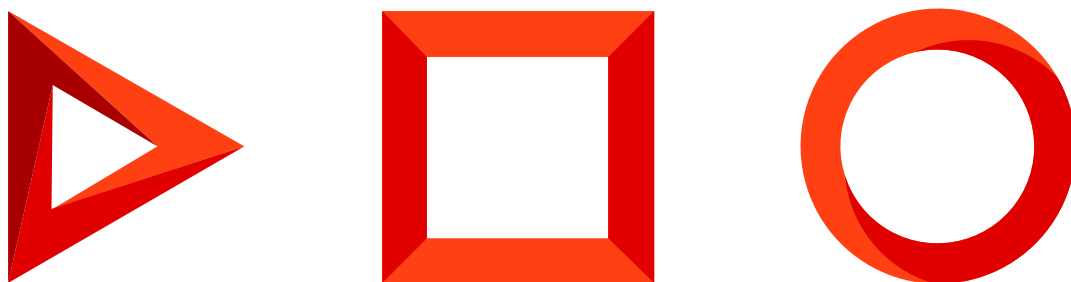


Creatio maintenance

Version 7.17



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

Clean up the drive space	4
Possible reasons for quick database growth	4
Creatio database cleanup tools	5

Clean up the drive space

PRODUCTS: ALL CREATIO PRODUCTS

It is important to delete outdated and irrelevant records in a timely manner when you maintain a large database. As a result, the database server will use less space on its drive, which improves the database performance

Possible reasons for quick database growth

The following factors may lead to rapid database growth:

- Incorrectly set up or redundant **record permissions**. For example, the record permissions are set individually for a large number of users not arranged in groups. In this case, we recommend changing the settings and updating Creatio access permissions. Learn more in a separate guide: [Access management](#).
- Lack of restrictions on **file upload** to Creatio. The files can be added by employees, uploaded as part of the email synchronization, or attached to self-service portal messages. We recommend limiting the size of uploaded files to 10 Mb. You can change this value in the “Attachment max size” (“MaxFileSize” code) system setting. We also recommend checking the uploaded files for relevancy and deleting the irrelevant files regularly. You can implement this feature using a business process.
- The **synchronization of all emails** from user mailboxes. We recommend selecting only the mailbox folders whose letters you must process in Creatio. For example, Important or Starred. Learn more in a separate article: [Receive emails in Creatio](#).
- **Processes traced** for prolonged periods. Processes are usually debugged on a development environment or a test site. If you must collect the debugging information on a production site, we recommend turning the tracing off as soon as you finish analyzing the process execution problems. Learn more in a separate article: [Trace process parameters](#).
- Incorrectly set up **business process execution logic** that causes the process to have the “Running” status for longer than needed. In this case, Creatio will keep the temporary files related to the process execution. We recommend modeling business processes so that they have definite completion conditions and do not have the “Running” status for longer than several hours. Learn more in a separate article: [View process execution data](#).
- Incorrectly set up **data reading in business processes**. Creatio stores the values the business process retrieves as part of the [*Read data*] element execution in temporary data tables until the process finishes. If the business process does not require the values of all object columns, we recommend specifying the exact list of values to read. This will help you to reduce the amount of temporary data stored in Creatio greatly. Learn more in a separate article: [\[*Read data* \] process element](#).
- Excessive **change logging**. We recommend turning on the change logging only for the sections in which you must track the data dynamics. For example, the product catalog. If you want to save the record change information, clean up the change log from the irrelevant data regularly. Learn more in a separate article: [Clear the change log](#).
- Incorrectly set up **integration of external services** with Creatio. If the external services send a Creatio request that lacks the ForceUseSession header, they will need to re-authenticate. Learn more in developer documentation: [EntityDataService.svc web service \(OData 3\)](#).

- **Connecting unnecessary cultures.** This will cause Creatio to download translation files, which are then updated in each new Creatio version. We recommend connecting only those cultures that your company's employees use and removing any obsolete cultures.

Creatio database cleanup tools

There are several ways to clean up the drive space in Creatio:

- Configure the process log archival and automatic cleanup.
- Clear the change log.
- Delete section records.
- Delete data as part of a business process.

Automatic process log cleanup

Creatio can log the processes it runs. Logs help to track diagram bottlenecks and optimize them, as well as to analyze the efficiency of your employees. Creatio automatically archives the logs older than 30 days to reduce the amount of used space. The archived records remain available for 360 days before Creatio deletes them automatically. You can manage when to archive the records and for how long to store them.

Learn more in a separate article: [Archive the process log records.](#)

Clear the change log

Clear the change log history to avoid storing irrelevant records in Creatio. We recommend clearing the change log regularly to ensure that the [*Change log*] section contains only the currently relevant information.

Learn more in a separate article: [Clear the change log.](#)

Delete section records

Creatio may store irrelevant section records. You can delete such records from any Creatio section individually or in bulk. If the record you want to delete is connected to other sections, Creatio will ask you to review the connections and confirm what to delete. You can delete all of the information or only the selected record and keep the connected data.

