

# Фоновое выполнение операций

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>Фоновое выполнение операций</b>	4
<b>Зарегистрировать фоновую операцию</b>	5
1. Создать класс для объекта активности	5
2. Создать класс для добавления активности	6
3. Создать бизнес-процесс для запуска фоновой операции	8
Результат выполнения примера	11

# Фоновое выполнение операций



**Фоновое выполнение операций** позволяет в фоновом режиме выполнять операции, которые требуют длительного времени выполнения, без задержек в работе пользовательского интерфейса.

Для запуска фоновых операций в классе `Terrasoft.Core.Tasks.Task` реализованы методы `StartNew()` и `StartNewWithUserConnection()`. В качестве параметров методов можно использовать базовые типы данных в .NET (например, `string`, `int`, `Guid` и т. д.) или пользовательские типы. **Отличие** метода `StartNewWithUserConnection()` — запуск фоновой операции, которая требует использования пользовательского соединения `UserConnection`.

Параметры, которые принимаются фоновой операцией, преобразуются в массив байт с помощью модуля `MessagePack-CSharp`. Реализация модуля `MessagePack-CSharp` содержится на [сайте GitHub](#). Если не удастся сериализовать или десериализовать значение параметра, могут сгенерироваться исключения.

**Важно.** В фоновой операции не рекомендуется использовать бесконечные циклы, поскольку это приводит к невозможности запуска в приложении других задач.

Действие асинхронной операции описывается в отдельном классе, который должен реализовать интерфейс `IBackgroundTask<in TParameters>`.

## Интерфейс `IBackgroundTask<in TParameters>`

```
namespace Terrasoft.Core.Tasks
{
    public interface IBackgroundTask<in TParameters>
    {
        void Run(TParameters parameters);
    }
}
```

Если для выполнения действия требуется пользовательское соединение, то класс должен дополнительно реализовать интерфейс `IUserConnectionRequired`.

## Интерфейс `IUserConnectionRequired`

```
namespace Terrasoft.Core
{
    public interface IUserConnectionRequired
    {
        void SetUserConnection(UserConnection userConnection);
    }
}
```

```
}
}
```

В классе, который реализует интерфейсы `IBackgroundTask<in TParameters>` и `IUserConnectionRequired` необходимо реализовать методы интерфейсов `Run` и `SetUserConnection`.

При **реализации методов** следует учесть:

- В метод `Run` не следует передавать `UserConnection`.
- В методе `Run` не следует вызывать метод `SetUserConnection` — ядро системы вызывает этот метод и инициализирует `UserConnection` при старте фоновой операции.
- В метод `Run` допустимо передавать структуры, состоящие только из простых типов данных. Если передавать сложные экземпляры классов, то с большой вероятностью произойдет ошибка сериализации параметров.

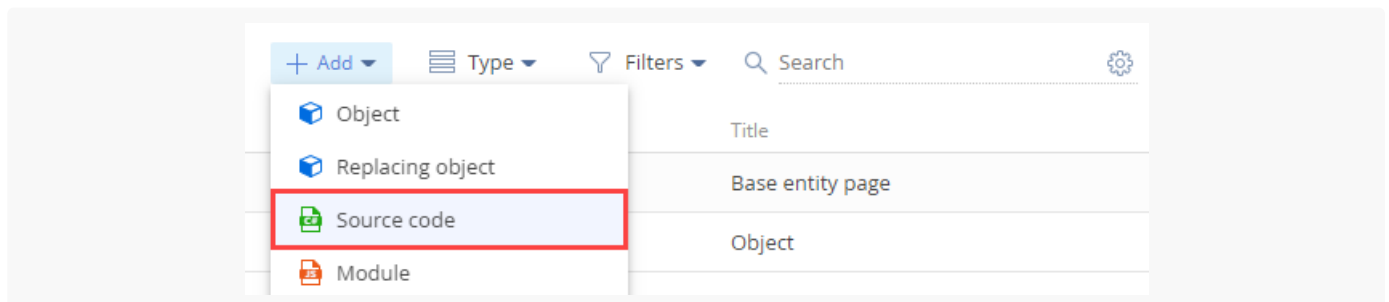
# Зарегистрировать фоновую операцию

 Средний

**Пример.** Создать бизнес-процесс, который регистрирует фоновую операцию. Фоновая операция выполняется около 30 секунд. По истечении этого времени в списочном представлении реестра раздела [ *Активности* ] ([ *Activities* ]) добавляется запись [ *Activity created by background task* ].

