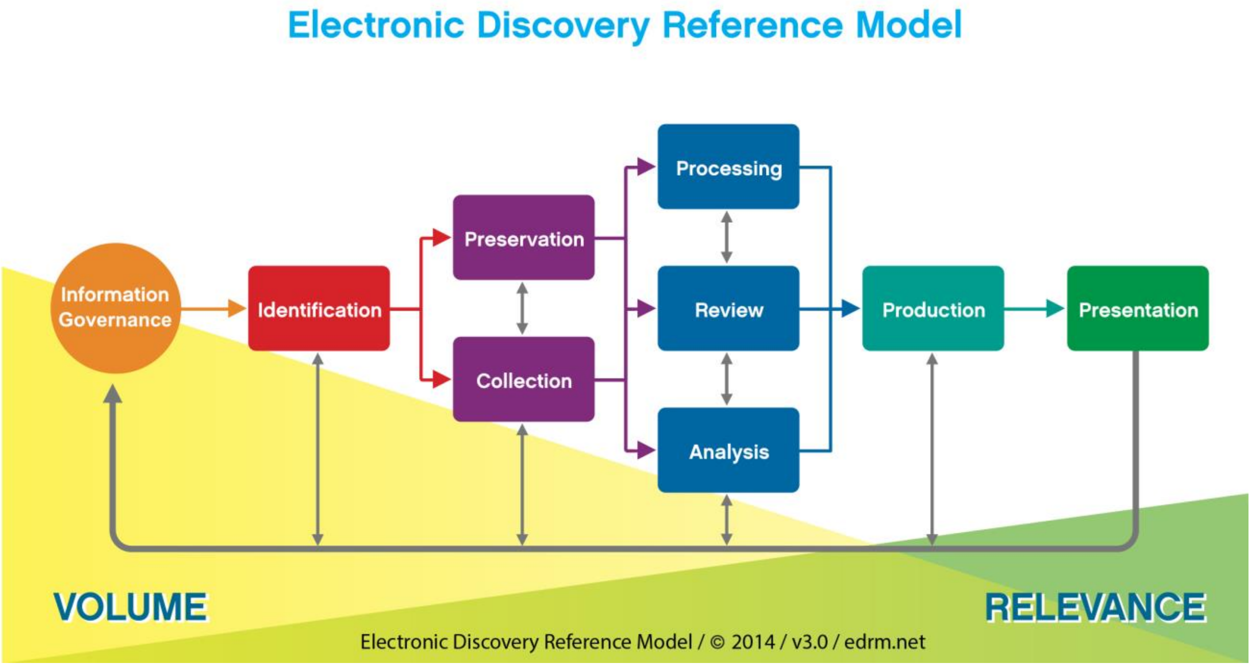
Guidelines for
E-Discovery Processing



EDRM Working Group
on E-Discovery Processing

Guidelines for E-Discovery Processing

(A Project of the EDRM Processing Guidelines Task Force)

Editors and Drafting Team Leaders

John Tredennick Merlin Search Technologies, Inc. JT@Merlin.Tech	**Craig Ball** University of Texas School of Law Tulane University School of Law craig@ball.net

Assistant Editor

Tammy Dahl, ACP, CEDS
Mesch Clark Rothschild
tdahl@mcrazlaw.com

Drafting Team

Christopher Cella Aon christopher.cella@aon.com	**Oran Sears** Relativity Oran.sears@relativity.com
Tracyann Eggen CommonSpirit Health tracyann.eggen@commonspirit.org	**Sarai Schubert** IPRO schubert.sarai@iprotech.com
William Hamilton University of Florida Levin College of Law hamiltonw@law.ufl.edu	**Jeffrey Wolff** ZyLAB jeffrey.wolff@zylab.com

Table of Contents

Introduction

E-discovery[1] involves the exchange, analysis and review of electronic files, email and other information stored on a computing device.[2] Over the past two decades, it has become a common part of U.S. litigation and regulatory investigations, spurring the growth of a multi-billion dollar litigation support market.[3] It has also spawned a new specialty for legal professionals, many of whom now focus their practices on, and get certified for, this aspect of litigation.

The goal of an e-discovery effort is to surface relevant information for trial, arbitration or a hearing. The process begins with the identification of electronically-stored information ("ESI") that may be relevant to the matter and ends with the production of responsive, non-privileged ESI to a requesting party. In between are a series of steps designed to move data from identification and collection through processing, review, analysis, and production.

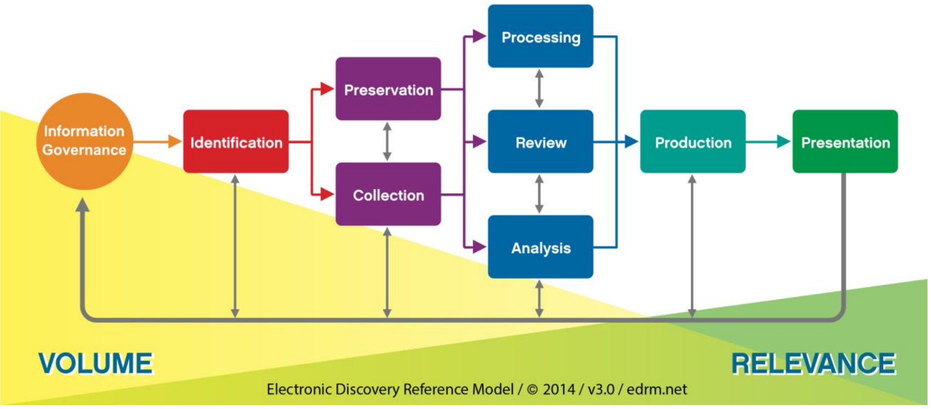
The stages through which ESI moves from its original location through trial or a hearing are depicted in the EDRM (Electronic Discovery Reference Model), which is a widely accepted conceptual model of the e-discovery process.

