d.velop

d.velop cloud migration: Administration

Table of Contents

1. d.velop cloud migration: Administration

1.1. Basic information about the application

This chapter provides you with product instructions and general information.

1.1.1. About d.velop cloud migration

d.velop cloud migration is a migration tool that lets you migrate documents and dossiers from a source system to a d.velop documents cloud system. Possible source systems are all d.velop documents systems (on-premises) from version 7.2.3 or an export from any third-party system in the prescribed format.
In this process, the application loads an export, transforms the metadata based on the configured mappings, performs the migration to the target cloud system and logs the migration in a migration report for documentation purposes. Only the categories to which all of the following properties apply are migrated:

- The categories have been provided via export.
- The categories have been configured in mapping.
- The categories have been configured in the migration.

1.2. Installation and uninstallation

This chapter provides you with information about installing the application.

1.2.1. System requirements

d.velop cloud migration is provided in the target cloud system as an app. When you book the d.velop cloud migration app, you can use the app. In the app, you can download d.velop cloud migration connector. This application is executed locally. The following specifications therefore only apply to d.velop cloud migration connector.

Operating systems

- Microsoft Windows Server 2012
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019
- Microsoft Windows Server 2022
- Microsoft Windows Server 2025
- Microsoft Windows 10, version 1607 or higher
- Microsoft Windows 11

1.2.2. Installing d.velop cloud migration connector

In the app, you can download d.velop cloud migration connector. This process involves a single EXE file. You save this file locally on a Windows client or Windows server. Start the file using the Windows command prompt (cmd.exe). Installation is not required.

You can find the call for starting the connector on the page for downloading the connector in the app.

1.2.3. Installing updates for d.velop cloud migration connector

d.velop updates and enhances the cloud app on an ongoing basis. In the cloud, you are notified when an update of the connector is available.

If you are using a very old version of the connector, you are prompted to update it before you can continue migrating documents. You update the connector using the latest EXE file, which is available in d.velop cloud migration.

This is how it works

1. Close the connector.
2. Download the new EXE file in d.velop cloud migration.
3. Save the old EXE file on your local system so you can reset the Connector if necessary.
4. Swap the EXE file.
5. Start the connector with the new file.

1.2.4. Rolling back an installation of d.velop cloud migration connector

Replace the new EXE file with the EXE file that you backed up previously and restart the connector.

1.2.5. Uninstalling d.velop cloud migration connector

You uninstall d.velop cloud migration connector by deleting the downloaded EXE file. In addition, delete the folder **cloudmigration** under C:/ProgramData/d.velop.

1.2.6. Enabling the default port for d.velop cloud migration connector

d.velop cloud migration connector accesses the target system through port 443. You do not need to enable an incoming port.

You can find more information about enabling IP addresses here: https://help.d-velop.de/docs/de/pub/leitfaden-cloudmigration-documents/cloud/detaillierte-anleitungen/welche-ip-adressen-muss-ich-freischalten-damit-ich-die-migration-api-verwenden-kann

1.3. Performing a document migration with d.velop cloud migration (d.velop documents on-premises)

1.3.1. Uploading the d.3 backup file in the cloud migration toolkit

In order for the migration system to know what the source system looks like, a d.3 backup file is required as an export from the source system. You can obtain the file in d.3 Administration under **System Settings > Data Backup > Export**.

Upload the d.3 backup file to the Cloud Migration Toolkit.

1.3.2. Performing the category mapping

1. Select the source and target in d.velop cloud migration.
2. Map the source categories to the categories for the target system in d.velop cloud migration.
   - For 1:1 mappings, use the option **Apply automatic mapping**.
     a. Map the properties for each category.
     b. Complete the mapping for each category using the lock icon.

> **Note**
> Only completed categories are taken into account when migrating documents and dossiers.

1.3.3. Mapping the users and groups

1. Map the users and groups in the cloud migration toolkit. Provide any missing users with a CSV file in the toolkit.

2. Open the user and group mapping in d.velop cloud migration
3. Apply the mapping from the cloud migration toolkit.
4. Configure a default user to prevent migration errors.

> **Note**
> The default user is a user that is configured if a user field in the document metadata is empty but is required for the migration. For example, dossier links in older systems may not have any users but may still be required in the cloud. If a default user has been configured, the default user is used in this case.

