

# Primo Publishing Platform

## 1. Primo Publishing Platform

### *1.1 Primo Publishing Platform*




#### **Notes:**

Hi and welcome to this session on Primo Publishing Platform.

## 1.2 Agenda

**Agenda**




- Introduction
- Definitions
- Publishing Process
- Primo Pipe Types
- Session Review and Quiz

### Notes:

By the end of this session, you will understand how your data is loaded into Primo, what happens to your records once they're loaded, and what these new records will look like.

## 1.3 Introduction

**Agenda**



- **Introduction**
- Definitions
- Publishing Process
- Primo Pipe Types
- Session Review and Quiz

### Notes:

## Introduction

### 1.4 Introduction



#### Notes:

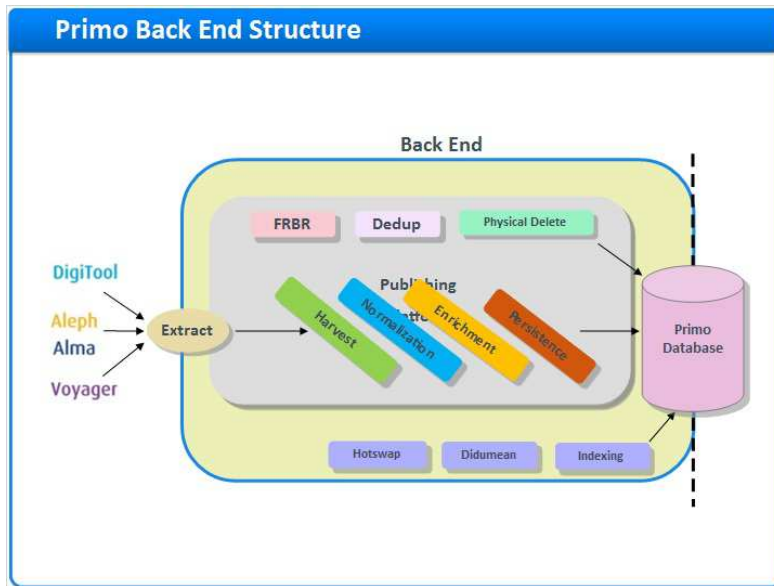
We'll start our discussion by discussing the Primo Publishing Platform.

When searching or looking at a record in Primo, we are in essence using the source metadata that was harvested into Primo and indexed or displayed.

Looking at the Details tab, some of the metadata has been made visible, including: title, author, subject, a description, related titles, the publisher, creation date, format, language, identifier and source.

This information was taken from the MARC record coming from the ILS, then loaded into Primo. It was then transformed from MARC into a different type of record and stored in the Primo database. In this session we will talk about this process - the flow of the record from your ILS or source system and into Primo to be searched and displayed to end users.

## 1.5 Primo Back End Structure




### Notes:

Let's take a brief look at the flow of a record that was created in the source system, such as Alma or any other ILS, loaded into Primo and put through a series of stages in order to make it discoverable in Primo.

We will go through each the stages in this session, but first, let's begin with some basic definitions.

## 1.6 Definitions

**Agenda**



- Introduction
- **Definitions**
- Publishing Process
- Primo Pipe Types
- Session Review and Quiz

### Notes:

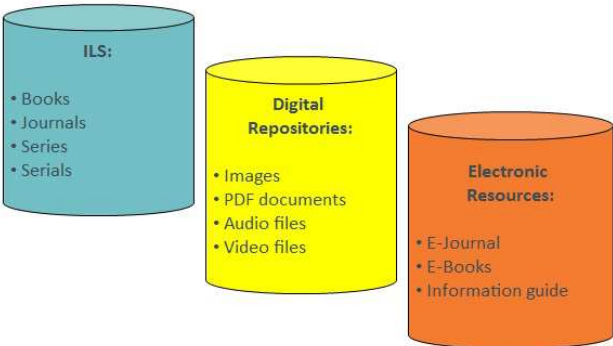
First, let's begin with some basic definitions

## 1.7 Data Source

**Data Source**

Data Source = Source System

The Primo database may contain data from various data sources:



The diagram illustrates three data source categories, each represented by a colored cylinder:

- ILS:**
  - Books
  - Journals
  - Series
  - Serials
- Digital Repositories:**
  - Images
  - PDF documents
  - Audio files
  - Video files
- Electronic Resources:**
  - E-Journal
  - E-Books
  - Information guide

### Notes:

Our first definition is "data source."

A data source is the term used to describe the source system and format of the records.

A data source could be an ILS system that stores books and journals, a Digital Repository for media, or an Electronic Resource, such as an E-Journal.

We will need to let Primo know some details about the origin and type of records we are about to load.

## ***1.8 Data Source Format***

**Data Source Format**

The source system provides data in an XML format

For example:

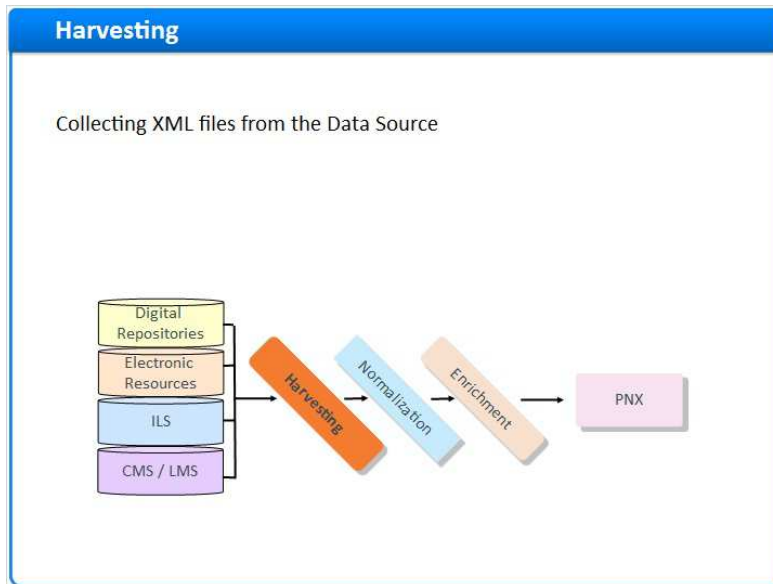
- MARC / MAB XML
- Dublin Core XML
- MARC Exchange
- Digital Entity
- Generic XML
- WARC

### **Notes:**

Here is a list of some of the Data source formats supported by Primo, out of the box.

Data sources may publish records in various formats, but they are always rendered as an XML file, which Primo can process.

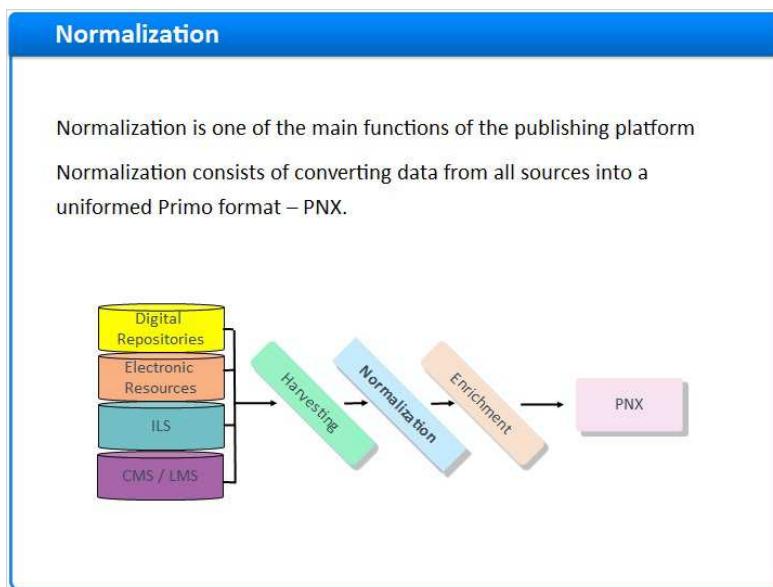
## 1.9 Harvesting



### Notes:

Once we have records from the data source, the **Harvesting** process will collect these records and import them into Primo.

