

# Отладка

Back-end отладка

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

<b>Back-end отладка</b>	<b>4</b>
Выполнение back-end отладки	4
Возможные проблемы при отладке	4
<b>Выполнить отладку серверного кода</b>	<b>6</b>
1. Выполнить настройку приложения	7
2. Выгрузить исходные коды конфигурации	7
3. Создать проект Visual Studio для отладки	8
4. Добавить в проект файлы с исходным кодом	9
5. Подключить проект к рабочему процессу IIS	10
6. Выполнить отладку	11

# Back-end отладка



Сложный

**Back-end отладка** — отладка серверных схем исходного кода. Это могут быть, например, существующие базовые схемы, пользовательские конфигурационные классы, веб-сервисы или скрипты бизнес-процессов, написанные на языке C#.

## Выполнение back-end отладки

Выполнять отладку серверных схем исходного кода Creatio можно исключительно с помощью интегрированных функций отладки **внешних IDE**, например, Visual Studio. Отладчики внешних IDE позволяют:

- приостанавливать выполнение методов;
- проверять значения переменных;
- изменять значения переменных;
- получать полное представление о том, что делает код.

В этом руководстве описано выполнение back-end отладки на примере Visual Studio.

**Необходимые условия** выполнения отладки с использованием Visual Studio:

- Приложение Creatio развернуто on-site.
- Режим разработки в файловой системе отключен.
- В Visual Studio включен признак [ *Suppress JIT Optimization* ] (меню [ *Options* ], вкладки [ *Debugging* ] —> [ *General* ]). Это позволяет во время отладки узнать значения переменных. Подробнее об оптимизированном и неоптимизированном коде во время отладки можно узнать из [документации Visual Studio](#).

**Последовательность** back-end отладки:

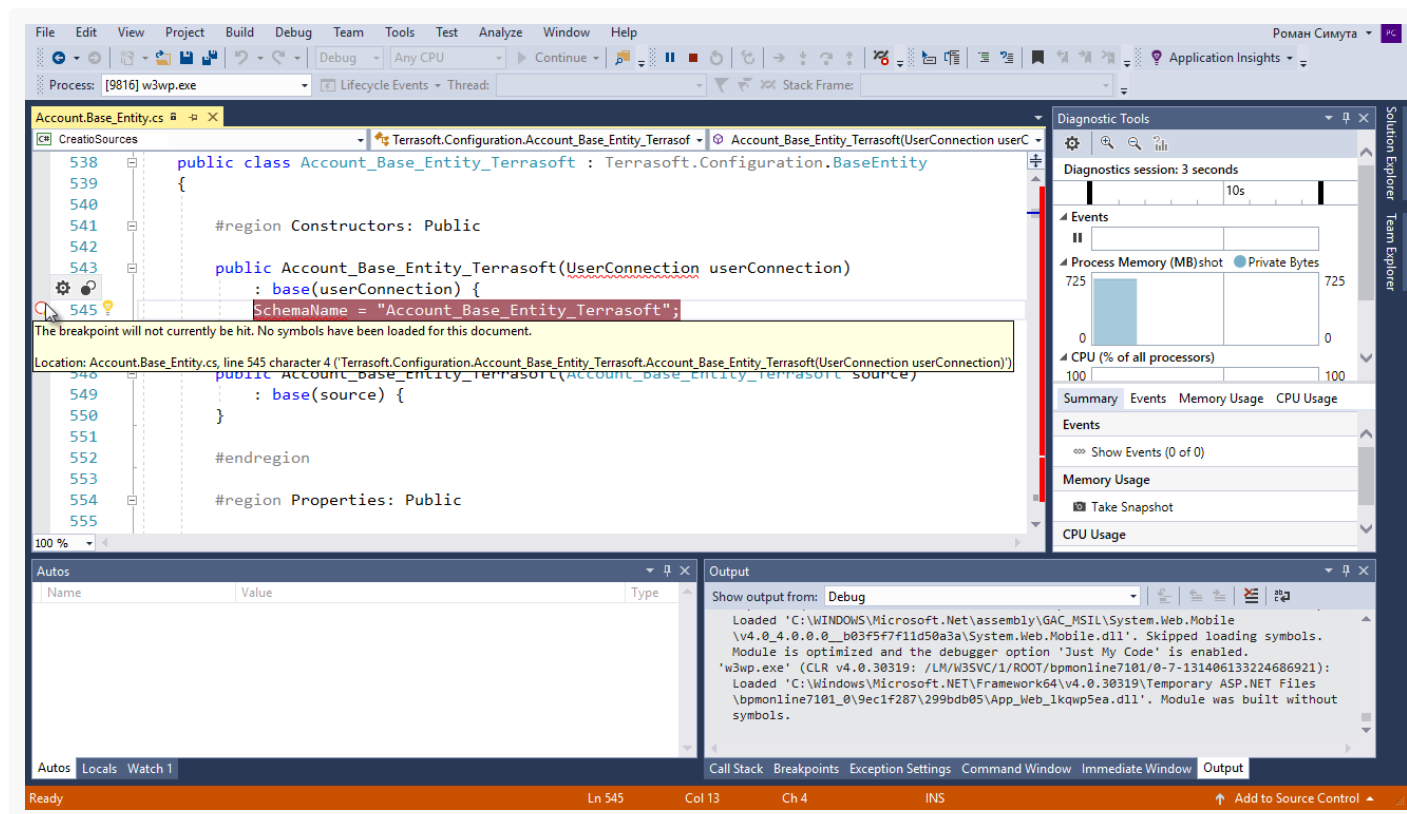
1. Выгрузить исходные коды конфигурации Creatio в файлы локального каталога.
2. Создать новый проект в Visual Studio, в котором будет выполняться отладка.
3. Добавить в проект Visual Studio выгруженные файлы с исходным кодом.
4. Подключить проект к рабочему процессу сервера IIS и начать процесс отладки.

