Content Migration -FileNet Image Service to P8

Megharaj G[1], A V Krishnamohan[2]

[1]*PG Schoolar, Department of Computer Science & Engineering, SIT, Tumakuru-572104, Karnataka, India*
[2]*Associate Professor, Department of Computer Science & Engineering, SIT, Tumakuru-572104, Karnataka, India*

---***---

Abstract - *IBM FileNet Image Service is a document management system which has been used by banking and insurance domain for long time. Since the launch of IBM FileNet P8, lot of customers wants to migrate their content from Image Services to P8. This paper suggests end to end methodology for content migration from FileNet Image Services to FileNet P8 including various approaches available for data extraction.*

Key Words: **FileNet P8, Image Services, Extraction-Staging-Loading**

[1] INTRODUCTION

IBM FileNet Image Services is one of the oldest document management systems. It has been popular for robust architecture, fast search & retrieval and support for large volume of documents. Many customers have been using Image Services for years for storing all types of content using storage device such as Magnetic Search and Retrieval (MSAR), Optical Search and Retrieval (OSAR), WORM (Centera) etc. Since IBM FileNet has launched P8 and there is no roadmap for Image Services in future, more and more customers are aiming to migrate their content from Image Services to P8.

The architecture of these two repositories is completely different. While Image Services is just a content management system, P8 is a full blown Enterprise Content Management and Business Process Management system which can be integrated to other systems like SAP, Siebel etc.

Since Image Services does not support workflows, migration will only target content and metadata which will be extracted from Image Services and imported to FileNet Content Engine. The core architecture of both the repositories is different, therefore extracted data requires some transformation prior to import to target repository.

[2] BUSINESS CHALLENGES

- Image Services is mainly used by customers having high transactions and heavy use of document such as banking and Insurance. For those which have been using the system for many years, document volume has grown up to millions. Migration of such a high volume of data is a time consuming activity and carries lots of risk
- Architecture of IS and P8 is completely different which makes it impossible to transfer data using out of the box export/import method.
- There is a difference in the way images are handled by IS and P8 so data needs to be transformed before it can be imported to P8 Content Engine.
- Compliance and regulations laws require that every activity on a document should be audited. This imposes challenges for migration initiatives because a document has to go through many phases during migration.
- Source system (and sometimes target system) should be available to business users during migration phase and there should not be or any impact on user experience.

[3] WHAT ARE WE GOING TO MIGRATE?

Before any migration exercise is started, it is very important to analyze structure of source data and objects which are required to be migrated. This is critical not just to estimate accurate development and migration timelines but it also ensures that core document management functionalities such as search and retrieval are intact in target repository after migration is complete. IBM FileNet Image Service repository typically consists of following type of objects/information –

3.1) Documents

Image Service considers all type of content as documents and assigns unique identifiers to each of them. These can include tiff, bmp, jpeg, office documents, pdf, text etc. Since Image Services does not support versioning, documents only have single instances in the repository. These documents can reside in OSAR (Optical Storage and Retrieval) ,MSAR(Magnetic Storage and Retrieval) ,NAS ,CENTERA etc.

3.2) Taxonomy & Metadata

Every document in Image Services is associated with some parameters which are stored in RDBMS. These parameters are used for search and retrieval of documents.

3.3) Annotations

Annotations are used to link additional information with documents .They are stored separately but displayed on top of documents using Image Viewer. Both Image Services and P8 support annotations

3.4) Security

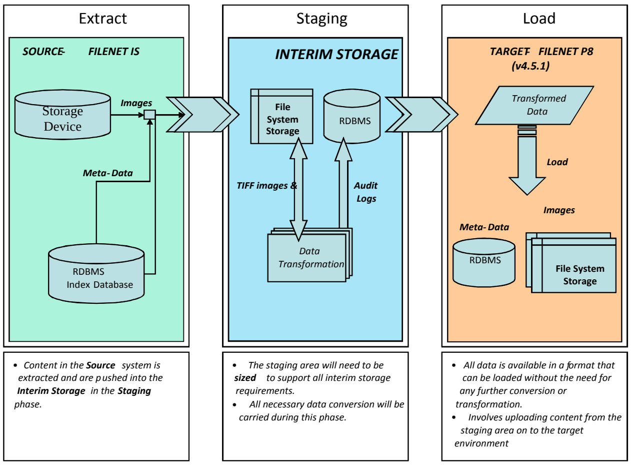
Image Services secure documents and annotations through its own database while P8 relies on LDAP for authentications. Security objects are important and require careful analysis to map with P8 security model