## 1.10 Normalization



### Notes:

Once the records are collected from the different data sources, we need to **Normalize** them so that they are all converted into a unified format, called a PNX.

Let's take moment to discuss why we need to transform these records.

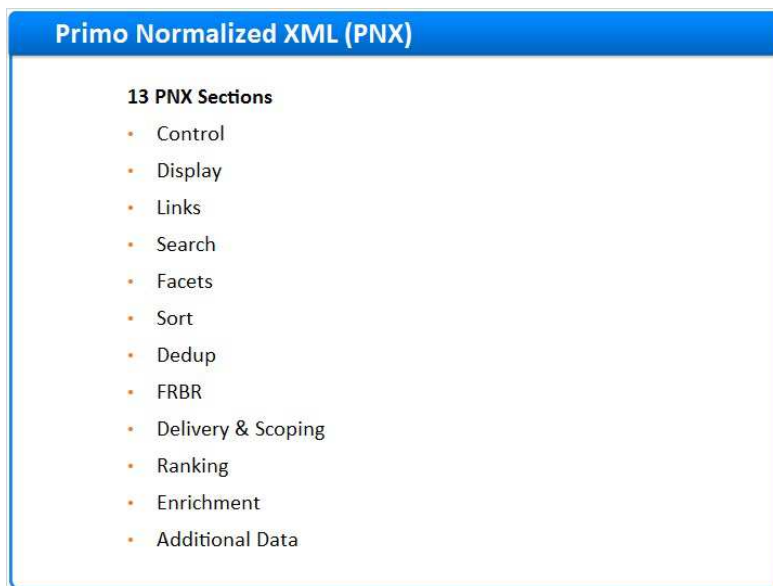
MARC or Dublin Core may be optimum for cataloging records, but they are not ideal for searching. Unless you know exactly what to search for, searching on the ILS system is very difficult. We will need to change the records to a format that is much more "search engine interface" friendly.

It's possible to extract data from library catalogs in various forms of MARC, and Primo can harvest data in other formats as well, such as Dublin Core, XML, and the Web Archive format. This allows us to load different types of data into Primo, and have all of the records end up in the same format.

Having all of the records formatted the same way allows Primo to search across all records at once, and display them consistently in the discovery layer. It also allows us to organize and group records in different ways to create search scopes, facets, and enable virtual shelf browsing.

Records will be normalized into a proprietary XML format, called Primo Normalized XML, which contains all the data shown in the Front End.

### ***1.11 Primo Normalized XML (PNX)***



#### **Notes:**

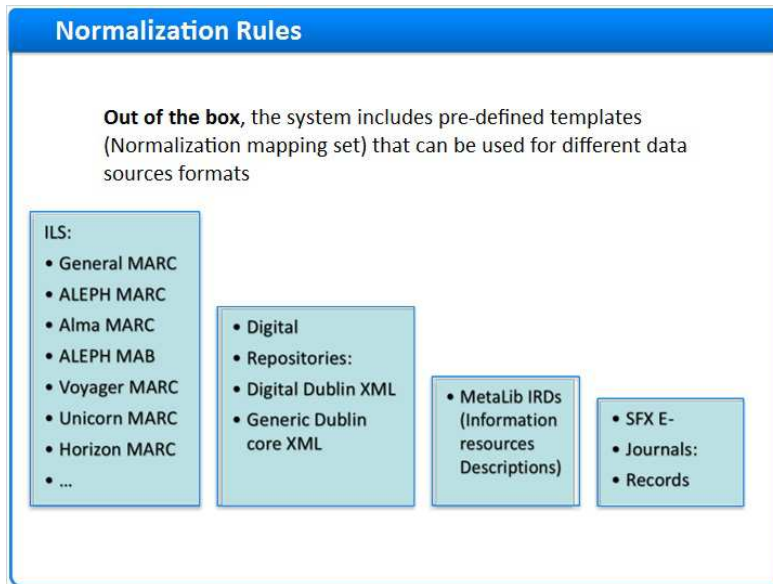
PNX stands for Primo Normalized XML

This is the standardized record format used in the Primo database.

13 different sections are created in the PNX during the normalization process. These sections are populated with the data from the original record in various ways. Each section serves a purpose in Primo, whether it's to present the record for display, or optimize it for searching.



## 1.12 Normalization Rules



### Notes:

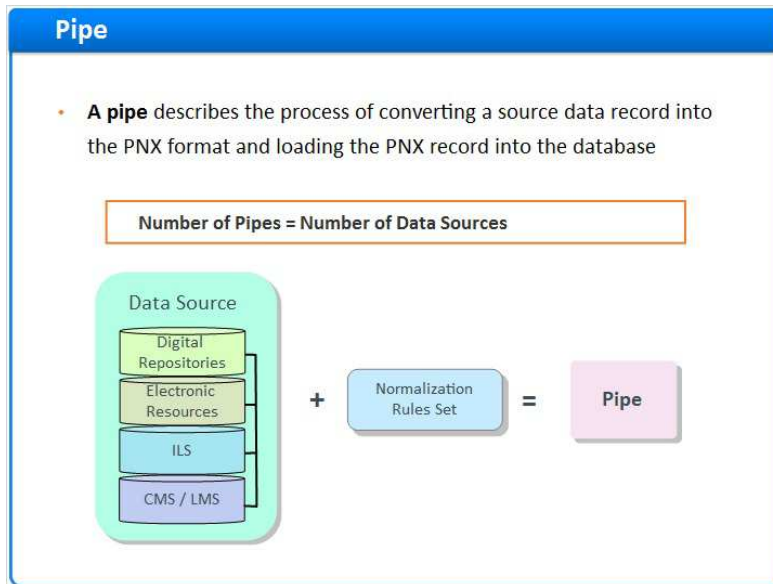
We will need a different sets of rules to convert different formats into PNX. A format like Dublin Core is very different from MARC and will need a different set of rules.

A rule can be something like: take MARC field 245\$a and use it for display/title in Primo or in a case of Dublin Core it could be: take DC field dc:title and use it for display/title in Primo

Primo includes out-of-the-box pre-defined templates for converting different source data formats.

You will later learn how these rules work and how to edit them to suit your needs.