## Возможные проблемы при отладке

Символ точки останова отображается в виде белого круга, ограниченного красной окружностью.

Такая точка останова является неактивной и на ней не будет прерываться выполнение скрипта. При наведении курсора на символ неактивной точки останова появится подсказка, уведомляющая о

проблеме.

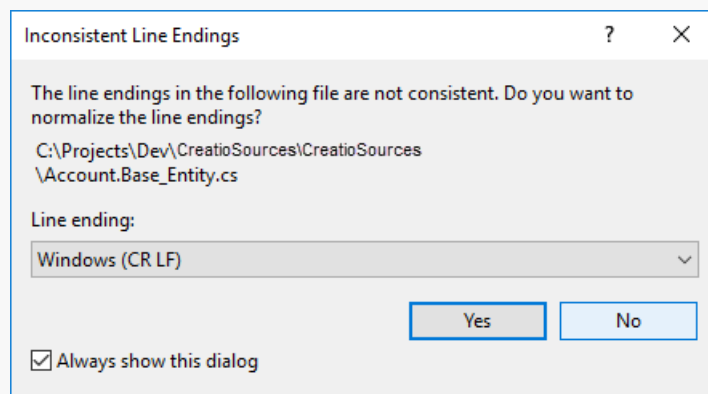


## Решение:

1. Завершить выполнение отладки ( **Debug** —> **Stop Debugging** ).
2. Закрыть файл с исходным кодом, для которого выполняется отладка.
3. В разделе [ *Конфигурация* ] ( [ *Configuration* ] ) приложения выполнить действие [ *Компилировать все* ] ( [ *Compile all* ] ).
4. Во время выполнения компиляции и перегрузки файлов с исходными кодами подключить проект к процессу IIS заново.
5. После выполнения компиляции переоткрыть файл с исходным кодом, для которого выполняется отладка.

**На заметку.** В некоторых случаях может помочь повторная компиляция без открепления и прикрепления к IIS.

После повторного открытия файла с исходным кодом появилось сообщение о неединообразных символах в конце строк.

