

# Управление поставками

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>Понятие рабочей среды</b>	<b>5</b>
Среда разработки	5
Предпромышленная рабочая среда	8
Промышленная рабочая среда	8
<b>Процесс управления поставками</b>	<b>8</b>
1. Разработать новую функциональность	9
2. Экспортировать пакет в *.zip-архив	9
3. Импортировать пакет в предпромышленную среду	10
4. Создать резервную копию базы данных промышленной среды	10
5. Импортировать пакет в промышленную среду	10
<b>Управление поставками в Creatio IDE</b>	<b>10</b>
Перенос пакета	11
Перенос пакета с использованием SVN	11
Перенос схемы конфигурационного элемента	12
<b>Перенести пакеты</b>	<b>12</b>
Экспортировать пакет	12
Импортировать пакет	14
<b>Перенести пакеты с использованием SVN</b>	<b>17</b>
1. Подключить хранилище системы контроля версий SVN	17
2. Включить автоматическое применение изменений	18
3. Проверить привязки данных	19
4. Проверить возможность переноса зависимостей пакета	19
5. Установить пакет из хранилища SVN	19
<b>Перенести схемы конфигурационных элементов</b>	<b>19</b>
Экспортировать схему конфигурационного элемента	19
Импортировать схему конфигурационного элемента	20
<b>Управление поставками в WorkspaceConsole</b>	<b>21</b>
Настройка утилиты WorkspaceConsole	22
Использование утилиты WorkspaceConsole	23
<b>Экспортировать пакеты из базы данных</b>	<b>32</b>
1. Сформировать команду для экспорта пакета из базы данных	33
2. Экспортировать пакет из базы данных	33
<b>Экспортировать пакет из SVN</b>	<b>34</b>
1. Сформировать команду для экспорта пакета из SVN-хранилища	35
2. Экспортировать пакет из SVN	35
<b>Импортировать пакет в базу данных</b>	<b>36</b>

1. Сформировать команду для импорта пакета в базу данных	37
2. Импортировать пакет в базу данных	37
3. Сформировать команду для генерации статического контента в файловую систему	38
4. Сгенерировать статический контент в файловую систему	38
<b>Параметры утилиты WorkspaceConsole</b>	<b>39</b>

# Понятие рабочей среды

## Основы

**Рабочая среда** представляет собой отдельное приложение Creatio со своей базой данных. Рабочая среда может быть дополнена системой контроля версий. **Назначение** рабочих сред — обеспечение поставок новой функциональности на разных этапах ее жизненного цикла: разработка, тестирование и использование.

**Виды** рабочих сред:

- Среда разработки.
- Предпромышленная рабочая среда.
- Промышленная рабочая среда.

## Среда разработки

**Среда разработки** (Development Environment) — отдельное приложение или несколько приложений Creatio, в которых выполняется разработка новой функциональности.

Для фиксации изменений рекомендуется дополнить среду разработки [системой контроля версий](#).

**Важно.** Систему контроля версий SVN разрешено использовать для переноса изменений только между [средами разработки](#). Запрещено использовать SVN на [предпромышленной](#) и [промышленной](#) среде, поскольку это может привести к неработоспособности или ухудшению производительности приложения. Подробнее читайте в статье [Контроль версий в Subversion](#).

Используйте систему контроля версий [Git](#), если:

- планируется вести разработку в **файловой системе**;
- используется **on-site** приложение.

Используйте систему контроля версий [Subversion \(SVN\)](#), если:

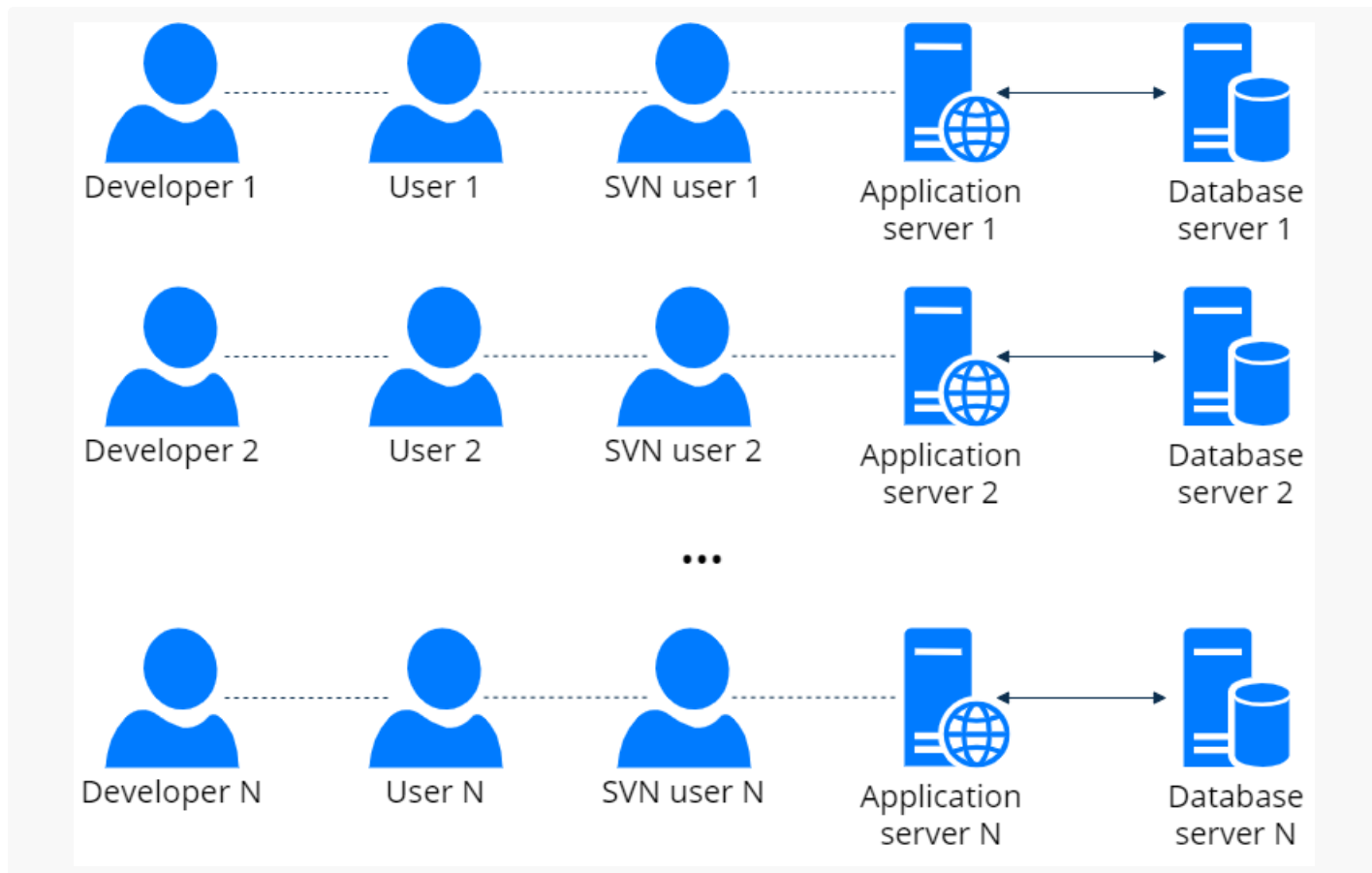
- планируется вести разработку **low-code** инструментами;
- используется **cloud** приложение.

Использование системы контроля версий SVN доступно только для приложения Creatio на платформе **.NET Framework**.

Настройка систем контроля версий описана в разделе [Системы контроля версий](#).

Для среды разработки рекомендуется выбрать [on-site развертывание](#).

При **on-site развертывании** для каждого разработчика разворачивается on-site приложение и база данных.



Варианты **cloud развёртывания** приложения:

1. Воспользоваться [страницей создания пробной версии](#). На протяжении 14-дневного пробного периода можно ознакомиться с основными возможностями приложения. По завершению пробного периода используемая демоверсия приложения может быть перенесена на основную площадку Creatio.
2. Обратиться к менеджеру по продажам Creatio о необходимости развернуть новое приложение в облаке или перенести приложение клиента в облачный сервис Creatio. После согласования условий с менеджером соответствующее подразделение компании выполнит необходимые работы.

При создании облачных приложений Creatio на площадке Terrasoft необходимо учитывать некоторые ограничения. Если эти требования не будут выполнены, продукт не может быть развернут.

#### 1. Запрещено использование SQL Agent

Нельзя создавать задания (Jobs) и другие действия, выполняемые SQL Agent. Вместо этого необходимо использовать [планировщик заданий Creatio](#).

#### 2. Запрещено использование DB Mail

Отправку Email-уведомлений необходимо делать с использованием возможностей платформы Creatio.

#### 3. Запрещено использование Extended Stored Procedure

Всю необходимую логику необходимо реализовывать или с использованием стандартных хранимых процедур на языке T-SQL, или с использованием возможностей сервера приложений.

#### 4. Запрещена привязка к именам пользователей СУБД

В СУБД на площадке Terrasoft не создаются пользователи базы данных. Вместо этого используются

доменные пользователи и доменная аутентификация.

5. Запрещено изменение файла `web.config` приложения

Все необходимые параметры необходимо хранить в системных настройках приложения Creatio.

6. Запрещена привязка к IP-адресам серверов приложений и СУБД

IP-адреса серверов могут меняться. Поэтому привязываться к ним нельзя. Необходимо всегда работать с доменными именами приложений.

7. Запрещена установка дополнительного ПО

Никакое дополнительное программное обеспечение не может быть установлено на серверах облачного сервиса Terrasoft.

8. Запрещена работа с файловой системой

Работа с файловой системой для сервера приложений и СУБД ограничена правами доступа, которые настроены в операционной системе. Вместо этого необходимо работать с файлами с использованием протоколов FTP и HTTP(S).

9. Запрещен запуск сторонних приложений на сервере

Возможность запуска сторонних приложений ограничена правами доступа, которые настроены в операционной системе. Вся необходимая логика должна быть реализована в приложении.

10. База данных должна работать на SQL Server 2016 и выше.

Для обеспечения совместимости с облачной инфраструктурой площадки Terrasoft предоставляемая база данных приложения должна быть создана в SQL Server 2016 и выше.

11. Приложение должно работать как по протоколу HTTP, так и HTTPS

Нельзя использовать логику, связанную с использованием конкретного протокола. Вместо этого необходимо определять текущий протокол приложения.

12. Приложение должно работать с правами обычного пользователя

Нельзя использовать функции, требующие административных привилегий.

13. Приложение должно работать от имени пользователя без профиля

На площадке создаются пользователи, не имеющие возможности фактического логина в ОС и не имеющие профиля.

### Дополнительные рекомендации:

- В качестве значения системной настройки [ *Издатель* ] (код [ *Maintainer* ]) необходимо установить название партнера, например FineSolution.
- Значение системной настройки [ *Префикс названия объекта* ] (код [ *SchemaNamePrefix* ]) должно характеризовать партнера, например, FS.
- Решение не должно использовать замещение модулей. Замещать можно только схемы.
- Серверная логика должна быть сосредоточена в C# классах и вызываться в нужных местах.
- Public API серверных классов и клиентских схем должен быть покрыт unit-тестами.
- Все необходимые данные, скрипты, библиотеки должны быть прикреплены к пакету.
- Разработка продукта должна вестись с использованием системы контроля версий и все пакеты должны храниться в системе контроля версий.

Для разработки сложных проектных решений можно воспользоваться рекомендациями, представленными в документации [Обзор методологии Project Life Cycle](#).

## Предпромышленная рабочая среда

**Предпромышленная рабочая среда** (Pre-Production Environment) — отдельное приложение, в котором выполняется тестирование функциональности, разработанной в среде разработки. Как правило, тестирование выполняется аналитиком группы разработки или заказчиком функциональности. Для предпромышленной среды разработки можно выбрать [on-site](#) или cloud развертывание приложения.

## Промышленная рабочая среда

**Промышленная рабочая среда** (Production Environment) — отдельное приложение Creatio, которое используется в повседневной работе. **База данных** промышленной среды идентична базе данных предпромышленной среды. Поскольку разработка практически всегда сопряжена с возникновением ошибок, их обнаружением, отладкой приложения, компиляцией и т. д., то запрещается вести разработку в промышленной среде.

Для промышленной среды разработки можно выбрать [on-site](#) или cloud развертывание приложения:

- При выборе варианта развертывания **cloud** приложения настройка промышленной среды не отличается от настройки среды разработки.
- При выборе варианта развертывания **on-site** приложения настройка промышленной среды не отличается от настройки предпромышленной среды.