[1] The term is shorthand for electronic discovery and is sometimes spelled eDiscovery or ediscovery. In the U.K. and some other jurisdictions, this process is known as e-disclosure. For convenience, we will use e-discovery to refer to all of its variants.

[2] See EDRM Glossary at 1106 (https://edrm.net/resources/glossaries/glossary).

[3] E-discovery is less common outside the U.S. Many civil law countries, including those in the EU, do not allow e-discovery or limit it to evidence needed to support a claim or defense at trial. In contrast, most common-law jurisdictions, such as the U.K., Australia or Canada, allow e-discovery but often with a more limited scope than in the U.S.

Electronic Discovery Reference Model



Over the years, the model has evolved and is often used to help people understand how the e-discovery process works.

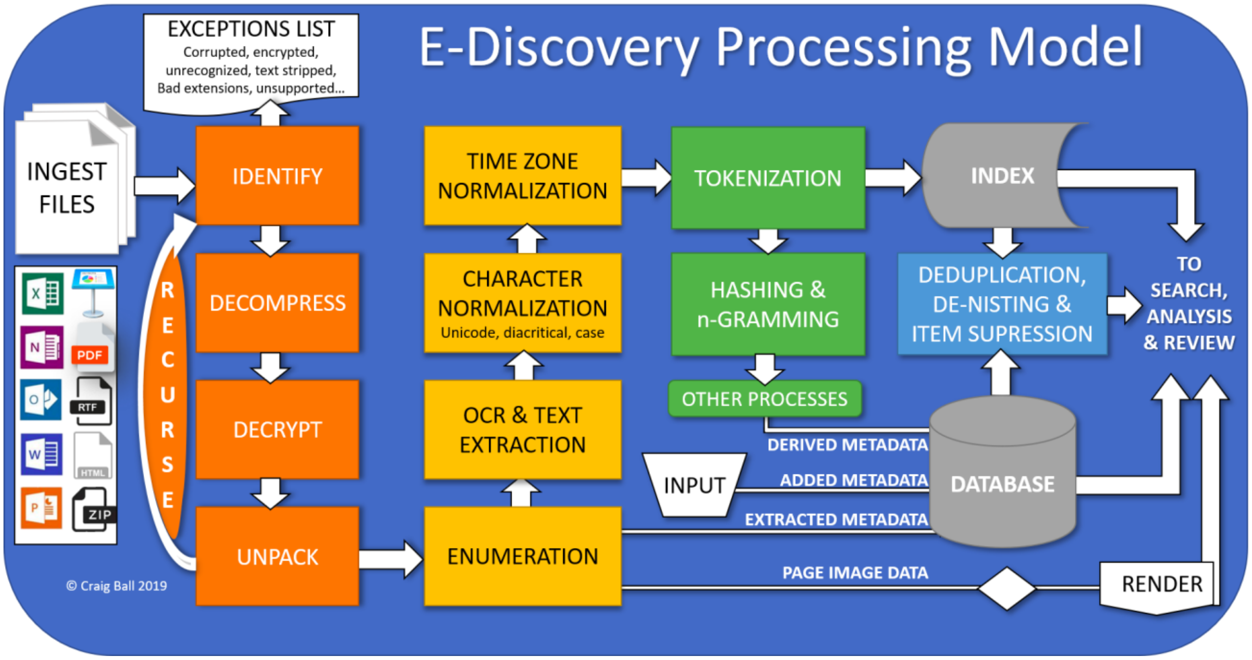
E-Discovery Processing

These guidelines, adapted from Craig Ball's seminal work, "Processing in E-Discovery, A Primer, ©2019, will focus on the processing stage of the EDRM. As you can see from the EDRM model, processing bridges the gap between preservation/collection and review/analysis.

During processing, ESI that has been collected using a variety of methods and tools is processed through a series of steps and prepared for review. In the early days, processing included printing ESI for manual review. As ESI volumes increased, printing to paper no longer made sense. Modern processing software converts ESI into a standard format for electronic review, with the resulting data often being loaded into specialized litigation support software.

Here is one look at the complexities underlying a standard processing workflow from Craig Ball's Processing in E-Discovery primer.[4]

[4] For more about ESI processing read Craig Ball's "Processing in E-Discovery, A Primer," published in 2019 and available here. It is an excellent source for those seeking to better understand this stage of e-discovery and to learn about industry requirements for processing functionality.



In sharing this diagram, we are not suggesting that it provides the only order of steps to perform this work (e.g. OCR is usually done after you filter (DeNIST, De-Dupe, Date), in order to reduce the amount of files it needs to run through). There are many ways to attack the problem. Rather, this is a visual view of processing, one that may help readers better appreciate the complexities underlying this stage of the EDRM.

Scope

These guidelines will help elucidate this often overlooked stage of the EDRM and will help readers better understand the basic steps involved in processing and the needs of the legal professions at this stage. In that regard, these guidelines are addressed to those who (a) use processing products, (b) support these products,(c) processing products, and (d) are new to e-discovery and want to learn about the processing stage.

