



Cloud Migrator

Prosperoware Cloud Migrator: Guidelines and Considerations





Introduction

This document provides an overview of Prosperware’s Cloud Migrator and sets forth guidelines and key considerations for migrating an on-premises Document Management System (“DMS”) to the Cloud.

What is Cloud Migrator? Cloud Migrator is a software product from Prosperware to move on-premises DMS content to iManage Cloud (both private and cloudmanage.com) rapidly and with minimal supervision. It is a multithreaded application designed with industry-standard ETL (“Extract, Transform, Load”) process that offers a fast and efficient method for consolidating databases, mapping metadata, and migrating content to iManage Cloud from iManage (on-premises), eDocs (requires additional scripting), Windows file shares (requires additional scripting).

How does Cloud Migrator work? Cloud Migrator brings data from the source DMS, or file share, into a staging database. In the staging database, data clean-up, consolidation, or re-organization can be performed before migrating to the Cloud. This prevents any accidental updates to the source database. Cloud Migrator assumes that migration is happening from a live system with users actively using the source system. Staging to the Cloud is accomplished through vendor-provided REST APIs.

Using Cloud Migrator, the migration should be minimally invasive and should not create any unnecessary performance impact to the source database. While the solution is designed to do the actual migration and move all data, careful planning and understanding is required before attempting any large and complex migrations.