## 1.13 Pipe

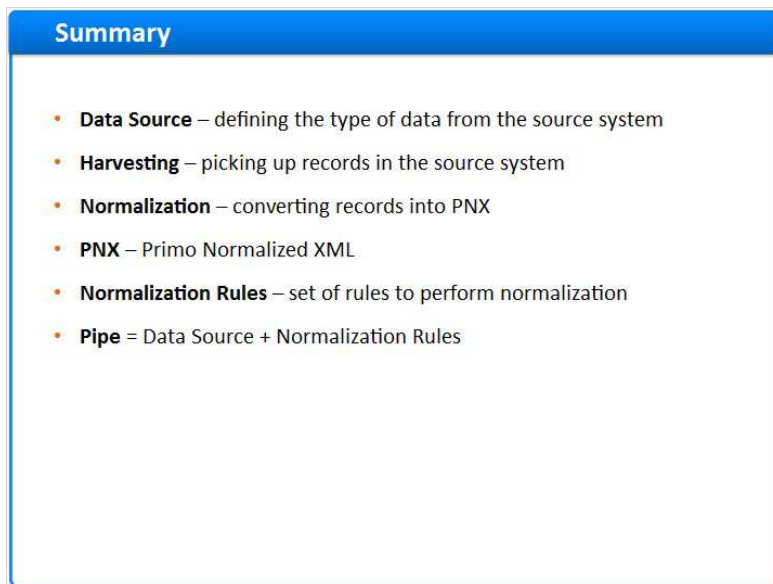


### Notes:

A pipe is the term for the entire process of harvesting the record from the source, converting into the normalized PNX format and loading it into the database.

In order to define a pipe you will need to have defined a data source and a set of rules to fit - then you are ready to tell Primo where to pick up the relevant source files.

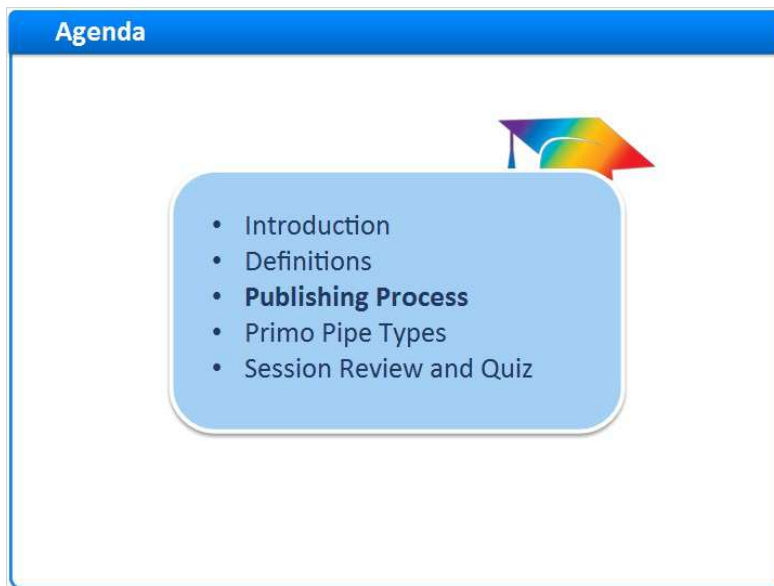
## 1.14 Summary



## Notes:

Let's review the different terms we have learned before moving on. We have **Data Source**, which is defining the type of data from the source system. **Harvesting**, picking up the files containing the records that were created by the source system. **Normalization** - converting the source format to the internal Primo format, the PNX. **PNX**, Primo Normalized XML. **Normalization Rules** - a set of rules to convert the source format fields to PNX, and **Pipe**, Data Source plus Normalization Rules.

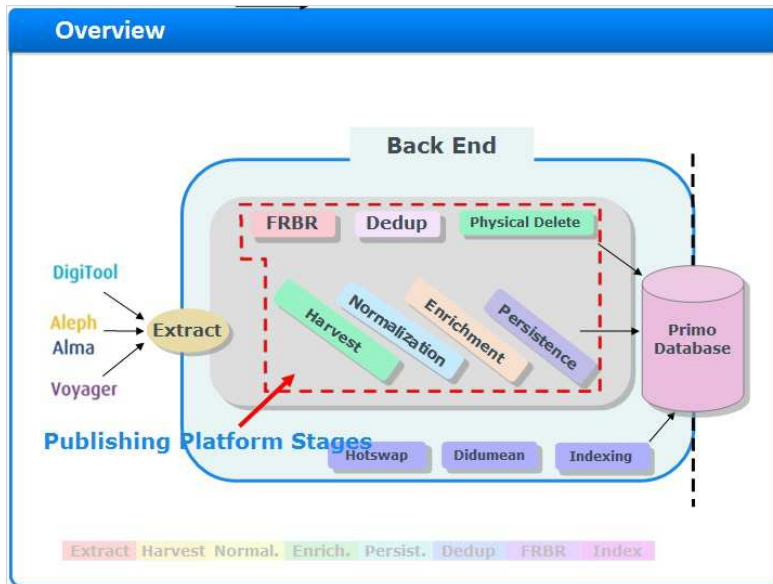
## 1.15 Publishing Process



## Notes:

So now that we know we're going to be converting your source records into PNX records, let's take a look at how the data is processed from start to finish.

## 1.16 Overview



### Notes:

Let's take a closer look at this diagram.  
Our focus will be on these stages.

As defined previously, the pipe is what a record goes through in order to become a PNX record in the Primo database.

We will now walk through this process together from beginning to end.

## 1.17 Data Source Extract

### Data Source Extract

```
<datafield tag="245" ind1="1" ind2="0">
<subfield code="a">E. & G. W. Blunt's map of Charleston and vicinity.</subfield>
</datafield>
<datafield tag="260" ind1=" " ind2=" " >
<subfield code="a">[New York]</subfield>
<subfield code="c">1862.</subfield>
</datafield>
<datafield tag="300" ind1=" " ind2=" " >
<datafield tag="AVA" ind1=" " ind2=" " >
<subfield code="a">PRM65</subfield><subfield code="b">NWILS</subfield>
<subfield code="c">James</subfield>
<subfield code="d">PL3001.L5 A3</subfield>
<subfield code="e">available</subfield>
<subfield code="f">1</subfield>
<subfield code="g">0</subfield>
<subfield code="h">N</subfield>
<subfield code="i">0</subfield>
</datafield>
<subfield code="a">Available also through the Library of Congress web site as raster image.</subfield>
</datafield>
<datafield tag="651" ind1=" " ind2="0">
<subfield code="a">Charleston Region (S.C.)</subfield>
<subfield code="x">History</subfield>
<subfield code="y">Civil War, 1861-1865</subfield>
<subfield code="v">Maps.</subfield>
</datafield>
```

Extract Harvest Normal. Enrich. Persist. Dedup FRBR Index

### Notes:

This stage is outside of the Publishing Platform, but it is important to understand. The Data source extract is the process of publishing records to Primo in different formats which are all XML files, as discussed in the Definitions section.

While your records are stored in MARC in the ILS, one of the first steps in your Primo implementation will be to generate an extract of this data.

Primo can process data in many different formats. Our Alma data, for example, will be extracted in MARC21 XML. Our sample record, once extracted would look like this.