3.5) Folder Structure

Documents are stored in folder structure which is driven by business requirements. If users browse the documents through the repository then these folder structures need to be maintained in P8 repository as well.

[4] MIGRATION METHODOLOGY

Image Service and P8, both are IBM FileNet offerings but due to significant differences in underlying architectures, IS to P8 migration is also done using ESL (Extraction-Staging-Loading) strategy which is the proven migration methodology for any medium to large volume migration exercise. However there are variations in the approaches involved for each of these phases. This paper is intended to cover all such possible approaches.



- Content in the **Source** system is extracted and are pushed into the **Interim Storage** in the **Staging** phase.

- The staging area will need to be **sized** to support all interim storage requirements.
- All necessary data conversion will be carried during this phase.

- All data is available in a format that can be loaded without the need for any further conversion or transformation.
- Involves uploading content from the staging area on to the target environment

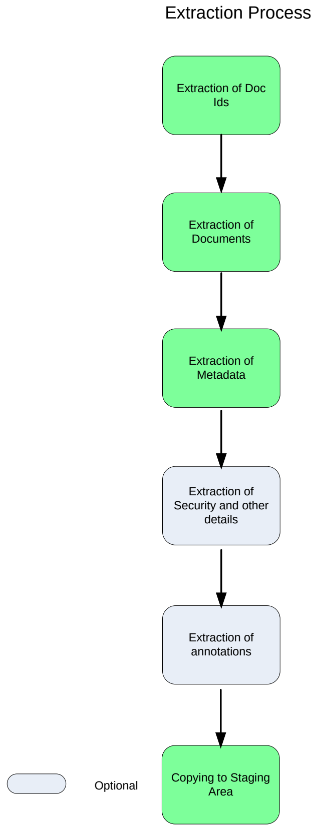
As described in the above diagram, the proposed approach comprises three phases –

4.1) Extraction

This phase involves extraction of objects from source repository i.e. Image Services and storing it in staging area for further processing. Extraction utility will primarily extract documents, metadata and annotations and if required may be extended to extract other information also.

4.1.1 Extraction Workflow

In order to optimize Extraction from Image Services, it is essential to carry out extraction steps in the manner shown in below diagram .Some of the steps are optional and required only in certain scenarios.



4.1.2 Extraction of Doc IDs

Every document in Image Service is assigned a unique identifier which is used for search and retrieval. Before extraction of documents is started, list of doc ids should be extracted from RDBMS. These doc ids can then be supplied to document extraction component to search documents in the repository.

Doc Ids based extraction can be very useful in improving performance if documents are stored in OSAR platters. . These platters are stored in Juke box which has drives to hold these platters and robotic arms to move these platters on and off the shelves. Whenever a document is requested by user and if it is in the platter which is not in the drive, robotic arm will have to move this platter into the drive so that document can be fetched into page cache. This is a time consuming process and hence platter swap should be minimized for better performance during extraction. It is recommended to get the list of doc ids on each platter so that document fetch can be done sequentially.

For storage area other than OSAR where document can be retrieved in any order without affecting the performance, document class wise extraction should be performed.

4.1.3 Extraction of Documents

There are many ways to extract documents from Image Services which may be suitable for different degree of requirements. These approaches also require different skills set hence selection of any approach should be done considering available skills, migration volume and timelines.

4.1.3.1 Extraction using automated IS native tool

MKF and CSM tool are supplied with Image service installation and used for Image Services administration. These can be used to copy images from Image Services to a staging area. These tools are basically set of commands to work with Image Services. These commands can be automated using a script to extract documents from Image Services.

