Creatio Academy

Update guide

Update guide

Version 8.0

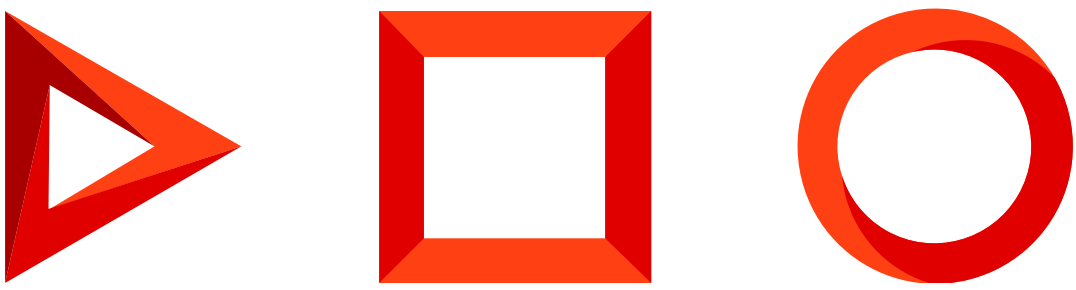


Table of Contents

Update guide

This guide covers the process of updating Creatio application to the latest available version. Our team at Creatio is constantly working to deliver advanced capabilities to automate your sales, service and marketing processes. You can learn more about the new features included in the Creatio latest version in the release notes.

These instructions are universal. You can use them to update to the latest version of the application regardless of which version you are using.

Please note that if you are using version 7.8.0 and lower, you should contact Creatio technical support for further instructions.

> **Note.** To update Creatio, you need .NET Framework of version 4.7.2 and Runtime. Update them first if required: download .NET Framework 4.7.2. Update the Developer pack: download Developer pack.
>
> Starting from version 7.16.0 you need to download and install .NET Core SDK 3.1 x64 to ensure correct compilation: download .NET Core SDK 3.1 x64. After you install the SDK, restart the host.

If your application has installed cultures that are not used by company employees, we recommend that you remove these cultures before starting the upgrade. This will reduce the amount of data that needs to be downloaded as part of the update.

Also, 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 taking these steps causes any errors, you will need to fix them before starting the update. Learn more in a separate article: Clean up the drive space.

> **Note.** Some of the steps of the process are different for Microsoft SQL Server, Oracle Database and PostgreSQL.

How to update

We recommend updating in two stages:

1. First, perform the update on a **pre-production site** with a copy of your current Creatio database.
2. If the first stage completes successfully, perform the update of the **production** site of the application.

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

The update process consists of the following 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. You can find out more about DB backups below – “Creating 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 "Creating 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 updated.
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 Creatio, transfer Windows authentication settings to the updated application. Learn more in the "How to setup Windows authentication" article.

Learn more about the web farm mode in the user documentation: Application server web farm.