## 1. Создать класс для объекта активности

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Исходный код* ] ([ *Add* ] —> [ *Source code* ]).



3. В дизайнера схем заполните свойства схемы:

- [ *Код* ] ([ *Code* ]) — "UsrActivityData".
- [ *Заголовок* ] ([ *Title* ]) — "ActivityData".

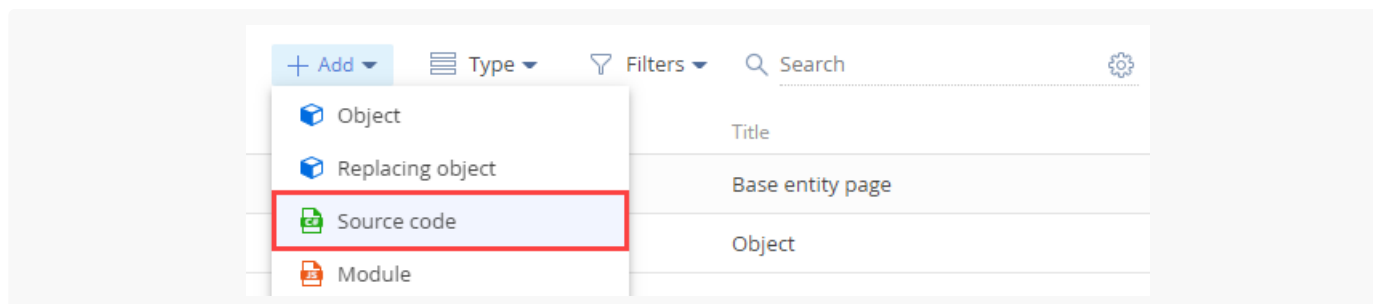
Для применения заданных свойств нажмите [ Применить ] ([ Apply ]).

4. В дизайнере схем добавьте исходный код.

5. На панели инструментов дизайнера нажмите [ Сохранить ] ([ Save ]), а затем [ Опубликовать ] ([ Publish ]).