The extraction process can be divided in steps which are as following-

1-Prefetch documents to Page Cache using *docfetch* command

2-Copying the pages to files using *csm_tool*

3-Identifying the format of each copied page/file and appending the extension in filename

4-Extraction of metadata using SQL queries in the script and store it in csv or xml file

This approach is good for low volume of documents but can be difficult to handle if documents belong to many different formats. Also it's not possible to fetch annotations using these tools hence a standalone component will be required to fetch annotations from Image Services. This component can be written using Visual Basic.

4.1.3.2 Extraction using IDM desktop APIs

IDM desktop is a thick client used to access documents in Image Services or Content Services. It also provides libraries to connect to repository from a custom application using VB or ASP. This approach allows rapid development of extraction utility but at the same time it delay overall extraction activity due to poor performance. This approach is only used when documents to be migrated are very less. If Image Services contains large files then application can also time out during extraction.

4.1.3.3 Extraction using WAL APIs

Image Service Toolkit allows access to Image Service object using WAL APIs which can only be compiled using C/C++/VC++ compilers. WAL APIs are server APIs and are fastest way of retrieving documents from Image Services which makes it ideal for large migration exercises. The WAL toolkit consists of ANSI C based libraries which interacts directly with the server and improves the performance of Image Search and Retrieval.

4.1.4 Extraction of metadata

FileNet Image Service stores metadata in a table called DOCTABA. Each document is assigned a unique identifier which links document in storage device to its metadata in RDBMS. The metadata can be extracted either using APIs or using any ETL tool i.e. Informatica. It will be used for indexing of document while uploading of documents to P8. it is recommended that these metadata are extracted using extraction utility and stored in staging area as xml .There can be one folder per document which will contain document as well as metadata xml.

4.1.5 Extraction of Security & folder structure

In most of the cases, security requirements in Image Services can be replicated in P8 Content Engine using Default Instance and Inheritance security types. However, it is possible that additional security layer needs to be added on documents/folders while importing them to P8 Content Engine. In such cases, security information should be extracted and stored in either database or xml file which can be easily read by Import utility. Folder structure in source system can also impose such similar limitation during migration and needs to be addressed using standard ETL methodology.

4.1.6 Extraction of annotations

Image Services support seven types of annotations i.e. Text, Highlight, Arrow, Sticky Note, Pen, Stamp and Custom. Each type of annotation supports different properties. The type of annotation can be determined by the F_CLASSNAME property of an annotation. The extraction process will generate an xml file containing all the properties of an annotation and will be kept in memory until it is saved in the same location as the document. A Separate xml file should be created for each annotation and name of xml should be in the format -

DocumenteId_PageNumber_ AnnotationId_ClassName.xml

4.1.7 Copying to Staging area

Extraction process will store documents, metadata and annotations on a staging area in appropriate folder structures. This can be a SAN or any secured storage area where document security and compliance requirements should be addressed carefully. Since extraction process is the longest process in migration activity, appropriate sizing is required for this staging area considering extraction capacity and avg size of documents to enable smooth execution extraction process for few days.

Although migration components will be tested for error handling and audit logging, possibility of operational errors can not be denied. Therefore it is important to design folder structure in staging area to ease reconciliation and troubleshooting .A sample folder structure is given below –

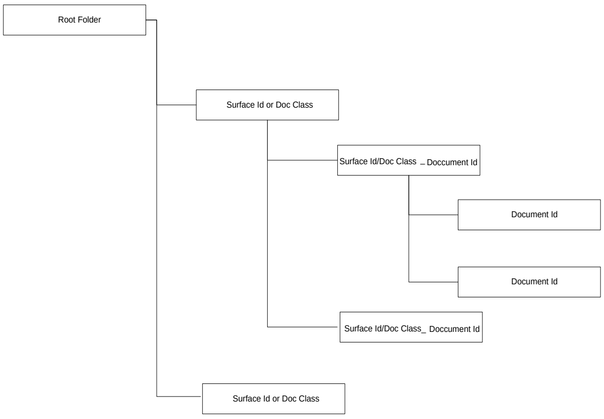


Image Server stores scanned images as single page TIFF files. Extraction process should loop through all the pages in a document and save them in the folder as shown in above diagram. Rename all the pages with a format 'Document ID_Page Number'. For example 324445_1, 32445_2 etc. Apart from images there can be files with other formats (word, excel, text, PDF etc.). These files should be saved as is in the document id folder with name same as document id. Metadata xml will also be stored here. Annotation xml files can be saved in this folder or there can be separate folder inside document folder. These folders should be accessible to only Administrators.