1.3.4. Downloading and configuring d.velop cloud migration connector

1. Download the connector application on the **Download and connect** page.
2. Save the .exe file that is provided on a Windows server or client (see system requirements).
3. Start the exe file using the Windows command prompt (cmd.exe).
4. Using the following command enables the initial configuration: **CloudmigrationConnector.exe --configure**

You then configure the API key to access the target system, repository ID, base URI and export directory in this sequence.

1.3.5. Configuring the migration

You configure and perform the migration via the **Document migration** page in the target system. On this page, you can select the categories to be included for the migration. There are also other configuration options that you can activate at the push of a button.

1. **Categories**: Open the categories and select the categories that you want to migrate. Only categories that were marked as completed in the category mapping are displayed.
2. **Common settings**: Define the maximum number of processes to be used in parallel to upload the documents. Also specify the number of files per upload batch that will be migrated simultaneously in a batch per process.

> **Note**
> No general recommendation can be made regarding the choice of number of processes and batch size. A larger number of processes and a larger batch size usually result in a higher migration speed. However, other factors such as the utilization of local hardware or the available Internet bandwidth should also be considered when making your selection so that the customer's day-to-day business is not restricted by the migration.

3. **Validation options**: **Ignore hash values** enables you to ignore a missing filehash or D3Hash so that documents without these hash values can also be migrated.

> **Warning**
> When you activate the option, it cannot be guaranteed that a hash value check is performed. In any case, the specified file size is compared with the real file size.

4. **Transformation options**:
   The transformation options add and correct specifications, such as file size and name. This resolves migration errors.
   - **Add file size, if missing**: The file size is calculated and saved.
   - **Correct file size, if incorrect**: The file size is corrected and saved.
   - **Add file name, if missing**: If a file name property is missing, the file name of the metadata file is used.

- **Correct version number**: Incorrect version numbers (file IDs) are corrected in the export before the migration.

5. **Advanced settings**
   a. **Re-upload previously migrated documents**: Activate this option if you want to migrate all documents again, including successfully migrated documents (e.g. for delta migrations). By default, successfully migrated documents are not migrated again when the connector is restarted.
   b. **Create new dossier scheme**: Activate this option after migrating the structure rules. This will create links and new dossiers.

> **Note**
> The generation takes time and delays the completion of the project. Existing links are retained. If necessary, new files are created or linked so that existing and new links exist. Link or parent properties can also be empty. In this case, the configured structure rule cannot be executed. If dossiers are migrated from the source system and the dossier links are recreated, duplicate dossiers will appear in the target system. In this case, dossiers should not be migrated.

   c. **Create new full text and rendition**: If you activate this option, the creation of a PDF and a full-text index is started after a document is migrated.

> **Note**
> The generation takes time and delays the completion of the project. This results in considerable additional costs. If the full-text index is available in the source system, this index should be adopted and generation deactivated.

6. **Debug settings**
   a. **Test run without uploading**: This option deactivates uploading to the target system so that no migration is performed; only the validation of the exported documents.
   b. **Create transformed metadata file**: Activate this option to analyse errors. A new file is created in the export directory for each document, in which the transformed metadata is saved as it is sent to the migration API.
   c. **Debug Log**: Activates the debug log for the connector to perform advanced logging. This log lets you track errors more effectively.

1.3.6. Starting the migration

1. Go to the directory of the file **CloudmigrationConnector.exe** and execute the file.
2. Start the migration from the user interface. To do so, the connector must already be configured and started.

If the connector has been started, the console displays the message “Waiting for jobs...” at regular intervals. The d.velop cloud migration app in the target system is notified about the start. The following status should then be displayed on the user interface: **Connector ready**

You can stop the connector in the Windows command prompt using **CTRL+C**.

1.3.7. Monitoring the migration

You can also monitor the document migration on the **Document migration** page without accessing the on-premises system. The page displays the progress of the migration as a percentage and shows a table list for the migration job, grouped according to categories.

1.3.8. Completing the migration and reporting

Once the migration is complete, the report is available for download in d.velop cloud migration under **Migration Report**. The report only covers the current run.

