

Инструменты тестирования

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

NUnit	4
Добавить тест для пользовательского класса	4
1. Установить адаптер NUnit для Visual Studio	5
2. Выгрузить пакет в файловую систему	6
3. Настроить проект Unit-тестов	7
4. Создать тесты	8
5. Выполнить тестирование	10

NUnit



Unit-тестирование ([модульное тестирование](#)) — процесс в программировании, позволяющий проверить работоспособность изолированных частей программы. Как правило, тесты пишутся разработчиками для каждого нетривиального метода разрабатываемого класса. Это позволяет обнаружить **регрессию исходного кода** — появление ошибок в уже протестированных частях программы.

Одним из фреймворков Unit-тестирования .NET-приложений является [NUnit](#) — среда Unit-тестирования с открытым исходным кодом. Для ее интеграции с Visual Studio разработан специальный **адаптер**.

Варианты установки адаптера:

- Расширение к Visual Studio.
- NuGet-пакет проекта, в котором реализованы Unit-тесты.

Особенности использования адаптера NUnit **как расширения для Visual Studio**:

- Доступность для любого проекта тестов, поскольку адаптер становится частью IDE.
- Автоматическое обновление расширения.
- Необходимость установки для каждого участника команды, работающего над проектом тестов.

Особенности использования адаптера NUnit **как пакета NuGet**:

- Пакет является частью проекта Visual Studio и доступен для всех разработчиков, использующих проект.
- Необходимость установки для всех проектов Unit-тестов.

Работа с NUnit описана в официальной [документации NUnit](#).

Чтобы **создать Unit-тесты** для методов или свойств класса пользовательского пакета необходимо:

1. Установить адаптер NUnit для Visual Studio.
2. Перейти в [режим разработки в файловой системе](#).
3. Настроить проект Unit-тестов.
4. Создать тесты.
5. Выполнить тестирование.

Добавить тест для пользовательского класса



Пример. Добавить тесты для пользовательского класса, реализованного в схеме типа [*Исходный*

код] ([*Source code*]) `UsrNUnitSourceCode` пользовательского пакета `sdkNUnit` .

1. Установить адаптер NUnit для Visual Studio

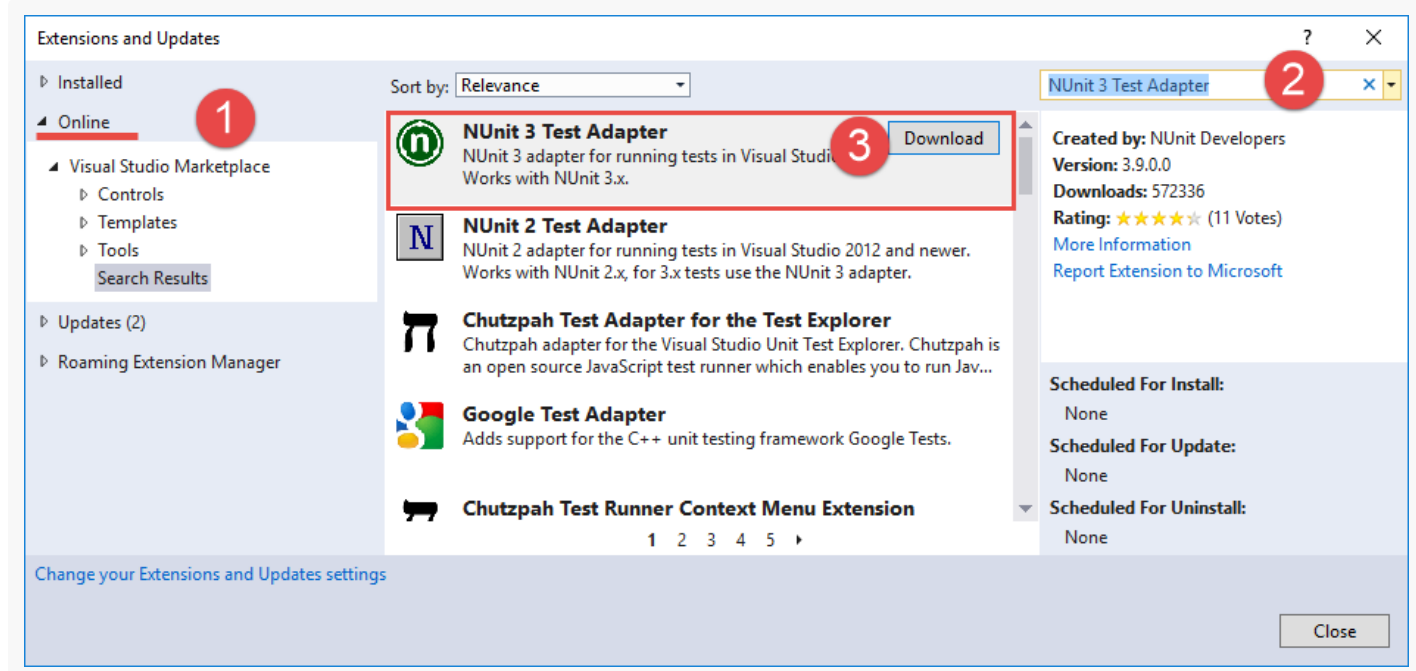
Установить адаптер NUnit как расширение Visual Studio

