



Creatio

Update guide

ACCELERATE 

Contents

1. How to update	2
2. How to create database backup.....	3
3. Installing updates.....	6
4. Stopping the site.....	12
5. Website starting, compilation and verification	13

This guide covers the process of updating Creatio to version 7.15.2. Our team at Creatio is constantly working to deliver advanced capabilities to automate your sales, service, and marketing processes. The new features included in Creatio version 7.15.2. can be found in the [release notes](#).

You can update to version 7.15.2 from any version.

NOTE

The update requires NET Framework of version 4.7.2 and Runtime and Developer Pack installed on the application server. [Download .NET Framework 4.7.2](#). Update the Developer pack. [Download Developer pack](#).

Before starting the update, go to the [Configuration] section and run the **[Generate source code for all items]** action, then run the **[Compile all items]** action. If executing these actions causes any errors, you will need to correct them before starting the update.

NOTE

The process differs for MS SQL Server, Oracle Database and PostgreSQL.

How to update

We recommend updating in two stages:

1. First update on a **pre-production site** with a copy of the current database
2. If the first stage was completed successfully, update the **production** version of the application.

ATTENTION

Update of the production site should not be carried out during business hours, as the site will be unavailable.

The update process consists of the following general steps:

1. **Create a copy of the database and the binary files** of the production site which will be required to deploy the pre-production site. To create a backup copy of the binary files, archive them in any other directory. Read more about creating a DB backup further - "[How to create database backup](#)".
2. **Create a new pre-production site in IIS**. Application deployment is described in the "[Deployment procedure](#)" article.
3. **Install the update** on the pre-production site. For more information, see "[Installing updates](#)".
4. **Verify that the pre-production site is fully operational**. If the primary and frequently used functions run without errors, you can begin updating the production site. For more information, see "[Website starting, compilation and verification](#)".
5. **Create copies of the database and application**. You will need them to return to a working version in case of problems. For more information, see "[How to create database backup](#)".
6. **Stop the production site of the application**. For more information, see "[Stopping the site](#)".
7. **Install the update on the production site**. For more information, see "[Installing updates](#)".
8. **Run the website and verify that the updated version is operational**. For more information, see "[Website starting, compilation and verification](#)".

If your application operates in the **web farm** mode, perform additional steps after you complete the update of the pre-production site and one of the production sites:

1. Set the **Data Source** and **Initial Catalog** values in the Terrasoft.Tools.WorkspaceConsole.exe.config file.
2. **Disable** all sites except for the ones that have been upgraded.

3. **Copy** the contents of the myapp\webapp\conf folder from the upgraded site to the disabled sites.
4. **Enable all sites.**

NOTE

To enable domain-based authentication in bpm'online, transfer Windows authentication settings to the updated application. Learn more in the ["How to setup windows authentication"](#) article.

ATTENTION

If the update process has failed, stop the procedure and contact the customer support. Provide the folder with update log: \Terrasoft.WebApp\DesktopBin\WorkspaceConsole\Log.

How to create database backup

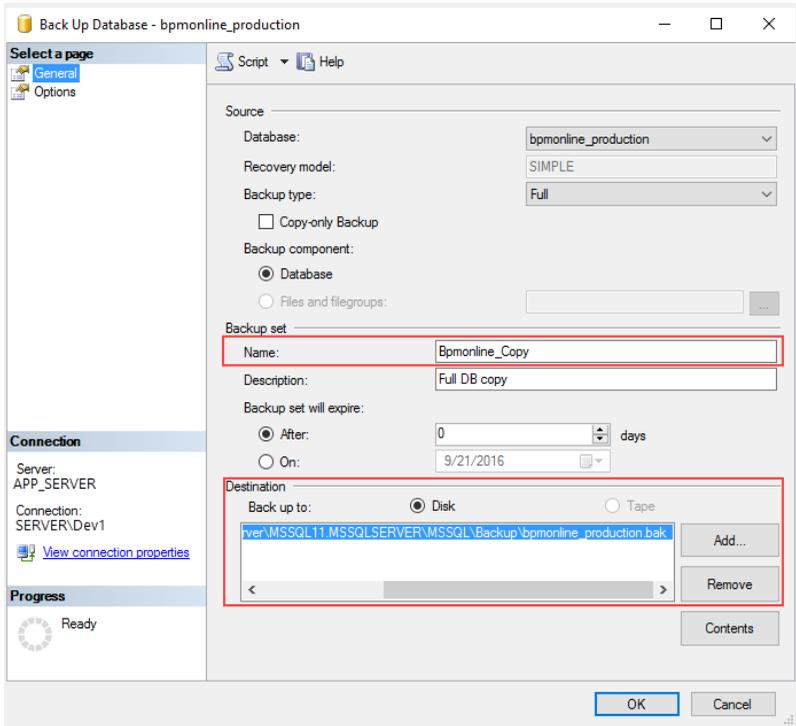
You will need a backup copy of your production database to create a test site before applying the update to the production site, as well as to roll back the update in case of problems (e.g., compatibility with customizations).