4.2) Transformation

The extraction process will extract documents, metadata, annotation and other information from Image Services and store it in staging area but this needs to be cleaned and transformed to a format which can be understood by FileNet P8 Content Engine. The steps in this phase can vary depending on the extracted objects, complexity and extraction mechanism but ultimate goal of this process will be the same i.e. to bring extracted objects to a state where they can be easily imported to FileNet P8 Content Engine.

At a minimum, transformation phase in IS to P8 migration should include these steps –

4.2.1 Merging of Single Page to Multi Page Tiff

Paper documents may consist of multiple individual pages which are scanned together and stored as multipage tiff in Image Service repository. While extraction ,these documents are extracted as original individual pages which are then required to be merged together as multipage tiff images before importing to FileNet P8 Content Engine. A transformation or Conversion utility will loop through all the single page tiffs of a document and merge them all as one multi page document. There are many third party Image processing tools (OCX or Dlls) which can be used with .NET or VB based component. Alternatively Java Advance Imaging Image I/O tools can be used which is freely available for download.

4.2.2 Parsing and Conversion of metadata xmls

Extraction process will extract and store metadata as xmls in staging area. These xmls need be parsed using SAX or DOM .SAX is recommended because it is fast, efficient and takes less memory than DOM .This is because unlike DOM, SAX does not construct internal representation of xml data. It also examines an incoming xml stream thus avoiding the need of keeping all xml data in memory

Any changes in the metadata structure can be done in this phase.

4.2.3 Conversion of Annotation XML

As stated earlier, annotation should be extracted and stored as xml files in staging area. These xmls contain annotation attributes for example height ,width, length etc.There is a difference in the way annotations are stored in Image Service and FileNet P8 Content Engine. Annotation xmls are required to be modified to make them suitable for P8 Content Engine.

4.3) Loading

Loading process involves uploading documents to FileNet P8 Content Engine with appropriate indexing, annotations and security. This can be done using a java component which will use Content Engine java APIs to access Object Store and other objects. For better execution, loading phase can be divided into following steps:

4.3.1 Creation of LDAP Users and Groups

FileNet P8 relies on LDAP for user authentication while Image Services maintains its own database for users and roles. Creation of correct users and groups in LDAP is essential activity before loading can be started. These Users and Groups will be mapped to Image Service Users and Groups. Careful analysis is required if new security requirements have to be implemented here.

4.3.2 Creation of Content Engine objects

Document class, properties, folder structure and other configurations need to be done in FileNet P8 Content Engine before loading can begin. Loading component will loop through all documents and upload it to corresponding document class and folders with appropriate indexing based on metadata xmls.

4.3.3 Loading of documents and metadata

There will be java component which will use Content Engine APIs to file documents to correct folders in FileNet Content Engine and link them with corresponding document class and metadata.

The data will be loaded in following sequence:

1. **Creation of Folders –** Loading utility will create folder structure based on the information available in metadata file. Folder structure can be a replica of Image Service or it can be modified depending on the requirements. Apart from default folder security additional Security can be implemented at folder level using templates.
2. **Document Filling –** Document from staging area will be filed in folders created as stated in the above step. Document will also be indexed using metadata xml in staging area and additional security can be implemented as per pre defined security model for Content Engine.
3. **Applying Annotation –** Annotations will be processed from each document folder in staging area and applied to their respective documents in P8 Content Engine using java APIs. Additional security needs to be implemented if annotation security differs from their parent documents.