Способ 1:

1. [Скачайте расширение](#) из Visual Studio Marketplace.
2. Двойным кликом по *.vsix-файлу запустите установку.
3. Во время установки выберите необходимые версии Visual Studio.

Способ 2:

1. В Visual Studio нажмите [*Tools*] —> [*Extensions and Updates*].
2. Выберите фильтр [*Online*] (1).
3. В строке поиска укажите "NUnit 3 Test Adapter" (2).
4. В результатах поиска выберите расширение `NUnit 3 Test Adapter` (3).
5. Для установки расширения нажмите кнопку [*Download*].

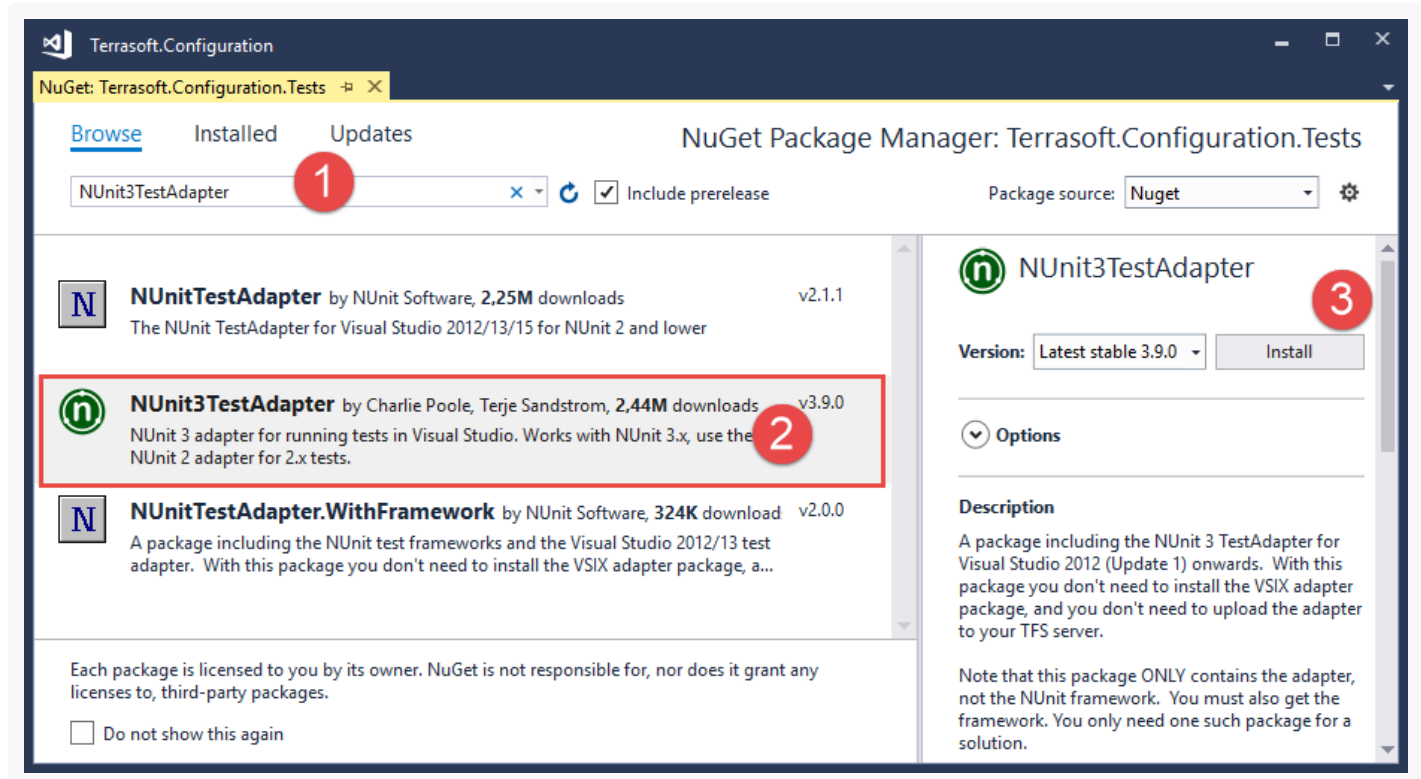


Установить адаптер NUnit как пакет NuGet

1. Кликните правой кнопкой мыши по названию проекта тестов (например, `Terrasoft.Configuration.Tests.csproj`) и выберите [*Manage NuGet Packages...*].
2. В строке поиска укажите "NUnit3TestAdapter" (1).

- В результатах поиска выберите пакет `NUnit 3 Test Adapter` (2).
- Для установки пакета нажмите кнопку [*Install*] (3).

Установка пакета NuGet подробно описана в официальной [документации Microsoft](#).