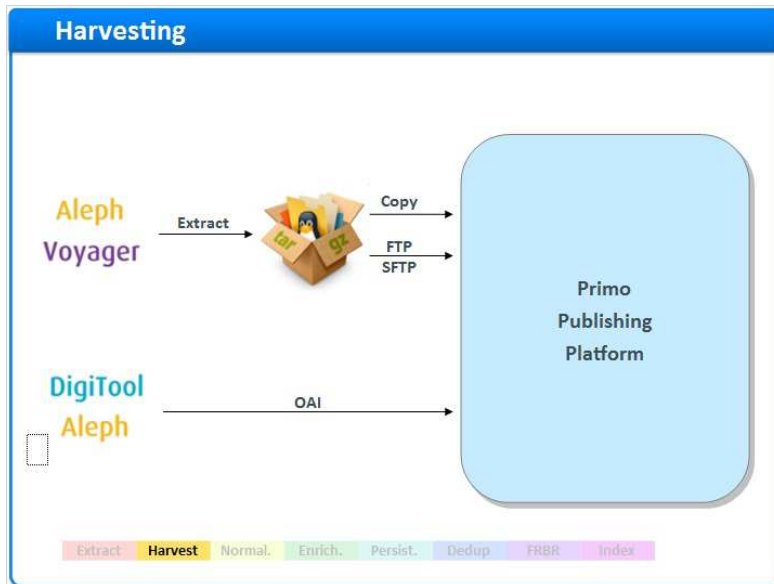
We have the 245 field, which contains the title and author information, the 650 field, which contains the subjects, and so on.

One thing to keep in mind about the extract is that each record will include both the MARC record itself, and the availability information, including the library, collection, and location codes associated with this record. In our example here, that information is present in the AVA section.

The first time you generate an extract, it will be for your entire data set. During your implementation, we'll load this initial extract, and, then we'll work with you to set up ongoing extracts to capture changes to your data on a regular basis.

Let's imagine for now that this is a new record, and we're loading it for the first time. The next step would be to harvest this record into Primo.

## 1.18 Harvesting



### Notes:

In the harvesting stage of the process, Primo harvests the extracted records from the ILS. Primo can harvest extracts in several ways.

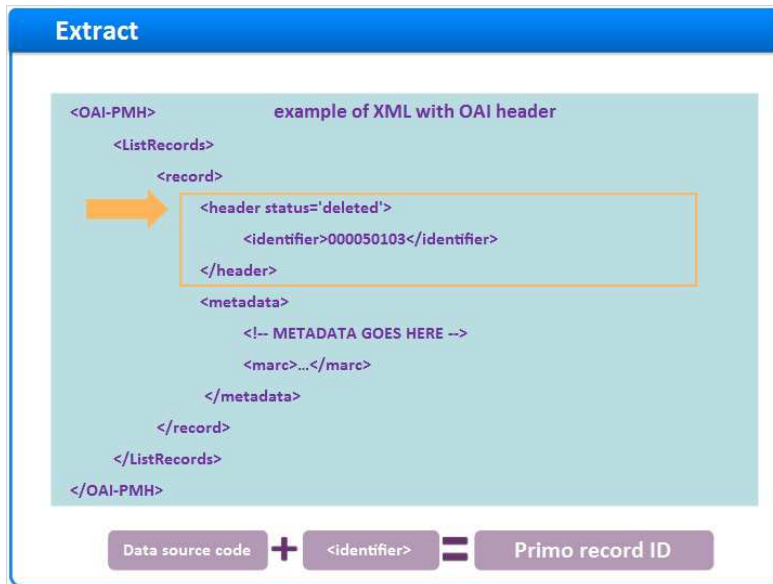
While it is possible to manually place the data on the Primo server, during your implementation, we'll configure an automated way to transfer these files.

Your ILS may be configured to drop the files onto the Primo server, or Primo can pick up the files from the ILS.

It's also possible to configure the ILS to move the files to a different location, and have Primo pick them up there.

All of these options use file transfer protocols, or FTP, and the secure file transfer protocol, SFTP. OAI protocols are used to harvest data from various digital repositories - instead of generating an extract file, Primo can request the data directly from the OAI server during the harvesting process. Once we determine the best method to transfer the data, we can schedule a task in Primo to automatically harvest new extracts on a regular basis.

## 1.19 Extract

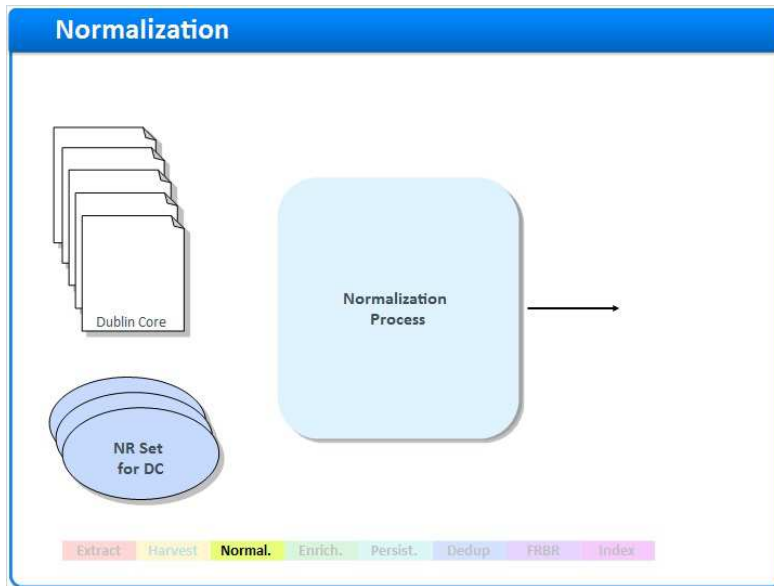


### Notes:

Records typically have indication coming from the source system to indicate this - a very common structure is the OAI-PMH header such as shown here. This is created by many ILS systems as part of extracting the data.

The Primo record ID is built by connecting the data source code with the source system number. This way we can have multiple systems loaded into Primo without the record ID clashing.

## 1.20 Normalization

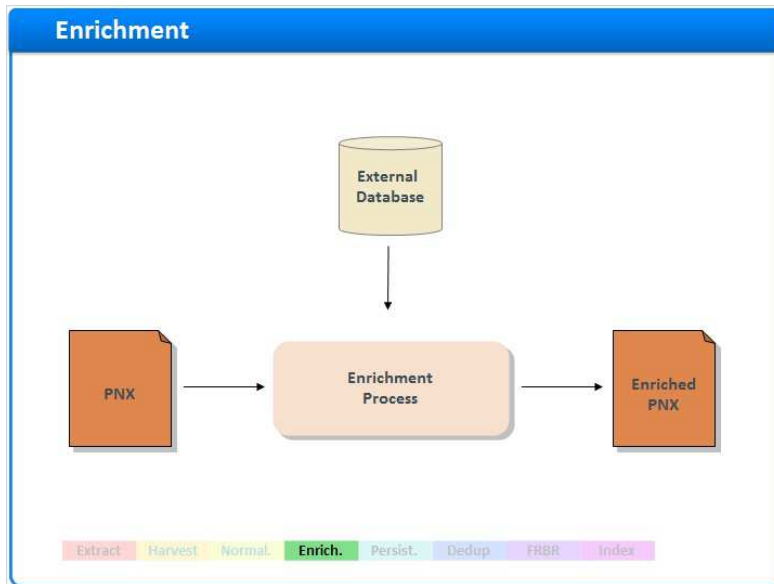


### Notes:

The next stage is to begin processing the individual records. Normalization Rules Sets are used to reformat the data from the source format into the fields of the Primo Normalized XML format, building the standardized PNX. Based on the source format, we apply the appropriate normalization rules and the result is a PNX record