Loading will be a continuous process for bulk uploading and should use FileNet batch APIs to optimize performance. It should also be able run multiple instances to save time and proper logging needs to be done to facilitate easy debugging.

[5] AUDIT

Auditing is an integral part of any migration exercise and should be designed carefully to gain an insight of entire migration activity. Since document is in the center of entire migration activity, it is essential to relate each audit entry with document id of source system. Each component or utility in migration should follow same standards for auditing and error handling.

There will be two types of audit logging in migration –

1- Operational Auditing

2- Application Auditing

Operational Auditing

Operational auditing is required to maintain the history of events for documents in various migration stages i.e. Extraction, Transformation and Loading. Whenever a document goes through any migration process or throws an error, an audit entry will be recorded in a RDBMS table with current timestamp. Therefore this table will contain at least 4-5 entries for each document.

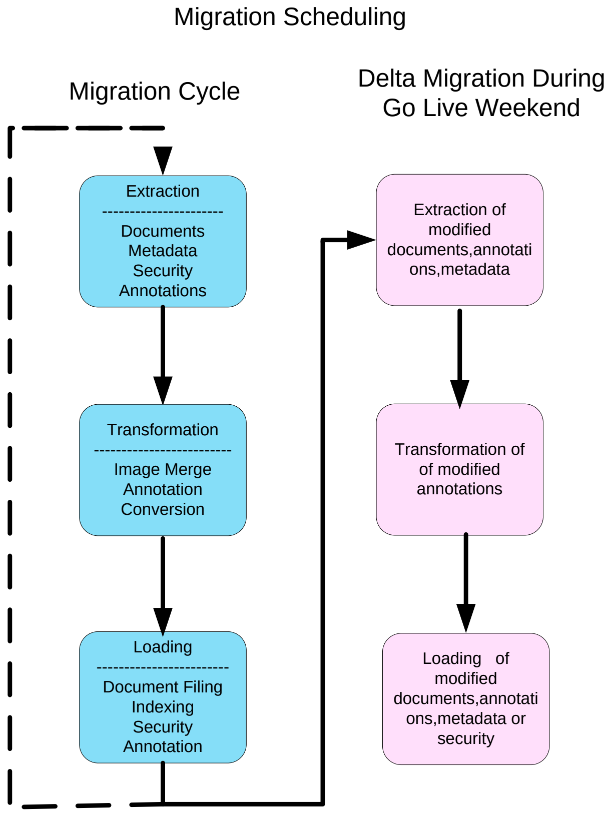
Application Auditing

Application auditing is required to log any activity performed by an application along with a timestamp. This helps in debugging any error and is very useful initially when the application is undergoing testing. Application logging should be configurable via the various configuration files and should be turned off when not required. Application logging will be done using log4j for java applications, which allows configuring of common logging parameters such as the frequency of log file generation and the maximum size of one log file

[6] MIGRATION SCHEDULING

Image Service can contain documents ranging from few hundreds to millions .Therefore it is important to do proper schedule planning of migration steps. Although all three processes (Extraction, Transformation and Loading) will run in parallel to each other, Extraction will be the longest leg of entire operation. Available operations window for extraction may also vary for each organization. Therefore it is important to calculate estimated time for each process and multiple instances should be run to save time.

The following diagram shows a typical migration cycle and other steps to complete Image Service to P8 migration –


Migration Scheduling
Migration Cycle
Delta Migration During Go Live Weekend
Extraction
Documents
Metadata
Security
Annotations
Transformation
Image Merge
Annotation
Conversion
Loading
Document Filing
Indexing
Security
Annotation
Extraction of modified documents,annotations,metadata
Transformation of of modified annotations
Loading of modified documents,annotations,metadata or security

As shown in the above workflow, migration cycle will be repeated many times because entire migration may take long time and users will keep on adding new documents to Image Service. It is also possible that users may modify already migrated document properties and annotations. Therefore Extraction, Transformation and Loading utilities should be able to handle such cases where only partial processing is required.

Delta migration will take place during GO LIVE weekend to cover changed documents and annotations. This will be done in two phases i.e. first to cover newly added documents just before GO LIVE documents and second to migrate modified annotations.