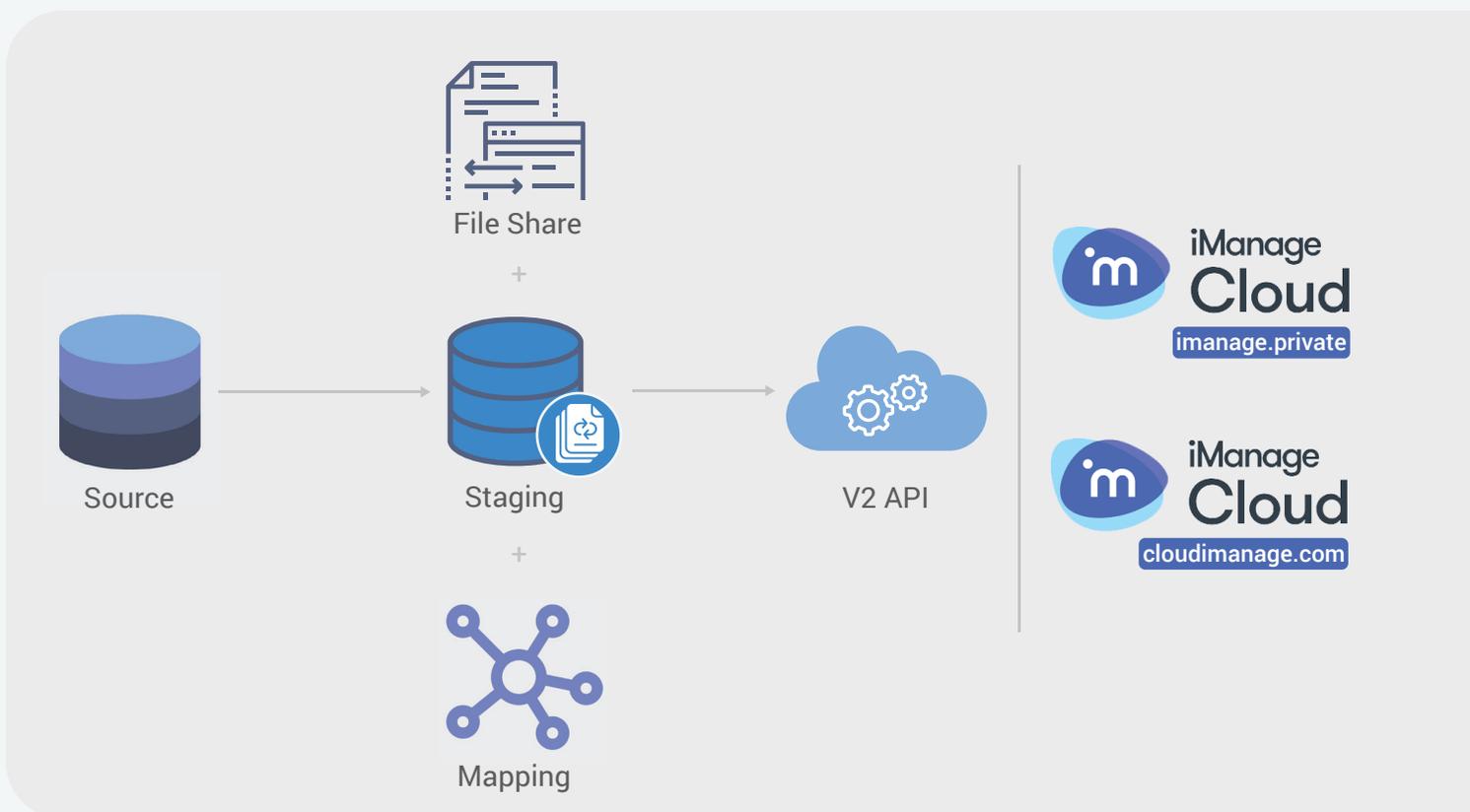


Figure 1: Prosperware Cloud Migrator Diagram



Migration to the Cloud

Migrating your Document Management System ("DMS") to the Cloud can be a sizable and complex task that requires careful planning and execution. Cloud migrations are often further complicated because they provide an opportunity for data-related housekeeping (which is particularly true when the project involves a legacy on-premises database).

Clients often want to:

- Remove old and unused data (users, groups, empty folders, or old document audit history),
- Re-arrange data (such as, flatspace filing, i.e., documents filed at the root of the database and not within any folder container), or do a matter-centric design (create workspaces),
- Consolidate data (going from multiple on-premises libraries to fewer cloud libraries).

Cloud migration projects are also complicated because migration happens on a live DMS. Users are actively using the source system during migration which means it cannot be shut down. All user activity that happens on the source production system during the migration will need to be applied to the cloud environment system via delta runs.

For these reasons and more, a successful migration to a cloud-based DMS system requires careful planning. Along with a considered migration plan, Prosperoware's Cloud Migrator has been designed to provide a solution to simplify and expedite these migration projects.



Preliminary Considerations

Before you start any migration project, we recommend creating an execution plan, listing the approach, assumptions, timeline and ownership of various tasks, all of which should be validated with the client. Some of the key things to preliminary consider include:

1. Health Check: A health check of the source database(s) is a good way to better understand the scope and type of data clean-up needed. If the source system is iManage, you can use a free tool provided by Prosperoware for this purpose. The goal of a health check is to measure the integrity of existing data and identify areas that would benefit from a data clean-up or reorganization. Key things to look for include:

- Unused and missing metadata,
- Empty folders,
- Orphaned documents,
- Flatspace documents, and
- Count of documents, folders, and workspaces.

Prosperoware provides a free DMS health-check application. This application generates a health report for all on-premises iManage database libraries. This utility can be requested by sending an email to healthcheck@prosperoware.com

2. SQL Hardware: The migration uses a SQL database as the staging database. Data from the source system is brought over to the staging database through SQL scripts. As such, it is important to have a SQL server with adequate hardware (RAM & CPU) to process data from the source system to the staging database (see attached hardware recommendations). Prosperoware recommends having a separate box with adequate processors and memory. Not having it could cause a bottleneck when processing data, e.g., data could be queued up and processed at a slower pace (rather than multithreaded), if RAM and CPU are below recommended specifications.

3. Timeline: It's important to understand the client's go-live plan. The go-live plan can impact the migration strategy. Knowing the go-live date helps in deciding what kind of data cleaning is possible or should be done. It helps set up realistic expectations with the migration project. The general rule of thumb is to determine the client's go-live date and work backwards when developing your migration plan.

4. Multiple databases: Multiple databases require careful planning. If all databases are moved individually to the cloud environment, it becomes less complicated. But if the goal is to consolidate databases into fewer cloud databases, then careful planning is required around document numbering, shortcuts, and NRL links (NRL links are specific to the iManage DMS). If consolidation is required, check each database for the maximum document numbers, and then come up with a range for offset numbers.