## Процесс управления поставками

### Основы

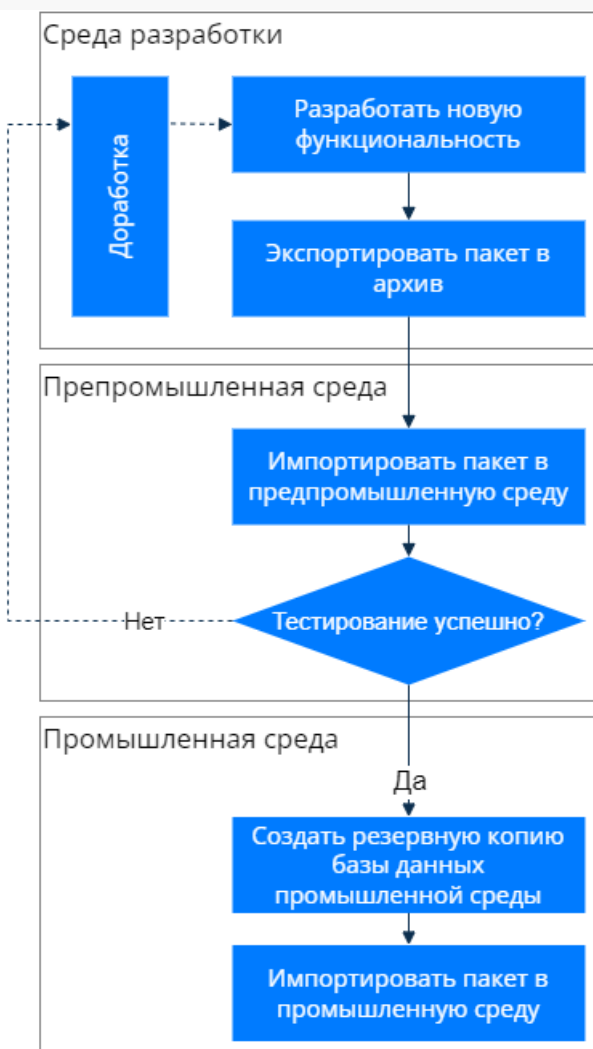
**Рабочие среды**, которые задействованы в процессе разработки и внедрения новой функциональности:

- Среда разработки.
- Предпромышленная среда.
- Промышленная среда.

Подробнее о рабочих средах читайте в статье [Понятие рабочей среды](#).

Чтобы избежать нарушений в работе системы и критических ошибок на промышленной среде, при переносе функциональности между рабочими средами необходимо придерживаться определенной последовательности действий, которая приведена на рисунке ниже.





## 1. Разработать новую функциональность

Разработку новой функциональности рекомендуется выполнять в [среде разработки](#) с персональной базой данных для каждого разработчика. Для переноса изменений между средами разработки рекомендуется использовать [систему контроля версий](#) ([Subversion](#), [Git](#) и т. д.).

**Важно.** Для переноса изменений в промышленную среду нельзя использовать SVN. Перенос изменений с помощью SVN следует использовать только для сред разработки.

## 2. Экспортировать пакет в \*.zip-архив

**Способы** экспорта пакета в \*.zip-архив:

- Из раздела [ *Конфигурация* ] ([ *Configuration* ]). Для этого воспользуйтесь инструкцией, которая приведена в статье [Перенести пакеты](#).
- С помощью утилиты WorkspaceConsole. Для этого воспользуйтесь инструкцией, которая приведена в статье [Управление поставками в WorkspaceConsole](#).

### 3. Импортировать пакет в предпромышленную среду

**Способы** импорта пакета в приложение:

- Из пользовательского интерфейса приложения. Удобен, если предпромышленная среда размещена в облаке. Для этого воспользуйтесь инструкцией, которая приведена в статье [Управление поставками в Creatio IDE](#).
- С помощью утилиты WorkspaceConsole. Удобен, если используются процессы непрерывной интеграции и предпромышленная среда размещена on-site. Для этого воспользуйтесь инструкцией, которая приведена в статье [Управление поставками в WorkspaceConsole](#).

**Важно.** Для переноса изменений в приложение, которое размещено в облаке, рекомендуется использовать возможности пользовательского интерфейса Creatio. Использование WorkspaceConsole невозможно, поскольку у пользователя нет непосредственного доступа к базе данных облачного приложения.

Импорт пакета отличается для рабочей среды с балансировщиком нагрузки. Чтобы **импортировать пакет на среду с балансировщиком**, воспользуйтесь инструкцией, которая приведена в статье [Установить приложение Marketplace](#).

В случае возникновения ошибок при тестировании разработанной функциональности, выполните ее доработку, устранив все ошибки. Затем повторить шаги 1—3.

### 4. Создать резервную копию базы данных промышленной среды

Перед поставкой пакетов с разработанной функциональностью в приложение промышленной среды, выполните резервное копирование базы данных. Для этого воспользуйтесь инструкцией, которая приведена в [инструкции по обновлению on-site](#). Этот шаг является обязательным, поскольку существует вероятность, что функциональность, разработанная разными сторонними разработчиками, может влиять на общую работоспособность приложения.

**Важно.** Чтобы создать резервную копию базы данных приложения, которое размещено в облаке, обратитесь в службу поддержки. При размещении приложения on-site резервная копия базы данных создается клиентом самостоятельно.

### 5. Импортировать пакет в промышленную среду

Способы импорта пакетов в промышленную среду аналогичны предпромышленной среде (шаг 3).

## Управление поставками в Creatio IDE



Легкий

Creatio предоставляет различные инструменты поставок функциональности.

**Инструменты управления поставками**, которые предоставляет Creatio:

- Creatio IDE.
- Утилита WorkspaceConsole.

В этой статье будет рассмотрено управление поставками с использованием Creatio IDE.

Creatio IDE предоставляет различные способы переноса функциональности в зависимости от этапа [процесса поставок функциональности](#).

**Использование** Creatio IDE в качестве инструмента переноса решений позволяет:

- Переносить пакеты в виде архивов — на этапе переноса функциональности между рабочими средами.
- Переносить пакеты с использованием системы контроля версий SVN — на этапе разработки новой функциональности или в процессе доработок.
- Переносить схемы конфигурационных элементов — на этапе разработки новой функциональности или в процессе доработок.

## Перенос пакета

**Перенос пользовательского пакета** — это процесс экспорта разработанной функциональности, представленной в виде пакета, из одной рабочей среды и импорта в другую рабочую среду. При многопользовательской разработке также можно выполнять перенос между средами разработки.

Экспорт пакетов выполняется в разделе [ Конфигурация ] ([ *Configuration* ]) дизайнера системы. Пакет скачивается в виде \*.zip-архива, который содержит \*.gz-архив пакета.

Импорт пакетов выполняется в разделе [ Установка и удаление приложений ] ([ *Installed applications* ]) дизайнера системы. Для выполнения импорта пакетов воспользуйтесь инструкцией [Перенос первого приложения](#).

**Важно.** Пакет `Custom` нельзя переносить между приложениями.

Также \*.gz-архив пакета можно [выгрузить из базы данных](#) или [из хранилища SVN](#) с помощью [утилиты WorkspaceConsole](#).

Не допускается создавать пакеты в промышленной среде, затем на основе промышленной среды создавать среду разработки, дорабатывать функциональность пакетов и переносить их обратно на промышленную среду. Подробнее о рекомендуемой последовательности разработки описано в статье [Процесс управления поставками](#).

## Перенос пакета с использованием SVN

Система управления версиями является опциональным компонентом приложения Creatio. Используется когда параллельно с эксплуатацией системы необходимо на платформе организовать разработку пользовательской конфигурации. Для **многопользовательской разработки** применение системы контроля версий является обязательным.

Creatio IDE настроена на работу с [системой контроля версий SVN](#). Одной из функций SVN является перенос изменений между средами разработки.

### Особенности использования SVN для переноса изменений:

- Возможность переноса между средами разработки пакетов и схем конфигурационных элементов.
- Возможность переноса привязанных к пакету данных, например, наполнение справочников, новые системные настройки, демонстрационные записи раздела и т. д.
- Автоматическая установка из SVN пакетов-зависимостей.
- Независимость от службы поддержки при переносе изменений для on-site приложений.

**Важно.** Систему контроля версий SVN разрешено использовать для переноса изменений только между [средами разработки](#). Запрещено использовать SVN на [предпромышленной](#) и [промышленной](#) среде, поскольку это может привести к неработоспособности или ухудшению производительности приложения. Подробнее читайте в статье [Контроль версий в Subversion](#).

Перед обновлением приложения с помощью SVN необходимо выполнить резервное копирование базы данных. Если приложение развернуто в облаке, то необходимо обратиться в службу поддержки.

## Перенос схемы конфигурационного элемента

**Перенос схем** — это процесс экспорта схемы из среды разработки и импорта в предпромышленную среду. При многопользовательской разработке также можно выполнять перенос между средами разработки.

Перенос схем конфигурационных элементов допускается использовать при:

- Передаче незавершенной разрабатываемой схемы от одного разработчика другому, поскольку заливка незавершенной функциональности в систему контроля версий является нецелесообразной.
- Сохранении промежуточных результатов разработки, если система контроля версий не используется, или при возникновении проблем с системой контроля версий.
- Быстром переносе схемы между средами разработки.

### Особенности использования механизма переноса схем конфигурационных элементов:

- Возможность заменить содержимое одной схемы.
- Механизм позволяет экспортировать и импортировать только схемы. Для переноса пакетов необходимо использовать процедуру, описанную в статье [Перенести пакеты](#).
- Требуются права доступа администратора в приложении.
- Нет возможности загрузить несколько схем одновременно.