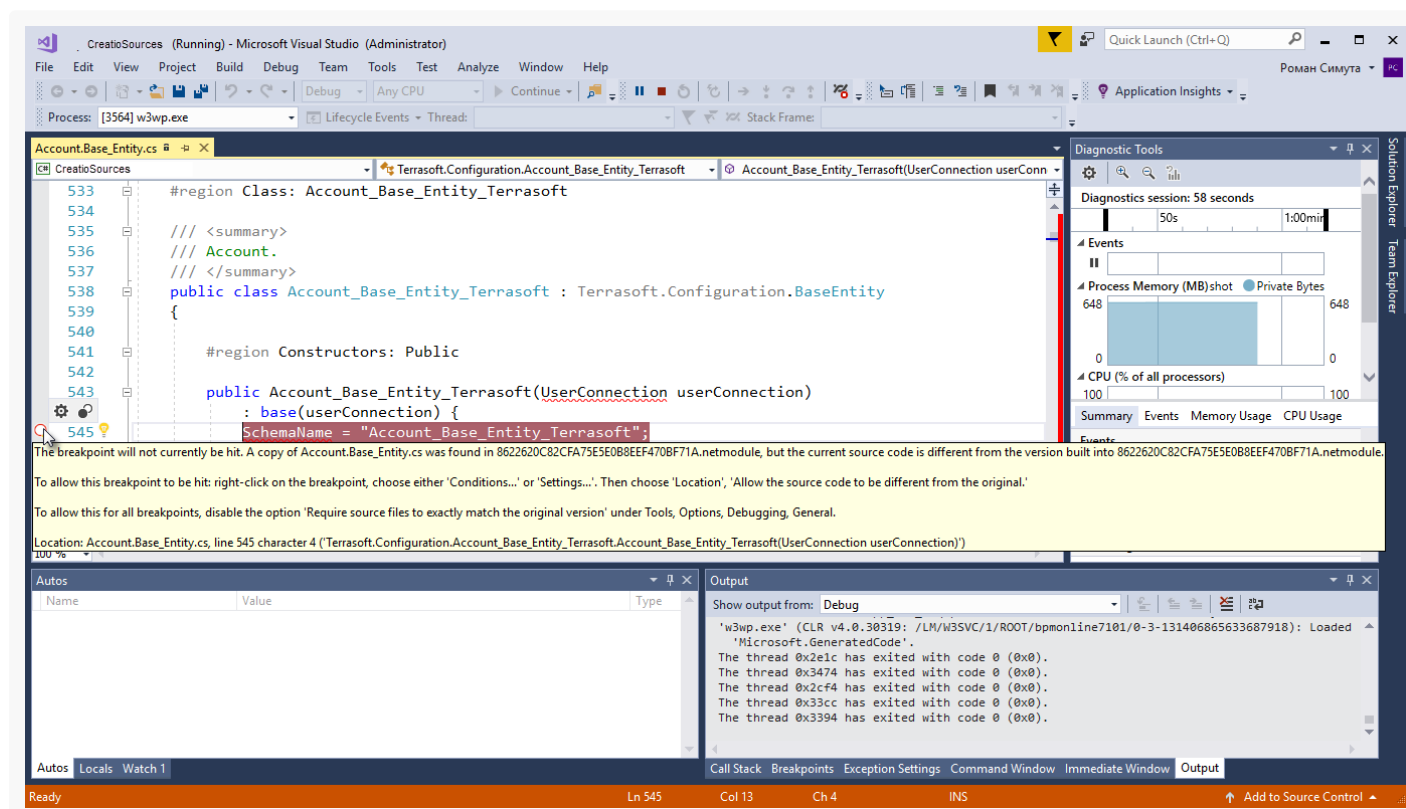


## Решение:

1. Нажать кнопку [ Нет ] ([ No ]). Если согласиться с нормализацией символов (кнопка [ Да ] ([ Yes ])), то точка останова снова может стать неактивной.

Причина проблемы отобразится в подсказке — несоответствие версий файла. Также в подсказке отображены варианты решения проблемы.

2. Выполнить один из предложенных вариантов решения проблемы.



# Выполнить отладку серверного кода



# 1. Выполнить настройку приложения

Для выполнения отладки необходимо внести изменения в конфигурационные файлы приложения.

Чтобы **выполнить настройку приложения**:

## 1. Настройте "внешний" `Web.config`.

В файле `Web.config`, расположенном в корневом каталоге приложения, для атрибута `debug` элемента `<compilation>` установите значение `true`.

`Web.config`

```
<compilation debug="true" targetFramework="4.5" />
```

После внесения изменений сохраните файл.

## 2. Настройте "внутренний" `Web.config`.

В файле `Web.config`, расположенном в каталоге `Terrasoft.WebApp` приложения, укажите значения для следующих элементов:

- `IncludeDebugInformation` — `true`.
- `ExtractAllCompilerSources` — `true`, если необходимо выгружать все схемы при выполнении действия [ *Компилировать* ] ([ *Compile* ]) раздела [ *Конфигурация* ] ([ *Configuration* ])
- `ExtractAllCompilerSources` — `false`, если необходимо выгружать только измененные схемы (значение по умолчанию).

