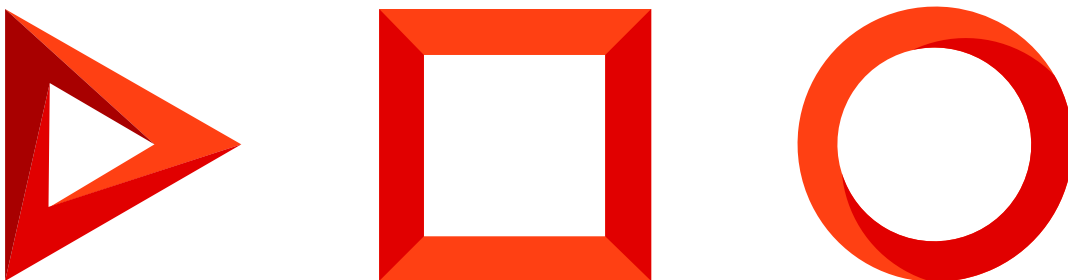


# Version control system

Version control in Creatio IDE

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

# Table of Contents

<b>Version control in Creatio IDE</b>	<b>4</b>
Install a package from the SVN repository	4
Update a package from the SVN repository	8
Commit a package to the SVN repository	9

# Version control in Creatio IDE



Creatio IDE lets you work with the SVN version control system.

**Attention.** Use the SVN version control system only to transfer changes between [development environments](#). Do not use SVN on a [pre-production](#) or [production](#) environment. This can render Creatio inoperable or decrease its performance. Learn more in a separate article: [Version control in Subversion](#).

The [out-of-the-box Creatio IDE tools](#) let you execute the following actions:

- install package from the SVN repository
- update package from the SVN repository
- commit package to the SVN repository

You cannot work with the SVN version control system using the out-of-the-box Creatio IDE tools when the [file system development mode](#) is turned on. By default, the file system development mode is turned off.

SVN integration is enabled in Creatio by default. To set up SVN integration, modify the value of the `defPackagesWorkingCopyPath` setting's `connectionString` attribute in the `ConnectionStrings.config` configuration file. This setting contains the path to the directory that stores working copies.

## ConnectionStrings.config

```
<add name="defPackagesWorkingCopyPath" connectionString="TEMP\APPLICATION\WORKSPACE\TerrasoftPac
```

Creatio makes and uses a working copy of each custom package with versioning enabled in SVN integration mode. The working copy contains custom packages organized as a set of directories and files. The Creatio SVN client synchronizes this data with an SVN repository. We recommend specifying the path to a permanent directory in the `defPackagesWorkingCopyPath` system setting. The OS might clear the temporary directory specified by default. Do not specify `...\Terrasoft.WebApp\Terrasoft.Configuration\Pkg` as the repository of working package copies in Creatio .NET Framework.

## Install a package from the SVN repository

**Package installation** involves adding the package and its dependencies from the SVN repository.

Install the package in the following **cases**:

- Multiple developers work on the package functionality.
- You need to transfer the changes between [environments](#).

If you use **Creatio in the cloud**, we recommend contacting Creatio support to install the package.