## Перенести пакеты



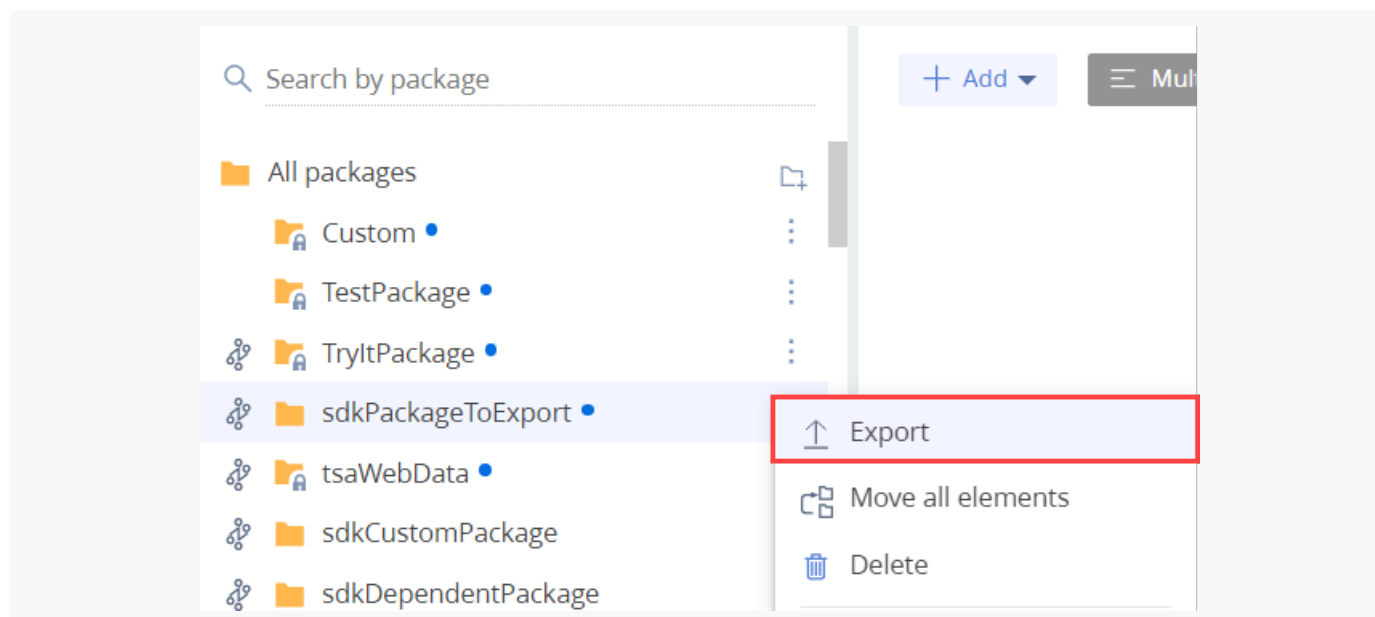
Легкий

### Экспортировать пакет

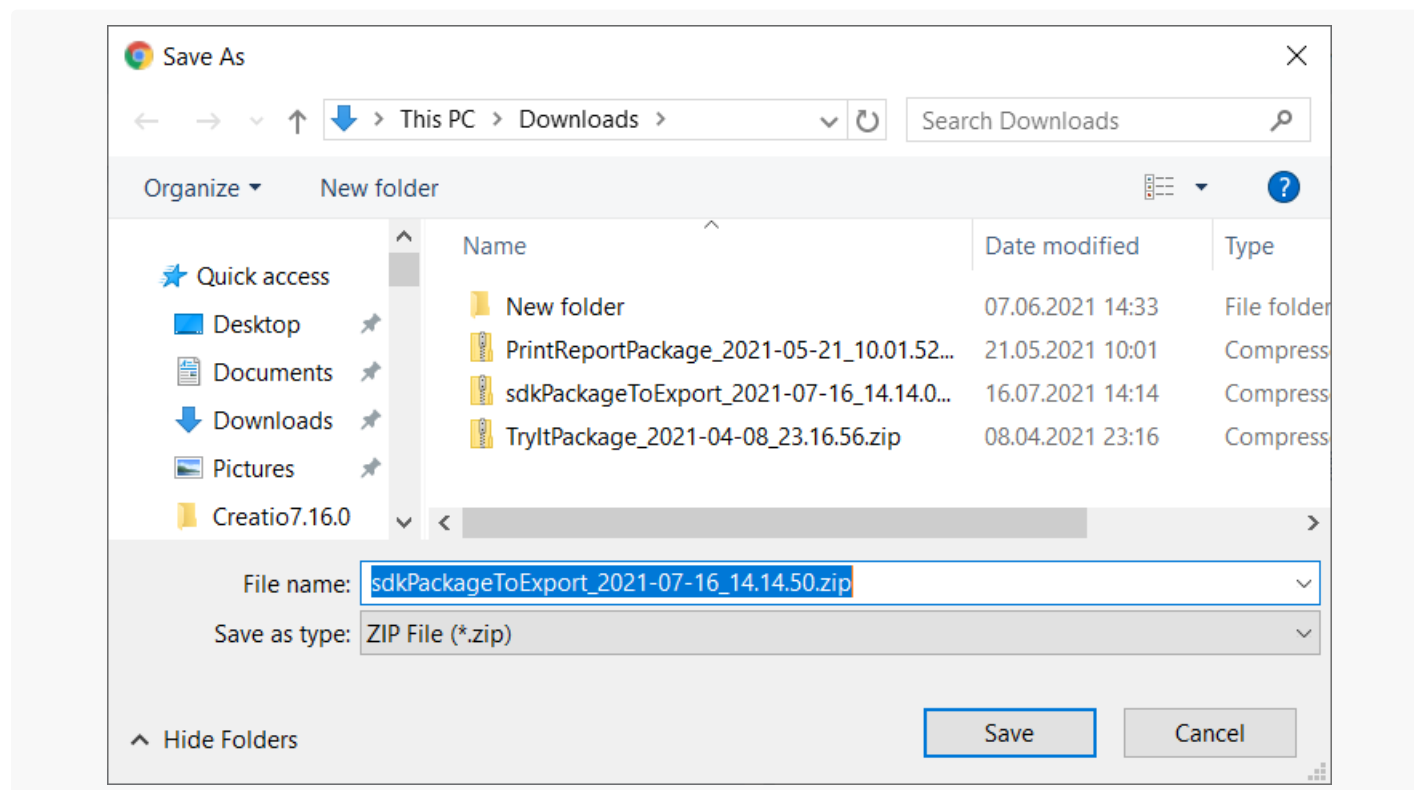
1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]) и выберите пользовательский [пакет](#), который

необходимо экспортировать.

2. В меню свойств пакета нажмите [ Экспортировать ] ([ Export ]).




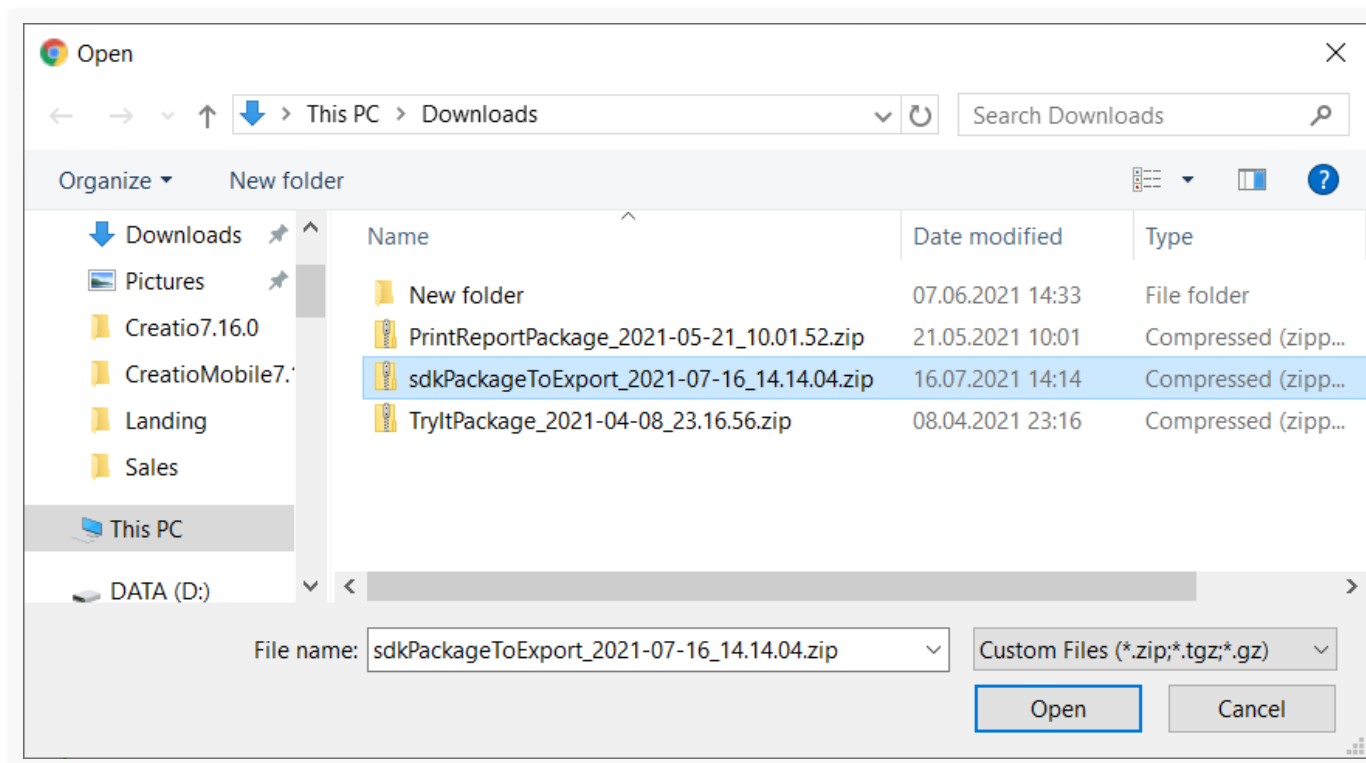
В зависимости от настроек браузера \*.zip-архив с \*.gz-архивом пакета будет выгружен в каталог по умолчанию или браузер отобразит диалоговое окно для выбора каталога, в который необходимо сохранить архив.



После этого пакет можно импортировать в другую рабочую среду.

## Импортировать пакет

1. Перейдите в дизайнер системы по кнопке .
2. В блоке [ Приложения ] ([ Applications ]) перейдите по ссылке [ Установка и удаление приложений ] ([ Installed applications ]).
3. На странице мастера установки приложений Marketplace нажмите [ Добавить приложение ] —> [ Установить из файла ] ([ Add application ] —> [ Install from file ]).
4. Нажмите [ Выбрать файл ] ([ Select file ]) и укажите путь к \*.zip-архиву, который содержит \*.gz-архив пакета.




После этого выполнится создание резервной копии базы данных приложения. Процесс может занять несколько минут.

После успешной установки пакета в приложение Creatio отобразится соответствующее сообщение.

## Application installation

CLOSE



Application installed successfully.

You can close the page and continue working with the Creatio.

[DOWNLOAD INSTALLATION LOG](#)

### Attention!


Ensure that the application you are installing is compatible with your version of Creatio. Description of the Marketplace application contains a list of compatible Creatio products.

The vendor guarantees the compatibility of these products. Do not install an application if your Creatio version is not present in the list.

If you install a Marketplace application to become familiar with its capabilities, it is better to install it on your sandbox version of the system (support service can provide it) or install it on 14-days trial version which is available on Creatio website.

Only after that, we recommend to install the product on your Creatio production environment.

Если установка пакета выполнена неуспешно также отобразится соответствующее сообщение.

Application installation


CLOSE

Application installation failed.  
Restore configuration from backup?

You can close this page and continue with the Creatio.

RESTORE PACKAGES FROM BACKUP

DOWNLOAD INSTALLATION LOG

Attention!

Ensure that the application you are installing is compatible with your version of Creatio. Description of the Marketplace application contains a list of compatible Creatio products.

The vendor guarantees the compatibility of these products. Do not install an application if your Creatio version is not present in the list.


If you install a Marketplace application to become familiar with its capabilities, it is better to install it on your sandbox version of the system (support service can provide it) or install it on 14-days trial version which is available on Creatio website.

Only after that, we recommend to install the product on your Creatio production environment.

Чтобы посмотреть причину неуспешной установки пакета, нажмите [ *Скачать лог установки* ] ([ *Download installation log* ]).

Чтобы восстановить предыдущее состояние конфигурации, нажмите [ *Восстановить пакеты из резервной копии* ] ([ *Restore packages from backup* ]). После восстановления резервной копии отобразится соответствующее сообщение.



Application installation


CLOSE

The configuration has been successfully restored from backup.  
You can close this page and continue with the Creatio.

DOWNLOAD INSTALLATION LOG

### Attention!

Ensure that the application you are installing is compatible with your version of Creatio. Description of the Marketplace application contains a list of compatible Creatio products.

The vendor guarantees the compatibility of these products. Do not install an application if your Creatio version is not present in the list.

If you install a Marketplace application to become familiar with its capabilities, it is better to install it on your sandbox version of the system (support service can provide it) or install it on 14-days trial version which is available on Creatio website.

Only after that, we recommend to install the product on your Creatio production environment.

# Перенести пакеты с использованием SVN

 Легкий

**На заметку.** При включенном [режиме разработки в файловой системе](#) механизм интеграции с SVN отключен.

## 1. Подключить хранилище системы контроля версий SVN

Если к приложению подключено хранилище системы контроля версий SVN, переходите к пункту [2. Включить автоматическое применение изменений](#).

Чтобы **подключить хранилище системы контроля версий SVN** к приложению Creatio:

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]).
2. В меню действий в группе [ *Хранилища SVN* ] ([ SVN repositories ]) выберите [ *Открыть список хранилищ* ] ([ Open list of repositories ]).
3. Нажмите [ *Добавить* ] ([ Add ]) и заполните **свойства хранилища**:
  - [ *Название* ] ([ Name ]) — название хранилища (обязательное свойство).

- [ Адрес SVN хранилища ] ([ *SVN storage address* ]) — адрес хранилища SVN (обязательное свойство).
- [ Логин ] ([ *Login* ]) — логин пользователя хранилища SVN.
- [ Пароль ] ([ *Password* ]) — пароль пользователя хранилища SVN.

4. Нажмите [ Аутентифицироваться ] ([ *Authenticate* ]) и авторизируйтесь в хранилище SVN.

В результате к приложению будет подключено хранилище системы контроля версий SVN. Подробнее о работе с системой контроля версий SVN с использованием встроенных средств Creatio читайте в статье [Контроль версий в Creatio IDE](#).

## 2. Включить автоматическое применение изменений

Чтобы изменения применились после выполнения переноса, необходимо включить автоматическое применение изменений.

Чтобы **включить автоматическое применение изменений**, в файле `..\Terrasoft.WebApp\Web.config` установите значение `true` для ключей элемента `<appSettings>`:

- `AutoUpdateOnCommit` — ключ отвечает за обновление пакетов из хранилища SVN перед их заливкой. Если для этого ключа установлено значение `false`, то перед заливкой в хранилище SVN приложение предупредит пользователя о необходимости обновления, если схемы пакета были изменены.
- `AutoUpdateDBStructure` — ключ отвечает за обновление структуры базы данных.
- `AutoInstallSqlScript` — ключ отвечает за установку SQL-сценариев.
- `AutoInstallPackageData` — ключ отвечает за установку привязанных данных.

```
..\Terrasoft.WebApp\Web.config
```

```
<appSettings>
  ...
  <add key="AutoUpdateOnCommit" value="true" />
  <add key="AutoUpdateDBStructure" value="true" />
  <add key="AutoInstallSqlScript" value="true" />
  <add key="AutoInstallPackageData" value="true" />
</appSettings>
```

### 3. Проверить привязки данных

Перед переносом пакета необходимо проверить, что все данные, которые необходимо перенести, привязаны к пакету. При создании раздела с помощью мастера к пакету автоматически привязываются данные, необходимые для регистрации и корректной работы раздела.

Описание привязки данных к пакету содержится в статье [Общие принципы работы с пакетами](#).

### 4. Проверить возможность переноса зависимостей пакета

Во время разработки к пользовательскому пакету могут быть добавлены зависимости от других пакетов. Если пакеты-зависимости разработаны сторонними разработчиками, то необходимо проверить, что они уже установлены в приложение, в которое будет осуществляться перенос пользовательского пакета. Если пакеты-зависимости находятся в доступном хранилище SVN, то они будут установлены автоматически.

Описание зависимостей пакетов содержится в статье [Общие принципы работы с пакетами](#).

### 5. Установить пакет из хранилища SVN


Описание установки пакет из хранилища SVN содержится в статье [Контроль версий в Creatio IDE](#).