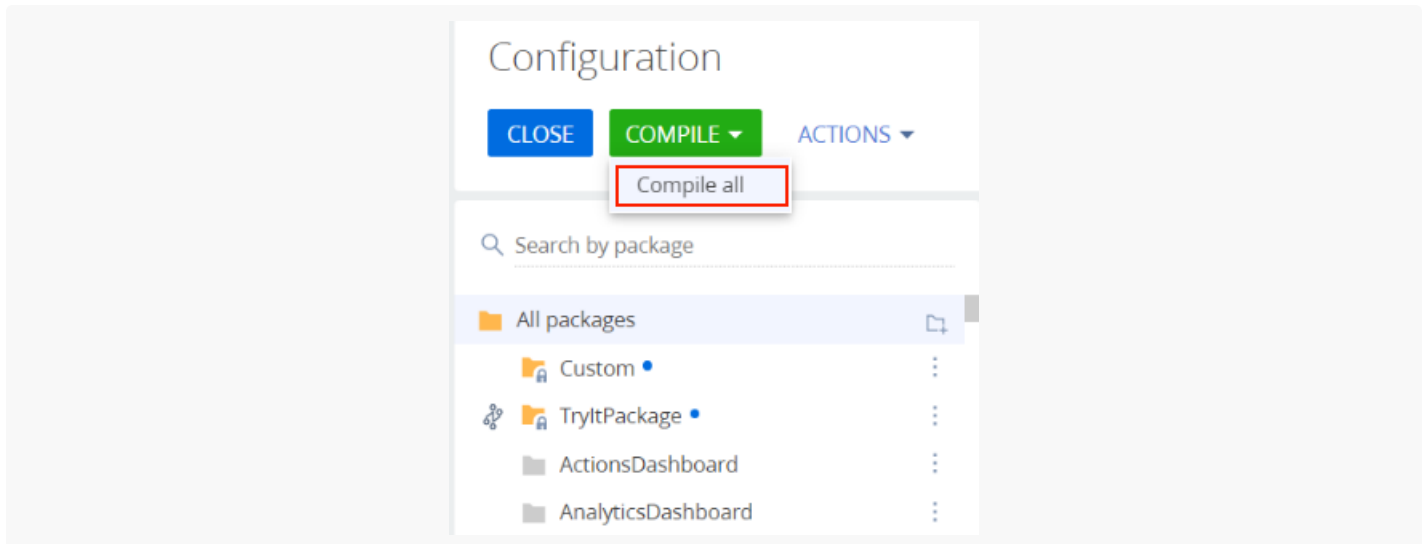
`Web.config`

```
<add key="IncludeDebugInformation" value="true" />
<add key="ExtractAllCompilerSources" value="false" />
```

После внесения изменений сохраните файл.

# 2. Выгрузить исходные коды конфигурации

В разделе [ *Конфигурация* ] ([ *Configuration* ]) приложения выполните действие [ *Компилировать все* ] ([ *Compile all* ]).



Во время компиляции в папку `../Terrasoft.WebApp/Terrasoft.Configuration/Autogenerated/Src` будут выгружены файлы с исходными кодами конфигурационных схем приложения, а также конфигурационные библиотеки, их модули и файлы с отладочной информацией (\*.pdb). Исходные коды схем будут выгружены заново при каждой последующей компиляции приложения.

Файлы выгруженных исходных кодов конфигурационных схем именуются в определенном формате:

[Название схемы в конфигурации].[Название пакета]\_[Тип схемы].cs .

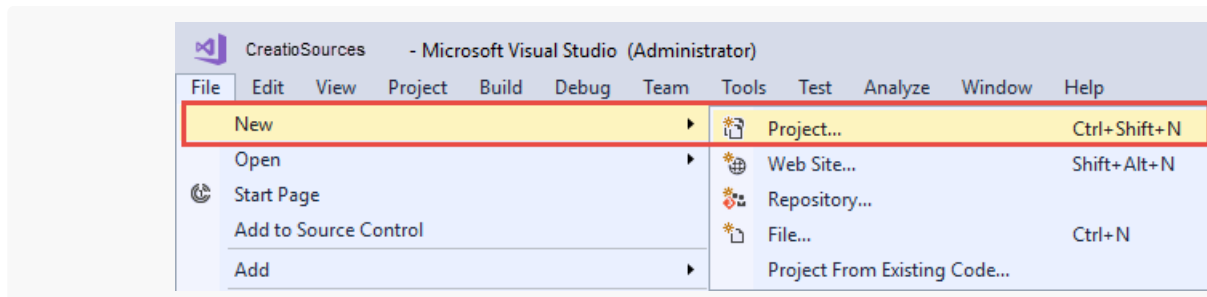
Например: `Contact.Base_Entity.cs` , `ContractReport.Base_Report.cs` .