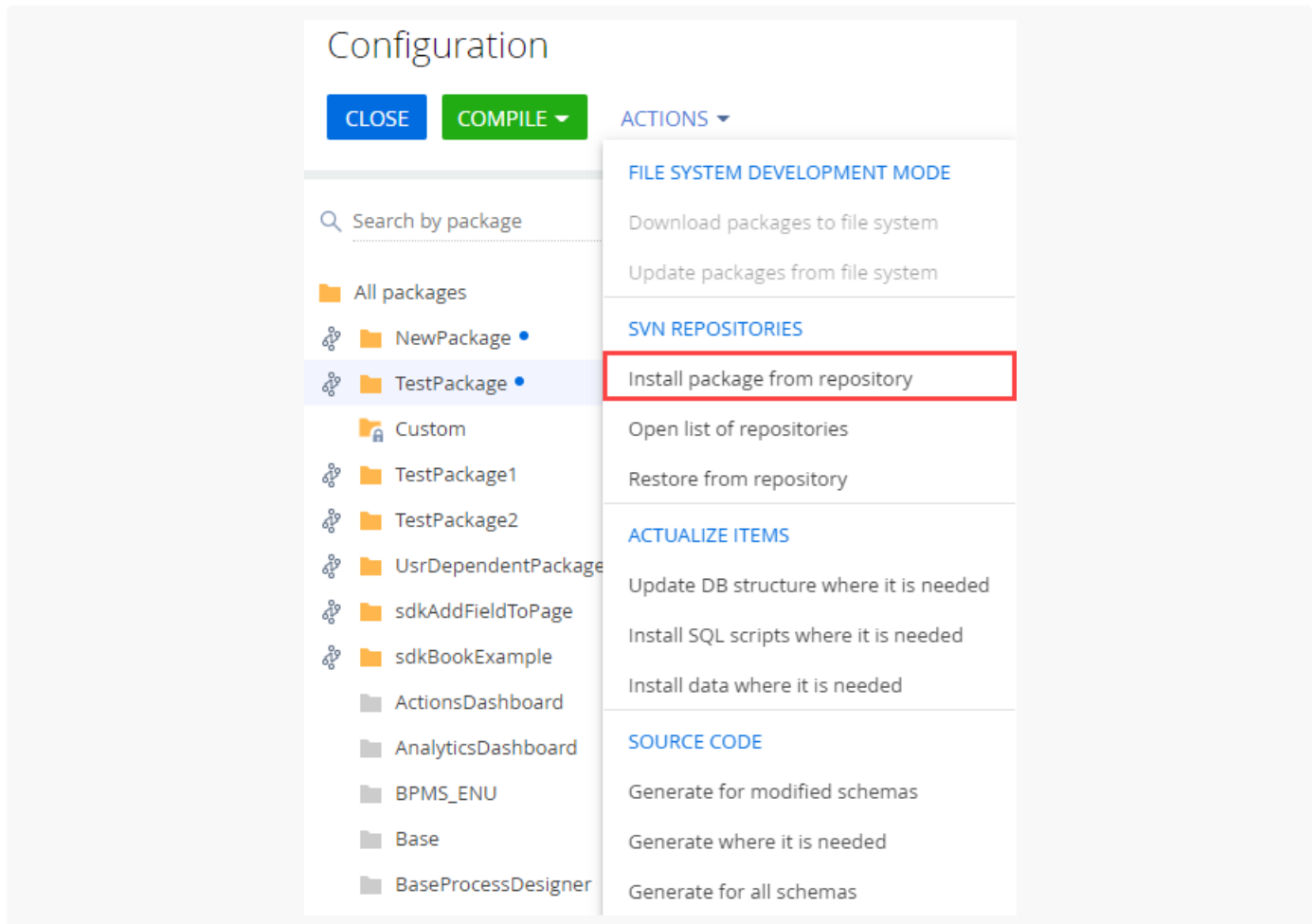
To **install a package from the SVN repository** in Creatio on-site:

1. Back up the database.

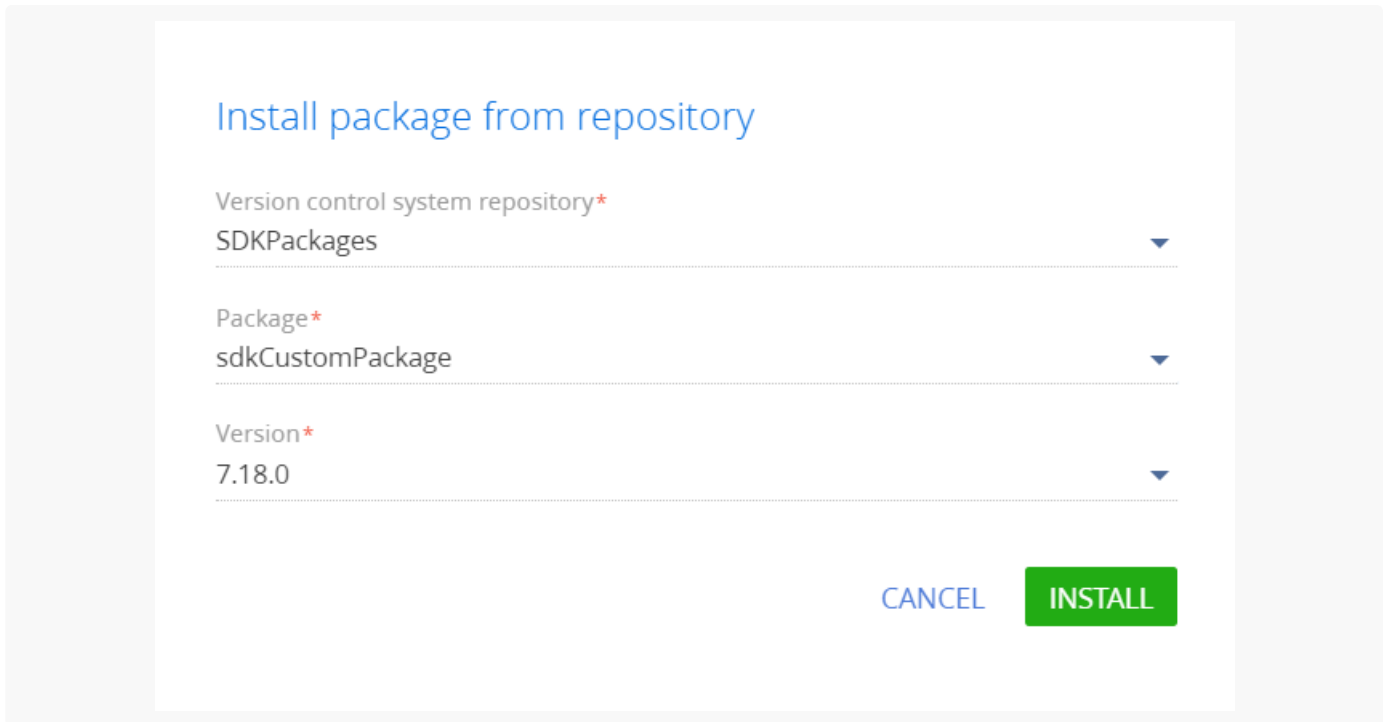
This is required because it is not possible to roll back to the previous version using the SVN version control system.

2. [Open the \[ Configuration \] section.](#)

3. Select [ *Install package from repository* ] in the [ *SVN repositories* ] group of the action menu.



4. Select the SVN repository as well as name and version of the package to install → [ *Install* ].



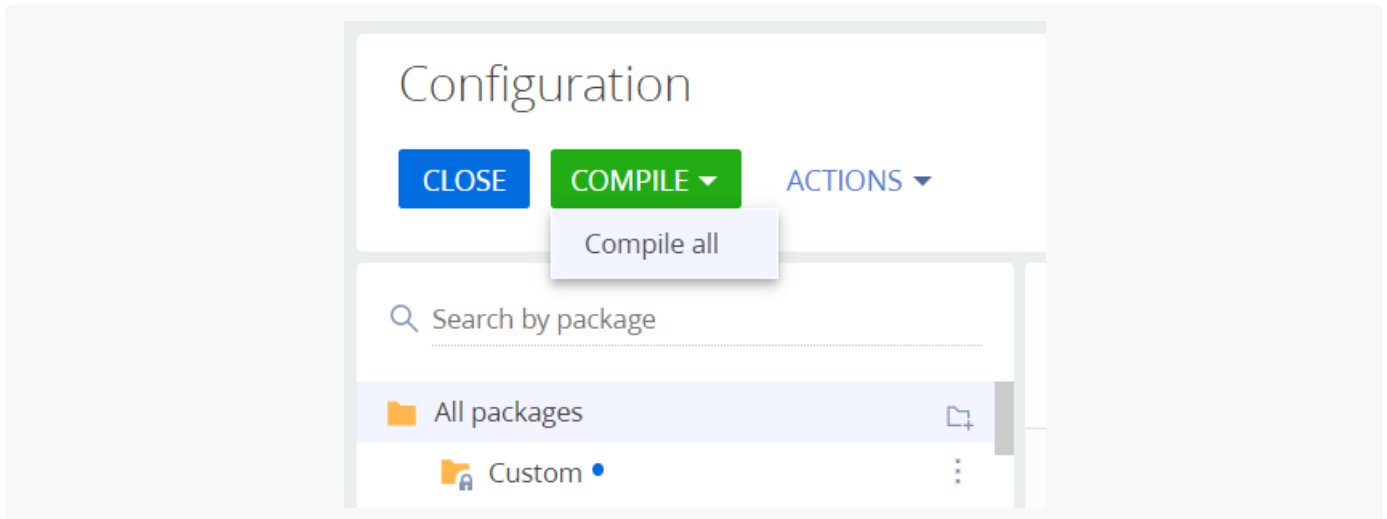
Creatio applies bound data and sets dependencies automatically as part of the package installation. In some cases, the changes are not applied automatically. If this happens, apply the changes manually.