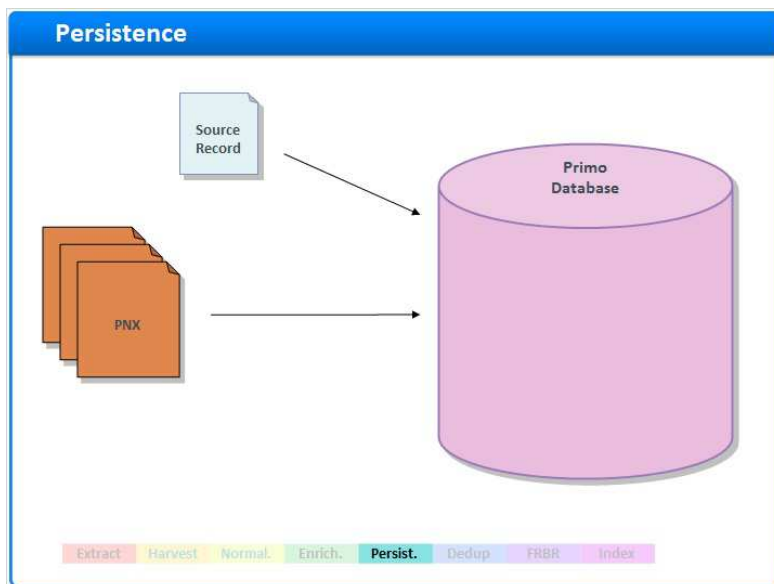
## 1.21 Enrichment



### Notes:

Enrichment enables adding additional data to the PNX records that were not included in the original Source Record, and therefore cannot be added to the PNX using Normalization rules. This data usually originates from external databases. The result is an Enriched PNX record.

## 1.22 Persistence

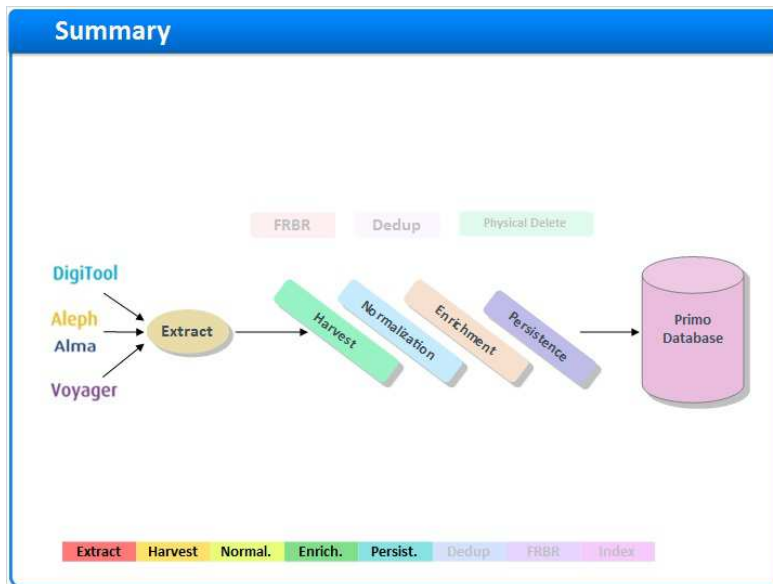


## Notes:

The Persistence stage is when the newly created PNX records are written to the Primo Database. It also deletes records which were marked for deletion during the Harvesting/File Splitting stage

In addition to the PNX, the source record is also loaded into the Primo Data base, this allows us, among other things, to later look at the source record and the PNX record side by side to help identify issues regarding data display or search in Primo.

## 1.23 Summary

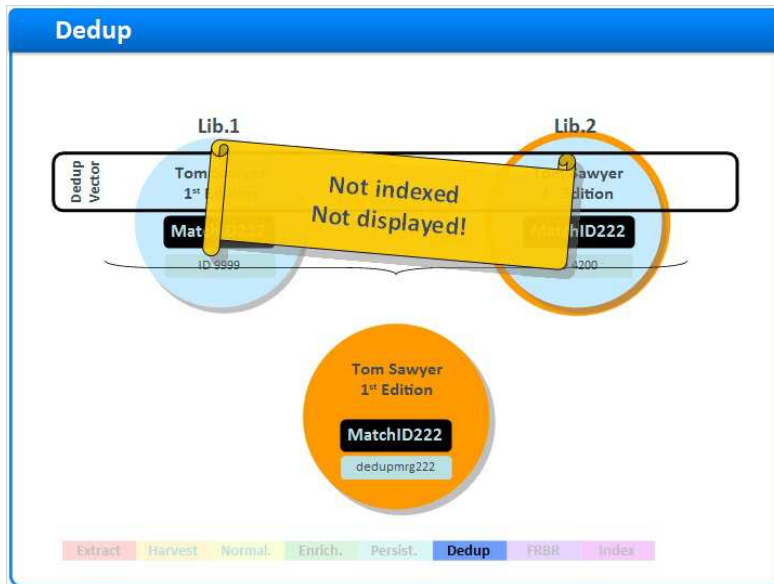


## Notes:

So far we have discussed these basic and main stages of a pipe.

Next we will discuss the Dedup process.

## 1.24 Dedup



### Notes:

The DEDUP process is intended to avoid the duplication of results when performing a search in the Front End.

The goal of this process is to avoid duplicated results for the same book in different libraries within the same institution, or consortium.

A Scoring Algorithm is used to determine whether two records represent the same book in different locations, based on their Dedup vector.

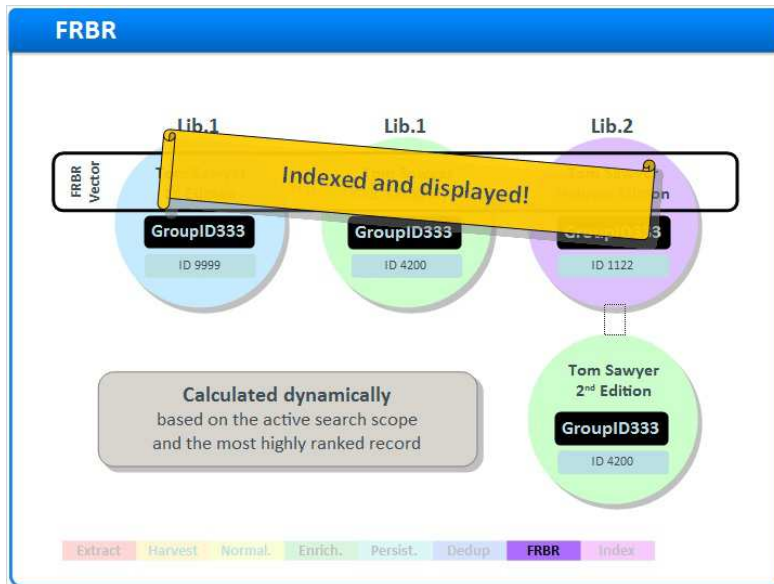
Once a match has been found, a MATCH ID is assigned to both records and one of the records is selected as the "Preferred record", based the "delcategory" that is part of the delivery section of the PNX.

A new record is generated and is known as a "dedup-merge" record, containing the same match ID as its originating records.

The dedup-merge contains all the data of the Preferred record, as well as additional data from the other dedup-members.

Now we have three records, so in order to ensure that only one result is displayed in the Front End, the dedup-members are not Indexed, and only the dedup-merge record is searchable.

## 1.25 FRBR



### Notes:

Similar to the Dedup process which removes duplications in the search results, the FRBR process is in charge of removing versioning of an item in the original set of search results, while not hiding it completely.