### 3. Создать проект Visual Studio для отладки

**Важно.** Для отладки исходного кода создавать проект Visual Studio не обязательно. Достаточно открыть в Visual Studio нужные файлы. Однако если отладка выполняется часто или необходимо работать с большим количеством файлов одновременно, то создание проекта сделает работу более удобной.

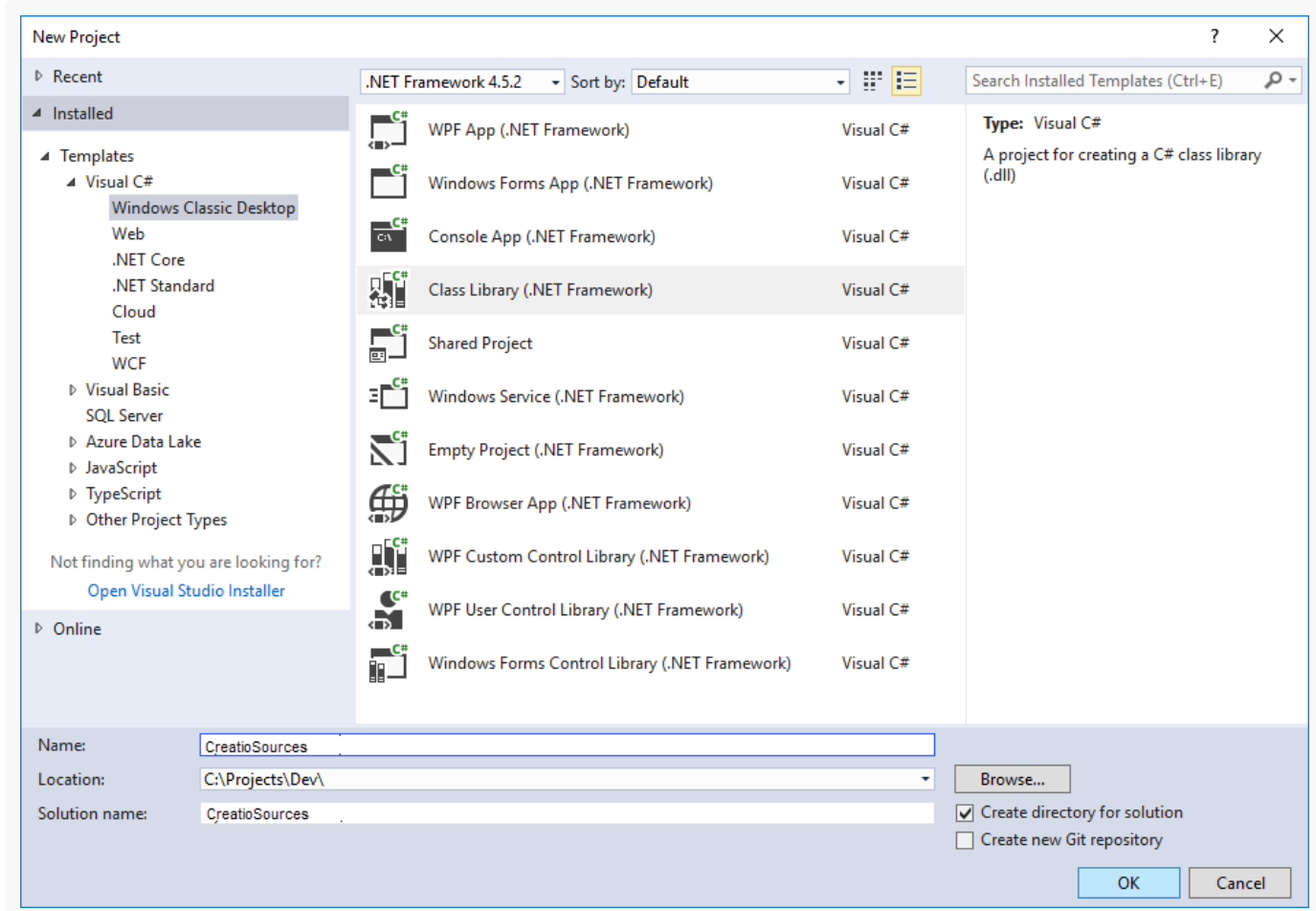
Чтобы **создать проект Visual Studio** для отладки:

1. В Visual Studio выполните команду меню [ File ] —> [ New ] —> [ Project ] .



2. В окне свойств создаваемого проекта выберите тип проекта [ *Class Library (.NET Framework)* ], укажите название и расположение проекта.