To **apply the changes manually** to an installed package:

- a. [Open the \[ Configuration \] section.](#)
- b. Generate source codes for configuration elements that require it.
- c. Compile the configuration.
- d. Update the database structure.
- e. Install SQL scripts if needed.
- f. Install bound data.

**Note.** You can check whether to update the database structure, SQL scripts, and bound data in the [ *Needs to be installed in database* ] and [ *Needs to be updated in database* ] properties. A dialog will appear if an error occurs.

5. Click [ *Compile all* ]. This is required to [generate static content](#).



When you install a custom package, Creatio checks its dependencies and additionally sets or updates the dependency packages of the current package. For example, when you install the [ *sdkCustomPackage* ] package from the SVN repository, the [ *sdkDependentPackage* ] dependent package is installed as well. This also changes the Creatio package hierarchy. If the [ *sdkCustomPackage* ] package was installed previously, it is changed.

### Changes

UsrEntitySchema	Schema	Added
ExternalAssembly.dll	External Assembly	Added
UsrCustomSQLScript	SQL script	Added
UsrEntitySchema	Data	Added
UsrClientUnitSchema	Schema Resource	Added
UsrClientUnitSchema	Schema Resource	Added
UsrClientUnitSchema	Schema Resource	Added
UsrClientUnitSchema	Schema Resource	Added
UsrEntitySchema	Schema Resource	Added
UsrEntitySchema	Schema Resource	Added
sdkDependentPackage	Package	Added

[CLOSE](#)

When you instal a custom package from the SVN repository, Creatio changes the **package hierarchy** as follows:

1. The dependencies of the installed package are defined. View the dependencies in the `DependsOn` property of the package metadata.

### Package metadata

```
{
  "Descriptor": {
    "UIId": "8bc92579-92ee-4ff2-8d44-1ca61542aa1b",
    "PackageVersion": "7.18.0",
    "Name": "sdkCustomPackage",
    "ModifiedOnUtc": "\\Date(1522671879000)\\/",
    "Maintainer": "Customer",
    "Description": "Package created by user",
    "DependsOn": [
      {
        "UIId": "51b3ed42-678c-4da3-bd16-8596b95c0546",
        "PackageVersion": "7.18.0",
        "Name": "sdkDependentPackage"
      },
      {
        "UIId": "e14dcfb1-e53c-4439-a876-af7f97083ed9",
        "PackageVersion": "7.18.0",
        "Name": "SalesEnterprise"
      }
    ]
  }
}
```

2. The installation of the dependent packages into the configuration is verified. If the packages are installed, Creatio updates them, otherwise, it installs them.