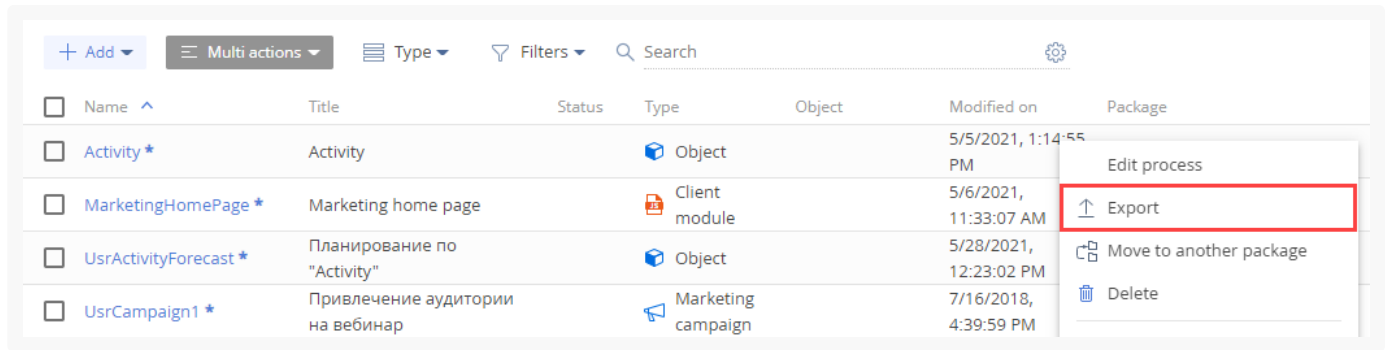
## Перенести схемы конфигурационных элементов



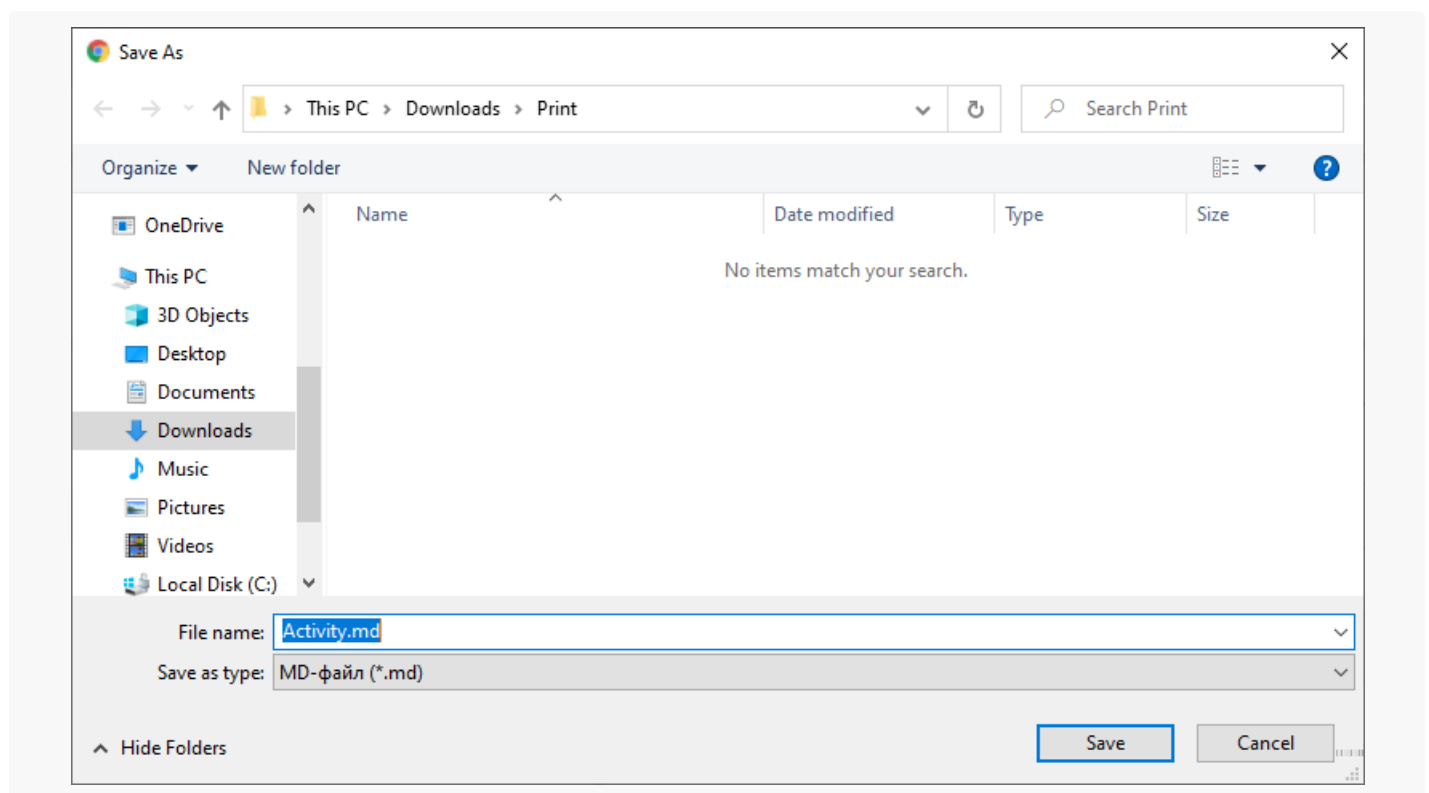
Легкий

### Экспортировать схему конфигурационного элемента

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]) и в пользовательском [пакете](#) выберите схему конфигурационного элемента, которую необходимо экспортировать.
2. В меню свойств конфигурационного элемента нажмите  —> [ Экспортировать ] ([ Export ]).

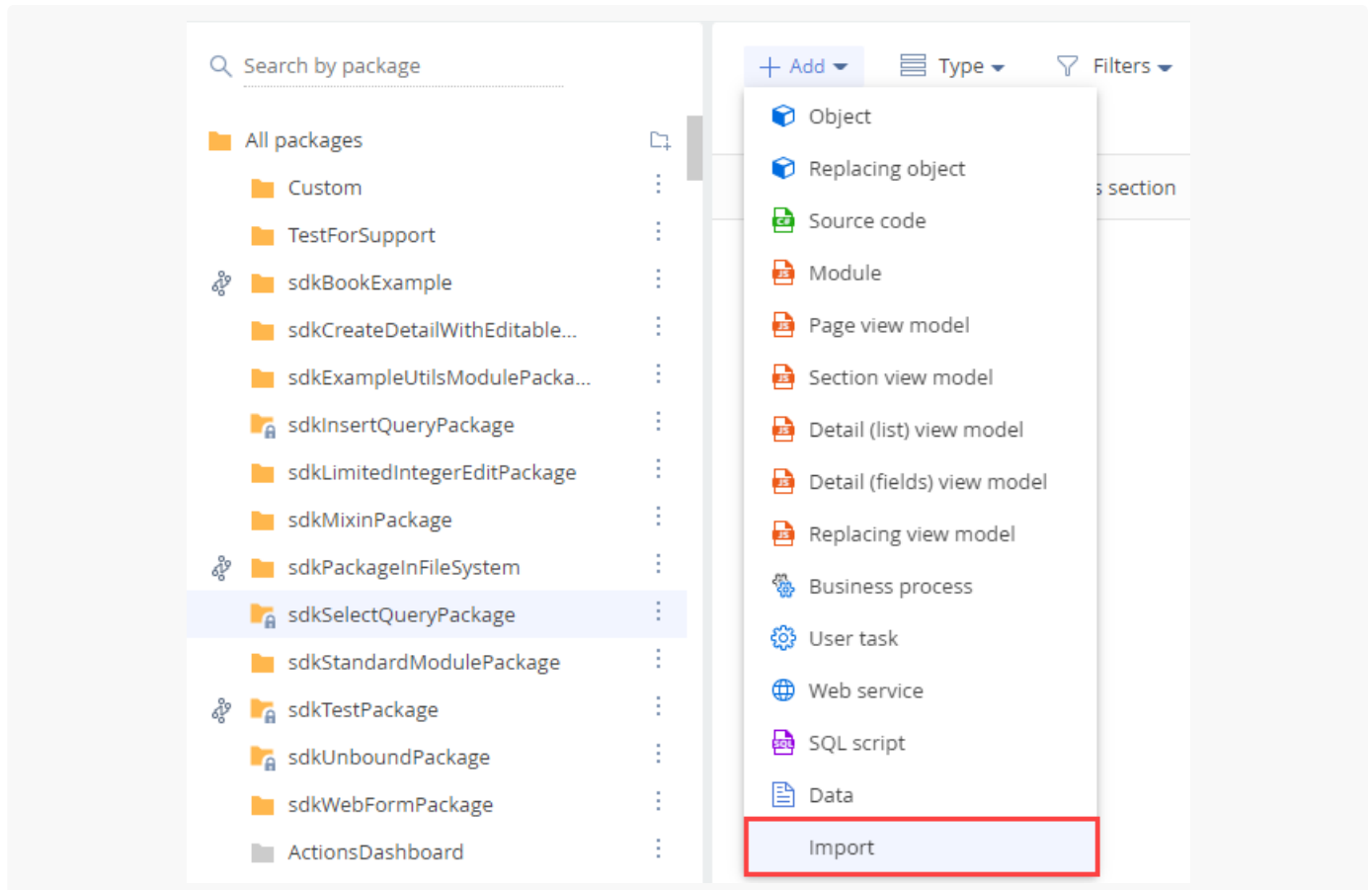


В зависимости от настроек браузера \*.md-файл схемы конфигурационного элемента будет выгружен в каталог по умолчанию или браузер отобразит диалоговое окно для выбора каталога, в который необходимо сохранить файл.



## Импортировать схему конфигурационного элемента

1. [Перейдите в раздел \[ Конфигурация \]](#) ([ Configuration ]) и выберите пользовательский [пакет](#), в который необходимо импортировать схему конфигурационного элемента.
2. На панели инструментов реестра раздела нажмите [ Добавить ] —> [ Импорт ] ([ Add ] —> [ Import ]).



3. Выберите ранее экспортированный \*.md-файл схемы конфигурационного элемента.
4. На панели инструментов нажмите [ *Компилировать* ] —> [ *Перекомпилировать все* ] ([ *Compile* ] —> [ *Compile all* ]) и скомпилируйте конфигурацию.

**Важно.** При импорте нескольких схем необходимо учитывать их зависимости друг от друга. Сначала необходимо импортировать все схемы-зависимости, после этого импортировать схему, зависящую от них. Например, сначала необходимо импортировать схему объекта, а затем схему модели представления страницы, зависящую от данного объекта.

## Управление поставками в WorkspaceConsole

 Средний

Creatio предоставляет различные инструменты поставок функциональности.

**Инструменты управления поставками**, которые предоставляет Creatio:

- Creatio IDE.
- Утилита WorskpaceConsole.

В этой статье будет рассмотрено управление поставками с использованием утилиты WorkspaceConsole.

**WorkspaceConsole** — это утилита, которая предназначена для выполнения операций с [пакетами](#) и [схемами конфигурационных элементов](#) Creatio.

**Использование** утилиты WorkspaceConsole в качестве инструмента переноса решений позволяет:

- Переносить [пакеты](#) и [схемы конфигурационных элементов](#) между [рабочими средами](#) и конфигурациями.
- Устанавливать новые пакеты при обновлении или при экспорте из среды разработки.
- Переносить привязанные к пакету данные, например, наполнение справочников, новые системные настройки, демонстрационные записи раздела и т. д.
- Переносить ресурсы для локализации.
- Создавать и переносить рабочие пространства между рабочими средами.

Перед использованием утилиты WorkspaceConsole необходимо выполнить ее настройку.

## Настройка утилиты WorkspaceConsole

### 1. Узнайте значение используемой строки подключения.

Для этого откройте файл

`..\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe.config`. Строка подключения содержится в атрибуте `connectionStringName` XML-элемента `<db>`.

**Файл** `Terrasoft.Tools.WorkspaceConsole.exe.config`

```
<terrasoft>
  ...
  <db>
    <general connectionStringName="db" securityEngineType="Terrasoft.DB.MSSql.MSSqlSecuri
  </db>
  ...
</terrasoft>
```

### 2. Отредактируйте строку подключения.

Для этого откройте файл `ConnectionStrings.config`, который находится в корневом каталоге приложения. Строка подключения содержится в атрибуте `name` XML-элемента `<connectionStrings>`. Значение атрибута `name` файла `ConnectionStrings.config` должно совпадать со значением атрибута `connectionStringName` файла `Terrasoft.Tools.WorkspaceConsole.exe.config`.

**Файл** `ConnectionStrings.config`

```
<connectionStrings>
  <add name="db" connectionString="Data Source=dbserver\MSSQL2016; Initial Catalog=YourDBNa
  <add name="dbOracle" connectionString="Data Source=(DESCRIPTION = (ADDRESS_LIST = (ADDRES
</connectionStrings>
```

**На заметку.** Как правило, в конфигурационном файле по умолчанию настроено только две строки подключения. Строка с названием "db" используется для подключения к базе данных MS SQL Server, а строка с названием "dbOracle" — к базе данных Oracle.