NOTE

If you deploy a backup copy of the production database and you want to disable your integrations in it, run the Disable_Synchronization.sql script. [Download the script.](#)

Creating database backup for MS SQL SERVER

1. Run Microsoft SQL Server Management Studio.
2. Select the [Back Up] command under the [Tasks] section of the context menu of the application database catalog.
3. Specify the name of the database copy and the directory in which the backup will be created. Click the [OK] button to start the backup process.

Fig. 1 Creating database backup**NOTE**

Make sure the directory for the database backup copy already exists. The SQL server has no rights to create catalogs.

When updating the Creatio production version, we recommend creating a copy of the application using any file manager.

To open a database backup:

1. Log in to Microsoft SQL Studio.
2. Create a new database if you need to extract only certain data from the backup, or select an existing database if you need to restore all data.
3. Select the [Restore database] command in the right-click menu of the database.
4. Specify path to the backup file in the opened window.
5. Click [OK] and wait for the restoration process to complete. Learn more in the "Installing Creatio" article.

Creating Oracle Database backup

1. Connect to the Oracle server using the SqlPlus utility.

```
sqlplus "SYS/SYS_PASSWORD@ORACLE_HOST:ORACLE_PORT/SERVICE_NAME AS
SYSDBA"
```

- SYS_PASSWORD – a password for authorization on the Oracle server
- ORACLE_HOST – Oracle server address
- ORACLE_PORT – Oracle server port
- SERVICE_NAME – Oracle service name

2. Execute the following SqlPlus commands:

```
CREATE OR REPLACE DIRECTORY DIRECTORY_ALIAS AS
'PATH_TO_BACKUP_DIRECTORY';
GRANT READ, WRITE ON DIRECTORY DIRECTORY_ALIAS to BACKUP_SCHEMA_NAME;
```

- DIRECTORY_ALIAS – an alias for the directory where the backup copy will be placed
- PATH_TO_BACKUP_DIRECTORY – full path to the directory where the backup copy will be placed
- BACKUP_SCHEMA_NAME – name of the schema for which the backup is made.

3. Back up your schema using the expdp utility:

```
expdp "BACKUP_SCHEMA_NAME/BACKUP_SCHEMA_PASSWORD@//
ORACLE_HOST:ORACLE_PORT/SERVICE_NAME" SCHEMAS=BACKUP_SCHEMA_NAME
DIRECTORY=DIRECTORY_ALIAS dumpfile=BACKUP_FILE_NAME NOLOGFILE=YES
```

- ORACLE_HOST – Oracle server address
- ORACLE_PORT – Oracle server port
- SERVICE_NAME – Oracle service name
- DIRECTORY_ALIAS – an alias for the directory where the backup copy will be placed
- BACKUP_SCHEMA_NAME – name of the schema for which the backup is made.
- BACKUP_SCHEMA_PASSWORD – password for the schema for which the backup is made.
- BACKUP_FILE_NAME – name of the file where the schema will be exported.

As a result, the expdp utility will create a backup copy of the BACKUP_SCHEMA_NAME schema with the BACKUP_FILE_NAME in the PATH_TO_BACKUP_DIRECTORY directory.

Deployment of the backup copy is covered in the ["installing Creatio"](#) article.

Creating the backup copy of PostgreSQL database

To create a backup copy of the database, you need to use the **pg_dump** utility. It is located in the PostgreSQL software setup folder.