5. Pilot run: It is also important to discuss with the client if they would prefer to do a pilot run. A pilot run is usually carried out with selected users or groups. The main purpose of the pilot is to get these selected users comfortable with the Cloud, and while they are using the new environment, address any issues that arise prior to the full cut-over to the cloud environment.

6. Manual creation of unsupported data: While Cloud Migrator is designed to migrate most of the data, some data migration is not supported by the underlying vendor's APIs, or it's simply too complex to handle. The following data are not supported by the Cloud Migrator:

- a. Roles
- b. DeskSite/FileSite Search Forms



Key Considerations to Successfully Migrate to the Cloud

Data Clean-up

Cloud Migrator handles most of the common clean-up problems, such as leading, trailing spaces, or null values. One of the most common issues with most cloud migrators is folder metadata. Since folder metadata in iManage does not have referential integrity, it potentially has bad or orphan data. Many times, you see custom values that do not exist in the corresponding custom tables. Cloud Migrator provides ready-available scripts to identify such data, but to clean it up, one may have to consult the client before making a decision.

Migration Order

The DMS has multiple objects with interdependency. It is important to understand the order of data migration to maintain the integrity of those interdependencies. Typically, the following order is applied:

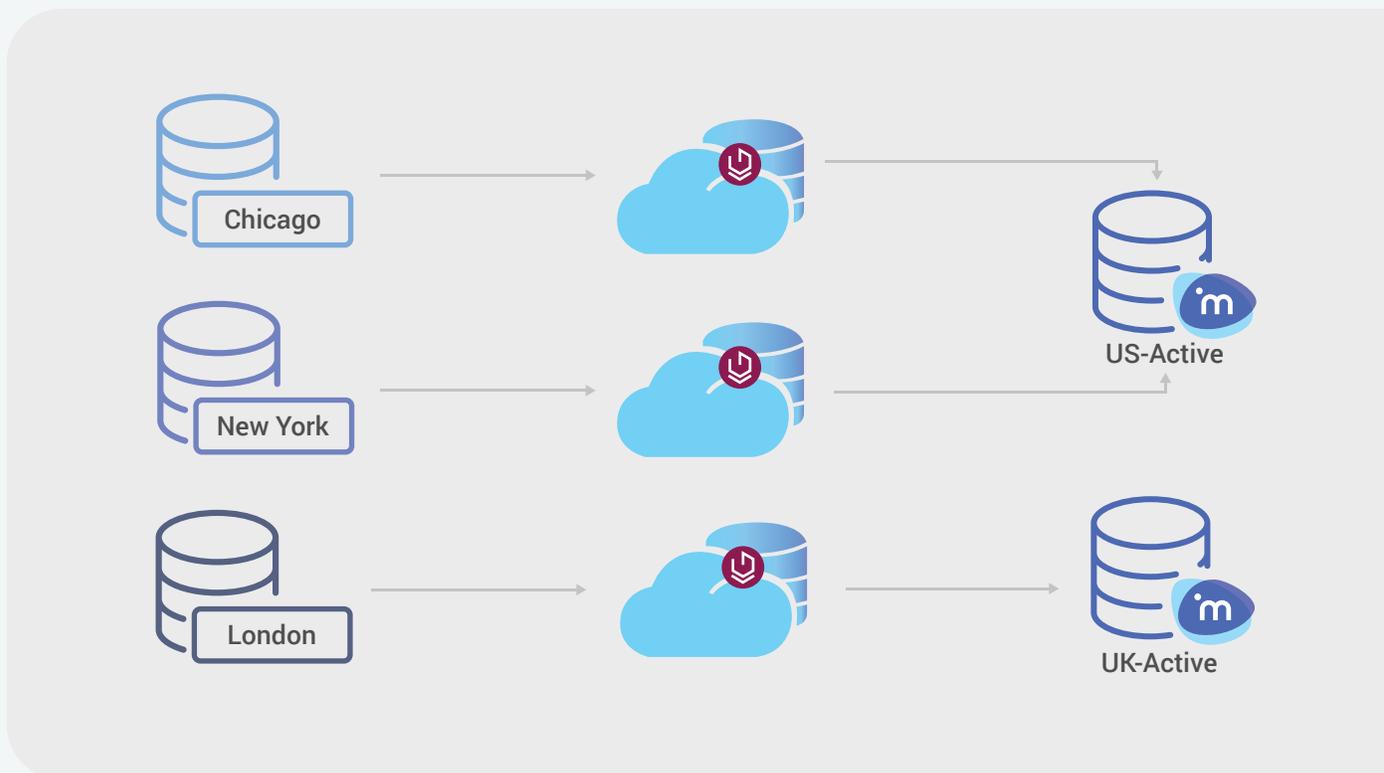
1. Metadata
2. Users & Groups
3. Workspaces
4. Folders
5. Documents
6. Shortcuts
7. History