In addition, the following files are included in a ZIP file to provide further information about the migration:

- List of documents with migration errors: **migration_error_documents.csv**
- List of successfully migrated documents: **migration_successfull_documents.csv**
- List of documents imported and ignored according to the migration configuration: **migration_ignore_documents.csv**
- Subfolder for each migration run performed. The name is incremented starting from 1. For each migration run, the following files that document the configuration selected for the migration run are created.
  - Configuration settings for the migration: **settings.csv**
  - Category and property mapping used: **category_mapping.csv**
  - User and group mapping used: **user_mapping.csv**

The document status is shown in the lists **migration_error_documents.csv**, **migration_successfull_documents.csv** and **migration_ignore_documents.csv**. The following statuses are possible:

- 0: Unknown
- 1: Transformed but not uploaded (e.g. for documents that were not uploaded but have been prepared for uploading)
- 2: Uploaded
- 3: Upload error
- 4: Error during transformation (e.g. when assigning the category)
- 5: Validation error before uploading (e.g. for metadata that does not match the binary file)
- 6: Error importing the file (e.g. for metadata files that do not correspond to the expected format)
- 7: Ignored (e.g. for unassigned categories)
- 8: Document has been uploaded and the correctness check, including metadata check, has been successfully completed
- 9: Correctness check failed (e.g. for uploaded documents for which there are differences in the metadata between the uploaded data and the data in the cloud)
- 10: Document has already been uploaded in a previous run

> **Note**
> It is strongly recommended that you download the migration report and the attachments provided and keep them for migration traceability purposes. If the migration is performed for a customer, these files should be provided to the customer so that the customer can verify the correctness of the migration.

1.3.9. Migration report

After completing one or more migrations, you can generate a complete report using the migration report interface. This function performs a comprehensive inventory check as follows:

- Existence check: Checks the availability of all uploaded documents in d.velop documents based on the document ID.
- Complete documentation: Creates a comprehensive overview of the status of all migrated data from all migration runs.
- Missing documents: Identifies documents that can no longer be found (e.g., due to deletion or automatic archiving).

> **Note**
> Only those documents that were previously made available by importing the export directories are listed. A comparison is made based on the migration database. No comparison is made between the source system and the target system.

The following files are provided:

- PDF report: Migration report on all migration runs as a protected PDF file
- ZIP archive containing the following CSV files:
  - All standard document lists (successfully migrated, faulty, and ignored documents)
  - **migration_missing_documents.csv**: List of documents that were originally migrated successfully but can no longer be found in the target system.

> **Note**
> It is strongly recommended that you download the migration report and the provided appendices and keep them for migration traceability purposes. If the migration is performed for a customer, these files should be provided to the customer so that the customer can verify the correctness of the migration.

1.4. Performing a delta migration

In some migration projects, the first documents are already migrated while you are still working with the source system. In the process, new documents are created or existing documents are changed. To ensure that these changes are also transferred to the target system, there are delta migrations. These delta migrations include the newly imported or changed documents.

> **Note**
> Using the **dateOverallProc** field in the metadata file for the exported documents as a basis, d.velop cloud migration automatically detects whether a newer version of the document is available. If so, the document is migrated again. Otherwise, documents that have already been migrated successfully are not migrated again.

This is how it works

1. Perform a new export of the documents to be migrated as part of the delta migration (for instance, using the d.velop documents repo export API).
2. If necessary, edit the mappings for the categories, properties, users and groups in d.velop cloud migration.
3. If necessary, update the connector.
4. Configure the connector and adapt the export directory for the new export.
5. Configure the migration.
6. Start the migration.

1.5. Performing a third-party system migration with d.velop cloud migration

1.5.1. Preparation in d.velop documents

In the target system, define the following data before you start the migration.