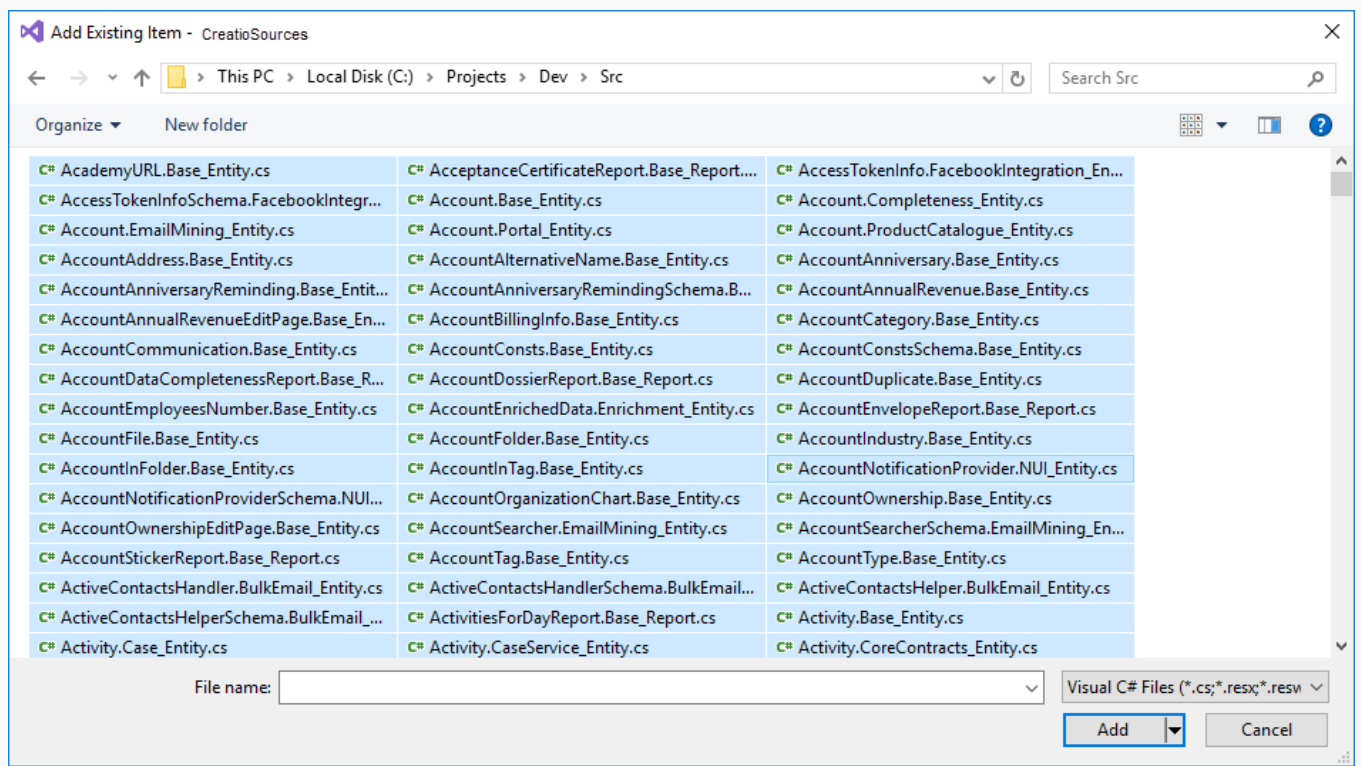




3. После создания проекта удалите из него файл `Class1.cs`, который был создан по умолчанию, и сохраните проект.

## 4. Добавить в проект файлы с исходным кодом

1. В контекстном меню проекта в проводнике решения выберите команду [ *Add* ] —> [ *Existing Item* ].
2. Перейдите в каталог с выгруженными файлами с исходным кодом и выберите все файлы.

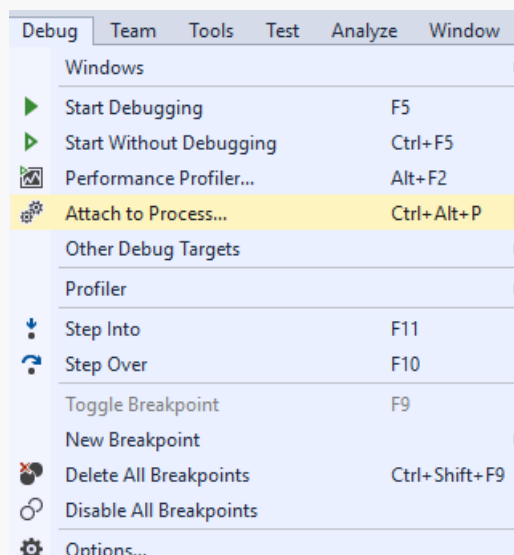


**На заметку.** В проект Visual Studio можно добавлять только необходимые для отладки файлы. Но тогда переход между методами при отладке будет ограничен только методами классов, реализованных в добавленных в проект файлах.