Principles

In preparing the guidelines, the drafting team worked from the following principles:

1. Our focus will be on processing functions rather than processing products. It will not discuss the individual strengths and weaknesses of specific products.

2. Our intent is to describe the processing functions required for successful e-discovery processing and to note functions deemed optional, albeit desirable, in some instances. Our goal is to set a floor, not a ceiling.
3. Processing consists of a number of tasks. A single product may attempt to accomplish all of these tasks or only a subset.
4. A processing product may also attempt to facilitate other functions, such as data review. Similarly, one may rely on a single product for processing or several products to achieve that aim.
5. As a cardinal rule, processing operations should not alter or destroy files or metadata.

Organization

Like the EDRM itself, processing functions proceed through a number of logical steps which, for convenience, are referred to as "phases." Following a lawyer's penchant for chronology, the phases are ordered to follow the ESI from the point where collection is completed to when processed data is ready for review.

This is not to suggest that a good processing engine must work in this order or that any product must perform every step in each phase. Rather, the goal is to present the key processing steps in a logical order to encompass all functions traditionally included in the processing stage of e-discovery.

The key phases for the processing stage of the EDRM are:

1.0 ESI Ingestion and File Extraction

2.0 Initial Filtering

3.0 Text, Metadata and Image Extraction

4.0 Output

5.0 Reporting

These guidelines will discuss each phase and how they may be used to meet legal industry needs.

1.0 ESI Ingestion and File Extraction

Ingestion is the first step in data processing. A processing system should have the ability to ingest a variety of ESI types such as emails, office documents (word processing, spreadsheets and slides), instant messages, social media, and audio and video files, as well as a variety of container formats, which are often used to package the collected files.

Container files are a type of compressed file created by such programs as WinZip or WinRAR and are often identified by their three-letter extensions (e.g., zip or rar). Container files are commonly used to make files easier to transport. When a container file is exploded or unzipped, it returns exact copies of the original compressed files.

Forensic software programs also create container files to make exact images of a drive or part of a drive for evidentiary or preservation purposes. These programs have their own proprietary formats with file extensions such as FTK or E01. Ultimately, a processing engine must be able to extract files from different kinds of container files so they can move further along the processing chain.

The basic steps to unzip container files are:

1.1 Receive and Extract Data from Common Container Formats

The first step is to explode the container files by extracting and identifying file contents. For ZIPs and RARs, the system must apply the proper decompression algorithm to properly extract those files. The same is true for forensic containers. In some cases, the forensic software may be used to extract container file content.

To receive and extract data, processing software should be able to:

- Extract data from common file transport container formats such as Zip and RAR
- Extract data in a recursive manner until all containers within containers have been addressed
- Identify and report on encrypted container files
- Extract data from forensic collection formats such as FTK, L01, DD, E01 and AFF and
- Make a record of container files and their contents for chain of custody purposes.

1.2 Identify and Extract Content from Email Containers

Email collections are typically transported in a container file known as a PST (personal storage table) or OST for local temporary copies, and NSF for Lotus Notes.[5] PSTs are created to hold Microsoft Exchange files such as Outlook emails, contacts, tasks and calendar items. To properly extract this information, the processing system must apply the appropriate encoding schema to segregate messages and other individual items to successfully extract their contents.

Individual emails are mini containers that often contain attachments and embedded objects with the body of the email message and its associated metadata. For example, Microsoft Outlook's email export format is typically identified as an .msg. Other standard Internet mail formats include eml, emlx, and mbx.[6] Gmail, for example, uses the eml extension.

To identify and extract content, processing software should be able to:

- Continue recursion until all message content and attachments are extracted, including standard container files attached to messages
- Track and report on family relationships, e.g. the attachments contained in an individual email message
- Extract embedded objects[7] from email messages and display them as attachments
- Extract inline images at the operator's request and
- Distinguish between recurring logos and other inline images.
- Extract text from OLE objects (Object Linking and Embedding) such as smart art graphics or icons within office files

[5] We note that some cloud-based email systems are beginning to offer in place ingestion and processing. In such a case, the ingestion stage remains, but the data may not be physically removed from its original cloud environment. Rather, the system acquires the data directly (ingestion) and processes it in the same environment. The interim stage of extracting the data and moving it through a PST or ZIP may be avoided.

[6] See https://en.wikipedia.org/wiki/Email for more information on email formats.

[7] Embedded objects are typical file attachments embedded as base 64 encoding or object linking and embedding (OLE, an early Microsoft format).

1.3 Identify Other Basic File Types

It is imperative that the processing system recognize the various file types received for ingestion. Office files must be properly identified; image files must be treated as images. Audio and video files must also be properly addressed. Misidentification of file types can quickly lead to processing failures.

File identification is part art and part science. In the Windows world, the computer's operating system usually identifies file types (and opens them with the appropriate program) based on their file extension, e.g., .doc or .docx for Word files and.xls or .xlsx for Excel files.

This can be misleading. A file can be renamed with any file extension, which may be done by a malefactor to hide a file's identity. As a result, other operating systems, including Mac, Linux and many Windows utilities, do not rely on the file extension for identification. Instead, the information in the file itself is used to identify the file.[8] Programmers place a binary file signature in the first few bytes of a file to identify its type. In cases where the software cannot confidently identify the file type, the system should report the file as part of an error listing.

To identify file types, processing software should be able to:

- Correctly identify file types based on multiple factors including header information, MIME types and file extensions and
- Identify common Office file formats.