1. Enter the data base connection password in the environment variable:

```
set PGPASSWORD=pg_password ("env PGPASSWORD=pg_password" - for linux)
```

2. Run the following command:

```
"C:\PostgreSQL\pg_dump.exe" --host=#ServerIP# --port #ServerPort# --
username #SysUserName# --format=c --blobs --verbose -c -f
#BackupFilePath# #DatabaseName#
```

- **ServerIP** – PostgreSQL server address.
- **ServerPort** – PostgreSQL server port.

- **SysUserName** — name of the PostgreSQL system user (you specify it when installing the PostgreSQL server).
- **SysUserPassword** — password of the PostgreSQL system user (you specify it when installing the PostgreSQL server).
- **BackupFilePath** – full path to the directory where the backup copy will be placed.
- **DatabaseName** – name of the data base, whose backup is being made.

As a result of the utility operation, the backup of the data base will be created in the **BackupFilePath** directory.

Deployment of the backup copy is covered in the “[installing Creatio](#)” article.

Installing updates

To install the update:

1. Delete the temporary Creatio files from the production site catalog.

The path to the files to be deleted is specified in the “tempDirectory” setting, the “compilation” block. You can find the setting in the web.config loader and web.app.

If the tempDirectory setting is not specified, clear the files stored at the following addresses (where %ApplicationName% is the name of website in IIS):

- 64 bit
 - %windir%\Microsoft.NET\Framework64\v4.0.30319\Temporary Files%ApplicationName% ASPNET
 - %windir%\Microsoft.NET\Framework64\v4.0.30319\Temporary Files%ApplicationName%_0 ASPNET
- 32 bit
 - %windir%\Microsoft.NET\Framework\v4.0.30319\Temporary Files%ApplicationName% ASPNET
 - %windir%\Microsoft.NET\Framework\v4.0.30319\Temporary Files%ApplicationName%_0 ASPNET

ATTENTION

You can install the update even if your application server is not connected to the Internet. In this case, you will need a computer that is connected to the Internet. Use this computer to perform all the instructions of the “Installing updates” step up to p.4 (including p.4). After you perform p.4, switch to the application server. To learn more about switching to the application server, use the instruction below.

2. Download the UpdaterService.ps1 for uploading distributions. [Download the service](#).
3. Open the **downloader.json** script file to edit. Populate its parameters with the corresponding values:
 - **WebRootDirectory** – path to the website root folder. Please do not specify this parameter if your application server is not connected to the Internet and you are downloading data from another computer.
 - **WorkDirectory** – path to the folder, where all installation packages and the update utility will be stored.
 - **Site** - name of your site in IIS.

- **Product** – name of the product where the website is deployed. Copy the needed name from the **enum Product** block in the Downloader.ps1 file.
- **DbEngineType** – type of the DBMS. Copy the needed name from the **enum Product** block in the Downloader.ps1 file.
- **VersionBuild** – current version of you application.
- **SkipBinary** – manually add this parameter with the “true” value to downloader.json if you already have distribution files and you do not want to download them again.
- **ConnectionString** – database connection string. Copy this string from your connection to the database. Please do not specify this parameter if your application server is not connected to the Internet and you are downloading data from another computer.
- **CurrentSchemaName** – current database schema. For MS SQL – “dbo”, for PostgreSQL – “public”, for Oracle – your schema.
- **RedisServer** – Redis server.
- **RedisDB** – the number of Redis database.
- **RedisPort** – Redis port.

Please note that the below parameters are required. If you do not specify them in downloader.json, they will be defined automatically at the **WebRootDirectory** path.

- **ConnectionString**
- **CurrentSchemaName**
- **DbEngineType**
- **RedisServer**
- **RedisDB**
- **RedisPort**

Example of a downloader.json script file configuration:

a. On your application server:

```
{
  "WebRootDirectory": "c:\\inetpub\\wwwroot\\delivery",
  "WorkDirectory": "c:\\temp\\delivery"
  "Site": "<name of your site in IIS>"
  "Product": "Studio"
  "VersionBuild": "7.14.1.935"
}
```

b. If your application server is not connected to the Internet and you are downloading data from another computer:

```
{
  "WebRootDirectory": "c:\\temp\\delivery",
  "Site": "<name of your site in IIS>"
  "Product": "Studio"
  "DbEngineType": "MSSQL"
  "VersionBuild": "7.14.1.935"
  "CurrentSchemaName": "dbo",
  "RedisServer": "localhost",
  "RedisDb": 1,
}
```

```
"RedisPort": 6379.
}
```

ATTENTION

Please note that in all paths the character “\” is indicated as “\\”.