Если с помощью WorkspaceConsole необходимо выполнить разовую операцию, то утилиту можно запустить с параметром `-webApplicationPath`. В этом параметре необходимо указать путь к каталогу с установленным приложением. В таком случае утилита самостоятельно определит все необходимые параметры подключения к базе данных из файла `ConnectionStrings.config`. При этом параметры подключения из файла `Terrasoft.Tools.WorkspaceConsole.exe.config` будут проигнорированы.

- В файле `Terrasoft.Tools.WorkspaceConsole.exe.config` установите значение `true` для атрибута `enabled` элемента `loadFromRemoteSources`.

```
<loadFromRemoteSources enabled="true" />
```

- Установите утилиту WorkspaceConsole.

Для этого с правами администратора запустите на выполнение предустановленный пакетный файл команд, который находится по пути `..\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\`. Это необходимо для установки версии исполняемого файла утилиты и библиотек, которые используются утилитой.

Пакетные файлы команд утилиты WorkspaceConsole:

- Для **32-битной операционной системы** необходимо запустить файл `PrepareWorkspaceConsole.x86.bat`.
- Для **64-битной операционной системы** необходимо запустить файл `PrepareWorkspaceConsole.x64.bat`.

- Настройте утилиту на выполнение операций с хранилищем SVN (опционально).

Для этого скопируйте файлы `SharpPlink-x64.svnExe`, `SharpSvn.dll` и `SharpSvn-DB44-20-x64.svnDll` из соответствующего каталога в каталог `...\Terrasoft.WebApp\DesktopBin\WorkspaceConsole`.

- Для **32-битной операционной системы** скопируйте файлы из каталога `...\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\x86`.
- Для **64-битной операционной системы** скопируйте файлы из каталога `...\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\x64`.

## Использование утилиты WorkspaceConsole

Исполняемый файл утилиты `Terrasoft.Tools.WorkspaceConsole.exe` расположен в каталоге `..\Terrasoft.WebApp\DesktopBin\WorkspaceConsole`, а версия утилиты совпадает с версией приложения.

**Важно.** Версия утилиты WorkspaceConsole должна совпадать с версией приложения Creatio. Например, если текущая версия приложения 7.18.1.1794, а пакеты необходимо обновить до версии 7.18.2.1658, то используйте утилиту версии 7.18.2.1658. Чтобы получить утилиту требуемой версии, обратитесь в службу поддержки.

Утилита WorkspaceConsole работает напрямую с базой данных приложения. Поэтому для корректной работы утилиты необходимо внести информацию о базе данных в конфигурационный файл `Terrasoft.Tools.WorkspaceConsole.exe.config` утилиты. Если приложение развернуто в облаке, то работать с утилитой могут только сотрудники отдела облачных сервисов. В таком случае для переноса изменений необходимо обратиться в службу поддержки.

Команды для `WorkspaceConsole` рекомендуется формировать в пакетном файле (\*.bat или \*.cmd), созданном в текстовом редакторе.

Чтобы **выполнить перенос решений с помощью утилиты WorkspaceConsole**:

1. Проверьте привязки данных.
2. Выполните резервное копирование базы данных.
3. Экспортируйте пакеты.
4. Импортируйте пакеты.
5. Перезапустите приложение в IIS.

## 1. Проверить привязки данных

Перед экспортом пакета проверьте правильность [привязки данных](#) к пакету. К привязанным данным относятся наполнение справочников, новые системные настройки, демонстрационные записи раздела и т. д.

Если раздел был создан при помощи мастера, то данные, необходимые для работы раздела, автоматически привязываются мастером раздела. Чтобы после импорта раздел отобразился в рабочем месте, необходимо привязать соответствующее значение объекта `SysModuleInWorkplace`.

## 2. Выполнить резервное копирование базы данных

Перед внесением изменений в приложение с помощью утилиты WorkspaceConsole выполните резервное копирование базы данных. Это позволит восстановить приложение при некорректном использовании команд и параметров утилиты.

## 3. Экспортировать пакеты

Утилита WorkspaceConsole позволяет экспортировать пакет из базы данных или из хранилища SVN.

### Экспортировать пакеты из базы данных

1. Сформируйте команду для экспорта пакетов.



Параметры WorkspaceConsole для экспорта пакетов из базы данных

Параметр	Значение	Описание
<code>-operation</code>	<code>SaveDBContent</code>	Сохраняет содержимое базы данных в файловую систему. Тип содержимого определяется значением параметра <code>-contentTypes</code> . Место в файловой системе, куда будет экспортировано содержимое, определяется параметром <code>-destinationPath</code> .  Требует указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code> .
<code>-contentTypes</code>	<code>Repository</code>	Тип содержимого, экспортируемого из базы данных на диск. Значение <code>Repository</code> определяет экспорт рабочего пространства, имя которого задается значением параметра <code>-workspaceName</code> , в каталог, путь к которому задается значением параметра <code>-destinationPath</code> .
<code>-workspaceName</code>	[ <i>Название рабочего пространства</i> ]	Название рабочего пространства (конфигурации), в котором выполняется операция. По умолчанию все пользователи работают в рабочем пространстве <code>Default</code> .
<code>-destinationPath</code>	[ <i>Путь к локальному каталогу</i> ]	Путь к локальному каталогу на диске. В этот каталог будут экспортированы *.gz-архивы пакетов.
<code>-webApplicationPath</code>	[ <i>Путь к локальному каталогу</i> ]	Путь к каталогу на диске, в который установлено приложение Creatio. По этому пути из файла <code>ConnectionStrings.config</code> будет считана информация по соединению с базой данных. Если параметр не указан, то будет установлено соединение с базой данных, указанной в строке соединения в конфигурационном файле утилиты.
<code>-configurationPath</code>	[ <i>Путь к локальному каталогу</i> ]	Путь к каталогу <code>..\Terrasoft.WebApp\Terrasoft.Configuration</code> . В этот каталог экспортируются исходные коды и ресурсы схем пользовательских пакетов в режиме разработки в <a href="#">файловой системе</a> .

Сигнатура команды, которую необходимо выполнить в интерпретаторе команд (консоли) Windows, для выполнения операции экспорта пакетов из базы данных:

```
[Путь к WorkspaceConsole]\Terrasoft.Tools.WorkspaceConsole.exe -operation=SaveDBContent -cont
```

## 2. Запустите утилиту.

**На заметку.** Во время выполнения операции экспорта пакетов, экспортируются все пакеты рабочего пространства. Процесс может занять несколько десятков минут.

В результате выполнения архивы пакетов указанного рабочего пространства будут экспортированы в локальный каталог.

## Экспортировать пакеты из хранилища SVN

### 1. Сформируйте команду для экспорта пакетов.

Параметры WorkspaceConsole для экспорта пакетов из SVN

Параметр	Значение	Описание
<code>-operation</code>	<code>SaveVersionSvnContent</code>	Выгружает иерархию пакетов в виде *.zip-архивов. Место в файловой системе, куда будет экспортировано содержимое, определяется параметром <code>-destinationPath</code> . SVN-хранилища определяются параметром <code>-sourcePath</code> .
<code>-workspaceName</code>	[ Название рабочего пространства ]	Название рабочего пространства (конфигурации), в котором выполняется операция. По умолчанию все пользователи работают в рабочем пространстве <code>Default</code> .
<code>-destinationPath</code>	[ Путь к локальному каталогу ]	Путь к локальному каталогу на диске. В этот каталог будут экспортированы *.gz-архивы пакетов.
<code>-workingCopyPath</code>	[ Путь к локальному каталогу ]	Локальный каталог для рабочей копии пакетов, которые хранятся в SVN.
<code>-sourcePath</code>	[ Путь к SVN-хранилищу ]	Адрес хранилища SVN для хранения структуры и метаданных пакетов. Может принимать несколько значений, указанных через запятую.
<code>-packageName</code>	[ Имя пакета ]	Имя пакета в SVN-хранилище, которое будет использоваться для экспорта. Все пакеты, от которых зависит

		текущий пакет, также будут задействованы.
<code>-packageVersion</code>	[ Версия пакета ]	Версия пакета в SVN-хранилище, которое будет использоваться для экспорта.
<code>-sourceControlLogin</code>	[ Имя пользователя SVN ]	Логин пользователя хранилища SVN.
<code>-sourceControlPassword</code>	[ Пароль пользователя SVN ]	Пароль пользователя хранилища SVN.
<code>-cultureName</code>	[ Языковая культура ]	Код языковой культуры. Например, <code>en-US</code> .
<code>-excludeDependentPackages</code>	<code>true</code> или <code>false</code>	Признак необходимости экспорта пакетов, от которых зависит пакет, указанный в параметре <code>-packageName</code> .
<code>-logPath</code>	[ Путь к локальному каталогу ]	Путь к каталогу, в который будет сохранен файл с логом операции. Название файла состоит из даты и времени запуска операции. Необязательный параметр.

Сигнатура команды, которую необходимо выполнить в интерпретаторе команд (консоли) Windows, для выполнения операции экспорта пакетов из SVN-хранилища:

```
[Путь к WorkspaceConsole]\Terrasoft.Tools.WorkspaceConsole.exe -operation=SaveVersionSvnConte
```

## 2. Запустите утилиту.

В результате выполнения архивы пакетов указанного рабочего пространства будут экспортированы в локальный каталог.

## 4. Импортировать пакеты

### 1. Сформируйте команду для импорта пакетов.

Параметры WorkspaceConsole для импорта пакетов в базу данных

Параметр	Значение	Описание
<code>-operation</code>	<code>InstallFromRepository</code>	Импортирует в конфигурацию содержимое и метаданные пакетов из *.zip-архивов. При необходимости

		<p>выполняются привязанные SQL-скрипты, регенерация исходных кодов, установка привязанных данных. Работает только с измененными или новыми пакетами и их элементами. Требуется указание одного из параметров</p> <p><code>-webApplicationPath</code> или <code>-configurationPath</code>.</p> <p>Требуется указание параметра <code>-confRuntimeParentDirectory</code>.</p>
<code>-packageName</code>	[ Имя пакета ]	<p>Имя пакета в конфигурации, которая указана в параметре <code>-workspaceName</code>.</p> <p>Все пакеты, от которых зависит текущий пакет, также будут задействованы. Если параметр не указан, то будут задействованы все пакеты конфигурации.</p>
<code>-workspaceName</code>	[ Название рабочего пространства ]	<p>Название рабочего пространства (конфигурации), в котором выполняется операция. По умолчанию все пользователи работают в рабочем пространстве <code>Default</code>.</p>
<code>-sourcePath</code>	[ Путь к локальному каталогу ]	<p>Путь к локальному каталогу на диске. В этом каталоге находятся *.gz-архивы пакетов, которые необходимо установить.</p>
<code>-destinationPath</code>	[ Путь к локальному каталогу ]	<p>Путь к локальному каталогу на диске. В этот каталог будут экспортированы *.gz-архивы пакетов, которые определены в параметре <code>-sourcePath</code>.</p>
<code>-skipConstraints</code>	false	<p>Пропустить создание внешних ключей в таблицах базы данных. Принимает значения <code>true</code> или <code>false</code>.</p>
<code>-skipValidateActions</code>	true	<p>Пропустить проверку возможности создания индексов таблиц при обновлении структуры базы данных. Принимает значения <code>true</code> или <code>false</code>.</p>
<code>-regenerateSchemaSources</code>	true	<p>Указывает на необходимость</p>

		<p>перегенерации исходных кодов после сохранения пакетов в базе данных. Принимает значения <code>true</code> или <code>false</code>. По умолчанию — <code>true</code>.</p>
<code>-updateDBStructure</code>	<code>true</code>	<p>Указывает на необходимость обновления структуры базы данных после сохранения пакетов. Принимает значения <code>true</code> или <code>false</code>. По умолчанию — <code>true</code>.</p>
<code>-updateSystemDBStructure</code>	<code>true</code>	<p>Указывает на необходимость изменения структуры базы данных системных схем перед выполнением установки пакетов. Также создает все отсутствующие индексы в системных таблицах. Принимает значения <code>true</code> или <code>false</code>.</p>
<code>-installPackageSqlScript</code>	<code>true</code>	<p>Указывает на необходимость выполнения SQL скриптов до и после сохранения пакетов. Принимает значения <code>true</code> или <code>false</code>. По умолчанию — <code>true</code>.</p>
<code>-installPackageData</code>	<code>true</code>	<p>Указывает на необходимость установки привязанных к пакету данных после сохранения пакетов. Принимает значения <code>true</code> или <code>false</code>. По умолчанию — <code>true</code>.</p>
<code>-continueIfError</code>	<code>true</code>	<p>Указывает на необходимость прервать выполнение процесса установки при получении первой ошибки. Если значение параметра — <code>true</code>, то процесс установки пройдет до конца, а пользователь получит список всех возникших ошибок. Принимает значения <code>true</code> или <code>false</code>. По умолчанию — <code>false</code>.</p>
<code>-webApplicationPath</code>	[ Путь к локальному каталогу ]	<p>Путь к каталогу на диске, в который установлено приложение Creatio. По этому пути из файла <code>ConnectionStrings.config</code> будет считана информация по соединению с базой данных. Если параметр не</p>

		указан, то будет установлено соединение с базой данных, указанной в строке соединения в конфигурационном файле утилиты.
<code>-confRuntimeParentDirectory</code>	[ Путь к локальному каталогу ]	Путь к родительскому каталогу директории <code>..\Terrasoft.WebApp\conf</code> .
<code>-logPath</code>	[ Путь к локальному каталогу ]	Путь к каталогу, в который будет сохранен файл с логом операции. Название файла состоит из даты и времени запуска операции. Необязательный параметр.
<code>-configurationPath</code>	[ Путь к локальному каталогу ]	Путь к каталогу <code>..\Terrasoft.WebApp\Terrasoft.Configuration</code> . В этот каталог экспортируются исходные коды и ресурсы схем пользовательских пакетов в режиме разработки в <a href="#">файловой системе</a> .