**Attention.** If the dependencies are not found in SVN repositories (e. g., the SVN repository is not registered in the repository list or is inactive), a message about package installation error appears. The package dependency hierarchy is updated when a package is installed. As such, add the SVN repositories that contain dependent packages to the configuration and activate these repositories.

3. When a package is installed, Creatio installs or updates only the dependencies installed in the SVN version control system. Packages installed from \*.zip archives and pre-installed packages are not updated.

Install the packages on which the custom package or its dependencies depend before you install the custom package. Installation fails if the workspace lacks any pre-installed dependency package installed from a \*.zip archive.

## Update a package from the SVN repository

**Package update** involves uploading the changes in both the package and package [dependencies](#) from the SVN repository into the Creatio. The dependencies of the updated package are defined as part of the update. View the



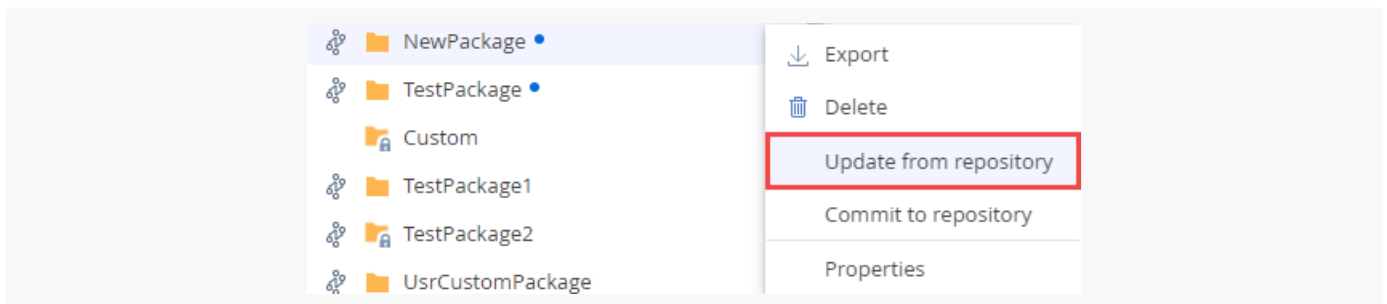
dependencies in the `DependsOn` property of the package metadata.

Update the package in the following **cases**:

- Multiple developers work on the package functionality.
- You need to transfer the changes between [environments](#).
- You are going to commit the changes.

To **update a package from the SVN repository**:

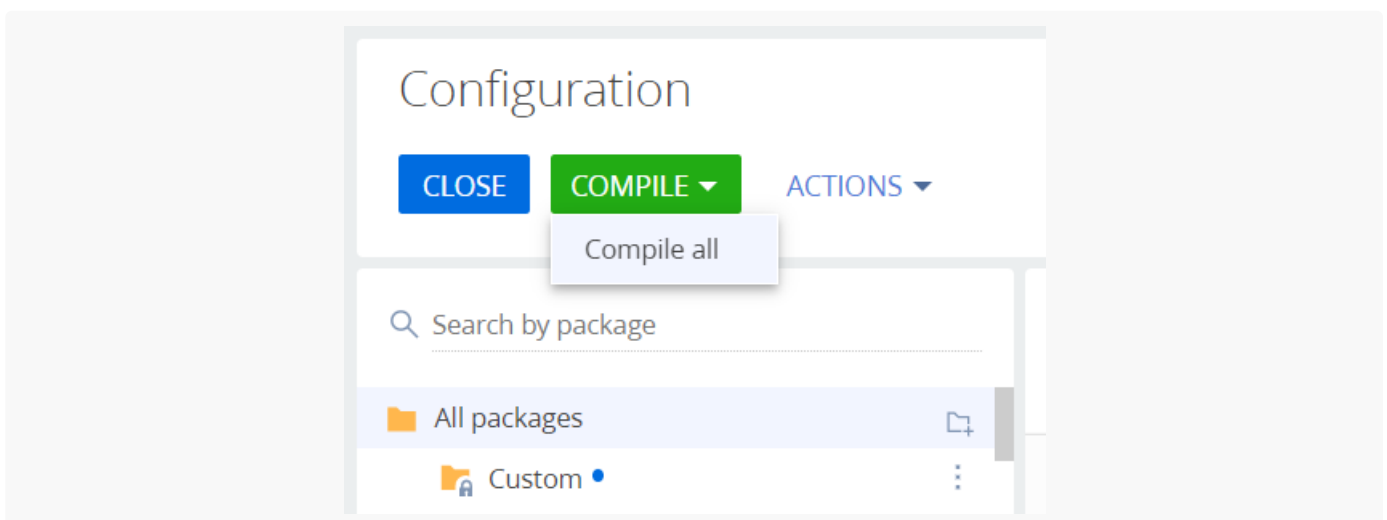
1. [Open the \[ Configuration \] section](#).
2. Select [ *Update from repository* ] in the package menu.