Metadata and Users & Groups are used as foreign key references for all objects, so typically you want to move them first. History and Shortcuts are not required for a test or pilot run. They can be configured as a separate job and moved after other objects have been migrated. Depending on the age of source DMS, the history database can be large. Due to this, moving history as a secondary job will improve the overall performance of the migration and time to go-live.

Considerations for Consolidating Multiple Libraries

Multiple libraries require special handling. If multiple source libraries are used and you want to consolidate them into a single library, you need to consider the following design decisions:

- 1. Document numbering:** There can be number overlapping between libraries. While you don't want to change any numbering in the source database during the migration, you can add an offset in the staging database. This way the new number will be assigned during uploading (to derive the offset, see the offset section below).
- 2. Duplicate workspaces:** If you have workspaces across libraries with same matter numbers, there are several choices you can make. First, you can leave them as they are, meaning you would have duplicate workspaces with the same number in the cloud environment. Secondly, you can merge them (i.e., move all folders from the second workspace into a single workspace). The third option is to change the metadata number so each workspace has unique metadata. All of these can be performed on the staging database with a SQL script.
- 3. Shortcuts:** If document numbers are changed, there is an implication for the existing links and document shortcuts. The existing links in emails will need to be resolved. This requires storing the old reference number in a custom field to enable users to search and find documents by the old number. It can potentially also resolve to the new number. Prosperoware has a utility called "NRLHandler" that resolves old NRL links to a new cloud link.
- 4. My Matters and My Favorites:** In a multi-database environment, users might have 'My Matters' and 'My Favorites' lists spanning across libraries. During migrations, these will need to be consolidated into a single 'My Matters' and 'My Favorites' list per user. It also implies dealing with duplicate folders and shortcuts, which can be addressed in the staging database.



Cloud Migrator supports consolidating with migration as a single process. Cloud Migrator staging database is required for each source library. One can consolidate one or more source databases to one or more target databases in the cloud. If workspaces content exists across multiple databases then after migrating the first database, all cloud IDs need to be copied to remaining staging databases. The Cloud Migrator process offers an out-of-the-box script for this. This will ensure that no duplicate workspaces are created.

Offsetting

When you have multiple libraries, the same document number might be in use across one or more libraries. Since the document number is a primary key, you cannot have duplicate document numbers when consolidating libraries. You must offset document numbers for each library by a certain increment to avoid this duplication.

For example, you have two libraries A & B, each with document numbers ranging from 1- 9,999. When consolidating, you offset library A by 10,000 so the new document numbers would range from 10,001 to 19,999. You could also offset library B by 20,000 so the new document number range would be 20,001-29,999. You could then start all new document numbers at 30,000, thus giving users a clear way to determine which documents came from library A or B, as well as see new documents consolidated in the cloud library. Using offsets helps avoid conflicting document numbering when consolidating two libraries during migration.