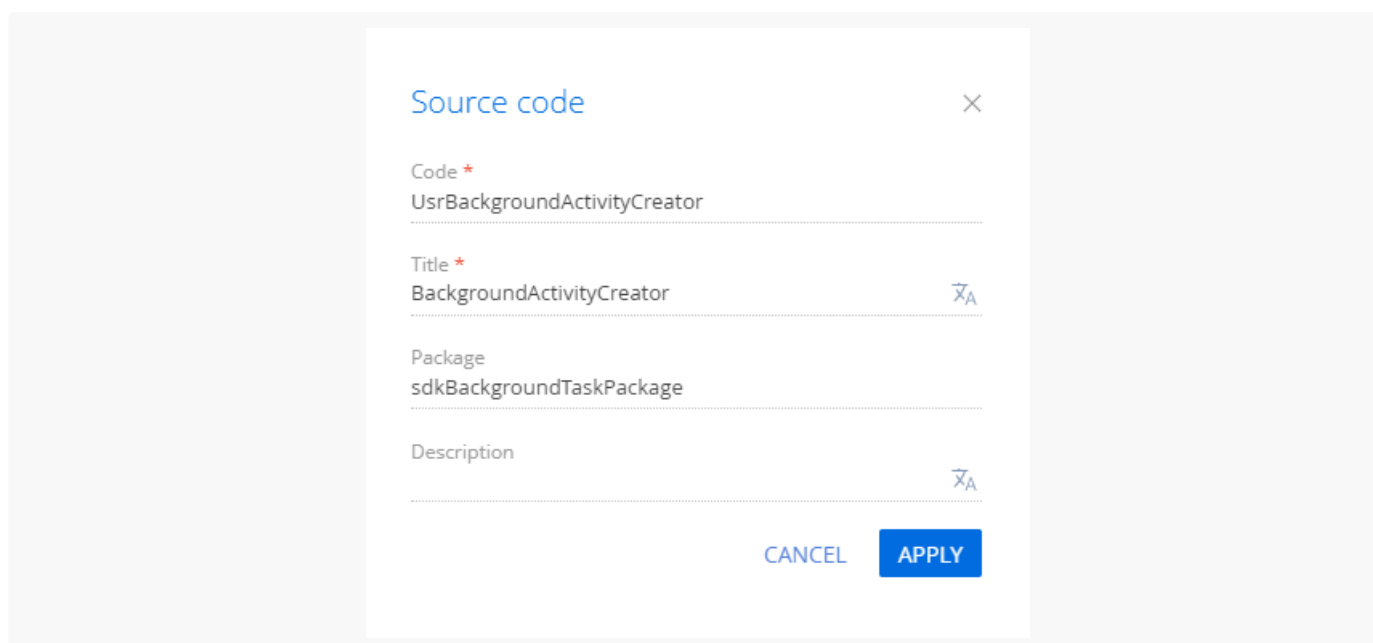
## 2. Создать класс для добавления активности

1. Перейдите в раздел [ Конфигурация ] ([ Configuration ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ Добавить ] —> [ Исходный код ] ([ Add ] —> [ Source code ]).



3. В дизайнере схем заполните свойства схемы:

- [ Код ] ([ Code ]) — "UsrBackgroundActivityCreator".
- [ Заголовок ] ([ Title ]) — "BackgroundActivityCreator".



Для применения заданных свойств нажмите [ Применить ] ([ Apply ]).

4. В дизайнере схем добавьте исходный код.

#### UsrBackgroundActivityCreator

```
namespace Terrasoft.Configuration
{
    using System;
    using Terrasoft.Common;
    using Terrasoft.Core;
    using Terrasoft.Core.DB;
    using Terrasoft.Core.Tasks;
    using System.Threading.Tasks;

    public class UsrBackgroundActivityCreator : IBackgroundTask<UsrActivityData>, IUserConnec
    {
```

```

private UserConnection _userConnection;

/* Implement the Run method of the IBackgroundTask interface. */
public void Run(UsrActivityData data) {
    /* Forced 30-second delay. */
    System.Threading.Tasks.Task.Delay(TimeSpan.FromSeconds(30));
    /* Creating activity. */
    var activity = new Activity(_userConnection){
        UseAdminRights = false,
        Id = Guid.NewGuid(),
        TypeId = data.TypeId,
        Title = data.Title,

        /* Activity category is "To do". */
        ActivityCategoryId = new Guid("F51C4643-58E6-DF11-971B-001D60E938C6")
    };
    activity.SetDefColumnValues();
    activity.Save(false);
}

/* Implement the SetUserConnection method of the IUserConnectionRequired interface. */
public void SetUserConnection(UserConnection userConnection) {
    _userConnection = userConnection;
}
}
}