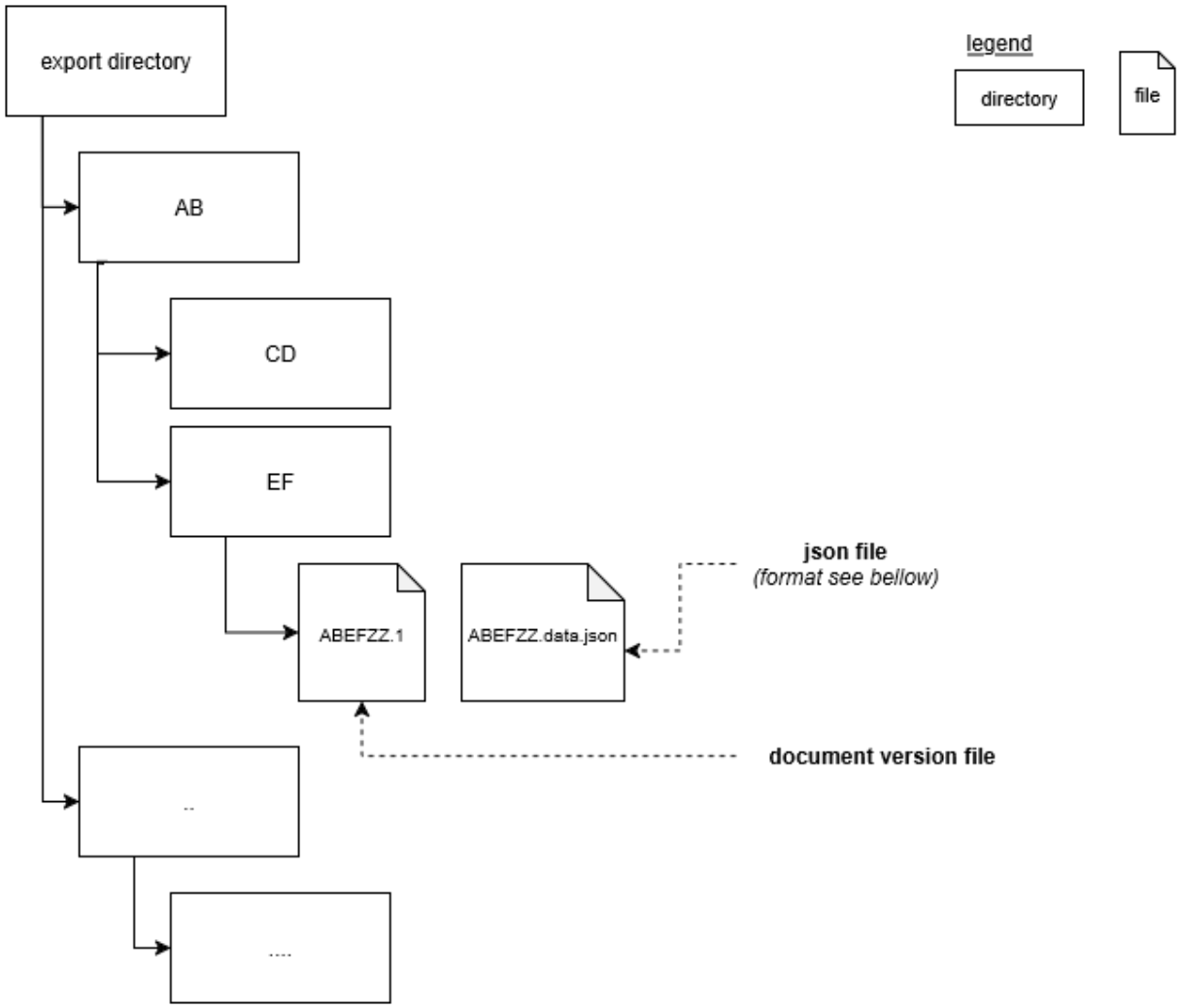
- Categories
- Properties
- User
- Groups

In the system, these data are given unique IDs. These IDs are essential as they must be used in the export of the third-party system in order to ensure the correct mapping.

1.5.2. Structure of the export directory

- The export directory can be structured in any desired manner. Recommendation: To make it easier to use, do not file too many documents in one directory. Use subdirectories instead.
- The file name is used as an ID and must therefore be unique over the complete migration. Where delta migrations are required, the same ID must be determined for one document for every export.
- A document always consists of at least one document version file (*.1 file) and a metadata file (*.data.json). If several versions belong to one document, there are “*.2”, “*.3”, *.n files as applicable.
- The file name is used to identify the metadata file which belongs to the document version. The names must therefore be the same. It is not permitted to create file names with other full stops (e.g. ABC.DEF.data.json).
  Examples of permitted file names:
  - If a document has one version and the ID **ABCD**, the file names are: **ABCD.1** and **ABCD.data.json**
  - If a document has two versions and the ID **CDEF**, the file names are: **CDFE.1, CDEF.2** and **CDEF.data.json**