4. Run the **Downloader.ps1** Powershell script. **Powershell 5.1** must be installed to run the script.

When you run the script, the following elements are created in the folder that you specified in the `workDirectory`: an “InstallPackages” folder containing a set of update packages, a folder containing the update utility and a `Start.bat` file.

The following folder/file structure will be used in the `InstallPackages` folder:

- A separate folder appears for each version in the alphabetical order as per the update schedule.
- Each version folder contains:
 - an archive with files of the corresponding version (the archive will be automatically unpacked to the `App`, `Pkg`, `Template` folders during the update process).
 - A “Scenario” folder.

Example of the file structure:

- 13/7 < 0/31.
 - 7.13.0.284_SalesTeam_SoftKey_MSSQL_ENU.zip
 - Scenario
- 13/7 < 1/31.
 - 7.13.1.769_SalesTeam_SoftKey_MSSQL_ENU.zip
 - Scenario
- 13/7 < 2/31.
 - 7.13.2.934_SalesTeam_SoftKey_MSSQL_ENU.zip
 - Scenario

ATTENTION

If your application server is not connected to the Internet and you are downloading data from another computer, switch to the application server. To do this:

1. Copy the created folder (the one that contains the `InstallPackages` set of update packages, the folder with the Updater update utility and the `Start.bat` file) and transfer it to the server where your application is placed.
2. Open the `install.xml` script file from the Updater folder for editing. Populate its parameters with the corresponding values:

`WebRootDirectory` – path to the website root folder.

`InstallPackagesPath` – path to the folder, where all installation packages and the update utility are stored.

`ConnectionString` – database connection string. Copy this string from your connection to the database.

3. Save the changes.

5. In the folder containing installation packages and the update utility (your “`workDirectory`” folder), find and run the **Start.bat** file.

During the execution, a number of commands will be run sequentially. Please wait for the entire process to complete.

If you receive the "Installed components/software have an outdated/incorrect version. In order to start the Creatio update, please upgrade it" error, check the list of software components (above the error text) for missing software and install it.

If the update process has failed, stop the procedure and contact the customer support. Provide the folder with update log: [The CreatioUpdater catalog]\InstallPackages\%Version%\Log.

NOTE

To restore the database, use the database copy created earlier.

6. If you previously made additional changes in your web.config files (e.g., integrations or external services), transfer these changes after a successful update manually.

Updating to 7.8.2

Preparing the development environment (for version 7.8.2)

If SVN is used in the development process, it is necessary to additionally run the FlatPackageConverter utility that:

1. Rebuilds the structure of the resources. Resources will be associated with a package, not a schema.
2. Corrects invalid SVN properties of resource files.
3. Corrects invalid names of cultures.
4. Removes invalid resource files (empty or without items) and resources that do not have schemas in the same package.

Download and unzip [FlatPackageConverter.rar](#). Open example.bat file that runs the FlatPackageConverter.exe utility in any text editor and edit required parameters:

```
FlatPackageConverter.exe relocateResources
--repositoryUri=<repositoryUri> --version=<version> --user=<user>
--password=<password> --commentFilePath=<commentFilePath>
--copyPath=<copyPath> --kind=<kind>
--needDeletedWrongResources=<needDeletedWrongResources>
```

Description of the used parameters is available in the table below.

Parameter	Function
<repositoryUri>	Path to your SVN repository.
<version>	Version of the packages to be updated.
<user>	SVN user login.
<password>	SVN user password.
<commentFilePath>	Path to the local file that contains a comment to commit changes in SVN.
<copyPath>	Path to a folder where the utility will copy its temporary files.