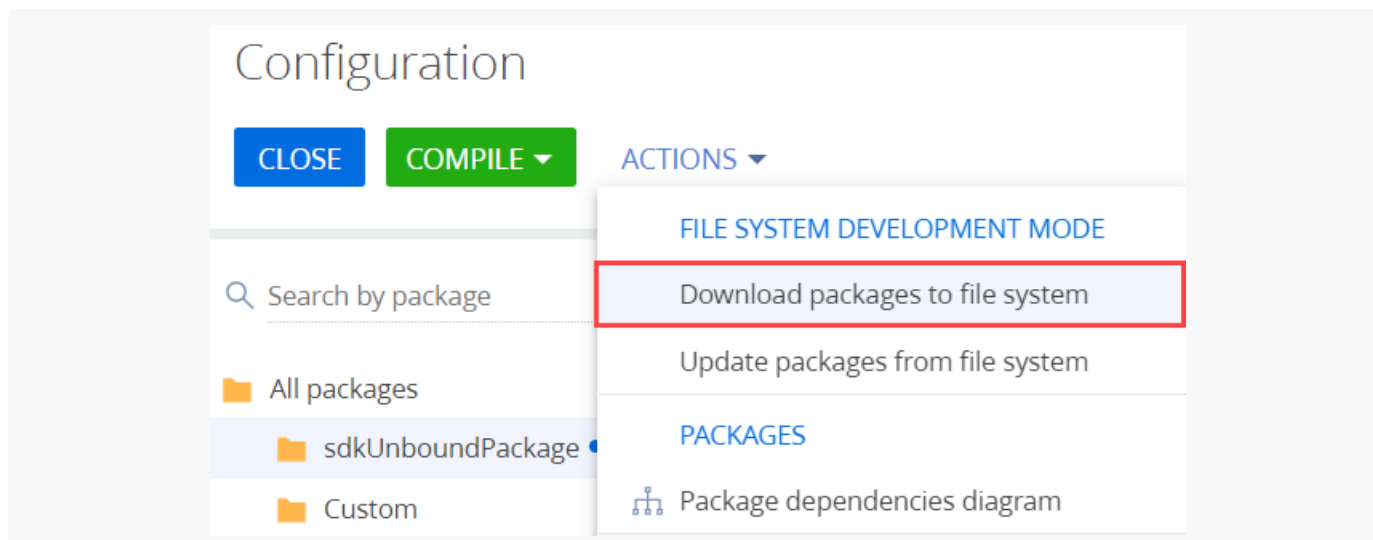
2. Выгрузить пакет в файловую систему

Создание Unit-тестов для .NET классов, реализованных в пакетах Creatio, возможно только в [режиме разработки в файловой системе](#).

В нашем примере используется пользовательский пакет `sdkNUnit`.

Чтобы **выгрузить пакет** в файловую систему:

- Настройте Creatio для работы в файловой системе. Настройка описана в статье [Внешние IDE](#).
- На панели инструментов в группе действий [*Разработка в файловой системе*] ([*File system development mode*]) выберите [*Выгрузить все пакеты в файловую систему*] ([*Download packages to file system*]).



В результате все пакеты будут выгружены по пути `..\Terrasoft.WebApp\Terrasoft.Configuration\Pkg` в каталог с соответствующим названием пакета. Пакет `sdnUnit` содержит схему типа [*Исходный код*] ([*Source code*]) `UsrNUnitSourceCode`.

В исходном коде схемы `UsrNUnitSourceCode` реализован класс `UsrNUnitSourceCode`, содержащий методы, для которых необходимо написать тесты.

UsrNUnitSourceCode

```
namespace Terrasoft.Configuration
{
    public class UsrNUnitSourceCode
    {
        // Строковое свойство.
        public string StringToTest
        {
            get
            {
                return "String to test";
            }
        }
        // Метод, проверяющий равенство двух строк.
        public bool AreStringsEqual(string str1, string str2)
        {
            return str1 == str2;
        }
    }
}
```