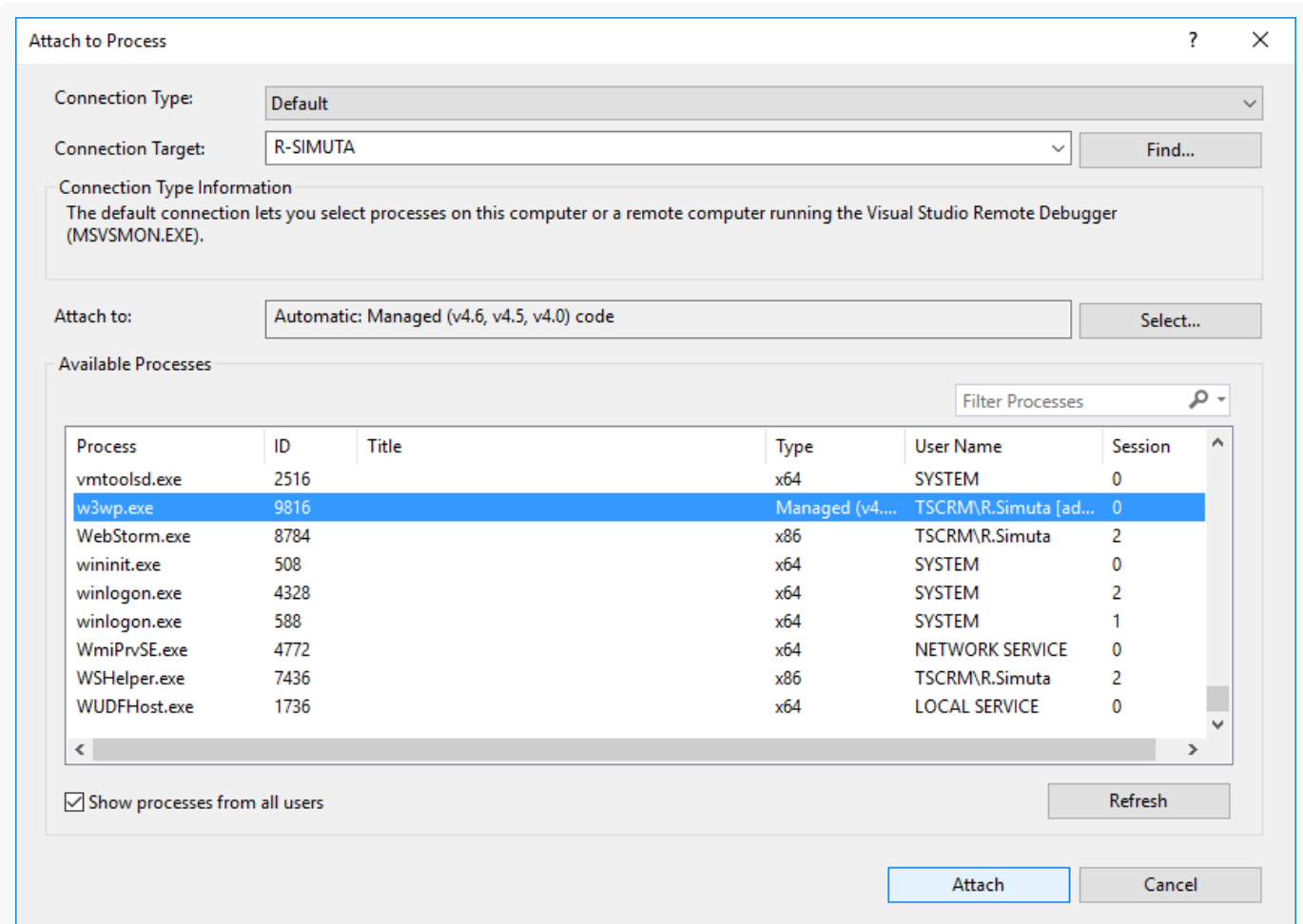
3. После добавления файлов сохраните проект.

## 5. Подключить проект к рабочему процессу IIS

1. В меню Visual Studio выберите команду [ *Debug* ] —> [ *Attach to process* ].



2. В списке процессов выберите рабочий процесс IIS, в котором запущен пул приложения Creatio.