Learn more in a separate article: [Delete linked records.](#)

Delete data as part of a business process

Automate the drive cleanup with business processes. Use the [*Delete data*] process element to delete one or more records that meet specific conditions from any Creatio object. For example, create a business process that deletes all scheduled activities that were canceled. Set this process to run:

- **On a timer**, at a specific time. This is useful if you want to run the process periodically, e. g. once a month, and during the minimum load period, e. g. at night.
- **After a specific event**. This is useful if you want to run the process automatically and only when there is Creatio data to delete.

- **Manually.** This is useful if you want to run the process at any required moment.

Learn more in a separate article: [\[Delete data \] process element](#).

Delete unused cultures

You can delete cultures that you no longer plan to use to free up additional disk space. To delete a culture, use the following script:

Microsoft SQL

```

IF OBJECT_ID('tempdb..#UsedCultures') IS NOT NULL
    DROP Table #UsedCultures
-- Get a list of used cultures
SELECT DISTINCT cult.Id
INTO #UsedCultures
FROM SysCulture cult
INNER JOIN SysAdminUnit au
    ON au.SysCultureId = cult.Id
INSERT INTO #UsedCultures
    (Id)
SELECT
    SysSettingsValue.GuidValue
FROM
    SysSettingsValue
INNER JOIN SysSettings
    ON SysSettings.Id = SysSettingsValue.SysSettingsId
WHERE
    SysSettings.Code = 'PrimaryCulture'

-- Get a list of tables from which to delete data
DECLARE TableNamesCursor CURSOR FOR
SELECT
    t3.TABLE_NAME AS ChildTableName
FROM INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS AS t1
    INNER JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE AS t2 ON t1.UNIQUE_CONSTRAINT_NAME = t2.C
    INNER JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE AS t3 ON t1.CONSTRAINT_NAME = t3.CONSTRAI
WHERE
    t2.TABLE_NAME = 'SysCulture'
    and t2.COLUMN_NAME = 'Id'
    and t3.COLUMN_NAME = 'SysCultureId'
DECLARE @TableName SYSNAME
OPEN TableNamesCursor
FETCH NEXT FROM TableNamesCursor INTO @TableName
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT @TableName
    DECLARE @Sql NVARCHAR(MAX);

```

```

SET @Sql = 'DELETE FROM ' + @TableName + '
          WHERE SysCultureId NOT IN (SELECT Id FROM #UsedCultures)';
PRINT @Sql
EXECUTE sp_executesql @Sql

    FETCH NEXT FROM TableNamesCursor INTO @TableName
END
CLOSE TableNamesCursor
DEALLOCATE TableNamesCursor
DELETE FROM SysCulture
WHERE Id NOT IN (SELECT Id FROM #UsedCultures)
IF OBJECT_ID('tempdb..#UsedCultures') IS NOT NULL
    DROP Table #UsedCultures

```

PostgreSQL

```

BEGIN;
-- Get a list of used cultures
CREATE TEMP TABLE "UsedCultures" ON COMMIT DROP AS
SELECT DISTINCT cult."Id"
FROM "SysCulture" cult
INNER JOIN "SysAdminUnit" au
ON au."SysCultureId" = cult."Id";
INSERT INTO "UsedCultures" ("Id")
SELECT "SysSettingsValue"."GuidValue"
FROM "SysSettingsValue"
INNER JOIN "SysSettings"
ON "SysSettings"."Id" = "SysSettingsValue"."SysSettingsId"
WHERE "SysSettings"."Code" = 'PrimaryCulture';
-- Get a list of tables from which to delete data
DO $$
DECLARE
TableNamesCursor REFCURSOR;
TableName varchar;
BEGIN
OPEN TableNamesCursor FOR
SELECT kcu.TABLE_NAME
FROM INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE ccu
INNER JOIN INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS rc
ON ccu.CONSTRAINT_CATALOG = rc.UNIQUE_CONSTRAINT_CATALOG
AND ccu.CONSTRAINT_SCHEMA = rc.UNIQUE_CONSTRAINT_SCHEMA
AND ccu.CONSTRAINT_NAME = rc.UNIQUE_CONSTRAINT_NAME
INNER JOIN INFORMATION_SCHEMA.KEY_COLUMN_USAGE kcu
ON kcu.CONSTRAINT_CATALOG = rc.CONSTRAINT_CATALOG
AND kcu.CONSTRAINT_SCHEMA = rc.CONSTRAINT_SCHEMA
AND kcu.CONSTRAINT_NAME = rc.CONSTRAINT_NAME
WHERE ccu.COLUMN_NAME = 'Id'

```

```
AND ccu.TABLE_SCHEMA = 'public'  
AND ccu.TABLE_NAME = 'SysCulture';  
  
--Delete localizations  
LOOP  
FETCH TableNamesCursor INTO TableName;  
EXIT WHEN TableName IS NULL;  
RAISE NOTICE 'Deleting from table %', TableName;  
EXECUTE format('DELETE FROM %I WHERE "SysCultureId" NOT IN (SELECT "Id" FROM "UsedCultures")', T  
END LOOP;  
CLOSE TableNamesCursor;  
END $$;  
DELETE FROM "SysCulture"  
WHERE "Id" NOT IN (SELECT "Id" FROM "UsedCultures");  
COMMIT
```