```

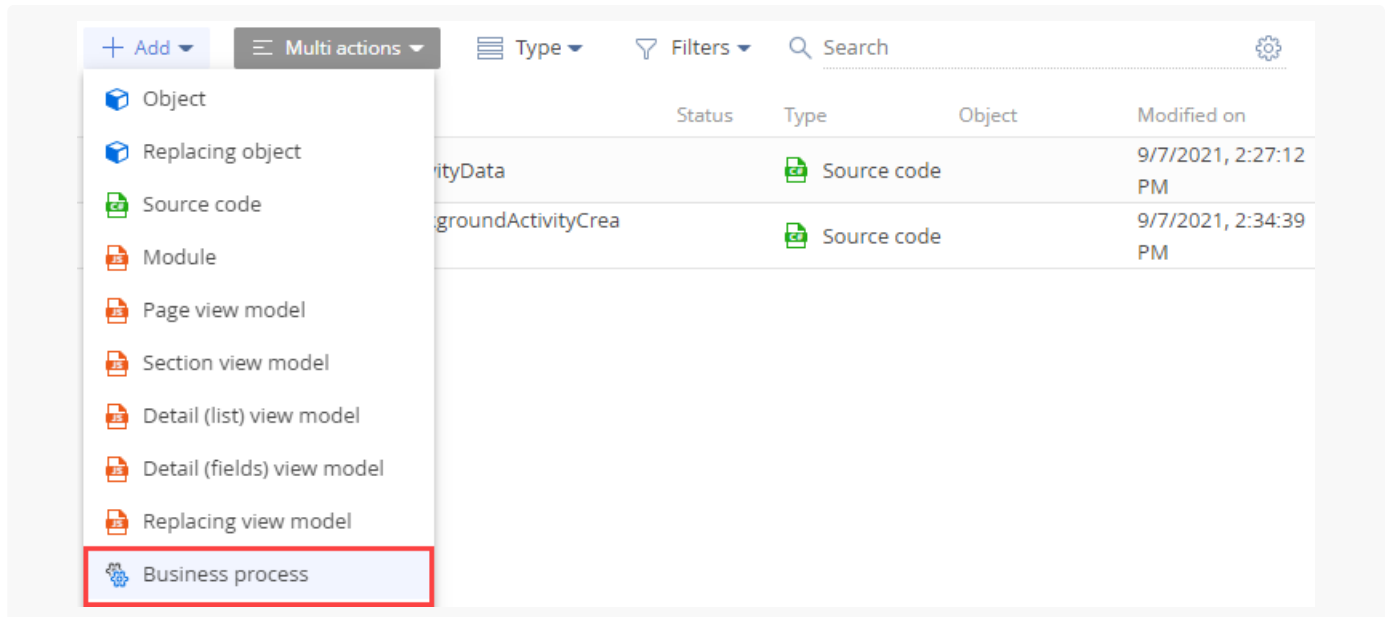
Класс `UsrBackgroundActivityCreator` реализует интерфейсы `IBackgroundTask<UsrActivityData>` и `IUserConnectionRequired`. В методе `Run()` после принудительной задержки в 30 секунд на основе предоставленного экземпляра `UsrActivityData` создается экземпляр объекта раздела [ *Активности* ] ([ *Activities* ]).

5. На панели инструментов дизайнера нажмите [ *Сохранить* ] ([ *Save* ]), а затем [ *Опубликовать* ] ([ *Publish* ]).