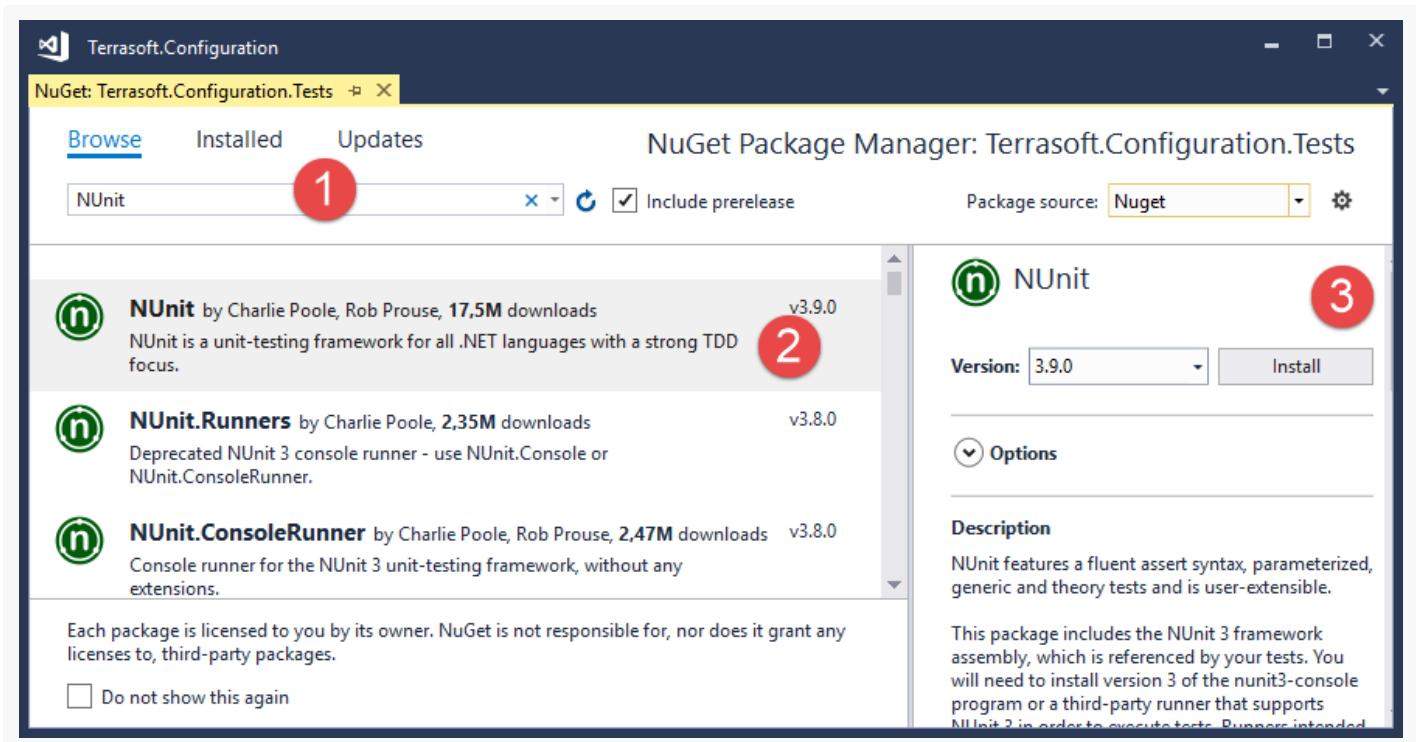
3. Настроить проект Unit-тестов

В нашем примере для создания Unit-тестов используется предварительно настроенный [проект](#)

`Terrasoft.Configuration.Tests.csproj`, поставляемый вместе с решением `Terrasoft.Configuration.sln`.

Чтобы использовать в проекте `Terrasoft.Configuration.Tests.csproj` фреймворк NUnit для создания тестов, необходимо **добавить NuGet-пакет NUnit в зависимости проекта**. Для этого:

1. В [*Solution Explore*] кликните правой кнопкой мыши по названию проекта тестов `Terrasoft.Configuration.Tests`.
2. Выберите команду [*Manage NuGet Packages...*].
3. В строке поиска укажите "NUnit" (1).
4. В результатах поиска выберите пакет `NUnit` (2).