1.4 Scan for Viruses

Ingested files should be scanned for viruses at an early stage. Infected files should be quarantined or removed from the system so they do not adversely affect the processing system, infect other data or be passed to other systems during later stages of the e-discovery process.

Ultimately pre-processing or processing software should include virus protection to scan files for viruses by:

[8] One commonly used mechanism is Media (MIME) detection. Multipurpose **I**nternet Mail Extensions (MIME) is a seminal Internet standard that enables the grafting of text enhancements, foreign language character sets (Unicode) and multimedia content (*e.g.,* photos, video, sounds and machine code) onto plain text emails. Virtually all email travels in MIME format. For more information on this, see Ball at 22.

- Working from a regularly updated virus signature database published by one or more reputable virus protection vendors
- Quarantining virus infected files for later handling
- Allowing virus infected files to be removed from the system or safely deleted and
- Reporting on files quarantined for virus issues.

Many processing systems do not include virus scanning as an integral part of the processing software. Rather, the belief is that an operator should maintain and run virus scanning software separately before loading the data into the system. This ensures that the virus scanning software is continuously updated, which is required to detect recent viruses.

At the least, if the processing software includes a virus protection component, there should be a mechanism to ensure that the virus signature files are current. A virus signature file is a security program used to detect and identify malware.

1.5 Hash Files for Identification and Comparison

Hashing is a process used to create a “digital fingerprint” for each individual file. This hash value can be used to identify duplicate files to reduce redundant files prior to review.

Hashing uses an algorithm that analyzes a file and its contents to calculate a unique file identifier. For example, a file hash value may resemble this:

5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8

The algorithm is designed to change values significantly when even a single byte of data is changed. This allows the processing system to confidently remove duplicate files without inadvertently removing similar but non-duplicates. It can also be used to determine whether a file has been changed at a later stage of the process, perhaps to alter its contents. If a claim is made that a file has been altered, hash values from the original and the suspect copy can be easily compared.

There are different hashing algorithms available for use in a processing system. One of the earliest was the MD5 (message digest five) algorithm, which created a 128-bit fingerprint. The secure hash algorithm (SHA) was developed by the National Security

Agency, and the current versions use 256 bits (SHA-256) or 512 bits (SHA-512) to create a digital fingerprint.[9]

The key is to use the same hash algorithm on both files when documents are duplicates or a file was later changed . Although it is not mathematically impossible, the chances of two non-identical files having the same hash value is extremely low.

To hash all files received, processing software should be able to:

- Create a hash value for each file using industry standarding hash protocols, such as MD5 or SHA-256;
- Store the hash values for chain of custody purposes with links to the corresponding hashed files.and
- Validate the files received for processing against the files delivered after processing.

Some products may also create a "family" hashing to provide an option to deduplicate at the family level and ensure that when you remove duplicates; the families stay intact.

Using hash file values to remove duplicates along with system and program files is discussed in the next section.

1.6 Create an Exceptions List

During this initial phase of processing, files may fail for a variety of reasons, including encryption, corruption, false identification or removal due to virus concerns. These files should be preserved, quarantined, or otherwise identified to maintain a proper chain of custody.

Many systems offer exception reports that can be retrieved at this stage of the process or later as the need arises. Exception reporting applies to every phase of processing. These guidelines will not include a discussion on the other phases. It is a critical part of the processing phase and its contents should be available through the end of the discovery process. Exception reports should include, but not limited to, the error or exception, its location, and its status (resolved, ignored, retried, unresolvable, etc.).

[9] Hashing is a unidirectional process that can never work backwards to retrieve the original data.Learn more about file verification and hashing at: https://en.wikipedia.org/wiki/File_verification.

2.0 Initial Filtering

Depending on the nature of the matter, the scope of the collection can vary greatly from narrow and targeted to broad and overly inclusive. As a result, processing software must filter files based on type, date range or other operator criteria. With broad collections, processing software should offer the ability to remove or hold collected but unwanted files rather than moving them to the next stage of processing.

Sections 2.1 through 2.3 discuss the types of filters a processing system should include.

2.1 Identify System and Program Files Based on the NIST List

System and program files[10] are rarely useful in e-discovery and are often removed (or at least not promoted further) during processing. System and program files are easily identified by comparing the hash value of each with an extensive hash list maintained by the National Institute of Standards and Technology ("NIST").[11] If the hash value matches, the file can safely be identified as a system or program file.

NIST files are typically withheld from further processing because they do not contain discoverable content and are not useful in the e-discovery process. This process is known as DeNISTing, and it reduces the volume of data to be hosted and later reviewed.

2.2 Identify and Remove Duplicate Files

As previously discussed, duplicate files may be identified by hash values. If two files have the same hash values, the content is identical. The process of removing or withholding duplicate files from further processing is known as "deduping" or "deduplication."

In some cases, deduplication by custodian is performed by identifying and removing or withholding all but one copy of each document maintained by the custodian. This reduces the volume of documents associated with that custodian, avoids repetitive review, and ensures the custodian is associated with the file that may have evidentiary significance.

[10] System files are files associated with and used by a computer's operating system, e.g. Windows, Linux or Mac OS. Program files are those associated with and used by the applications we run on computers.

[11] See https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl.

In other cases, deduplication is performed across all custodians in a process known as global deduping. This process leaves one copy of the file to be promoted while withholding the rest.

With either approach to deduplication, it is customary to include information with the file known as a load file that demonstrates where else it appears in the larger collection. Information should include two fields "ALL CUSTODIANS" and "ALL SOURCE PATHS" where the files reside. The load file may ultimately be loaded into a litigation support database. This information should be updated every time new data is processed.