### 3. Создать бизнес-процесс для запуска фоновой операции

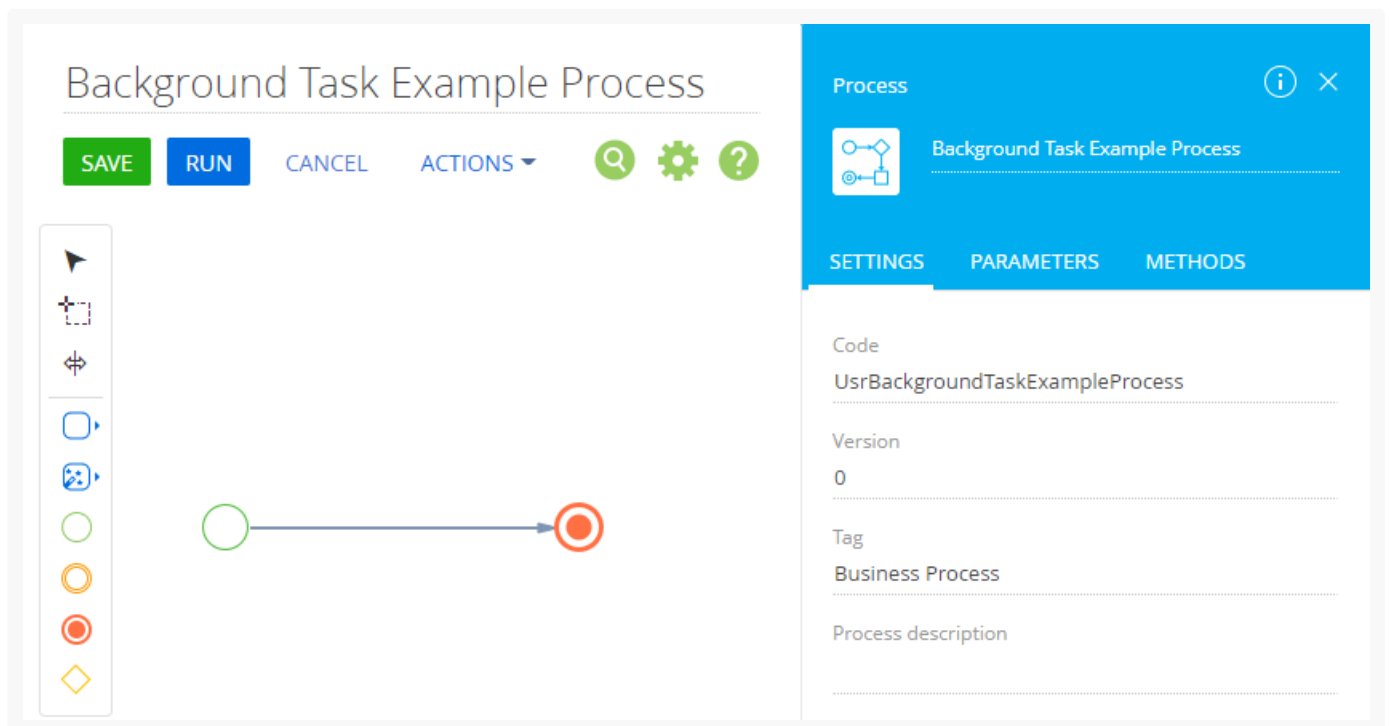
1. [Перейдите в раздел \[ \*Конфигурация\* \]](#) ([ *Configuration* ]) и выберите пользовательский [пакет](#), в который будет добавлена схема.
2. На панели инструментов реестра раздела нажмите [ *Добавить* ] —> [ *Бизнес процесс* ] ([ *Add* ] —> [ *Business process* ]).