Сигнатура команды, которую необходимо выполнить в интерпретаторе команд (консоли) Windows, для выполнения операции импорта пакетов в базу данных:

```
[Путь к WorkspaceConsole]\Terrasoft.Tools.WorkspaceConsole.exe -operation=InstallFromRepository
```

- Запустите утилиту.
- Сформируйте команду для генерации статического контента в файловую систему.

Параметры WorkspaceConsole для компиляции конфигурации

Параметр	Значение	Описание
<code>-operation</code>	<code>BuildConfiguration</code>	Выполняет <a href="#">генерацию статического контента в файловую систему</a> . Требует указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code> .
<code>-workspaceName</code>	[ Название рабочего пространства ]	Название рабочего пространства (конфигурации), в котором определены экспортируемые пакеты. По умолчанию все пользователи работают в рабочем пространстве <code>Default</code> .
<code>-destinationPath</code>	[ Путь к	Путь к локальному каталогу на диске. В

	локальному каталогу ]	этот каталог будут экспортированы *.gz-архивы пакетов, которые определены в параметре <code>-sourcePath</code> .
<code>-webApplicationPath</code>	[ Путь к локальному каталогу ]	Путь к каталогу на диске, в который установлено приложение Creatio. По этому пути из файла <code>ConnectionStrings.config</code> будет считана информация по соединению с базой данных. Если параметр не указан, то будет установлено соединение с базой данных, указанной в строке соединения в конфигурационном файле утилиты.
<code>-confRuntimeParentDirectory</code>	[ Путь к локальному каталогу ]	Путь к родительскому каталогу директории <code>..\Terrasoft.WebApp\conf</code> .
<code>-logPath</code>	[ Путь к локальному каталогу ]	Путь к каталогу, в который будет сохранен файл с логом операции. Название файла состоит из даты и времени запуска операции. Необязательный параметр.
<code>-force</code>	<code>true</code> или <code>false</code>	<p>Задаёт условия генерации файлового контента. Если значение параметра равно <code>true</code> , то генерация файлового контента выполняется по всем схемам. Если значение параметра равно <code>false</code> , то генерация файлового контента выполняется по изменённым схемам. По умолчанию — <code>false</code> .</p> <p>Требует указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code> .</p>
<code>-configurationPath</code>	[ Путь к локальному каталогу ]	<p>Путь к каталогу <code>..\Terrasoft.WebApp\Terrasoft.Configuration</code> .</p> <p>В этот каталог экспортируются исходные коды и ресурсы схем пользовательских пакетов в режиме разработки в <a href="#">файловой системе</a>.</p>

Сигнатура команды, которую необходимо выполнить в интерпретаторе команд (консоли) Windows, для выполнения операции генерации статического контента в файловую систему:

```
[Путь к WorkspaceConsole]\Terrasoft.Tools.WorkspaceConsole.exe -operation=BuildConfiguration
```

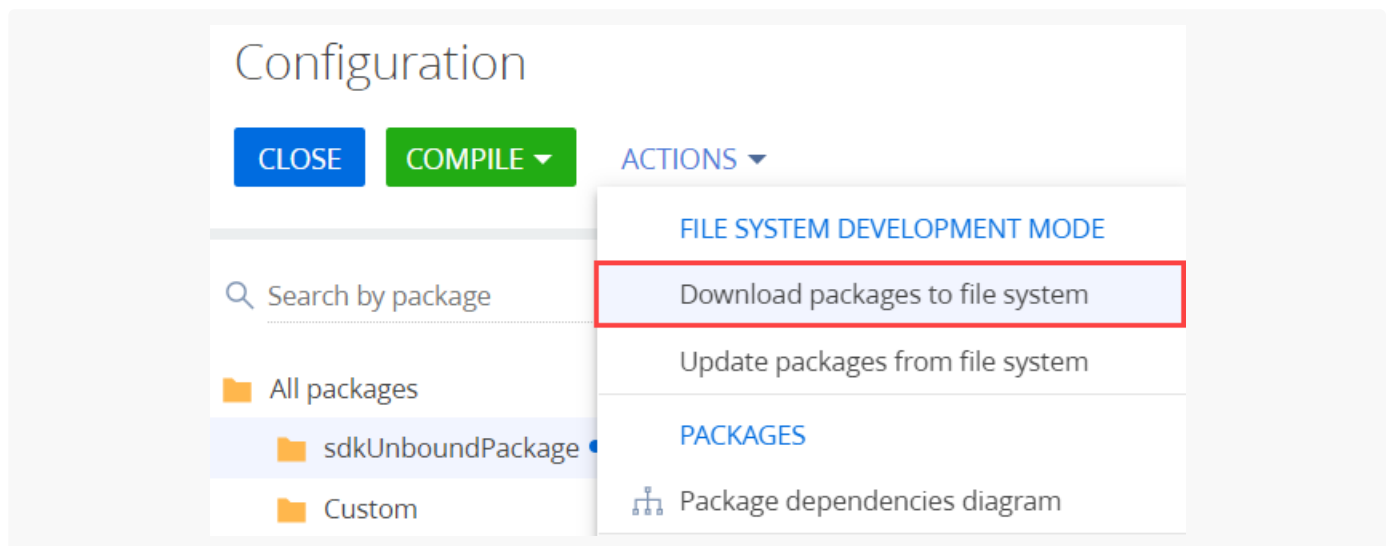
#### 4. Запустите утилиту.

**На заметку.** Пакеты, импортируемые в приложение с помощью WorkspaceConsole, считаются предустановленными и недоступны для изменения.

Не рекомендуется использовать утилиту WorkspaceConsole для импорта пакетов в базу данных при **включенном режиме разработки в файловой системе**. Если утилита будет использована, то исходный код измененных схем будет изменен в базе данных, но останется без изменений в файловой системе. Т. е. при открытии схемы конфигурационного элемента в Creatio IDE отобразится неизменный код из файловой системы. При этом дата модификации схемы конфигурационного элемента будет обновлена. Это приводит к ложному ощущению, что перенос схемы прошел корректно.

Чтобы импортировать пакеты в базу данных при **включенном режиме разработки в файловой системе**:

1. Воспользуйтесь [инструкцией](#) и импортируйте пакеты в базу данных с помощью утилиты WorkspaceConsole.
2. 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.

#### 5. Перезапустить приложение в IIS

Для применения изменений перезапустите приложение в IIS. Это необходимо, поскольку утилита WorkspaceConsole вносит изменения напрямую в базу данных.



# Экспортировать пакеты из базы данных

 Средний

**Пример.** Экспортировать в каталог все пакеты рабочего пространства `Default`.

- `C:\creatio` — каталог с установленным приложением.
- `C:\SavedPackages` — каталог для экспорта пакетов.
- `C:\Logs` — каталог для экспорта файла с логом операции.

## 1. Сформировать команду для экспорта пакета из базы данных

1. Создайте файл пакетных команд Windows (\*.bat или \*.cmd), используя текстовый редактор.
2. В созданный файл добавьте команду для запуска утилиты.

### Команда для запуска утилиты

```
C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe
pause
```

Сохраните пакетный файл.

## 2. Экспортировать пакет из базы данных

Чтобы **экспортировать пакет из базы данных** дважды кликните по имени пакетного файла.

После этого появится консольное окно, в котором будет отображаться процесс выполнения операции, указанной в соответствующей команде `WorkspaceConsole`.

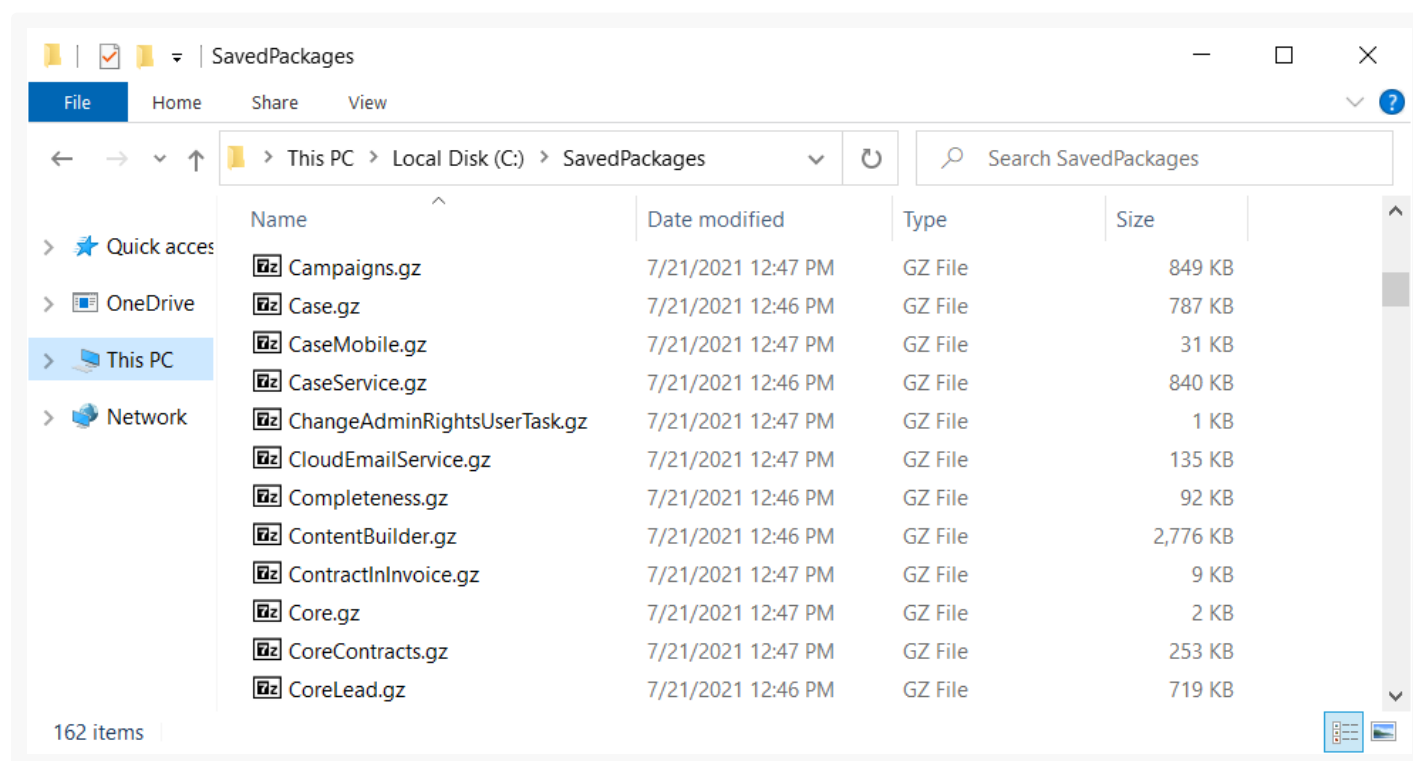
```

C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe
operation=SaveDBContent -contentType=Repository -workspaceName=Default
destinationPath=C:\SavedPackages -webApplicationPath=C:\creatio
confRuntimeParentDirectory=C:\creatio\Terrasoft.WebApp -logPath=C:\Logs

=== 09:25:34.5812 (UTC) ===
Saving 'Default' workspace in repository 'C:\SavedPackages'
Added - schema 'AttributeValue' in package 'Base'
Added - schema 'AcademyURL' in package 'Base'
Added - schema 'ActivityResultEditPage' in package 'Base'
Added - schema 'Department' in package 'Base'
Added - schema 'ProductType' in package 'Base'
Added - schema 'Product' in package 'Base'
Added - schema 'CreateSocialAccountUserTask' in package 'Base'
Added - schema 'VwSysEntitySchemaInPackage' in package 'Base'
Added - schema 'SysModuleAnalyticsReportLczOld' in package 'Base'
Added - schema 'CommunicationType' in package 'Base'
Added - schema 'SysSSPEntitySchemaAccessList' in package 'Base'
Added - schema 'BaseAdministrativeGridPage' in package 'Base'
Added - schema 'BaseObjectRecordRightsPage' in package 'Base'
Added - schema 'EmailTemplateFolder' in package 'Base'

```

В результате выполнения команды в каталог `C:\SavedPackages` будут экспортированы \*.gz-архивы всех пакетов конфигурации `Default`.