2.3 Filtering by Date Range or File Types

Excluding files by criteria such as date range or file type is another method of reducing the volume of files processed for review. Depending on the issues involved and if the matter has a defined date range, it may be prudent to exclude files outside the date range from further review.

Processing software should include a master date field in order to filter consistently. The master date (or DocDate) may reflect different dates and times from different source files, e.g. the SentDate for email, the LastSavedDate, etc..

Processing software should permit identification and inclusion (or exclusion) of files by date range and file type with the understanding that the system uses an appropriate process that goes beyond the use of the file extensions to identify file types.

3.0 Text, Metadata and Image Extraction

Once files have been extracted, analyzed and filtered, the next step is content extraction. Processing software should extract text from emails, office documents and other file types, often as text files, and move them forward for indexing and search later.

Reputable processing software should extract fielded information, known as metadata, from the files such as the from, to, cc, bcc, subject, sent date and time, and custodian fields from emails. In addition, the software should also offer the option to include tracked changes in a document, hidden content, and activity. The extracted information may be crucial in later stages of the e-discovery process, particularly search and review.

Sections 3.1 through 3.10 address the key stages of this phase of processing.

3.1 Access File Content

First, processing software must access the content of hundreds if not thousands of different file types, including the most common ones, such as email messages, Microsoft Office files and PDFs. Most processing platforms integrate specialized software for this purpose as it would be nearly impossible to build such a wide range of document filters independently.

Leading programs used for accessing file content include: Oracle's Outside In, Hyland Document Filters, dtSearch, Aspose, and Tika, a widely-used open source software product.

3.2 Detect Encrypted and Corrupt Files

Document filters cannot extract text or metadata from files that are encrypted or corrupt. The processing system should report the files as exceptions or errors to be addressed separately. Obtain passwords for password-protected files. Some processing systems permit the administrator to list known passwords that will be tried automatically when password-protected files are detected.

There is little to be done with corrupt files. Most processing systems identify the files in an error report so they may be recollected or otherwise addressed.

3.3 Detect Encoding

Once a file's contents are accessed, the software accessing the file must determine how the contents have been encoded. The earliest e-discovery processing tools assumed files were encoded using the ASCII character set that was developed in 1963. ASCII supported 128 characters that sufficiently expressed the English alphabet (upper and lowercase letters), the numbers 0 to 9, and a few dozen other symbols necessary for basic computing.

As computing reached beyond U.S. borders, it was quickly realized that 128 characters were insufficient to encode other languages. Different standards bodies, the International Organization for Standardization (ISO) and American National Standards Institute (ANSI), began creating extended ASCII character sets to address the problem and expressed through different code pages.[12] As a result, content extraction software needed to know which encoding set was used to properly extract text.

[12] Read more about extended ASCII at: https://en.wikipedia.org/wiki/Extended_ASCII

In the 1990s, a worldwide consortium of computer scientists developed the universal encoding standard Unicode, which integrated the original ASCII characters and today supports approximately 150,000 characters plus multiple symbol sets and emojis. Now in version 13, Unicode provides a basis to encode over 150 modern and historic languages.[13] The most widely used version of Unicode is UTF-8 (Unicode Transformation Format); some systems also use UTF-16.

For processing purposes, the text extraction system must recognize how a system was encoded, whether it was in ASCII, one of the extended ASCII sets or in Unicode. If the latter is used, it must also distinguish between UTF-8 or UTF-16. If the system mistakes the encoding, it is likely the extracted text will be partially or completely unusable and searches will be inaccurate..

3.4 Detect Language

While many e-discovery matters involve documents that are only in English, others involve documents written in different languages. Detecting which language is used in ESI is important to processing for several reasons.

First, language identification ensures the right filters are used for text extraction, tokenization and diacritical handling. Second, language identification may be important during the review stage when documents are assigned by language to ensure they are properly analyzed by foreign language reviewers. These foreign language documents may also be identified for machine translation but this should not be used as replacement for translation by a linguist.

3.5 Extract and Normalize Text

During the text extraction, the processing engine makes a series of decisions about how that text should be stored for indexing. First, the text is normalized to ensure it is stored consistently and understandably for later searches. The following are key considerations for normalization.

Case Normalization: Should capital letters in text be reduced to lower case? Some systems are case sensitive, ensuring that a search for RAM (memory) will not return Ram (sheep) or rams (car accidents). To locate all variants of RAM regardless of case, normalize the text to lower case.

[13] Read more about Unicode at: https://en.wikipedia.org/wiki/Unicode

Diacritical Normalization: Many languages use accents and other diacritical marks to distinguish between otherwise equivalent characters. For example, a job applicant may submit a résumé or resume (or even a resumé). A search for “resume” would return results for “resume” but not the other two variants. Likewise, a search for “résumé” would only find that variant. Some languages also include ligatures that are used to combine two letters. For example, “æ” is sometimes used in the phrase “curriculum vitæ.”

Normalizers used in many processing systems convert these Unicode characters into their ASCII equivalents. For example, instances of résumé, resume and resumé are converted to “resume” for purposes of search. “Curriculum vitæ” would be converted to “curriculum vitae” be retrieved regardless of how it was searched (assuming the search system similarly normalized the text input).