Creating database backup

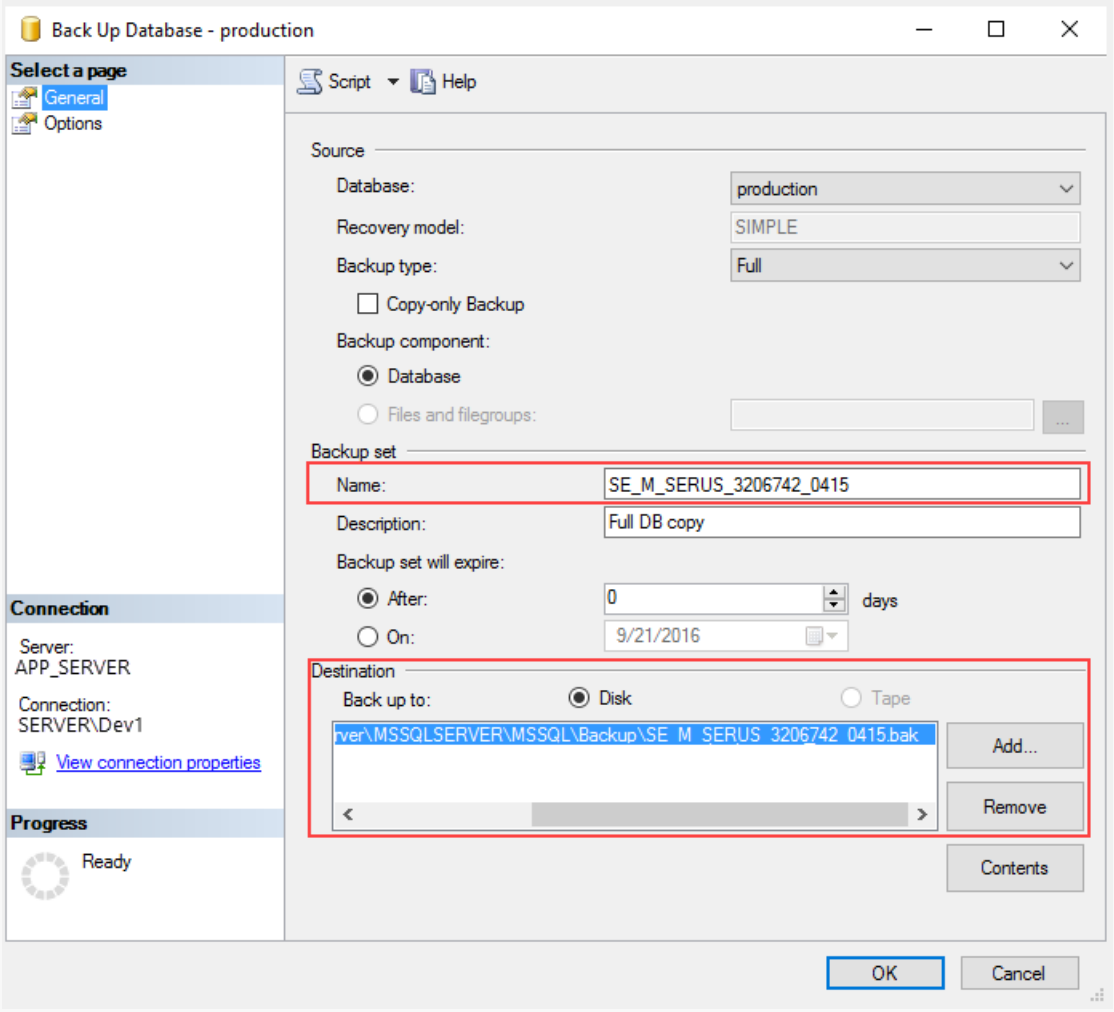
You will need a backup copy of your production database to create a test site before applying the update to the production site, as well as to roll back the update in case of problems (e.g., compatibility with customizations).

> **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 Microsoft SQL Server database backup

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 [*OK*] to start the backup process (Fig. 1).

Fig. 1 Back up the database



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 the path to the backup file in the opened window.
5. Click [*OK*] and wait for the restoration process to complete.

Learn more in the "Deploy Microsoft SQL Database for 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" SC
```

- 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_SCHEMANAME schema with the BACKUP_FILE_NAME in the PATH_TO_BACKUP_DIRECTORY directory.

Deployment of the backup copy database is covered in the "Deploy Oracle Database for Creatio" article.

Creating PostgreSQL database backup

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 ("export PGPASSWORD=pg_password" – for linux)
```

2. Run the following command:

```
"C:\PostgreSQL\pg_dump.exe" --host=#ServerIP# --port #ServerPort# --username #SysUserName#
```

- **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 database is covered in the "Deploy PostgreSQL Database (Linux)" and "Deploy PostgreSQL Database (Windows)" articles.

Installing updates

To install the update:

When your application server is not connected to the Internet

You need a machine that is connected to the Internet to run the update process. Follow the instructions below up to and including step 3 on that machine. After you complete step 3, switch to the application server.

1. Get the distribution downloading service: download the service.
2. Open the **downloader.json** script file to edit. Populate its parameters with the corresponding values:

   Example of a downloader.json script file configuration:

```
{

     "webRootDirectory": "c:\\inetpub\\wwwroot\\delivery",

     "WorkDirectory": "c:\\temp\\delivery",

     "Site": "<name of your site in IIS>",
```

```
    "Product": "Studio",

    "VersionBuild": "7.14.1.935"

}
```

- **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 DbEngineType** block in the Downloader.ps1. file.
- **VersionBuild** – current version of your application.
- **SkipBinary** – if you already have distribution files and you do not want to download them again, you can manually add this parameter value to the downloader.json file (if not already included) and set it to “true”.
- **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. It is “dbo” for Microsoft SQL, “public” for PostgreSql and your schema for Oracle .
- **RedisServer** – Redis server.
- **RedisDB** – the number of Redis database.
- **RedisPort** – Redis port.
- **UseAWS** – if you use AWS, manually add this parameter value and set it to “true”.

