

# OpenText Documentum In-Place Migration

## Summary

Replacing an OpenText Documentum based business application is a major challenge for IT departments. In-place migrations with fine migration-center are a cost effective and fast alternative to change the documents and their metadata in the repository to meet the requirements of the new application.

## Why in-place migrations?

Imagine you are responsible to run the OpenText Documentum Enterprise Content Management (ECM) system in your company. The system is a couple of years in place, with many business applications on top of it. You just finished a huge update project (moved it to new hardware and the latest version of the OpenText Documentum software) when the owner of the largest business application informs you about the exchange of that application with a very different one. This means to change several millions of documents in the OpenText Documentum repository to match the requirements of the new business application – a classical starting point of a migration project. However, with an in-place migration approach, you do not have to migrate those documents into a new system: You can modify them, i.e. move into a different folder structure, adjust the access permissions, change the metadata and so on, in the existing system.

## How does it work?

The typical migration scenario with migration-center consists of the steps extract, transform and load (ETL), as shown in figure 1.

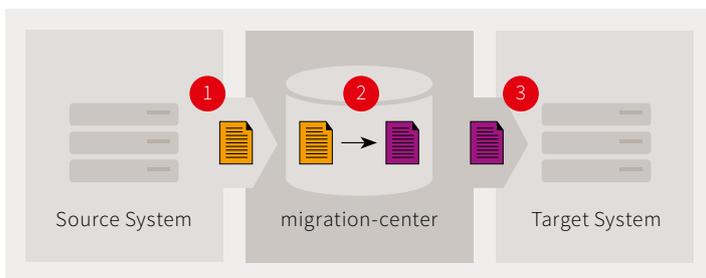


Fig. 1. Typical migration scenario

The first step is to extract documents and their metadata from the source Documentum repository (1). Then you perform necessary transformations on those documents in order to fit the target ECM system (2). Finally, you load them into the target system (3).

In case of an in-place migration, on the other hand, source and target are one. The first step is the same: You extract the documents and their metadata from the source Documentum repository (1). The second step also remains the same: You do the necessary transformations of the documents' metadata as usual (2). Instead of loading those transformed documents into a new ECM system, you update them in their source Documentum repository (3) using the migration-center's in-place importer.

A nice side effect of this approach is that it does not break any references from third-party systems to the documents because they keep their internal identifiers – something that is not possible when migrating into a new system.

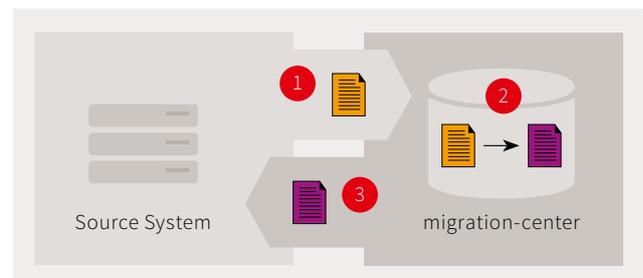


Fig. 2. In-place migration scenario

## Benefits

The benefits of an in-place migration with migration-center for your organization are:

- **Reduced migration costs**  
No additional hardware costs because there is no need to change or extend the existing environment.  
No additional development costs because there is no need to change third-party systems that reference documents in the repository.
- **Increased migration speed**  
The extract and load steps run much faster because the content stays in the repository and only the metadata is transferred.  
There is no need to develop complex migration scripts because migration-center offers powerful transformation functionality.
- **Traceable migration process**  
migration-center's reporting lets you trace the changes of each extracted and loaded document.
- **High quality**  
The incremental approach for the transformation rules and its ability to validate the results of the transformation against the new object model leads to a high quality of the migrated data.

## Features

The OpenText Documentum in-place importer supports the following features:

- Fill missing values for existing attributes or for newly created attributes so that all mandatory attributes are filled in the new object model.
- Replace values of existing attributes with new values resulted from the transformation rules according to the business requirements. This is usually done with mapping lists that you can load into and manage in the migration-center.
- Change the permissions of existing documents according to the new business policies.
- Change the object type of existing documents, i.e. transform all documents of my\_old\_business\_type into documents of my\_new\_business\_type.
- Reorganize the existing folder structure, i.e. create new folders as needed and link the transformed documents into them.
- Retain the existing object IDs (r\_object\_id). Even if you change the object type of a document, it will keep its ID.