Unicode Normalization: The Unicode standard allows accented characters to be identified in two ways: (1) the character is represented by a specific code value—a precomposed character and (2) the character is split into two parts -- the ASCII equivalent and the accent character. Thus, the “é” in résumé could be represented by its specific code value or as two characters—the letter “e” and the accent “aigu.”

Unicode normalization reduces these equivalent character values to the same values. Splitting out the accents broadens the search but is less efficient if the searcher wants to locate only the accented version.

Time Zone Normalization: The dates and times of sent and received emails can often be important to an ESI investigation. When email collections involve different time zones, it is important to normalize the time zones to a single standard. Otherwise, it may appear that a reply email sent from a California location (Pacific) to Boston (Eastern) was sent several hours before the original email from Boston. This may cause confusion when legal professionals prepare a timeline of events.

Most email processing engines normalize date and time values against the common standard, often Greenwich Mean Time (GMT), aka Coordinated Universal Time (UTC). Doing so provides a later viewer to review email communications against UTC or to convert it to a different but consistent format.

Text Tokenization: This step is critical because the search engine used for keyword searching must rely on an index of tokenized text to retrieve results quickly and efficiently.

A token is the lexical unit placed in the search index. It may be a word or any combination of letters or numbers that are grouped together in a wordlike construct. Thus, a set of letters or numbers or even a misspelled word is treated as a token and placed in the index.

During text normalization, the system separates the words to be properly indexed later. For English and most western languages, this is done by removing most punctuation and using the spaces between each token to define it as a separate unit. Thus, the phrase “natural-born citizen” is tokenized into three separate tokens: natural, born and citizen. Likewise, the phone number 303-777-1245 is treated as three separate tokens, 303, 777 and 1245.

Tokenization is more difficult for many Asian languages and others that do not use punctuation or spacing between words. In Japanese, the phrase “You have breached the contract” is written as あなたは契約に違反しました. How does the computer separate these characters into individual words for later searching?

Special tokenizing software is used to break apart the individual word tokens used in these languages. Modern processing software should include appropriate tokenizing software for Asian and other languages that do not use punctuation marks or spacing to define words.

Stop Word Normalization: Many search engines do not index certain commonly-used words such as a, an, and, the, etc. Others do not index numbers, symbols or two-letter combinations. Most do not index standard punctuation characters, such as quotation marks, hyphens, and percent characters that are also stripped for tokenization purposes. For example, if the “&” symbol is omitted, you will not easily be able to search for “AT&T”

Processing engines often follow these rules but should provide the administrator with a choice in this regard. If your search engine indexes every character, the processing engine should offer this option as part of text normalization.

Special Settings for Office Files: Modern Office files and their equivalents from other publishers, provide a wide range of options for preservation during text extraction. Some record the editing history of the file; many permit the addition of comments or notes. In most cases, this information is not presented when the Office document is printed or converted to PDF.

The text extraction software may be set to extract this additional information and include it for indexing, searching and reviewing later. These are typically included as special settings that may or may not be requested depending on the matter and its needs.

3.6 Extract Metadata

Processing software must extract metadata, defined as information about the file itself. Specifically, if possible it should t extract information about the file maintained in the operating system as well as internal information maintained in the file. In the case of email, the processing software should also track information collected from the email database (e.g., MS Exchange or Office 365) and information maintained within the file.[14]

Modern processing systems can track hundreds of metadata fields ranging from the original file name and basic creation information to internal fields such as author, date last saved and date printed. Emails contain hundreds of metadata fields including basics such as from, to, cc, bcc, subject, and sent date/time. Outlook files contain more than a hundred metadata fields, most of which are not relevant to an e-discovery investigation.

A processing engine must extract the basic metadata fields and should provide choices to the administrator as to which fields should be included in the processing output.

3.7 Extract Images

Email and other file types often allow content creators to embed images and other programs within the file. Photos and other graphics are common additions to email and office documents. Spreadsheets are often embedded in Word or PowerPoint files in addition to text or email messages.

Processing software must be able to separate embedded files, often treating them similarly to email attachments and allow the administrator to choose not to extract pictures or other image files because they have little to no value outside of the original file. A logo attached to an email has little probative value when it is separated from its original message. These extractions can substantially add to the number of records exported to the litigation support system, making review more difficult because of the increased volume of largely irrelevant records.

[14] Email and other files maintained in the cloud, for example Office 365 files, may not allow the extraction of operating system metadata.

3.8 Handling Mobile Devices: SMS and IM

Smartphones and other mobile devices have exploded in popularity over the past two decades. They now record a large amount of human and social activity, in many cases supplanting traditional desktop computers and email communication.

SMS (Short Messaging Service) and IM (Instant Messaging) involve non-document data types that may be relevant to an e-discovery matter, and which require special treatment during the processing phase.

When working with mobile devices, there are two areas of interest:

1. **Text Messaging:** Mobile devices offer a variety of means to send communications to others. SMS and MMS (Multimedia Messaging Service) are the most universal of these referred to as “text messages”. Closely related is iMessage data, Apple’s proprietary messaging service that works alongside SMS and MMS data on devices such as iPhones and iPads.
2. **Third Party Messaging Software:** Third party software such as WhatsApp and Facebook Messenger offer their own messaging platforms. Their content is typically stored in proprietary databases and may be extracted using specialized software.

Mobile device collection often involves a forensic component. Typically, an examiner will collect data from the device itself or from a backup server using specialized software to extract the available data and export it in a usable format.