5. Для установки пакета нажмите кнопку [*Install*] (3).

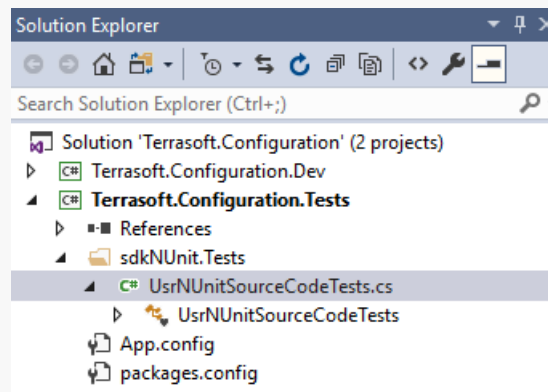


4. Создать тесты

Имя содержащего тесты класса должно состоять из имени тестируемого класса с добавлением окончания "Tests". Также для **группировки тестов в проекте** удобно помещать их в каталог, название которого совпадает с названием тестируемого пакета с добавлением окончания ".Tests".

Чтобы **создать тесты**:

1. В проекте `Terrasoft.Configuration.Tests.csproj` создайте каталог `sdnNUnit.Tests`.
2. В каталоге `sdnNUnit.Tests` создайте новый класс `UsrNUnitSourceCodeTests`. Исходный код этого класса будет сохранен в файле `UsrNUnitSourceCodeTests.cs`.