Example of an export directory



1.5.3. Metadata format (JSON file)

Recommended scope of a document

```
{
    "documentType": {
        "id": "296d9d61-56b2-476e-a72c-3cbf147d98cc"        // Document
```

```
category (e.g., invoice) (ID from the target system) [Required]
    },
    "attributesById": {                                     // Document
properties [Required]
        "39d41784-eb8b-43e9-851c-c3ec3c13873c": {           // Numerical
property (e.g., invoice number) (ID from the target system)
        "number": 12345                                     // Data type
and property content
        },
        "1145c6ba-fbc8-4e5c-be53-61d60080c58a": {           // Alphanumeric
property (e.g., customer name) (ID from the target system)
        "string": "Firma GmbH"
        },
        "11a752ba-fbc8-4e5c-be53-61d60080c58a": {           // Date
property (e.g., invoice date) (ID from the target system)
        "date": "2020-10-28"
        },
        "87a752ba-fbc8-4e5c-be53-61d60080c58a": {           // Date-time
property (ID from the target system)
        "datetime": "2020-10-28T10:50:12.345Z"
        },
        "4745c614-fbc8-4e5c-be53-61d60062c58a": {           // Numeric
multiple property (e.g., order numbers) (ID from the target system)
            "numbers": {
                "1": 12345,
                "3": 456789
            }
        }
    },
    "versions": [                                          // Versions of a
document (at least one version)
      {
        "status": "DOC_STAT_RELEASE",                      // Document
status (default = DOC_STAT_RELEASE)
        "physicalVersion": {
            "fileId": 1,                                   // Version ID
(ascending, starting at 1) [Required]
            "file": {
                "sizeInByte": "49",                        // [Required]
                "fileHash": "RIPEMD256:uq03qJ9j/
h7yjOjAwwRtfICOSvVQxQIeu4Wk6rC4Cz8="   // Hash value for the correctness
of the transmission; structure: “RIPEMD256:”; algorithm: RIPEMD256, Base64
encoded
            },
            "extension": "PDF"                             // File extension
[Required]
        },
        "create": {
            "user": {
                "idpId": "090e1771-d050-46f9-8874-dcc2e425d485"       //
Creator of the version, user ID from the target system [Required]
            },
            "timestamp": "2020-11-02T15:50:37.177679Z"                //
Version timestamp [Required]
        }
```

```
      }
    ],
    "systemAttributes": {
        "number": "12345ABCDE",                                    // ID of the
source system (unique), recommended for completeness check
        "varNumber":  1,                                           // Variant
number of a document (default = 1)
        "filename": "myDocument",                                  // File name
of the document [Required]
        "dateRetention": "2027-10-28T09:50:12.345Z",               // End of
document retention period (must not be in the past)
        "dateOverallProc": "2022-10-28T09:50:12.345Z",             // Last
modification date of the document (used to identify changes when the same
document is migrated again)
        "create": {
          "user": {
              "idpId": "090e1771-d050-46f9-8874-dcc2e425d485"   // Created
by, user ID from the target system [required]
          },
          "timestamp": "2020-11-02T15:50:37.177679Z"            // Original
creation date of the document [Required]
        }
    }
}
```

Possible data types of document properties

- Single-value properties: **string, number, date, datetime**
- Multi-value properties: **strings, numbers, dates, datetimes**

JSON format with note

```
{
 "documentType": {},
 "attributesById": {},
 "versions": [],
 "systemAttributes": {},
 "notes": [
    {
     "create": {
      "user": {
       "idpId": "090e1771-d050-46f9-8874-dcc2e425d485"
      },
      "timestamp": "2023-10-28T09:50:12.345Z"
     },
     "message": "New message"
    }
 ]
}
```

JSON format with dependent file (e.g. R1) and plain text

```
{
    "documentType": {},
    "attributesById": {},
    "systemAttributes": {},
    "versions": [
```

```
    {
    "status": "DOC_STAT_RELEASE",
        "physicalVersion": {},
            "extension": "PDF"
            "dependentFiles": {
                "R1": {                          // Dependent file R1
                 "file": {
                  "sizeInByte": "256",
                  "fileHash": "RIPEMD256:uq03qJ9j/
h7yjOjAwwRtfICOSvVQxQIeu4Wk6rC4Cz8="
                 }
                }
              },
              "ocr": "The extracted fulltext"     // Full text, if
available in the source system
        },
        "create": {
            "user": {
                "idpId": "090e1771-d050-46f9-8874-dcc2e425d485" // User ID
from the target system
            },
            "timestamp": "2020-11-02T15:50:37.177679Z"
        }
      }
    ]
}
```