<kind>	Determines the type of resource movement:
	<ul style="list-style-type: none"> • "package" — transfer resources on the package level (7.8.1 → 7.8.2) with the addition of the manager name. • "schemas" — transfer resources from schemas to packages with the addition of the manager name (7.7.x → 7.8.2). • "oldschemas" — transfer resources from schemas to packages (7.7.x → 7.8.0).
<needDeletedWrongResources>	True — deletes all invalid resource files; records all information in deletion log. Recommended option.
	False — displays information about errors and warnings that you must then manually correct.

Example of updating Creatio from 7.8.1 to 7.8.2:

```
FlatPackageConverter.exe relocateResources
--repositoryUri=http://tscore-svn:8050/svn/ts5conf/branches/TestPS_14
--version=7.8.0 --user=<Login> --password=<Password>
--commentFilePath=C:\Temp\111.txt --copyPath=C:\Temp\1 --kind=package
--needDeletedWrongResources=True
```

NOTE

In this example the parameter `--version=7.8.0` is not an error, as it specifies the version of the packages that need to be changed to update.

ATTENTION

After the update is complete, the utility displays a list of errors and warnings, as well as records detailed information on each package in the log. Errors are resources in packages without schemas, warnings are empty resource files that do not contain a single element. To remove invalid resource files, use the `needDeletedWrongResources=True` parameter.

Updating to 7.8.4

Preparing the development environment (for version 7.8.4)

If SVN is used in the development process, it is necessary to additionally run the FlatPackageConverter utility that will delete outdated schema resource files.

Download and unzip [FlatPackageConverter.rar](#). Open `example.bat` file that runs the FlatPackageConverter.exe utility in any text editor and edit required parameters:

```
<Full path to FlatPackageConverter.exe> deleteSchemaLegacyResources
--repositoryUri=<repositoryUri> --version=<version> --user=<user>
--password=<password> --commentFilePath=<commentFilePath>
--copyPath=<copyPath>
```

Description of the used parameters is available in the table below.

Parameter	Function
<repositoryUri>	Path to your SVN repository.

<version>	Version of the packages to be updated.
<user>	SVN user login.
<password>	SVN user password.
<commentFilePath>	Path to the local file that contains a comment to commit changes in SVN.
<copyPath>	Path to a folder where the utility will copy its temporary files.

Example of updating Creatio from 7.8.3 to 7.8.4:

```
FlatPackageConverter.exe deleteSchemaLegacyResources
--repositoryUri=http://tscore-svn:8050/svn/ts5conf/branches/TestPS_14
--version=7.8.0 --user="Login" --password="Password"
--commentFilePath=C:\Temp\111.txt --copyPath=C:\Temp\1
```

NOTE

In this example the parameter `--version=7.8.0` is not an error, as it specifies the version of the packages that need to be changed to update. Be sure to specify your repository login and password.

ATTENTION

After the update is finished, the utility will display a list of errors and warnings. Detailed information on each package will be available in the `Out.txt` file in the same directory. The error "svn: E195022: is locked in another working copy" in the log indicates that some files in the repository are locked in the version control system. To correct the error, unlock the files and run the utility again.

Updating to 7.9.0

Preparing the development environment (for version 7.9.0)

If SVN is used in the development process, it is necessary to additionally run the `FlatPackageConverter` utility that will remove incorrectly generated metadata for localized strings.

Download and unzip [FlatPackageConverter.rar](#). Open `example.bat` file that runs the `FlatPackageConverter.exe` utility in any text editor and edit required parameters:

```
<Full path to FlatPackageConverter.exe> fixResourcesInMetadata
--repositoryUri=<repositoryUri> --version=<version> --user=<user>
--password=<password> --commentFilePath=<commentFilePath>
--copyPath=<copyPath>
```

Description of the used parameters is available in the table below.

Parameter	Function
<repositoryUri>	Path to your SVN repository.
<version>	Version of the packages to be updated.
<user>	SVN user login.
<password>	SVN user password.

<commentFilePath>	Path to the local file that contains a comment to commit changes in SVN.
<copyPath>	Path to a folder where the utility will copy its temporary files.