This starts the update of the selected package and dependent packages from the active SVN repositories.

**Attention.** If the package dependencies are located in inactive SVN repositories, a message about package update error appears. The package dependency hierarchy is updated when a package is updated. As such, activate the SVN repositories that can contain dependent packages.

3. Click [ *Compile all* ]. This is required to [generate static content](#).




## Commit a package to the SVN repository

**Package committing** involves saving the package changes to the SVN repository. Only the package for which

you execute the commit action is committed to the SVN repository. Changes to other configuration packages are not committed.

Commit the package in the following **cases**:

- You [create](#) a custom package.
- You add new configuration elements to the package.
- You modify the existing package configuration elements.
- You delete the package configuration elements.
- You [modify the package properties](#).

The **repository** appears next to the package name. Hold the pointer over the  icon to view the **name of the connected repository**.

The following data is displayed for **uncommitted custom packages**:

- Package name.
- Name of the SVN repository where to commit the package. The revision number of the package in the SVN repository is not specified and added after the package is committed.

Uncommitted custom packages are locked by default.




The following data is displayed for **committed custom packages**:

- Package name.
- SVN repository name.
- Last revision number of the package in the SVN repository

The display style of a committed custom package that remains unchanged is the same as a base package.

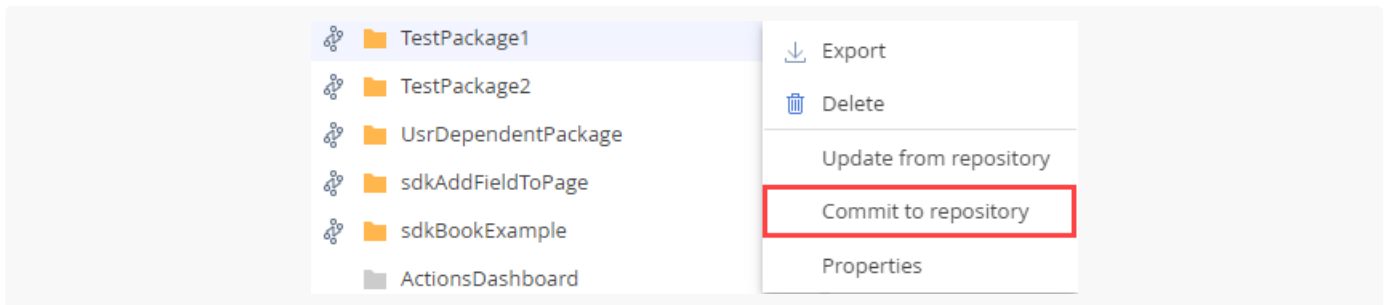


If you change a custom package, for example, add schemas or change properties, Creatio displays  next to the package name.