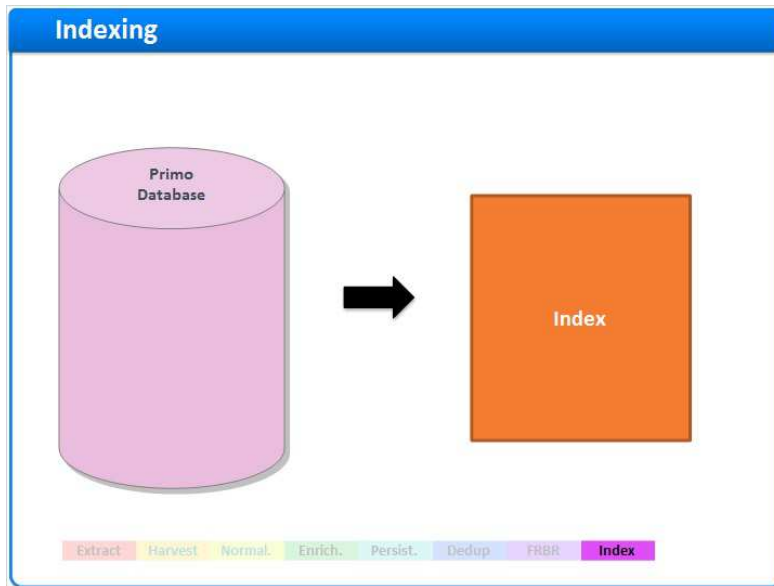
Versioning means that the same record has several editions, publications, languages or formats.

There is also a FRBR vector, and a FRBR scoring algorithm based on which groups are created. When a match is found, the members will receive the same group ID number.

Note that the records are not changed, nor is a new record created. In fact, all records are indexed and retrievable by the end user.

However one result is displayed in the original result set. This record is selected dynamically based on the search term. All the other records are available when clicking on the "Show more versions" option.

## 1.26 Indexing



### Notes:

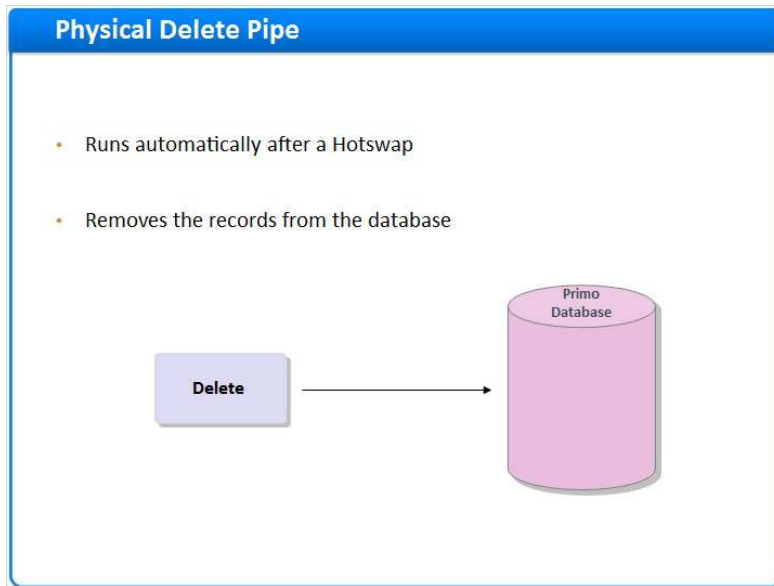
The Primo database itself will have a copy of the record in its original format as well as the PNX record.

Now that we have the normalized PNX records in our database, we are ready for indexing. The indexing process is separate from the normalization process.

During the indexing process, certain sections in the PNX records are used to generate index files that are referenced when a user searches Primo. Without the indexing process, the records in the database are not available to the Primo front end for searching.

Usually indexing is set to run on the Primo server on a daily basis so that it picks up any changes to the data that was harvested and normalized from all of your sources.

## 1.27 Physical Delete Pipe



### Notes:


The last stage we will discuss is the Physical Delete Pipe, which deletes PNX records which were marked for deletion and as a result removed from the Index file.

The "physical delete" pipe cannot be run manually, and this to ensure that it is only run after the Index file was properly updated.

As mentioned, the publishing process must be followed by running the Indexing process, which updates the Index file, and in turn is followed by a process called "hotswapping". Once the index file is updated, those records marked for deletion can be physically deleted safely from the database.

## 1.28 Primo Pipe Types

**Agenda**

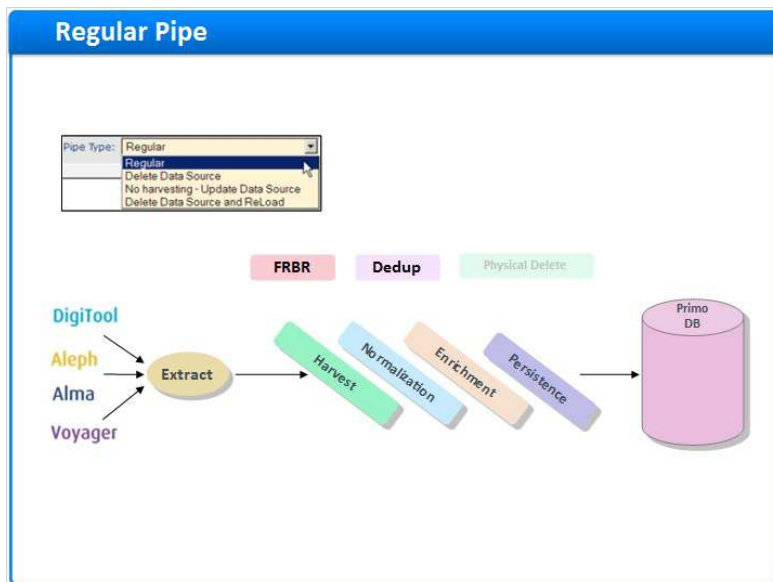


- Introduction
- Definitions
- Publishing Process
- **Primo Pipe Types**
- Session Review and Quiz

### Notes:

Now we can shortly discuss the types of pipes in Primo

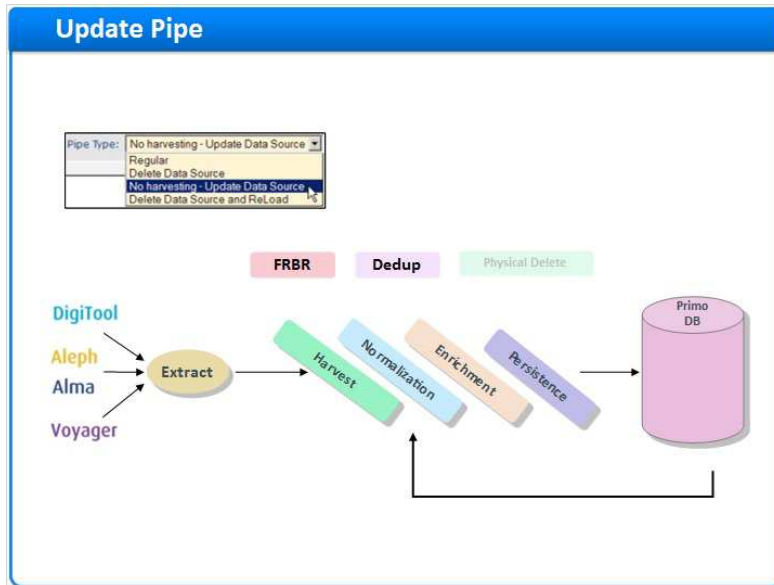
## 1.29 Regular Pipe



### Notes:

The Regular pipe is the most basic pipe and follows all the stages reviewed during this session.