## Экспортировать пакет из SVN

Средний

**Пример.** Экспортировать пакет из SVN-хранилища в каталог рабочего пространства `Default`.

- `C:\creatio` — каталог с установленным приложением.
- `C:\SavedPackages` — каталог для экспорта \*gz-архива пакета.
- `C:\WorkingCopy` — каталог для экспорта структуры пакета из SVN-хранилища.
- `http://server-svn:8050/Packages` — адрес SVN-хранилища.
- `sdkTestPackage` — пакет для экспорта из SVN-хранилища.
- `7.18.1` — версия пакета для экспорта из SVN-хранилища.
- `"User"` — логин пользователя хранилища SVN.
- `"Password"` — пароль пользователя хранилища SVN.
- `ru-RU` — языковая культура.
- `C:\Logs` — каталог для экспорта файла с логом операции.

## 1. Сформировать команду для экспорта пакета из SVN-хранилища

1. Создайте файл пакетных команд Windows (\*.bat или \*.cmd), используя текстовый редактор.
2. В созданный файл добавьте команду для запуска утилиты.

### Команда для запуска утилиты

```
C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe
pause
```

Сохраните пакетный файл.

## 2. Экспортировать пакет из SVN

Чтобы **экспортировать пакет из SVN** дважды кликните по имени пакетного файла.

После этого появится консольное окно, в котором будет отображаться процесс выполнения операции, указанной в соответствующей команде WorkspaceConsole.

```

C:\WINDOWS\system32\cmd.exe

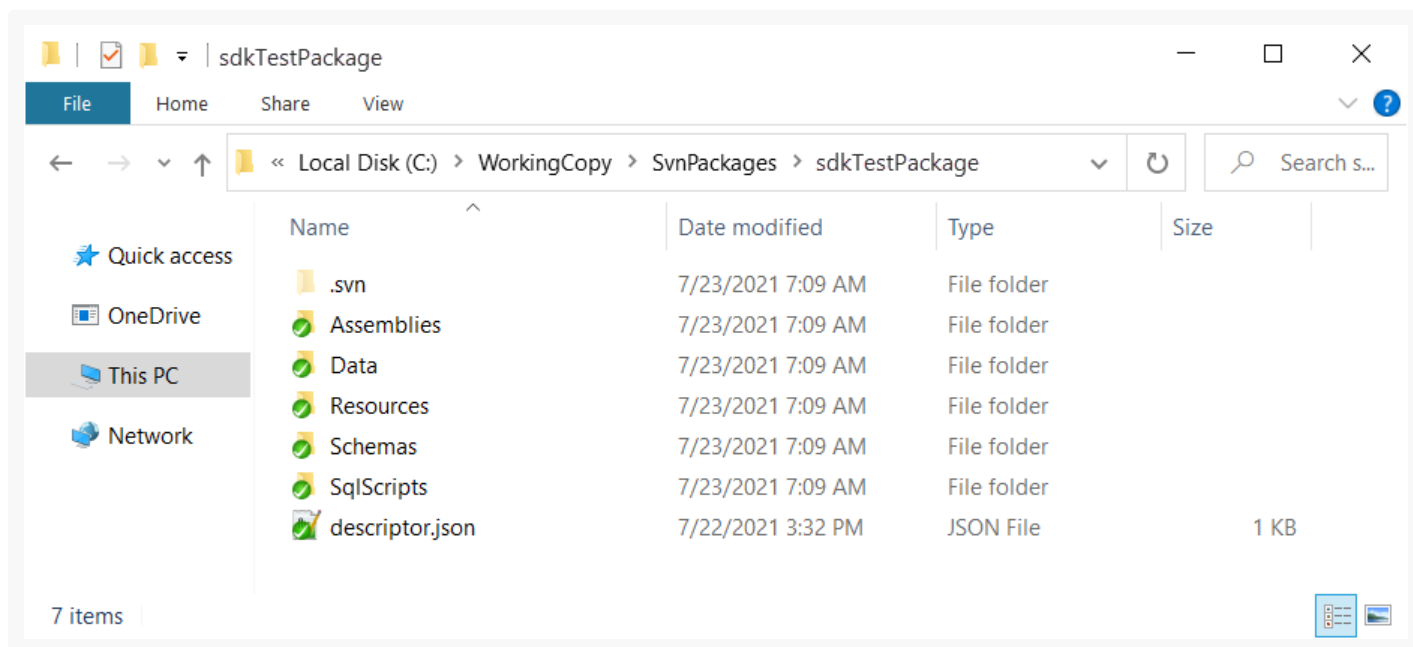
C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe
operation=SaveVersionSvnContent -workspaceName=Default -destinationPath=C:\SavedPackages
workingCopyPath=C:\WorkingCopy -sourcePath=http://server-svn:8050/Packages
packageName=sdkTestPackage -packageVersion=7.18.1 -sourceControlLogin=User
sourceControlPassword=Password -cultureName=ru-RU -excludeDependentPackages=true -logPath=C:\Logs

=== 12:30:38.3038 (UTC) ===
Receiving sdkTestPackage package data from repository
IsUnderSourceControl [C:\WorkingCopy\Packages\sdkTestPackage] = []
Compression of files in final repository
Added - package 'sdkTestPackage'
Utility finished working.

=== 12:30:39.0498 (UTC) ===

```

В результате выполнения команды в каталог `C:\SavedPackages` будет экспортирован пакет `sdkTestPackage` конфигурации `Default`. Структура папки с именем пакета описана в статье [Общие принципы работы с пакетами](#).



## Импортировать пакет в базу данных

Средний

**Пример.** Импортировать пакет из каталога в рабочее пространство `Default`.

- `C:\creatio` — каталог с установленным приложением.
- `sdkTestPackage` — пакет для импорта в приложение, который находится по пути `C:\SavedPackages`

- `C:\SavedPackages` — каталог с импортированными пакетами.
- `C:\TempPackages` — каталог для импорта пакета.
- `C:\Logs` — каталог для экспорта файла с логом операции.

## 1. Сформировать команду для импорта пакета в базу данных

1. Создайте файл пакетных команд Windows (\*.bat или \*.cmd), используя текстовый редактор.
2. В созданный файл добавьте команду для запуска утилиты.

### Команда для запуска утилиты

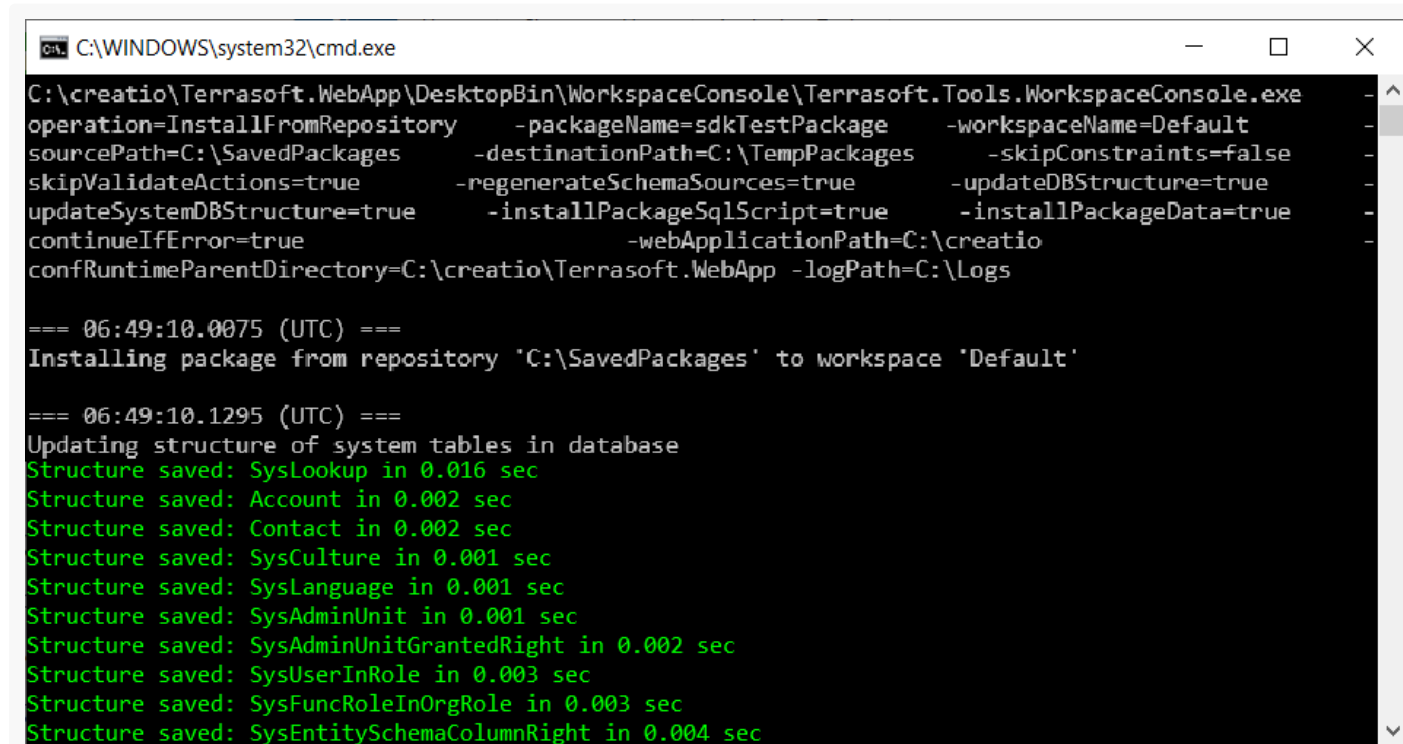
```
C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe
pause
```

Сохраните пакетный файл.

## 2. Импортировать пакет в базу данных

Чтобы **импортировать пакет в базу данных** дважды кликните по имени пакетного файла.

После этого появится консольное окно, в котором будет отображаться процесс выполнения операции, указанной в соответствующей команде WorkspaceConsole.

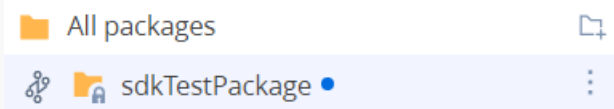


```
C:\WINDOWS\system32\cmd.exe
C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe
operation=InstallFromRepository -packageName=sdkTestPackage -workspaceName=Default
sourcePath=C:\SavedPackages -destinationPath=C:\TempPackages -skipConstraints=false
skipValidateActions=true -regenerateSchemaSources=true -updateDBStructure=true
updateSystemDBStructure=true -installPackageSqlScript=true -installPackageData=true
continueIfError=true -webApplicationPath=C:\creatio
confRuntimeParentDirectory=C:\creatio\Terrasoft.WebApp -logPath=C:\Logs

=== 06:49:10.0075 (UTC) ===
Installing package from repository 'C:\SavedPackages' to workspace 'Default'

=== 06:49:10.1295 (UTC) ===
Updating structure of system tables in database
Structure saved: SysLookup in 0.016 sec
Structure saved: Account in 0.002 sec
Structure saved: Contact in 0.002 sec
Structure saved: SysCulture in 0.001 sec
Structure saved: SysLanguage in 0.001 sec
Structure saved: SysAdminUnit in 0.001 sec
Structure saved: SysAdminUnitGrantedRight in 0.002 sec
Structure saved: SysUserInRole in 0.003 sec
Structure saved: SysFuncRoleInOrgRole in 0.003 sec
Structure saved: SysEntitySchemaColumnRight in 0.004 sec
```

В результате выполнения команды в конфигурацию `Default` будет импортирован пакет `sdkTestPackage`.



### 3. Сформировать команду для генерации статического контента в файловую систему

1. Создайте файл пакетных команд Windows (\*.bat или \*.cmd), используя текстовый редактор.
2. В созданный файл добавьте команду для запуска утилиты.

#### Команда для запуска утилиты

```
C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe  
pause
```

Сохраните пакетный файл.

### 4. Сгенерировать статический контент в файловую систему

Чтобы **сгенерировать статический контент в файловую систему** дважды кликните по имени пакетного файла.