We offer this simple exercise to help you determine what the offset numbers should be:

Database	Total Documents	On-Prem Document Number Range	Cloud
Chicago	5 M	1 - 15 M (min and max document number)	1 - 15 M
NY	10 M	1 - 20 M	20 - 30 M (Add 10 M offset)

Flatspace Filing

When a document is not inside a workspace, flatspace filing is required. In most cloud DMSs, flatspace document storage is disabled by default.

To accomplish flatspace filing, Cloud Migrator comes with an out-of-the-box script. This script takes several parameters that need to be configured based on client requirements. Typically, you can first identify the workplace by client and matter number of a document. If a workspace matching that client and matter number is not found, we need to determine if a new workspace should be created. If the workspace is found, you need to determine in which folder should the document be filed. Generally, documents are filed into a folder with the same document type (+ sub type). Sometimes the folder is at the top level or at the second level.

For example, you have a document in the flatspace that has a client matter number of 1001-001 and a document type (class) of pleading. We can search first for a workspace matching that client matter number. If we find one, then we need to see if there is a folder for pleadings. If we find both those matching items, we can file the document in that folder. If the workspace or folder does not exist, you have the option to create them and then file the document. In the alternative, organizations often elect to simply create a folder called 'Unfiled Documents' in every workspace and file the document there if no matching document type folder exists.

Workspace Creation for Non-matter Centric Databases

Many source libraries are not matter-centric while some of them are only partially matter-centric. It means a workspace does not exist for every client or matter. Cloud Migrator provides sample scripts to create missing workspaces in the staging database during migration.

Empty Folders and Workspaces

If the database has a significant number of empty folders and workspaces, it might be a good idea to delete them in the staging database before moving to the Cloud. Creating folders and workspaces in the Cloud has significant overhead, so cleaning them helps improve the speed. Another benefit to deleting empty folders is to improve the end user experience. Workspaces with many empty folders can be a cause of frustration to the end user in a couple of ways. First, it can cause confusion as to where documents should be filed. Users are more likely to file a document in the wrong folder if there are too many choices. Second, if the screen is full of folders, it makes it more difficult to navigate the UI to find what you need. Caution needs to be taken, not to delete recently-created workspaces as they are likely to be empty.

On-premises Test Run

It is recommended to do a complete end-to-end testing on the local instance of iManage Work 10 server before running an actual production cloud. Test runs do not only identify any data glaring issues, but also help to understand the speed of the actual cloud migration. Cloud Migrator supports a test without the actual file share. Most migration-related issues happen in the data. Hence by not uploading the actual file, you don't have to provision large data for testing and quickly identify data issues.

Full Sync

The full sync is the major portion of any migration project. Depending on the size and number of databases to be migrated, this will be the biggest task. While there are certain key data you must migrate, other data may be either not migrated or migrated after the key data (based on client direction). For migration, here is our general guidance:

1. First Migration

- a. Metadata
- b. Users and Groups
- c. Workspaces
- d. Documents

2. Second Migration

- a. My Matters & My Favorites (iManage)

3. Third Migration

- a. Document History

Cloud Migrator does not stage physical documents, but uploads them during the staging to phase. As such, having the server or staging box in the same data center where the file store is located is key to processing the migration as quickly as possible. **The account running the Cloud Migrator application must have read access to the iManage document file share.**

Incremental (Delta) Sync

From a pure process prospective, there is no material difference between full sync and incremental sync; both need to run the same numbers of steps. Cloud Migrator keeps track of items previously synced in the source database. When the initial sync is performed, the tracking tables are also updated in the source database. When an incremental sync is running, the Cloud Migrator queries join with the tracking tables so as to only select data that is truly modified, inserted or deleted from the source table since the last sync.

The more frequently you run incremental sync, the faster the sync will run as there will be less data to be uploaded.

Similarly, when an incremental job from staging to cloud is run, it only picks the items from the staging database that are not yet uploaded. This will identify and select any previously failed items, plus new items brought by the incremental run from source to staging.