### 1.30 Update Pipe

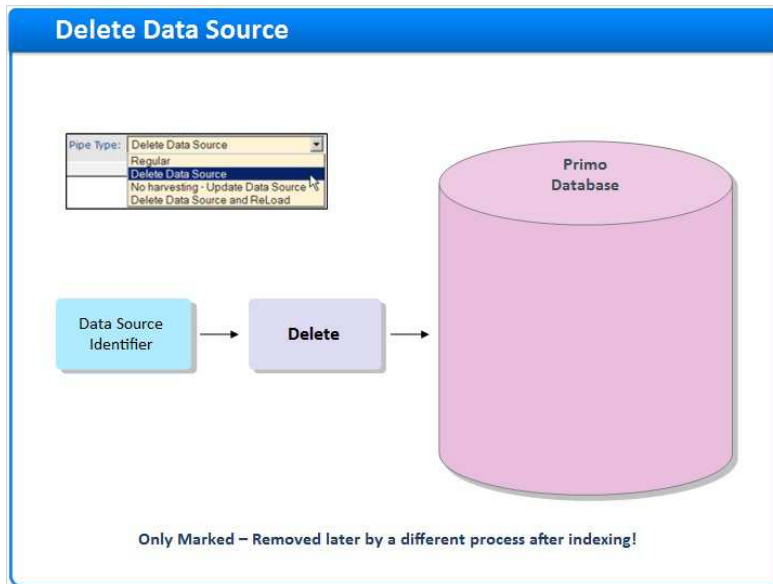


#### Notes:

The Update pipe, or the "No Harvesting - update data source" pipe, has fewer stages. This pipe uses the existing records in their source format and generates new PNX records. This means that no harvesting is necessary. The update pipe is usually used for correcting normalization rules, running tests, and other operations which do not require extracting the records from the data source.



### 1.31 Delete Data Source

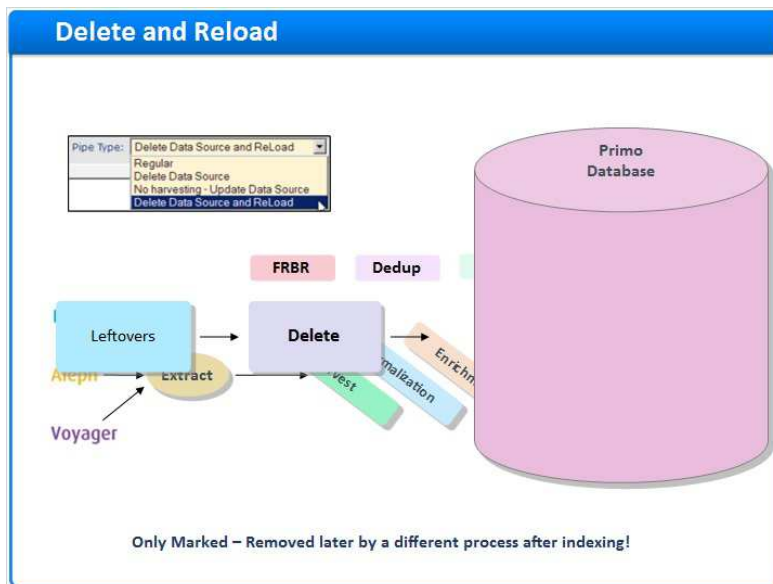


#### Notes:

The "delete data source" pipe removes ALL records from the Primo database, which are associated with the specified Data Source.

Note that the records are only marked for deletion at this point.

### 1.32 Delete and Reload

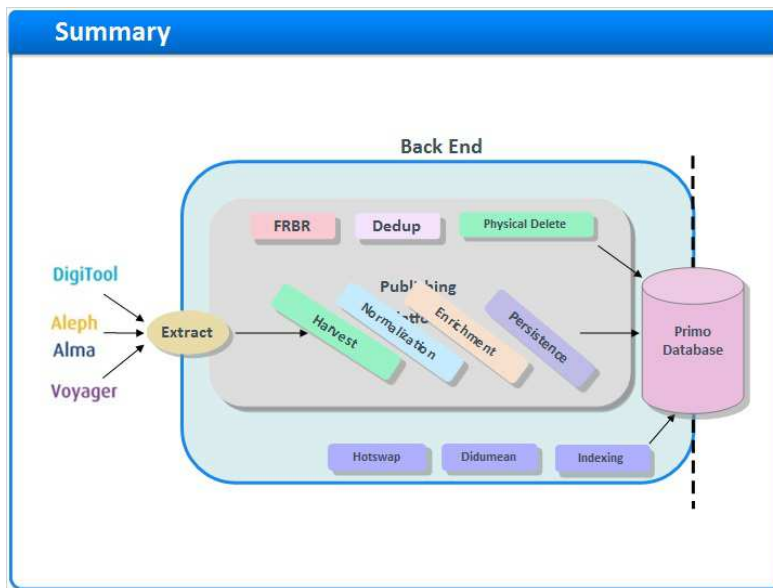


## Notes:

The “Delete Data source and reload” pipe actually works in reverse. First the records are loaded into the database, like in the Regular Pipe, and THEN

The old records are marked for deletion. Notice that the records are only MARKED for deletion and not properly deleted from the database.

## 1.33 Summary

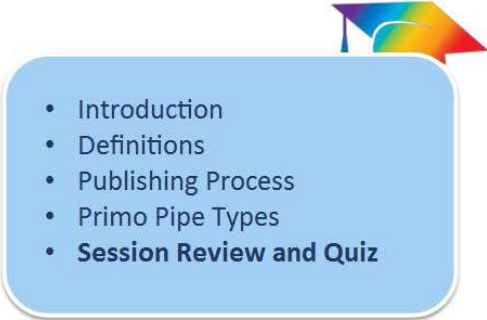


## Notes:

From the original record in the source system to the search result in Primo, the publishing process ensures that your users will be able to search across all of your various data sources and retrieve results that are consistently organized and displayed.

### ***1.34 Session Review and Quiz***

**Agenda**



- Introduction
- Definitions
- Publishing Process
- Primo Pipe Types
- **Session Review and Quiz**

#### **Notes:**

Session Review and Quiz

### ***1.35 Session Review***

**Session Review**

**In this session we covered:**

- How your data is loaded into Primo
- What happens to your records once they're loaded
- What these new records will look like

**Next Steps:**

- Review the recommended articles.
- Take the quiz
- Continue to the next training session

#### **Notes:**


In this session we covered how your data is loaded into Primo, what happens to your records once they're loaded, and what these new records will look like

Please refer to the Recommended Articles section for additional information. After the brief quiz that begins on the next page, you can continue on to the next training session in this series.

### 1.36 Question 1

(Multiple Choice, 10 points, 1 attempt permitted)

**Quiz: Publishing Platform**

 Please select the correct response. Normalization rules

☒ reformat the source data into Primo Normalized XML

☐ reformat the source data into MARC XML

☐ reformat XML records into PNX, but leave MARC records alone

☐ reformat MARC records into MARC XML

☐ reformat data in DublinCore and MARC21 into MARC XML

Correct	Choice
X	reformat the source data into Primo Normalized XML
	reformat the source data into MARC XML
	reformat XML records into PNX, but leave MARC records alone
	reformat MARC records into MARC XML
	reformat data in DublinCore and MARC21 into MARC XML

**Feedback when correct:**

Normalization rules reformat the source data into Primo Normalized XML. All source data gets normalized into PNX during the loading process.