[7] MIGRATION REPORTING

Audit table in RDBMS will store details of each document in each phase. Each process (ETL) will have different set of status codes used to record status of document in all phases. This table will have columns to store Document ID, time stamp, activity and status. A report can be generated by querying this table to produce statistic on various parameters. Apart from producing daily migration statistics there should be two more reports –

1) Migration Error Reports to indicate errors occurred during any phase of migration. It should provide complete information of the objects i.e. Documents, folders where error has occurred.
2) A Comparison report to verify and validate that each object that has been extracted from FileNet IS have indeed been migrated to P8 Content Engine. When the data is migrated from FileNet IS to P8, Image Service Doc Id will be mapped to custom defined field in P8 Content Engine Document class. During this report generation FileNet IS DoctId will be checked in P8, if that object is not found it will be logged in the report

[8] IMPLEMENTATION CHALLENGES

Image Services and P8 are based on completely different architecture; hence this migration may face some challenges during implementation –

1) No single technology for development of extraction, transformation and loading utilities which makes it difficult to do resource loading for implementation.
2) Some of the customers have been using Image Services for ages, huge volume of documents and obsolete infrastructure may cause extraction activity to take long time to complete. Therefore it is important to do a small POC before actual migration. This will help determining the actual timelines to complete entire migration.
3) Many customers who are using Image Services also use workflow tools like FileNet Visual Workflow, FileNet eProcess or any third party Business Process Management product. In such cases, the requirement would be to migrate documents and an associated work item to FileNet P8.This requires thorough analysis of workflow processes and Work item states. These process need to be created in FileNet Process Engine and Work item from Source system should be migrated in such a way that they reside in the same state in FileNet P8.
4) Images will be fetched to Page Cache before they can be extracted .Due to limitation of cache space, it is possible that the extraction application will fill up cache space and remove existing images in cache. In this case users will experience some delay in retrieving their documents. This issue can be addressed by following ways-
   i) Add another cache which will only be used for Image extraction

ii) Take the back up of cache, extract the images using original cache and restore the back up each morning

iii) Take the snapshot of cache prior to extraction i.e. list of Doc IDs, prefetch documents from a storage device for migration .Once all the documents are extracted or extraction is stopped due to production uses(Whichever happens first) 'Prefetch' all the existing documents back to cache so that user wont experience any delay in retrieval

5) Images will be extracted and stored in a Staging Area until they are imported into the FileNet Content Engine. During this period, images will have to be secured so that it cannot be accessed by any user other than Administrators.

[9] BENEFITS

The approach suggested in this paper has several benefits –

1) This is a standard approach (ETL) for any migration exercise and provides better control on every activity done on a given document. An extensive auditing and migration reporting makes it easier to troubleshoot any errors.

2) Use of WAL APIs for extraction improves performance and is suitable for large migration programs. Using Doc ID as search criteria for extraction is very efficient when OSAR is used for storage device. This helps identifying documents on each platter which should be extracted in one extraction cycle without affecting performance.

3) Loading utility is fully reusable for importing images to any other Content engine based system. All the information is stored in the database table and it does not contain any business logic. Therefore it can be reused to add documents with indexing and annotation data for any other project in conjunction with the Image Extraction process.

[10] CONCLUSION

The suggested approach is based on a POC and can be used for any IS to P8 migration exercise. There are many various ways to extract data from Image Services but it is important consider volume of documents and skill set of resources before choosing any one approach. For large volume, it is advisable to use WAL APIs as IDM COM APIs will be slow and csm_tool will not provide adequate control over extraction. These two approaches can be used for low volume and if average image size is not much of concern.

Scheduling of migration should be done keeping in mind volume of images and expected downtime during production roll out. Often Delta migration window will be very small hence regular migration cycle should try to cover as much as possible.

Audit and Validation reports should be as extensive as possible to track each and every document. Migration is a long exercise and prone to errors.Sugfficient auditing and reporting will minimize troubleshooting time.

[11] REFERENCES

1. IBM FileNet Documentation (www.ibm.com)