int value) =>
27        Assert.True(_primeService.IsPrime(value), $"{value} should be prime");
28
29    [
30        Theory,
31        InlineData(4), InlineData(6), InlineData(8), InlineData(9)
32    ]
33 }
```

Code Coverage Results

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
DESKTOP-2022-06...	0	0.00%	40	100.00%
numbers.dll	0	0.00%	10	100.00%
xunit.coverlet.collector.dll	0	0.00%	15	100.00%
xunit.coverlet.msbuild.dll	0	0.00%	15	100.00%

3) Чтобы изменить цвета или использовать полужирный шрифт, последовательно выберите **Сервис>Параметры>Среда>Шрифты и цвета>Параметры для: текстовый редактор**. В разделе **Отображаемые элементы** настройте параметры для элементов покрытия, например **Области вне покрытия кода**.



4) Если результаты показывают малый объем протестированного кода, определите, какие части кода не обрабатываются, и напишите дополнительные тесты для их охвата. Команды разработчиков обычно стремятся покрыть около 80 % кода. В некоторых случаях допустимо более низкое покрытие. Например, более низкое покрытие допустимо, когда некоторый код создается из стандартного шаблона.

Не забудьте снова выполнить покрытие кода после обновления кода. Когда вы модифицируете код или запускаете тесты, предыдущие результаты и цвета объема протестированного кода автоматически не обновляются.

Отчеты в блоках или строках

Покрытие кода измеряется в *блоках*. Блок — это часть кода строго с одной точкой входа и точкой выхода. Если поток управления программы проходит через блок во время тестового запуска, то этот блок учитывается как *покрытый*. То, сколько раз используется блок, не влияет на результат.

Можно также отобразить результаты для строк, выбрав *Добавить* или *удалить* столбцы в заголовке таблицы.

Отчеты в блоках или строках

Покрытие кода измеряется в *блоках*. Блок — это часть кода строго с одной точкой входа и точкой выхода. Если поток управления программы проходит через блок во время тестового запуска, то этот блок учитывается как *покрытый*. То, сколько раз используется блок, не влияет на результат.

Можно также отобразить результаты для строк, выбрав *Добавить* или *удалить* столбцы в заголовке таблицы.

Объединение результатов различных запусков

В некоторых ситуациях будут использоваться разные блоки в коде в зависимости от данных теста. Поэтому может потребоваться объединить результаты различных тестовых запусков.

Для этого используйте **Объединение результатов покрытия кода**. Можно выбрать любые сочетания последних запусков или импортированных результатов. Если требуется объединить экспортированные результаты, в первую очередь необходимо их импортировать.

Щелкните **Экспортировать результаты** покрытия кода, чтобы сохранить результаты операции слияния.

Анализ объема протестированного кода в командной строке

Для выполнения тестов из командной строки используйте служебную программу `vstest.console.exe`. Объем протестированного кода вызывается в `vstest.console.exe` с помощью параметра `/EnableCodeCoverage`.

1. Запустите командную строку разработчика Visual Studio.
В меню Пуск Windows выполните поиск по тексту Developer Command Prompt for VS и выберите соответствующее приложение в результатах.
2. В командной строке выполните следующую команду:

```
vstest.console.exe MyTestAssembly.dll /EnableCodeCoverage
```