JSON format with history

```
{
    "documentType": {},
    "attributesById": {},
    "versions": [],
    "systemAttributes": {},
    "history": [
        {
            "eventName": "myNewEvent",
            "timestamp": "2021-08-13T09:36:13.008445Z",
            "user":
            {
                "idpId": "090e1771-d050-46f9-8874-dcc2e425d485"     // User
ID from the target system
            },
            "details": [
                {
                "detailName": "myNewDetail",
                "string": "MyString"
                },
                {
                "detailName": "myNewDetail1",
                "numeric": 1.1
                },
                {
                "detailName": "myNewDetail2",
                "integer": "1"
                },
```

```
                {
                "detailName": "myNewDetail3",
                "datetime": "2021-08-13T09:36:13.008445Z"
                }
            ]
        }
    ]
}
```

1.5.4. Configuration in d.velop cloud migration

1. Open the source and target user interface.
2. Select **Third party system** as source.
3. The pages for mapping categories and users are skipped, as the prepared JSON files already contain the correct IDs. Manual mapping is therefore not required.
4. Then, continue with the regular migration steps from Downloading and configuring d.velop cloud migration connector.

> **Important**
> During a third-party system migration, no assignment of users, groups, categories, and properties from the source system to the target system is performed in d.velop cloud migration. The IDs of the target system must already be defined in the existing export. To ensure the traceability of the migration from a third-party system, it is necessary to log the mappings used when creating the export from the source system and to store them with the migration reports from d.velop cloud migration.

1.6. Technical process for a document migration

In this section, you can learn more about the technical process involved in a document migration with d.velop cloud migration. It will give you a better understanding of the order in which the steps are performed.

Once the migration starts, the following steps are performed:

1. The connector queries the mapping from the source category to the target category from the d.velop cloud migration app.
2. The connector queries the mapping from the source users to the target users from the d.velop cloud migration app.
3. The connector queries the other configurations (e.g. **Ignore hash values**, etc.) from the d.velop cloud migration app.
4. The mappings and configurations are saved in the migration database.
5. The connector imports the document export from the hard disk.
6. The imported documents are divided into batches that are then processed in multiple steps.
7. For each batch, the connector transforms the metadata in the export based on the mappings of the categories, the mappings of the users and the other migration settings in the memory. Advanced transformations must be explicitly activated via the d.velop cloud migration user interface and are also logged in the migration report. The following transformations are possible:
   - Defining the file size based on the export, if the file size is empty.
   - Correcting the file size, if the file size in the metadata differs from the actual export.
   - Setting the file name property on the file name for the metadata file, if the file name is empty.
   - Bringing the numbering for the document versions into sequence (e.g. 1,2,3,4 instead of 1,3,5,6).
8. The connector validates the metadata and document data for each batch. If there is any invalid data, validation errors occur and the affected documents are not included in the further migration process, without correction. The errors are logged in the migration database. Possible validation errors:

- The documents or metadata are not in the correct format (e.g. corrupt JSON file, incorrect data types in the metadata, incorrect JSON format).
- The documents and metadata do not match (e.g. the number of exported versions does not match the number in the metadata or the file size in the metadata does not match the file size in the export).
- Mandatory fields such as the hash values, file size or file name are missing in the metadata.
  Exception: The hash values of each individual version of a document are unavailable and you explicitly set missing hash values to be ignored. In this case, validation errors do not occur and documents without hash values are still included in the further process. This is recorded in the migration report.

9. The connector uploads the documents and metadata for each batch to the target system via the migration API of the d.velop cloud. The documents and metadata are transmitted as they are provided by the export and adjusted using the migration configuration. You can activate two options for the upload on the user interface:
   a. The full-text index and the dependent PDF files for the documents can be recreated in the cloud. In this case, any full-text indexes and dependent PDF files in the export are discarded. The use of this option is only recommended if there is not yet any plain text.
   b. The links to documents and dossiers can be recreated based on the structure definition defined in the cloud. Existing links are maintained. New dossiers are created or linked so that there are pre-existing and new links.