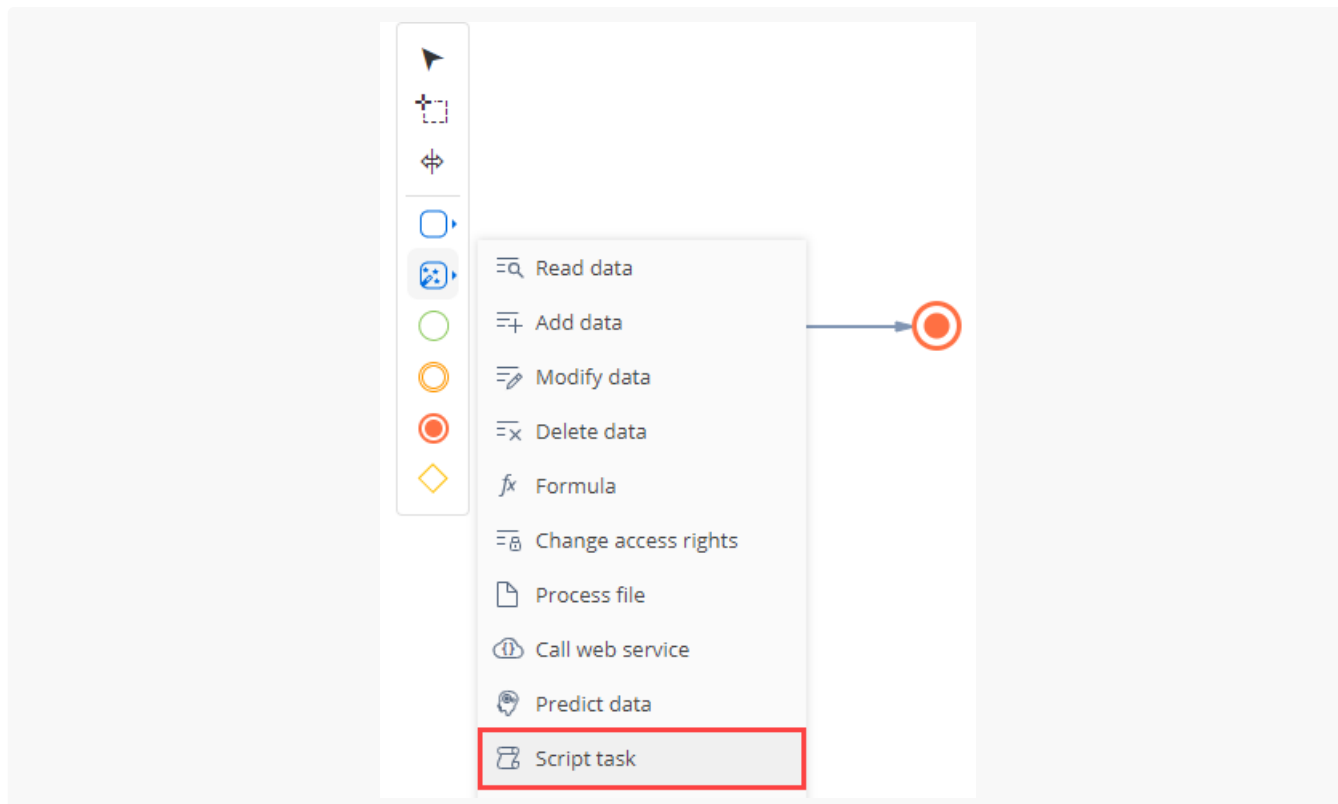
3. В дизайнере процессов заполните свойства процесса:

- На панели настройки элементов заполните свойство [ *Заголовок* ] ([ *Title* ]) — "Background Task Example Process".
- На вкладке [ *Настройки* ] ([ *Settings* ]) панели настройки элементов заполните свойство [ *Имя* ] ([ *Code* ]) — "UsrBackgroundTaskExampleProcess".

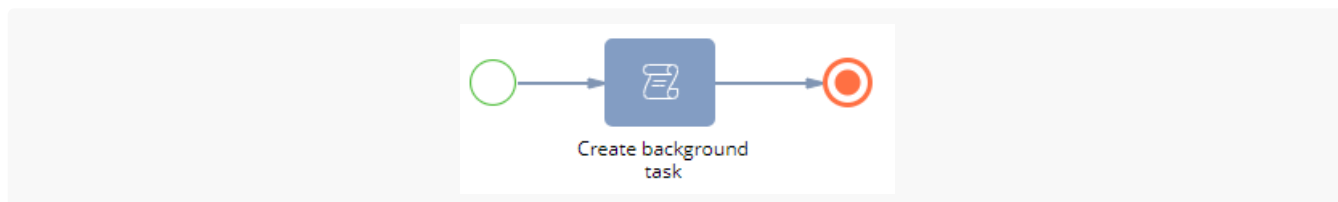


4. Реализуйте бизнес-процесс.

- В области элементов дизайнера нажмите [ *Действия системы* ] ([ *System actions* ]) и разместите элемент [ *Задание-сценарий* ] ([ *Script task* ]) в рабочей области дизайнера процессов между начальным событием [ *Простое* ] ([ *Simple* ]) и завершающим событием [ *Останов* ] ([ *Terminate* ]).




- b. Элементу [ *Задание-сценарий* ] ([ *Script task* ]) добавьте имя "Создать фоновую операцию" ("Create background task").