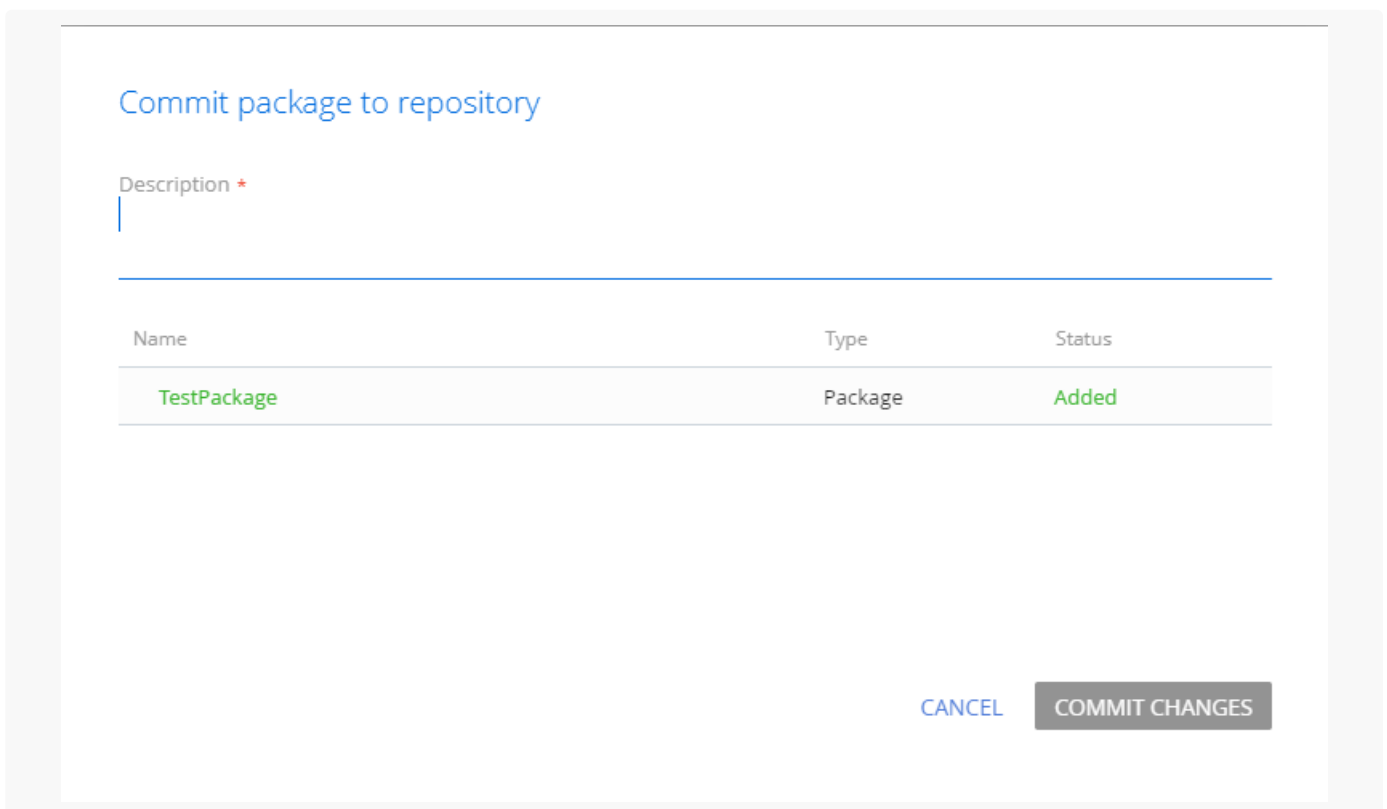
**Attention.** If you remove a configuration element from a package, the package looks like it was not changed.

To **commit a package to the SVN repository**:

1. [Open the \[ Configuration \] section.](#)
2. Select [ *Commit to repository* ] in the package menu.



This opens the change committing dialog.



3. Add a comment to the package commit in the required [ *Description* ] field. We recommend describing the changes in the package compared to the last commit. The changes in the commit package are displayed in the bottom.

After you click the [ *Commit changes* ] button, the package is committed and changes become available to other Creatio users.

**Attention.** The package is committed to the SVN repository specified in the package properties. You can only commit a package to an active SVN repository.

After the package is committed to the SVN repository, the package as well as its configuration elements are unlocked. The configuration elements also become editable for other Creatio users.