После этого появится консольное окно, в котором будет отображаться процесс выполнения операции, указанной в соответствующей команде `WorkspaceConsole`.

```
C:\WINDOWS\system32\cmd.exe

C:\creatio\Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Terrasoft.Tools.WorkspaceConsole.exe
operation=BuildConfiguration -workspaceName=Default -destinationPath=C:\creatio\Terrasoft.WebApp
webApplicationPath=C:\creatio -confRuntimeParentDirectory=C:\creatio\Terrasoft.WebApp
configurationPath=C:\creatio\Terrasoft.WebApp\Terrasoft.Configuration -logPath=C:\Logs

=== 07:35:34.6758 (UTC) ===
Configuration build started
Operation [BuildChanged] started
Operation [GenerateFileContentDescriptors] started
Operation [GenerateFileContentDescriptors] took 5 s 270 ms
Operation [GenerateFileContentBootstraps] started
Operation [GenerateFileContentBootstraps] took 0 s 39 ms
Ordered by CPU time
[ClientUnitSchemaManager.AccountPageV2] - Total: 5899 ms, GetInstance: 564 ms, Less: 1 ms, Images: 1559 ms,
SourceCode: 3610 ms, Resources: 165 ms
[ClientUnitSchemaManager.UsrBook1Section] - Total: 4952 ms, GetInstance: 611 ms, Less: 55 ms, Images: 3886 m
s, SourceCode: 77 ms, Resources: 323 ms
[ClientUnitSchemaManager.AccountSectionV2] - Total: 4318 ms, GetInstance: 321 ms, Less: 1 ms, Images: 0 ms,
SourceCode: 3277 ms, Resources: 719 ms
```

В результате выполнения команды в файловую систему будет сгенерирован статический контент измененных схем.

## Параметры утилиты WorkspaceConsole CLI

 Средний

-help

Справка с полным перечнем параметров утилиты и их кратким описанием. Если указать другие параметры, то они будут проигнорированы.

-operation

Название операции, которую необходимо выполнить. Обязательный параметр. Значение по умолчанию — `LoadLicResponse`.

### Возможные значения

LoadLicResponse	Загружает лицензии в базу данных, указанную в строке подключения. Единственная операция, которая не требует указания параметра <code>-workspaceName</code> .
SaveRepositoryContent	Сохраняет указанное в параметре <code>-contentTypes</code> содержимое *.zip-архивов пакетов из каталога, указанного в параметре <code>-sourcePath</code> , в каталог, указанный в параметре <code>-destinationPath</code> .

SaveDBContent	<p>Сохраняет содержимое базы данных в файловую систему. Тип содержимого определяется значением параметра <code>-contentType</code>. Место в файловой системе, куда будет экспортировано содержимое, определяется параметром <code>-destinationPath</code>.</p> <p>Требуется указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code>.</p>
SaveVersionSvnContent	<p>Выгружает иерархию пакетов в виде *.zip-архивов. Место в файловой системе, куда будет экспортировано содержимое, определяется параметром <code>-destinationPath</code>. SVN-хранилища определяются параметром <code>-sourcePath</code>.</p>
RegenerateSchemaSources	<p>Выполняет регенерацию исходных кодов и их компиляцию.</p> <p>Требуется указания параметра <code>-confRuntimeParentDirectory</code>.</p>
InstallFromRepository	<p>Импортирует в конфигурацию содержимое и метаданные пакетов из *.zip-архивов. При необходимости выполняются привязанные SQL-скрипты, регенерация исходных кодов, установка привязанных данных. Работает только с измененными или новыми пакетами и их элементами. Требуется указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code>.</p> <p>Требуется указания параметра <code>-confRuntimeParentDirectory</code>.</p>
InstallBundlePackages	<p>Устанавливает набор пакетов, перечисленных через запятую в параметре <code>-packageName</code>, в рабочее пространство, указанное в параметре <code>-workspaceName</code>.</p> <p>Требуется указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code>.</p> <p>Требуется указания параметра <code>-confRuntimeParentDirectory</code>.</p>
PrevalidateInstallFromRepository	<p>Проверяет возможность установки пакетов из *.zip-архивов.</p>
ConcatRepositories	<p>Слияние нескольких репозиториях.</p>
ConcatSVNRepositories	<p>Слияние нескольких хранилищ SVN.</p>
ExecuteProcess	<p>Выполняет запуск бизнес-процесса в конфигурации, если он будет найден.</p> <p>Требуется указания параметра <code>-confRuntimeParentDirectory</code>.</p>
UpdatePackages	<p>В базе данных выполняет обновление пакетов (параметр <code>-packageName</code>).</p>



	<p><code>-packageName</code> ), которые присутствуют в иерархии пакета продукта (параметр <code>-productPackageName</code> ).</p> <p>Требуется указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code> .</p> <p>Требуется указания параметра <code>-confRuntimeParentDirectory</code> .</p>
BuildWorkspace	<p>Компилирует рабочее пространство (конфигурацию). Используется при <a href="#">разработке схем с помощью Visual Studio</a>.</p> <p>Требуется указания параметра <code>-confRuntimeParentDirectory</code> .</p>
ReBuildWorkspace	<p>Перекомпилирует рабочее пространство (конфигурацию). Используется при <a href="#">разработке схем с помощью Visual Studio</a>.</p> <p>Требуется указания параметра <code>-confRuntimeParentDirectory</code> .</p>
UpdateWorkspaceSolution	<p>Обновляет решение и файлы <a href="#">проекта Visual Studio</a>.</p>
BuildConfiguration	<p>Выполняет <a href="#">генерацию статического контента в файловую систему</a>.</p> <p>Используемые параметры:</p> <ul style="list-style-type: none"> <li><code>-workspaceName</code></li> <li><code>-destinationPath</code></li> <li><code>-webApplicationPath</code></li> <li><code>-logPath</code></li> <li><code>-force</code></li> <li><code>-configurationPath</code></li> </ul> <p>Требуется указания одного из параметров <code>-webApplicationPath</code> или <code>-configurationPath</code> .</p>

**Важно.** Для корректной работы приложения после выполнения операций `InstallFromRepository` , `InstallBundlePackages` , `UpdatePackages` необходимо выполнить операцию `BuildConfiguration` .

`-user`

Имя пользователя для выполнения авторизации. Указывается, если эта информация отсутствует в конфигурационном файле утилиты или необходимо выполнить операцию от имени другого пользователя.

---

`-password`

Пароль пользователя для выполнения авторизации. Указывается, если эта информация отсутствует в конфигурационном файле утилиты или необходимо выполнить операцию от имени другого пользователя.

---

`-workspaceName`

Название рабочего пространства (конфигурации), в котором определены экспортируемые пакеты. По умолчанию все пользователи работают в рабочем пространстве `Default`.

---

`-autoExit`

Указывает, завершать ли автоматически процесс утилиты после выполнения операции. Принимает значения `true` или `false`. По умолчанию — `false`.

---

`-processName`

Имя процесса, который необходимо запустить.

---

`-repositoryUri`

Адрес хранилища SVN для хранения структуры и метаданных пакетов. Необязательный параметр. При указании параметра его значение используется вместо значения параметра, указанного в параметре `-workspaceName`.

---

`-sourceControlLogin`

Логин пользователя хранилища SVN.

---

`-sourceControlPassword`

Пароль пользователя хранилища SVN.

---

`-workingCopyPath`

Локальный каталог для рабочей копии пакетов, которые хранятся в SVN.

---

`-contentTypes`

Тип содержимого, экспортируемого из базы данных на диск.

[Возможные значения](#)

SystemData	Данные системных схем в формате JSON. Выгружаются все системные схемы и их колонки, кроме указанных в параметре <code>-excludedSchemas</code> .
ConfigurationData	Данные схем конфигурационных элементов в формате JSON. Выгружаются все схемы, кроме указанных в параметре <code>-excludedSchemas</code> .
Resources	Ресурсы схем конфигурационных элементов для локализации в формате XML.
LocalizableData	Наполнение схем конфигурационных элементов для локализации в формате XML. Выгружаются только текстовые колонки. Дополнительные ограничения указываются в параметрах <code>-excludedSchemas</code> и <code>-excludedSchemaColumns</code> .
Repository	Данные рабочего пространства в формате zip.
SqlScripts	SQL-скрипты, привязанные к пакетам.
Data	Данные системных схем и схем конфигурационных элементов в формате JSON. Комбинация значений <code>SystemData</code> и <code>ConfigurationData</code> .
LocalizableSchemaData	Данные локализуемых объектов.
All	Все типы содержимого.

#### -sourcePath

Путь к локальному каталогу на диске, из которого необходимо забрать данные (например, пакеты, схемы или ресурсы). Для операций `ConcatRepositories` и `SaveVersionSvnContent` может принимать несколько значений, указанных через запятую.

#### -destinationPath

Путь к локальному каталогу на диске, в который необходимо сохранить данные (например, пакеты, схемы или ресурсы).

#### -webApplicationPath

Путь к каталогу на диске, в который установлено приложение Creatio. По этому пути из файла `ConnectionStrings.config` будет считана информация по соединению с базой данных. Если параметр не указан, то будет установлено соединение с базой данных, указанной в строке соединения в конфигурационном файле утилиты.

Для операций `BuildWorkspace`, `ReBuildWorkspace` и `UpdateWorkspaceSolution` параметр `-webApplicationPath` должен указывать на путь к каталогу `Terrasoft.WebApp`.

---

`-configurationPath`

Путь к каталогу `..\Terrasoft.WebApp\Terrasoft.Configuration`. В этот каталог экспортируются исходные коды и ресурсы схем пользовательских пакетов в режиме разработки в [файловой системе](#).

---

`-filename`

Имя файла. Обязательный параметр для операции `LoadLicResponse`.

---

`-excludedSchemas`

Названия схем конфигурационных элементов, которые необходимо исключить при выполнении операции.

---

`-excludedSchemaColumns`

Названия колонок схем конфигурационных элементов, которые необходимо исключить при выполнении операции.

---

`-excludedWorkspaceNames`

Названия рабочих пространств (конфигураций), которые необходимо исключить при выполнении операции.

---

`-includedSchemas`

Названия схем конфигурационных элементов, которые принудительно используются при выполнении операции.

---

`-includedSchemaColumns`

Названия колонок схем конфигурационных элементов, которые принудительно используются при выполнении операции.

---

`-cultureName`

Код языковой культуры. Обязательный параметр, если используются значения `Resources` и/или `LocalizableData` для параметра `-contentTypes`.

---

`-schemaManagerNames`

Имена менеджеров схем конфигурационных элементов, которые используются для операций. Значение по умолчанию — `EntitySchemaManager`.

---

#### `-packageName`

Имя пакета в конфигурации, которая указана в параметре `-workspaceName`. Все пакеты, от которых зависит текущий пакет, также будут задействованы. Если параметр не указан, то будут задействованы все пакеты конфигурации.

---

#### `-clearWorkspace`

Указывает на необходимость очистки рабочего пространства перед обновлением. Принимает значения `true` или `false`. По умолчанию — `false`.

---

#### `-installPackageSqlScript`

Указывает на необходимость выполнения SQL скриптов до и после сохранения пакетов. Принимает значения `true` или `false`. По умолчанию — `true`.

---

#### `-installPackageData`

Указывает на необходимость установки привязанных к пакету данных после сохранения пакетов. Принимает значения `true` или `false`. По умолчанию — `true`.

---

#### `-updateDBStructure`

Указывает на необходимость обновления структуры базы данных после сохранения пакетов. Принимает значения `true` или `false`. По умолчанию — `true`.

---

#### `-regenerateSchemaSources`

Указывает на необходимость регенерации исходных кодов после сохранения пакетов в базе данных. Принимает значения `true` или `false`. По умолчанию — `true`.

---

#### `-continueOnError`

Указывает на необходимость прервать выполнение процесса установки при получении первой ошибки. Если значение параметра — `true`, то процесс установки пройдет до конца, а пользователь получит список всех возникших ошибок. Принимает значения `true` или `false`. По умолчанию — `false`.

Для операций `InstallFromSvn` и `InstallFromRepository` необходимо использовать параметр `-continueOnError=false`. Поскольку эти операции работают с измененными или новыми пакетами и их элементами. Информация об измененных элементах получается на основании сравнения новой и существующей структур пакетов. По этой причине, если пользователь выполняет команду без указания

ключа `-continueIfError=true` и получает ошибку, то повторный запуск команды для той же конфигурации выполнится без ошибок, но не внесет изменения в базу данных. Причина — предыдущая операция синхронизировала структуры пакетов указанной конфигурации и хранилища, и измененных элементов пакетов в этой операции нет.

---

#### `-skipCompile`

Указывает на необходимость выполнения компиляции. Работает, если для параметра `-updateDBStructure` установлено значение `false`. Принимает значения `true` или `false`. По умолчанию — `false`.

---

#### `-autoUpdateConfigurationVersion`

Обновляет значение версии конфигурации в базе данных до версии приложения Creatio. Принимает значения `true` или `false`. По умолчанию — `false`.

---

#### `-warningsOnly`

При обнаружении ошибки выполнения операции утилита информирует об ошибке. Принимает значения `true` или `false`. По умолчанию — `false`.

---

#### `-confRuntimeParentDirectory`

Путь к родительскому каталогу директории `..\Terrasoft.WebApp\conf`. Параметр необходимо использовать в командах, которые вызывают компиляцию приложения или генерацию статического контента:

- `RegenerateSchemaSources`
- `InstallFromRepository`
- `InstallBundlePackages`
- `UpdatePackages`
- `BuildWorkspace`
- `RebuildWorkspace`
- `ExecuteProcess`