**Feedback when incorrect:**

Normalization rules reformat the source data into Primo Normalized XML. All source data gets normalized into PNX during the loading process.

**Notes:**

**Correct (Slide Layer)**

**Quiz: Publishing Platform**

 Please select the correct response. Normalization rules

☒ reformat the source data into Primo Normalized XML

☐ reformat the source data into MARC XML

☐ reformat XML records into PNX, but leave MARC records alone

☐ reformat MARC records into


☐ reformat data in DublinCore

Normalization rules reformat the source data into Primo Normalized XML. All source data gets normalized into PNX during the loading process.

**Correct** Continue

## Incorrect (Slide Layer)

**Quiz: Publishing Platform**

 Please select the correct response. Normalization rules

☒ reformat the source data into Primo Normalized XML

☐ reformat the source data into MARC XML

☐ reformat XML records into PNX, but leave MARC records alone

☐ reformat MARC records into PNX, but leave XML records alone

☐ reformat data in DublinCore XML. All source data gets normalized into PNX during the loading process.


Incorrect

Continue

### 1.37 Question 2

(Multiple Response, 10 points, 1 attempt permitted)

**Quiz: Publishing Platform**

 Please select the correct responses. Select all that apply. The publishing platform includes the following processes:

☒ Harvesting

☒ Normalization

☐ Calibration

☐ Transformation

☐ Selection

Correct	Choice
X	Harvesting

X	Normalization
	Calibration
	Transformation
	Selection

**Feedback when correct:**

The Publishing Platform contains the following stages: extraction, harvesting, normalization, enrichment, persistence, dedup, FRBR, indexing, and physical deletion.


**Feedback when incorrect:**

The Publishing Platform contains the following stages: extraction, harvesting, normalization, enrichment, persistence, dedup, FRBR, indexing, and physical deletion.

**Notes:**

### Correct (Slide Layer)

**Quiz: Publishing Platform**

 Please select the correct responses. Select all that apply. The publishing platform includes the following processes:

☒ Harvesting

☒ Normalization

☐ Calibration

☐ Transformation

☐ Selection


The Publishing Platform contains the following stages: extraction, **harvesting, normalization**, enrichment, persistence, dedup, FRBR, indexing, and physical deletion.

**Correct**

Continue

### Incorrect (Slide Layer)

**Quiz: Publishing Platform**

 Please select the correct responses. Select all that apply. The publishing platform includes the following processes:

☒ Harvesting

☒ Normalization

☐ Calibration

☐ Transformation

☐ Selection

The Publishing Platform contains the following stages: extraction, **harvesting, normalization**, enrichment, persistence, dedup, FRBR, indexing, and physical deletion.

**Incorrect**

Continue

### 1.38 Question 3

(True/False, 10 points, 1 attempt permitted)



Quiz: Publishing Platform

?

The only way to update data in Primo is to delete and reload it.

☐ True

☒ False

Correct	Choice
	True
X	False

**Feedback when correct:**

Data that has already been loaded into Primo may be updated by running a No Harvest pipe.


**Feedback when incorrect:**

Data that has already been loaded into Primo may be updated by running a No Harvest pipe.

**Notes:**

## Correct (Slide Layer)

**Quiz: Publishing Platform**

 The only way to update data in Primo is to delete and reload it.

☐ True


☒ False

Data that has already been loaded into Primo may be updated by running a No Harvest pipe.

**Correct** Continue

## Incorrect (Slide Layer)

**Quiz: Publishing Platform**

 The only way to update data in Primo is to delete and reload it.

☐ True

☒ False

Data that has already been loaded into Primo may be updated by running a No Harvest pipe.

**Incorrect** Continue

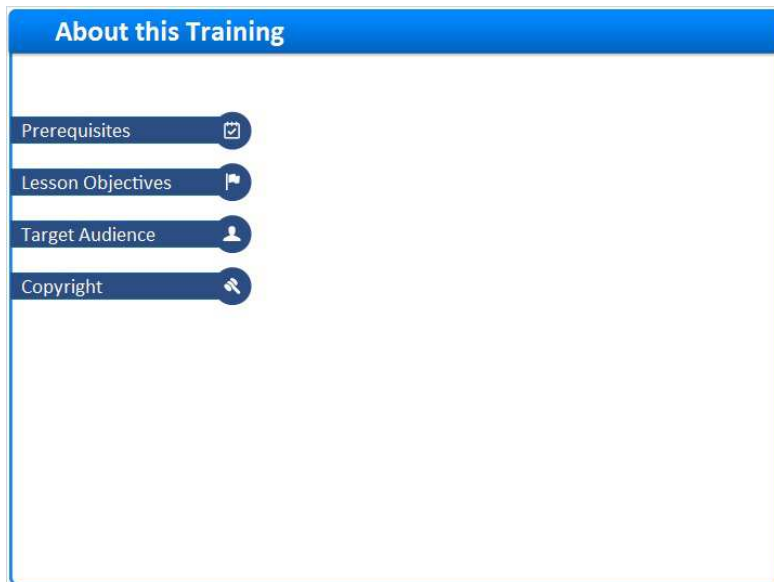
### **1.39 Thank You!**



#### **Notes:**

Thank you for joining this session!





### **1.40 About this Training**



#### **Notes:**

## 1 (Slide Layer)

### About this Training

- Prerequisites 
- Lesson Objectives 
- Target Audience 
- Copyright 

This presentation assumes that you have viewed these training sessions:

- Primo Getting Started Sessions
- Primo Back Office Overview
- Primo Administrative Structure

## 2 (Slide Layer)

### About this Training

- Prerequisites 
- Lesson Objectives 
- Target Audience 
- Copyright 

By the end of this training session, you will be able to:

- Understand how Primo harvests records using the pipe process
- Describe the different types of pipes and their stages

### 3 (Slide Layer)

#### About this Training

Prerequisites

Lesson Objectives

Target Audience

Copyright

**Target Audience:**

- Staff responsible for implementing and maintaining Primo

### 4 (Slide Layer)

#### About this Training

Prerequisites

Lesson Objectives

Target Audience

Copyright

All of the information and material inclusive of text, images, logos, product names is either the property of, or used with permission by Ex Libris Ltd. The information may not be distributed, modified, displayed, reproduced – in whole or in part – without the prior written permission of Ex Libris Ltd.

**TRADEMARKS**  
Ex Libris, the Ex Libris logo, Alaph, Alma, SFX, SFXIT, MetaLib, DigiTool, Verde, Primo, Voyager, MetaSearch, MetaIndex, Rosetta, Leganto and other Ex Libris products and services referenced herein are trademarks of Ex Libris, and may be registered in certain jurisdictions. All other product names, company names, marks and logos referenced may be trademarks of their respective owners.

**DISCLAIMER**  
The information contained in this document is compiled from various sources and provided on an "AS IS" basis for general information purposes only without any representations, conditions or warranties whether express or implied, including any implied warranties of satisfactory quality, completeness, accuracy or fitness for a particular purpose.

Ex Libris, its subsidiaries and related corporations ("Ex Libris Group") disclaim any and all liability for all use of this information, including losses, damages, claims or expenses any person may incur as a result of the use of this information, even if advised of the possibility of such loss or damage.

© Ex Libris Ltd., 2018