The most common export format for phone systems is an Excel workbook comprised of various worksheets, each of which corresponding to a data type from the device, such as messages, voicemails, call logs, etc. These worksheets contain rows of data that correspond to individual messages, voicemails, etc. The worksheet columns provide the metadata for each item—sender, recipient(s), timestamp, body text, date sent, etc.

While messaging is a core function of mobile phones and other smart devices, the devices also function as powerful, handheld computers in their own right, as well as repositories for files, pictures and traditional emails. Relevant information that may be extracted includes call logs, contacts, calendar items, voicemails, and more.

3.9 Handling Data Collected from Social Media and Collaboration Platforms

Closely related to, but distinct from, mobile data is data generated from various interactive social platforms that often contain a strong messaging/chat component, as well as other forms of communication, such as social media posts or file sharing.

Below are several examples of data that may require ediscovery processing.

Workplace Collaboration Data: This form of data has risen in prominence with the proliferation of work-from-home-policies in the wake of COVID-19.

Essentially, workplace collaboration software provides a platform for interaction and collaboration among teams of individuals, usually in a workplace setting. Three of the most well known examples are Slack, Microsoft Teams and Google Chat. These tools provide instant message functionality and hosted chat rooms to facilitate file sharing and other collaborative activities. Data from these platforms is typically exported in CSV, JSON, or an XML format.

Social Media: With billions of users, social media sites, such as Facebook, Twitter, and LinkedIn, store content that may be relevant to a wide range of matters from criminal investigations, to family matters and business disputes. The content can be similar to the workplace collaboration platforms with an emphasis on social interactions, rather than workplace interactions.

Social media sites frequently involve public or semi-public posts, which often appear on a timeline. Some sites have a specific focus. Instagram, for example, focuses on media content, such as photos and videos. There are a number of specialty software products used to collect data and to “scrape” public information from the sites.

Website Content: In some cases, the website content can provide relevant information for an e-discovery matter. There are a number of collection software programs that can extract information from websites, whether text or page representations. These collections are typically date and time stamped and exported in different formats such as PDF or HTML.

3.10 Exporting Mobile, Collaboration and Website Data

Traditional ESI for e-discovery consists of document formats such as PDF, email and Office files. The non-traditional formats extracted from mobile devices, collaboration software and websites more closely resemble rows in a table and may be understood as streams of event data or activity logs, rather than static files. As a result, the process of loading the data into a review platform can present challenges.

Often special software is needed to convert the exported data into a more traditional load file that can be imported into a document review platform. In such cases, it may be necessary to transform discussion segments, such as a day of conversations, into a single document that can be loaded, searched and viewed in a document review platform.

While a more traditional load file format may make sense based on your document review platform, vendors are starting to create alternatives to a traditional load file that enables reviewers to view and search the conversation in a format that more closely resembles how the custodian actually interacted within the conversation.

A unique component of these non-traditional formats is the various metadata that may or may not be included for each of these events. Given this, production requirements must be aligned upon sooner than with traditional document production, or mobile data productions. It is advised that prior to processing parties understand what data is available for production from collaboration and website data. As mentioned previously, the definition of a document may be more fluid with this type of data, though the main goal is to help reviewers sift through portions of a conversation, rather than attempting to review an entire conversation over many years.

4.0 Processing Output

The goal of e-discovery processing is to convert selected files, images, text and metadata into a format appropriate for loading into litigation support software by delivering one or more load files, along with the associated native, image and text files that result from processing. In some cases, native files are converted into an image format and delivered with the natives.

Many of the standard steps for delivering processing output are optional depending on the needs of the matter. The following are typical options in e-discovery processing software:

4.1 Keyword and Metadata Filtering

In some cases, legal professionals may want to further filter and reduce the volume of files being promoted for review through metadata filtering based on criteria such as date ranges, custodians, recipients, and subjects. Keyword searching for terms and phrases in documents or message text relevant to issues that may be used to support the claims and defenses of a party may be used to filter and reduce data volume.

Metadata searches can be run against the processing database. Keyword search requires the processing software index text extracted from processed files. Often, the search engine used for this purpose will be similar to the one used during the search and review phase.

In most cases the assignment is to run relatively simple metadata and keyword searches that can be used to safely cull the processing output before promoting it to the next EDRM stage. Culling searches reduce the volume of files for review and analysis. Note that keyword searches can miss relevant documents for a variety of reasons including poor search construction, misspellings of key terms, the use of acronyms, synonyms or code names not included in the search syntax, etc. Thus, there is always a tradeoff between what information data scientists refer to as “precision” and “recall” when using keyword search to find relevant information or to reduce the review population.

4.2 Developing Load Files

Litigation support software used during the review phase needs metadata for a number of reasons. Metadata is used to provide information about the document or message to reviewers. The resulting metadata fields are used to filter searches and to sort results. The database must include appropriate information to link each record to the underlying native, image and text documents that are output during processing. This allows the reviewer to not only search against fields and text but to review that information together on a computer screen.

Thus, the output of a processing system must include a load file that provides important information about each document and also facilitates data and file loading into the system. There are a number of standard load files used in e-discovery processing including:

- Concordance load file (DAT and LFP/OPT for images)
- Other Custom Delimited Files[15]

[15] These are often referred to as CSV files, a reference to a delimiter format involving comma separated values. In practice, the use of a comma to separate values can cause problems with data because there

- Microsoft Access database file (MDB)
- Summation load file format and
- EDRM load file format.

These standard formats may not be sufficient for all types of processing output and particularly for SMS and IM collections. These formats often include emojis and various picture formats which might not be rendered consistently with the original content. Likewise, the conversational format may be lost if the system tries to render the data in a traditional page format. As a result, new load file types are being developed for these conversational formats by different processing companies.