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

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

Example of the downloader.json file:

```
{
"WebRootDirectory": "c:\\inetpub\\wwwroot\\delivery",
"WorkDirectory": "c:\\temp\\delivery",
```

```
"Site": "site name in IIS",
"Product": "Studio",
"VersionBuild": "7.14.1.935"
}
```

When your application server is not connected to the Internet

Use the following example of the downloader.json file if your application server is not connected to the Internet and you are transferring data from another machine.

```
{

     "workDirectory": "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 the “\” character is escaped with an extra backslash. It should look like this: “\\”.

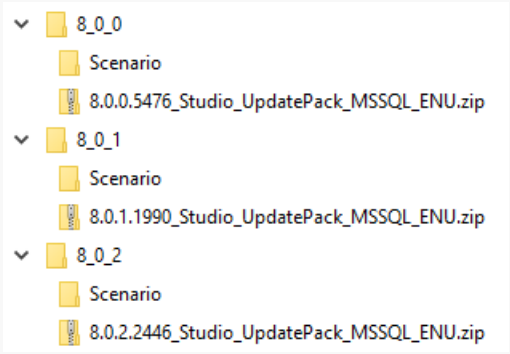
3. Run the **Downloader.ps1** Powershell script. **Powershell 5.1 or higher** must be installed to run the script.

   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 Updater utility.

   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 and Pkg folders during the update process) and a “Scenario” folder.

Fig. 2 Example of the file structure



When your application server is not connected to the Internet

Before you take step 4, switch to the application server. To do this:

a. Copy the created folder (the one that contains the InstallPackages set of update packages, the folder with the Updater utility and the Start.bat file) and transfer it to the server where your application is placed.

b. 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.

f. Save the changes.

Before updating to version 7.15.3

> **Note.** Take these steps before you update from Creatio version 7.15.0 to version 7.16.0 as well.

Before you update to version 7.15.3, make sure no ReportService customization has been performed in your configuration.

To do this, download and execute one of the following scripts depending on which DBMS you use: download the script for Microsoft SQL, download the script for PostgreSQL, download the script for Oracle.

If the script returns no results, proceed with the following update steps.

If the script returns a list of replaced schemas, compare them with the out-of-the-box schema.

a. Check which packages contain the schemas from the selection below:

- If the selection contains any of the schemas from the base packages: verify whether the base package is blocked. The "Maintainer" value of the base package should be "Terrasoft".
- If the selection only contains schemas from custom packages, analyze the selection schemas.

d. If the custom schema contain "using Terrasoft.Reports:", delete "using" (the action is recommended but will not affect the update).

e. If the replaced schema references other report designers, but the logic of using it does not change as compared to the out-of-the-box logic, use the instruction.

f. If the replaced schema references other report designers and the logic of issuing the reports is changed, analyze the replaced schemas and adapt them to work with the package (the package is part of your configuration starting from version 7.15.0).

4. We recommend to revise “prerequisites” container in the Updater.dll.config file (located in the Updater folder) before running the update.

```
<prerequisites>

      <add name="NetFramework" version="4.7.2"/>

      <add name="NetCore" version="2.2.300"/>

      <add name="VisualCpp" version="Microsoft Visual C++ 2010 Redistributable (x64)"/>

      <add name="Database" version="mssql=2012 SP3;oracle=10;postgresql=9.6"/>

      <add name="IIS" version="7.0.0"/>

</prerequisites>
```

Each entry corresponds to the minimum required version of a Creatio component:

- NetFramework – .NET Framework version
- NetCore – .NET Core version
- VisualCpp – Visual C++ version
- Database – database version
- IIS – IIS server version

The Updater utility runs version checks based on the “version” values of these entries.

5. Find and run the **Updater.exe** file in the folder with installation packages and the update utility (your “workDirectory” folder) as an administrator.

Please note that the user must have permissions to modify the directories for an update to run smoothly.

Note. To run the file as an administrator, right-click it and select «Run as administrator» in the context menu.

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

If the update completed sucessfully, flush the Redis server cache.

Make sure you are using the latest versions of microservice components after the successful update. If not, update them as well.

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 Updater catalog\InstallPackages\%Version%\Log**.

Updating to version 7.18.0 and later

You need to re-order licenses with every update to version 7.18.0 and later. Order and upload licenses after updating to **version 7.18.0**. You can also order and upload licenses before initiating the update to **version 7.18.1** and later. Learn more about generating a license request and uploading licenses to Creatio: Creatio licensing.

Rolling back the Creatio application before re-updating

1. Delete all application files and copy the binary files to the directory from the backup that you made before the update.

2. Restore the database using the database backup that you made before the update.

3. Go to the InstallPackages directory. In its child directories with the version numbers, delete the App, Log, Pkg, Temp, and Template directories if they exist.

4. Make the necessary corrections that you were recommended by the Creatio technical support.

After you complete all of the steps above, you can restart the system update process.

Updating to 7.8.2

If you use SVN in the development process, additionally run the FlatPackageConverter utility, which:

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.zip. Open the example.bat file that runs the FlatPackageConverter.exe utility in any text editor and edit the 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:
	• "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 to the deletion log. Recommended option.
	False – displays information about errors and warnings that you should then fix manually.

Example of updating Creatio from 7.8.1 to 7.8.2:

```
FlatPackageConverter.exe relocateResources

--repositoryUri=http://svn-server/svn/ts5conf/branches/TestPS_14

--version=7.8.0 --user=login --password=password
```

```
--commentFilePath=C:\Temp\111.txt --copyPath=C:\Temp\1 --kind=package --needDeletedWrongResou
```

Note. In this example, the “--version=7.8.0” parameter is not an error, as it specifies the version of the packages that need to be changed to update. Specify your repository login and password.

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

If you use SVN in the development process, additionally run the FlatPackageConverter utility after you perform the update. It will delete the outdated files for schema resources.

Download and unzip FlatPackageConverter.zip. Open the example.bat file that runs the FlatPackageConverter.exe utility in any text editor and edit the 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 can be view on the table.

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://svn-server/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 “--version=7.8.0” parameter is not an error, as it specifies the version of the packages that need to be changed to update. 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 as the utility. 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 fix the error, unlock the files and run the utility again.

Updating to 7.9.0

If you use SVN in the development process, additionally run the FlatPackageConverter utility after you perform the update. It will remove the incorrectly generated metadata for localized strings.

Download and unzip FlatPackageConverter.zip. Open the example.bat file that runs the FlatPackageConverter.exe utility in any text editor and edit the 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 can be view on the table.

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://svn-server/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 “--version=7.8.0” parameter is not an error, as it specifies the version of the packages that need to be changed to update. 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 as the utility. 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 fix the error, unlock the files and run the utility again.

If you use Redis Sentinel

If your Redis configuration is fail-proof, please contact Creatio support for more information about Redis Sentinel (that ensures Redis remains reliable) before updating to version 7.15.2 and later.

The Redis Sentinel mechanism will be retired in Creatio version 7.18.3. We recommend switching to Redis Cluster after updating Creatio to version 7.18.0 and later.

Updating with product upgrade

Note. To perform a product upgrade, e.g., from Sales Creatio Enterprise edition to the CRM product

lineup, use the distribution files of the new product of the same version as the current product.

1. Download the updater service: download.
2. Locate the **Updater.dll.config** file in the Updater folder.
3. Add the following entry to the appSettings block:

```
<add key="Feature-ExtendProduct" value="true"/>
```

4. Update Creatio as per the standard procedure. The old product will be upgraded to a new product of the same version.

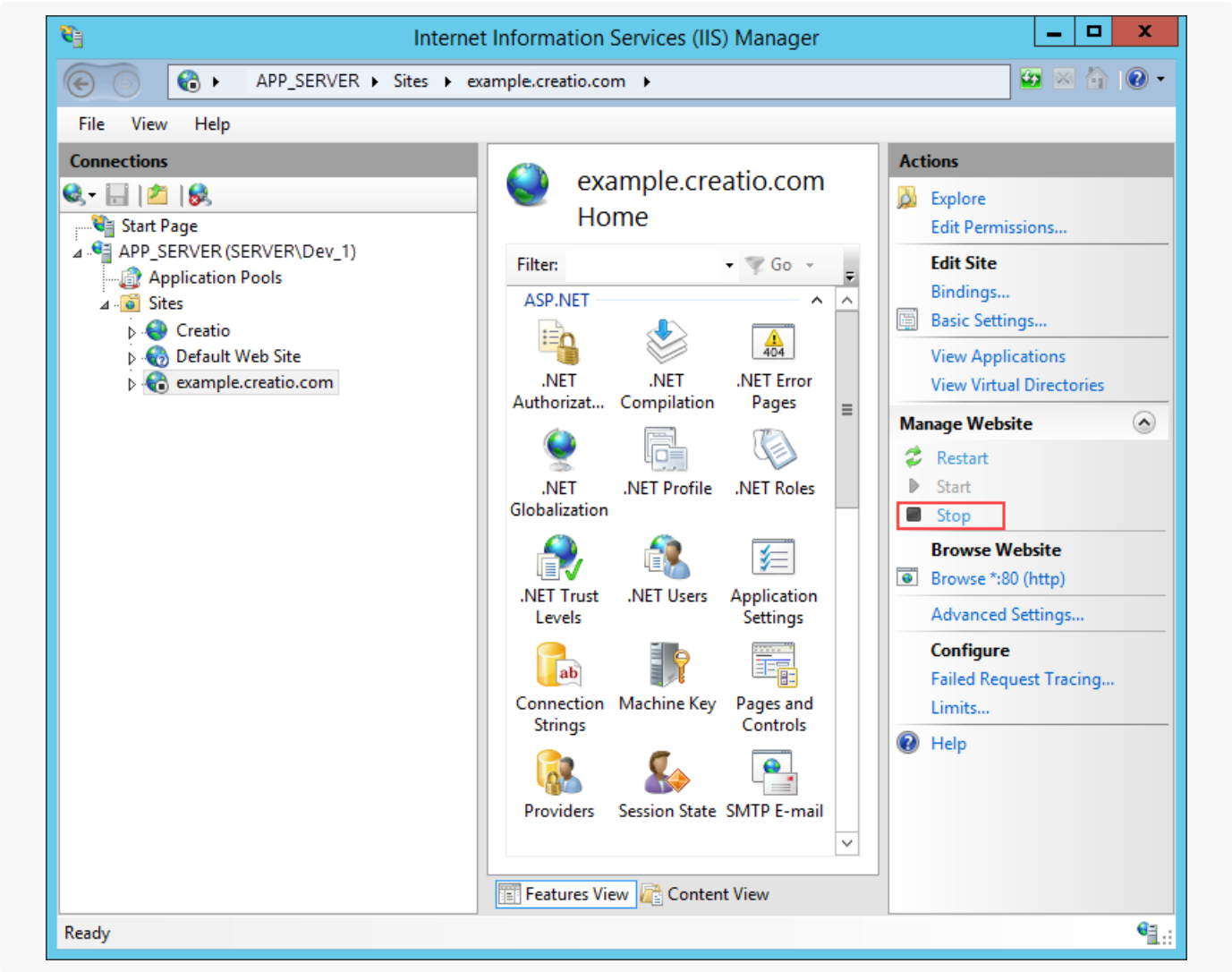
If you previously made additional changes in your web.config files (e.g., integrations or external services), transfer these changes after a successful update manually.

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 production web site using the [*Stop*] command in the [*Actions*] area (Fig. 3).

Fig. 3 Stop the website in IIS

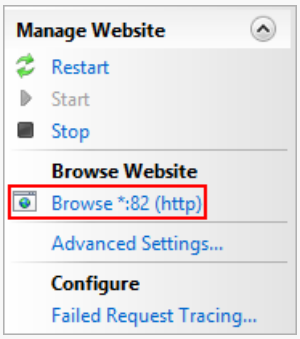


Website starting, compilation and verification

After the update process is complete, start the Creatio website in IIS, compile the application and test if the website functions as intended.

1. Open the Internet Information Services Manager (IIS).
2. Start the web site using the [*Start*] command in the [*Actions*] area.
3. Open the web site using the [*Browse*] command in the [*Actions*] area (Fig. 4).

Fig. 4 Open a 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.

1. To re-generate client static content, run the [*Compile all items*] action in the [*Configuration*] section.
2. Open the application in a web browser and verify that your routine operations function correctly.

   Note. An update can upset ISS session state settings. This will make impossible to log in to the mobile application. After an update, makes sure the “Use Cookies” mode is enabled in the ISS cookies settings.
3. If everything works properly, you can delete the backup application and database.

Updating to 7.10.1

Note. Take these steps before you update from Creatio version 7.10.0 to version 7.11.0 as well.

After updating to 7.10.1, the data enrichment service address and IP will change. The new address will be: **api.creatio.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 Creatio Marketing (such as for bulk email).

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

Note. Take these steps before you update from Creatio version 7.11.0 to version 7.12.0 as well.

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.

Updating to 7.16.0

After you update to version 7.16.0, deploy the Exchange Listener synchronization service to make sure the IMAP/SMTP and Exchange services work correctly.

Updating to 7.16.1 and 7.16.2

Note. Take these steps before you update from Creatio version 7.16.0 to version 7.17.0 as well.

Before you update to versions 7.16.1 and 7.16.2 make sure you do not have any customization using the obsolete library and the Terrasoft.Mail.SmtpClient class in your configuration.

To do this, run the script.

Script for Microsoft SQL:

```
SELECT SysSchema.Name AS SchemaName FROM SysSchema (NOLOCK)

WHERE SysSchema.Id IN (

SELECT SysSchemaId FROM SysSchemaSource

WHERE SysSchemaId IN (

SELECT

ss.Id

FROM SysSchema ss WITH (NOLOCK)

INNER JOIN SysPackage sp WITH (NOLOCK) ON ss.SysPackageId = sp.Id

WHERE sp.Name NOT IN ('Base', 'ProcessDesigner', 'NUI', 'SSP')

AND sp.Maintainer != 'Terrasoft'

AND ss.ManagerName NOT IN ('ClientUnitSchemaManager', 'DcmSchemaManager', 'PageSchemaManager'

)

AND (Source LIKE '%MailBe%' OR (Source like '%SmtpClient%' AND Source like '%Terrasoft.Mail%'
```

```
)
```

Scripts for Oracle:

First, run the below script to create a procedure:

```
BEGIN
EXECUTE IMMEDIATE 'CREATE OR REPLACE FUNCTION blob_to_clob (blob_in IN BLOB)
RETURN CLOB
AS
     v_clob    CLOB;
     v_varchar VARCHAR2(32767);
     v_start      PLS_INTEGER := 1;
     v_buffer  PLS_INTEGER := 32767;
BEGIN
 IF blob_in IS NULL THEN
  RETURN NULL;
    END IF;
     DBMS_LOB.CREATETEMPORARY(v_clob, TRUE);

     FOR i IN 1..CEIL(DBMS_LOB.GETLENGTH(blob_in) / v_buffer)
     LOOP

        v_varchar := UTL_RAW.CAST_TO_VARCHAR2(DBMS_LOB.SUBSTR(blob_in, v_buffer, v_start));
           DBMS_LOB.WRITEAPPEND(v_clob, LENGTH(v_varchar), v_varchar);
          v_start := v_start + v_buffer;
     END LOOP;

   RETURN v_clob;

END blob_to_clob;';
END;
```

Afterward, run the main script:

```
SELECT "SysSchema"."Name" AS "SchemaName" FROM "SysSchema"
 WHERE "SysSchema"."Id" IN (
  SELECT "SysSchemaId" FROM "SysSchemaSource"
  WHERE "SysSchemaId" IN (
   SELECT
    ss."Id"
   FROM "SysSchema" ss
   INNER JOIN "SysPackage" sp ON ss."SysPackageId" = sp."Id"
   WHERE sp."Name" NOT IN ('Base', 'ProcessDesigner', 'NUI', 'SSP')
    AND sp."Maintainer" != 'Terrasoft'
    AND ss."ManagerName" NOT IN ('ClientUnitSchemaManager', 'DcmSchemaManager', 'PageSchemaMa
```

```
    )
    AND (blob_to_clob("Source") LIKE '%MailBe%'
     OR (
      blob_to_clob("Source") LIKE '%SmtpClient%'
      AND blob_to_clob("Source") LIKE  '%Terrasoft.Mail%'
     )
    )
  );
```

If the script does not return anything, proceed with the following update steps.

If the script returns a list of schemas:

1. Check the packages for the schemas from the selection:
   - If the selection contains base package schemas, check whether the base package is blocked and if the "Maintainer" value of the base package is set to "Terrasoft".
   - If the selection only contains the custom package schemas, analyze the selection schemas.
2. If the custom schema contains the "using" directive but does not involve its types, delete "using".
3. If the custom schema contains obsolete directives, change the mechanism as per the following instructions:
   - Sending emails from existing accounts
   - Sending emails using the explicit account credentials

Updating to 7.17.1

Note. Take these steps before you update from Creatio version 7.17.0 to version 7.18.0 as well.

If your application contains Marketing Creatio, make sure you re-license your web site before you update to version 7.17.1. Otherwise, the update can lead to errors in the calculation of active contact licenses which may affect sending marketing emails. Learn more about application licenses in the Creatio licensing article.

Enjoy the new version of Creatio!