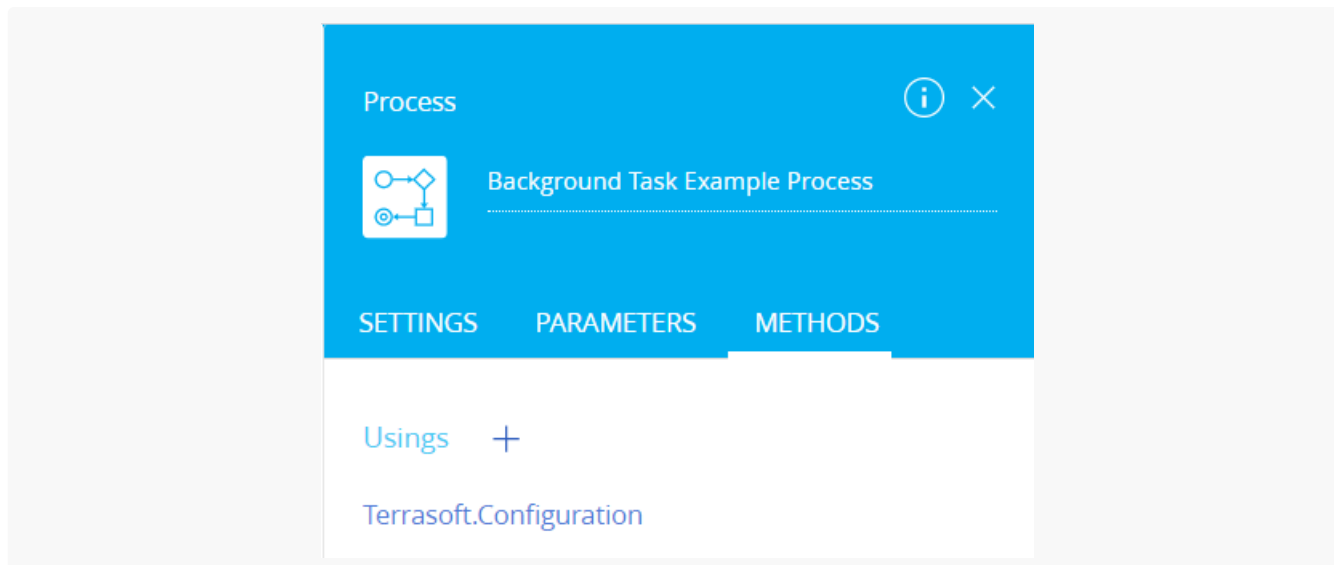


- c. Добавьте код элемента [ *Задание-сценарий* ] ([ *Script task* ]).

#### Код элемента [ *Задание-сценарий* ] ([ *Script task* ])

```
var data = new UsrActivityData {
    Title = "Activity created by background task",
    TypeId = ActivityConsts.TaskTypeUid
};
Terrasoft.Core.Tasks.Task.StartNewWithUserConnection<UsrBackgroundActivityCreator, UsrActivi>
return true;
```

- d. В дизайнера процессов на вкладке [ *Методы* ] ([ *Methods* ]) в блоке [ *Usings* ] нажмите кнопку  и добавьте пространство имен `Terrasoft.Configuration`. Это необходимо для использования в бизнес-процессе реализации [класса для объекта активности](#) и [класса для добавления активности](#).



5. На панели инструментов дизайнера нажмите [ Сохранить ] ([ Save ]).
6. Во всплывающем окне нажмите [ Опубликовать ] ([ Publish ]) для компиляции кода элемента [ Задание-сценарий ] ([ Script task ]).

## Результат выполнения примера

Чтобы **запустить бизнес-процесс** `Background Task Example Process`, на панели инструментов дизайнера процессов нажмите [ Запустить ] ([ Run ]).

В результате выполнения бизнес-процесса `Background Task Example Process` в списочном представлении реестра раздела [ Активности ] ([ Activities ]) добавляется запись [ *Activity created by background task* ].