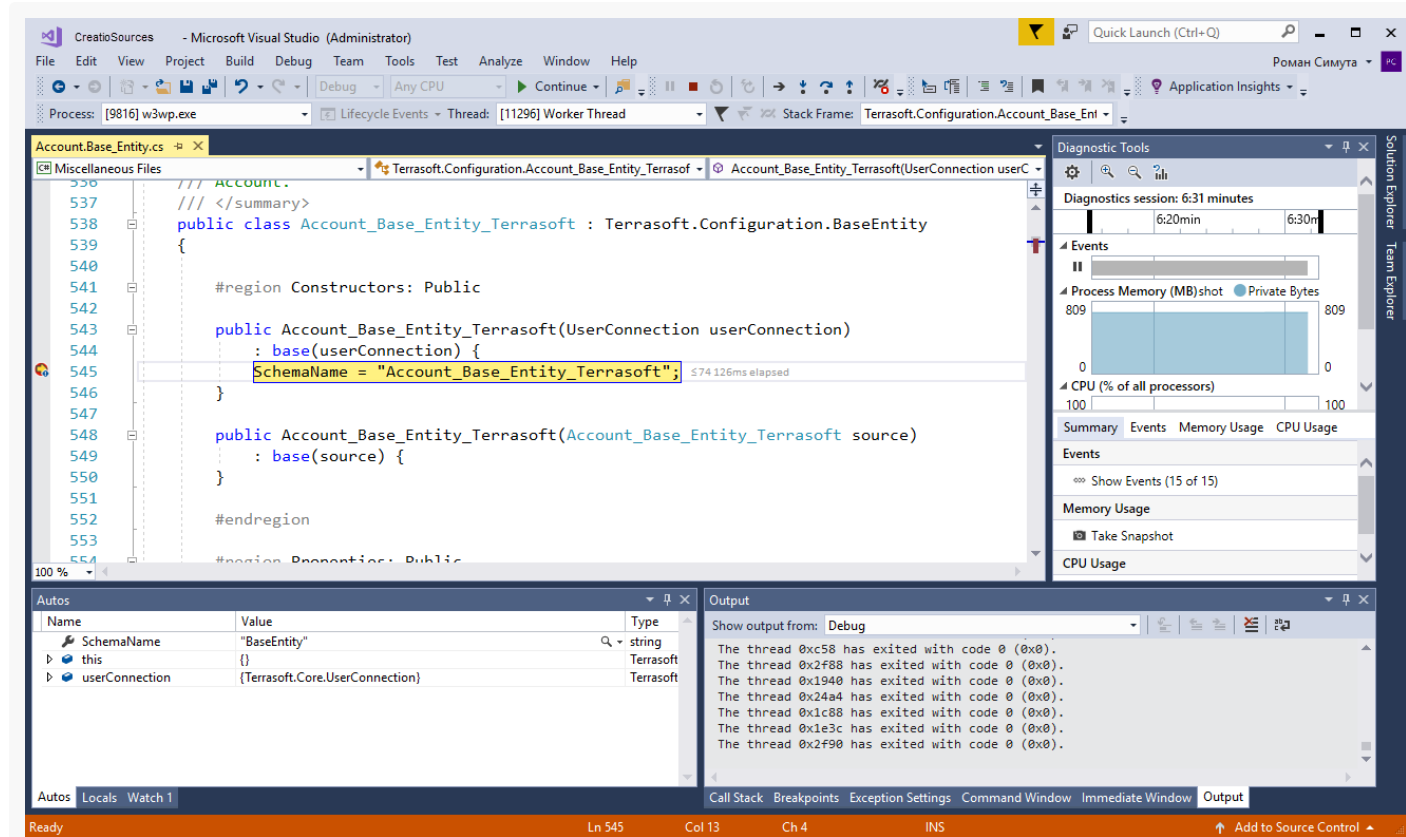


**Важно.** Название рабочего процесса может различаться в зависимости от конфигурации используемого сервера IIS. Для полнофункционального IIS Web Server процесс называется `w3wp.exe`, для IIS Express — `iisexpress.exe`.

По умолчанию рабочий процесс IIS запущен под учетной записью, имя которой совпадает с именем пула приложения. Чтобы отобразить процессы всех пользователей, а не только текущего, необходимо установить признак [ *Show processes from all users* ].

## 6. Выполнить отладку

Откройте файл с необходимым исходным кодом и установите точку останова.



Как только будет задействован метод, на котором была установлена точка останова, программа будет остановлена и можно будет проверить текущее состояние переменных и выполнить трассировку кода.