The incremental sync needs to run until there is no data to process. If the incremental sync from source to staging is returning data, it means the source database is still active. At some point, users' access to the source database needs to be cut-off. This typically should happen during the last weekend before the go-live date. If users' access is cut off on Friday evening, then between Friday and Sunday night multiple incremental runs should be performed to do any final catchup from source to staging, and then staging to the cloud environment. Depending on the number of databases, locations and the size of the firm, one may need to plan the "go-live" date over multiple weeks or over a long weekend.

For multiple databases, multiple instances of Cloud Migrator should be run.

Failures and Error Handling

There are two kinds of failure that could happen; from source to staging and from staging to cloud. Source to staging failures are typically SQL-level failures. Cloud Migrator uses select queries to select data from source database and perform a bulk copy to the staging database. Typical failures include network or data issues, which can be rectified by tweaking the SQL scripts or adding indexes or fixing data.

Out of the box, cloud Migrator handles most common issues such as spacing or null values. Cloud Migrator is also optimized to include all necessary indexes to optimally run all queries.

Staging to Cloud is done via the iManage V2 REST APIs. Typical failures can happen due to network failure, timeout or data issues. All errors during both source to staging and staging to cloud are logged. Most of the errors can be fixed by re-running the sync. Data issues such as missing required fields or incorrect data in fields that iManage won't accept, need to be fixed in the staging database before running the next staging to cloud sync.

For most common errors and troubleshooting visit support.prosperoware.com and click on the Cloud Migrator page.

Scaling and Throttling

Everybody wants to run migrations as fast as possible. While Cloud Migrator's implementation is optimized to take advantage of modern V2 APIs availability and technology in SQL and .NET, the actual output depends on many factors. This includes factors like source and staging database hardware, volume of data, network speed and latency, hardware and scalability of the iManage Cloud and throttling put in place by the vendor.

Cloud Migrator provides options to run multiple connections (multi-threading) in parallel. Out-of-the-box we set different multiple connections value for documents and workspaces. This number can be adjusted based on a variety of factors, including available network speed and hardware provisioned by iManage for the client's instance. Consult iManage before increasing connection numbers. The general rule of thumb is to increase as many parallel connections as possible for documents, provided you don't receive timeout errors and document indexing in iManage Cloud is performed in a reasonable timeframe.

Reconciliation

Reconciliation is an important part of any cloud migration. The goal is to move the entire database and all its contents. The Cloud Migrator application provides important reporting in the UI, e.g., total document and workspace count in the source and staging databases, as well as total failed jobs/documents.

In some cases, the total count of objects from the source database may not match the staging and cloud environments. Cloud Migrator will automatically skip orphan documents. An orphan document is a document with a missing file or zero-byte file. Also, due to clean-up scripts like empty folders and flatspace filing, the actual object count in the staging database may be higher or lower than in the source database.

Two important things to look for are:

1. **Objects brought by incremental runs:** Incremental runs need to happen until the day you go live. The goal is to run an incremental run until it returns “zero” objects.
2. **Migrated data in the staging database:** The goal is to move everything from staging to the Cloud and then re-try any failed items until everything is processed.

Post Migration

Sometimes, data issues are discovered after go-live. If the issues are related to missing data or incorrect data, you can re-run the Cloud Migrator to update these data in the cloud. Cloud Migrator will not create duplicate data as long as the data was uploaded through it. You will have to manually update the last modified date in tables if you are directly updating data in the staging database.

For complex clean-ups or data transfer, Prosperoware CAM platform can also be used. CAM can sync all iManage databases into the CAM cloud database. This make it easy to run SQL queries to identify data and then use APIs to bulk update.



About Us

We are a thought-leading software enterprise company for a digitized world. Our core competency is expert understanding of enterprise systems, data and processes in organizations and, building technology for transformative change. We develop software for privacy and regulatory compliance for content systems, and financial matter management.

We empower organizations to better compete through:

- Significant increases in profitability
- Meaningful improvements to client experiences
- Higher user adoption of systems
- Reliable organizational and client risk reduction

Our impact is evident from 270+ customers including 50% of G20 and AmLaw200, 64% of AmLaw100, many global corporations, and the Big Four.

Contact Us Today

Email: partners@prosperoware.com
www.prosperoware.com