3. В класс `UsrNUnitSourceCodeTests` добавьте методы, реализующие тесты.

`UsrNUnitSourceCodeTests`

```
using NUnit.Framework;

namespace Terrasoft.Configuration.Tests.sdkNUnitTests
{
    [TestFixture]
    class UsrNUnitSourceCodeTests
    {
        // Экземпляр тестируемого класса.
        UsrNUnitSourceCode objToTest = new UsrNUnitSourceCode();
        // Строка для тестирования.
        string str = "String to test";

        [Test]
        public void ClassReturnsCorrectStringProperty()
        {
            // Тестирование значения строкового свойства.
            // Значение должно быть не пустым и совпадать с требуемым.
            string res = objToTest.StringToTest;
            Assert.That(res, Is.NotNull.And.EqualTo(str));
        }

        [Test]
        public void StringsMustBeEqual()
        {
            // Тестирование на равенство значений двух строк.
            bool res = objToTest.AreStringsEqual(str, "String to test");
            Assert.That(res, Is.True);
        }

        [Test]
        public void StringsMustBeNotEqual()
        {

```

```

        // Тестирование на неравенство значений двух строк.
        // Этот тест будет провален, т.к. значения равны.
        bool res = objToTest.AreStringsEqual(str, "String to test");
        Assert.That(res, Is.False);
    }
}

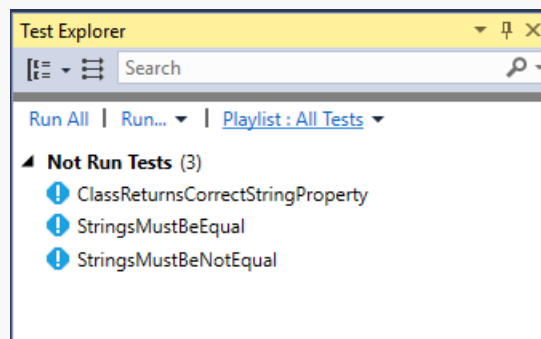
```

Атрибут `[TestFixture]` показывает, что класс `UsrNUnitSourceCodeTests` содержит тесты. Для каждого метода, тестирующего определенную функциональность этого класса, необходимо добавить атрибут `[Test]`. Атрибуты фреймворка NUnit описаны в [документации NUnit](#).

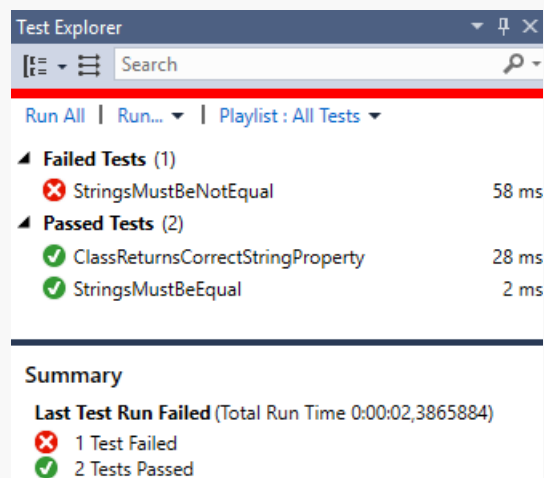
Тестирование выполняется с помощью метода [Assert.That\(\)](#), который принимает тестируемое значение. В качестве аргументов используются объекты, ограничивающие тестируемое значение. Модель ограничений описана в [документации NUnit](#).

5. Выполнить тестирование

1. В Visual Studio нажмите `[Test]` —> `[Windows]` —> `[Test Explorer]`.



2. Для запуска тестов выполните команду `[Run All]`. Успешно пройденные тесты будут перемещены в группу `[Passed Test]`, а не пройденные тесты — в группу `[Failed Test]`.



Функциональность окна [*Test Explorer*] описана в [документации Visual Studio](#).