Example of updating Creatio from 7.8.4 to 7.9.0:

```
FlatPackageConverter.exe fixResourcesInMetadata
--repositoryUri=http://tscore-svn:8050/svn/ts5conf/branches/TestPS_14
--version=7.8.0 --user="Login" --password="Password"
--commentFilePath=C:\Temp\111.txt --copyPath=C:\Temp\1
```

NOTE

In this example the parameter `--version=7.8.0` is not an error, as it specifies the version of the packages that need to be changed to update. Be sure to specify your repository login and password.

ATTENTION

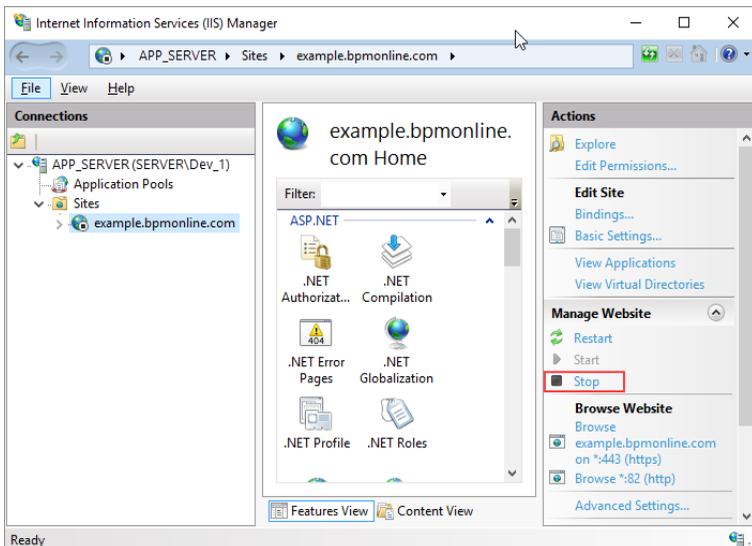
After the update is finished, the utility will display a list of errors and warnings. Detailed information on each package will be available in the `Out.txt` file in the same directory. The error "svn: E195022: is locked in another working copy" in the log indicates that some files in the repository are locked in the version control system. To correct the error, unlock the files and run the utility again.

Stopping the site

To avoid data loss, we recommend you to stop the production website before updating.

1. Open the Internet Information Services Manager (IIS).
2. Stop the web site using the [Stop] command in the [Actions] area;

Fig. 2 Stopping the site in IIS



Website starting, compilation and verification

After the update process has completed, start the Creatio website in IIS, compile the application and test the website by opening it in a web browser.

1. Open the Internet Information Services Manager (IIS).
2. Start the web site using the [Start] command in the [Actions] area.
3. Open the test site with the help of the [Browse] command in the [Actions] area;

Fig. 3 Opening test website in a web browser



ATTENTION

If the update process has failed, stop the procedure and check the update logs located at: [The CreatioUpdater catalog]\InstallPackages\%Version%\Log. If you receive the "Error:Error" type of a bug in base packages, contact the support and provide the folder with log records. If errors occur in custom packages, contact the developers of these packages.

4. Generate client static content by running the [Compile all items] action in the [Configuration] section.
5. Open the application in a web browser, verify that your routine bpm'online operations are performed correctly.
6. If everything works properly, you can delete the backup application and database.

Updating to 7.10.1

Additional settings

After updating to 7.10.1, data enrichment service address and IP has changed. The new address is: **api.bpmonline.com** (ip: 185.3.141.228). You must open the access to the service on your servers (port 443).

The previously valid address **cloud-service.bpmonline.com** (ip: 188.99.10.125) is now used only for the Marketing Creatio (for example, for bulk email).

Updating to 7.11.1 (for the Financial Services Creatio, lending edition)

Additional actions for the Financial Services Creatio, lending edition

If you are using Financial Services Creatio, lending edition and have a custom application page (FinApplicationPage), perform the following steps after the base update scenario completes.

1. Update packages from SVN.

2. Run the UpdateFinAppLendingPage utility.

Running the UpdateFinAppLendingPage from Windows command prompt:
UpdateFinAppLendingPage.exe "Path to the downloaded operational copy of svn".

Example: UpdateFinAppLendingPage.exe C:\MyPackagesFromSvn\.

3. Commit changes to SVN.
4. Update the configuration from SVN by running the [Restore from repository] command in the [Configuration] section.

Enjoy the new version of Creatio!