   These options are not activated by default. If you activate these options, this activation is also logged in the migration report. During the upload, the connector evaluates the response of the migration API (success or error) and saves these responses in the migration database.
10. Once the documents and metadata have been successfully uploaded to d.velop cloud, an additional check is performed. The connector queries the export API from the target system to obtain the metadata stored in the cloud for the uploaded documents.
    A comparison is then performed between the metadata that was originally exported and the metadata currently stored in the cloud. Among others, the following properties are checked:
    - Hash values for each document version
    - Retention date
    - Import date
    - Technical document properties

    If there are any discrepancies between the metadata that is delivered and the metadata that is stored, the corresponding errors are displayed in the correctness check. You can view the errors in d.velop cloud migration. You can also see the errors in the error log for the connector as well as in the migration report.
    This extended validation enables downstream quality assurance and ensures that the migration was not only technically successful, but also carried out correctly in terms of content.
11. Once a migration run is complete (all the batches have been processed), the connector creates the migration report locally as a PDF file, along with additional attachments (e.g. list of successfully migrated documents, list of incorrect documents, list of ignored documents) in a zip file. The checksums for the attachments are recorded in the PDF file so that any change to the files can be tracked later. All the described activities are logged in a migration database. For each document, a **<filename>.data.log** file that logs all the activities in the documents on an ongoing basis is created in the export directory.
12. Under **Migration Report**, you can manually start a complete migration report. The connector performs a comprehensive check of all documents ever recorded in the migration database, regardless of whether these documents were successfully migrated, are faulty, or were ignored.
    - All documents from the migration database are written to CSV files.
    - For all documents that have been successfully uploaded, the connector checks in batches whether their document ID (DocID) exists in the target system.
      - Documents that are available are listed in the **migration_success_documents.csv** file.
      - Documents that are not (or no longer) available appear in the file **migration_missing_documents.csv**.

- A complete migration report will be generated as a PDF file, along with a ZIP file containing all CSV files.
- The PDF file and the ZIP file are being uploaded and will be availöable for you to download.

> **Note**
> Only documents that are available in the database are taken into account. There is no comparison between the source and target systems.

1.7. Understanding migration errors

The following measures can help to understand migration errors:

Viewing the migration interface and the migration report (PDF file)

You can obtain an overview of the number of errors that occurred in the migration interface, where the errors that occurred during the last migration run are displayed in a table. You can find a complete overview of all errors in all migration runs in the migration report within the connector's working directory. The following section provides more details about how to understand migration errors.

Viewing the log and the error log of the connector

The connector logs provide further help when errors occur. The connector uses d.3 logview for logging as standard. A log can also be written to a file.

Viewing the document logs

For each document, a **<filename>.data.log** and a **<filename>.replaced.json** file (if activated in the debug settings, see **Create transformed metadata file**) is created in the export directory during the migration process. The LOG file logs all the activities and error messages that occurred while the document was being processed. The REPLACED.JSON file logs the document metadata record that was created and transferred to the cloud.

Activating the debug log

If the current log is not sufficient for understanding why an error occurs, you can increase the log level. You can activate the debug log while configuring the migration.

Deactivating the virus scanner for the export directory

A virus scanner may cause performance problems. To increase performance, you can add an exception for the export directory. However, you must assess the risk of deactivating it beforehand.

1.8. Procedure when reaching the document limit

d.velop cloud migration defines a limit for the maximum number of documents that you can migrate. You can find the limit in the product description for d.velop cloud migration.

The connector logs when the limit is close to being reached. A warning message is displayed in good time on the migration interface. If you continue with the migration anyway, it will stop automatically when the limit is reached.

If you have reached the document limit, create a support ticket under support@d-velop.de, specifying your tenant name. d.velop AG will then support you with the next steps.

1.9. Frequently asked questions

You can find answers to frequently asked questions in the d.velop documents cloud migration guide at https://help.d-velop.de/docs/de/pub/leitfaden-cloudmigration-documents/cloud/haufig-gestellte-fragen.

1.10. Additional information sources and imprint

If you want to deepen your knowledge of d.velop software, visit the d.velop academy digital learning platform at https://dvelopacademy.keelearning.de/.

Our E-learning modules let you develop a more in-depth knowledge and specialist expertise at your own speed. A huge number of E-learning modules are free for you to access without registering beforehand.

Visit our Knowledge Base on the d.velop service portal. In the Knowledge Base, you can find all our latest solutions, answers to frequently asked questions and how-to topics for specific tasks. You can find the Knowledge Base at the following address: https://kb.d-velop.de/

Find the central imprint at https://www.d-velop.com/imprint.