Ultimately, a good processing system will allow administrators to choose between different types of load file formats based on the chosen litigation support software.

4.3 Converting Native Files to Images

Some litigation support software requires files be converted to images for viewing. There are several standard image formats available, including:

- Single-page Tiffs: This format includes one image per file with no text included.
- Multi-page Tiffs: This format includes multiple images per file but does not include text. These are typically not used in litigation support software.
- PDFs: This is a multipage, color file format which may include text information and is used in many modern litigation support systems.
- JPEGs or PNGs: These are typically used for color images in systems that rely on TIFFs for basic image format.

In the early days of e-discovery, single page TIFFs were the standard. Today, many systems prefer PDFs or even a near native format such as a SVG (support vector graphic) because they can display color and the image files are compressed.

can be commas occurring within field elements. For example Smith, John may be the value in a Name field. The better practice is to use other delimiters, e.g. pipes |, carets ^, that are unlikely to appear naturally in the data.

4.4 Creating Text Files

Many litigation support software systems require separate text files for each image or native file output. The review systems then index and make searchable the separate text files. Once users retrieve a document through search, the associated image or native rendition of the document is analyzed for relevance to the inquiry.

4.5 OCR Image Files

With the exception of some types of PDF files, most images do not have extractable text and are not keyword searchable. As a result, many processing systems include the ability to OCR (optical character recognition) files so that text can be extracted for later search.

While OCR does not always capture image text correctly, modern OCR software does a good job of correctly identifying text from scanned images. At the least, OCR'd text is better than having nothing.

4.6 File Names

File naming is an important output function for processing software. While each file has an original name, it is common to rename files to correspond with an assigned control number.[16] Control numbers are often issued consecutively as files are processed which may provide some information as to file origin and proximity to other files. Many processors include a text prefix or suffix to provide further information about a file's origin or purpose.

Many call these IDs "Bates numbers." Bates referred to a popular band of a hand number stamper that was used to identify paper documents that were being produced. Computers typically overlay these numbers on document images for production purposes while inserting them in a field for the associated database record.

4.7 Family Relationships

As mentioned earlier, many email files act as containers for their attachments. Processing software must extract these attachments, which can include additional container files each of which must be exploded recursively. In doing so, the software

[16] In such cases, it is important to save the original file name as part of the processing metadata.

must number the attachments consecutively and keep a record of the control numbers for the parent email and its attachments.

When files are output, it is important to have a record showing which files were part of an email or container family.

Most litigation support software will preserve the family relationship such that a reviewer can review emails by family and, in some cases, tag both the parent email message and its attachments at one time.

4.8 Email Threading

Many email messages are part of a larger conversation, involving the original message and one or more replies. When an email is sent to multiple recipients, the number of replies and further replies held by the recipients can be large and spread throughout the collection. Review of these files can become repetitive and inefficient.

Many processing systems include the ability to analyze certain components of an email message to determine whether it is part of a larger conversation. If so, it will provide links to the larger conversation so that it can be reviewed in an integrated manner.

4.9 Near Deduplication

Some processing systems offer the ability to identify files that are similar in content although they do not match out using the hash algorithms described above. In some cases, they are similar but for a slight change in a metadata value, perhaps in a message header. In other cases, the body text may be different because the documents are different but highly-similar drafts of the same content.

Grouping these documents together through links or other reference information can be valuable in processing because the files can later be reviewed, and sometimes tagged, as a group. Doing so, promotes review efficiency particularly when compared to the prospect of reviewing each file separately. The potential for inconsistent tagging is reduced as well.

5.0 Reporting

Reporting is a critical part of e-discovery processing. From the beginning, the software should track the files received from collection and the actions taken on those files. An initial container file, for example, should be linked to the identity of the files extracted from it and those files should be tracked so an administrator can see every step taken on the files from ingestion to system output. All of this information should be available for searching, sorting and filtering, with the results available in a standard format for export or printing.

5.1 File Inventory Reporting

Processing systems should provide inventory reports showing the number of files contained on a given piece of media, the type of files contained on the media, and the size of the data contained on the media. In addition, directory lists of the file names should also be available and is generally referred to as a file inventory report.

5.2 Custodian Reporting

Custodian level reports provide data regarding files received for each custodian. A typical report will include the custodian’s name, records received and processed, file dates, types and sizes along with exception information for files that could not be processed.

5.3 Filtering Reports

Filtering reports are designed to show the volume of files removed or not promoted as a result of the different filters run against the data. This could include virus and NIST removal, along with date range and file type searches run.

5.4 Chain of Custody

Chain of custody is a term often used in criminal matters to reflect the rule that evidence should not be altered during the time it is in police hands. During processing, chain of custody refers to a report showing how each file was handled from reception to output. The purpose is to provide assurance that the file and its associated metadata has not been altered to the benefit of the party offering it as proof.

Chain of custody also refers to the receipt and maintenance of drives and other electronic media holding collected files and other data, which should be stored securely in a tamper proof vault when not in use.

Processing software should record all of the file handling steps taken during this phase of the EDRM for purposes of chain of custody tracking.

5.5 Exception Reporting

Files which cannot be processed should be identified in the processing database as exceptions. These are files for which no text or metadata can be extracted or for which no image can be rendered. This category may include encrypted or corrupted files, system files, program files or some other type that will not render information.

Exceptions information should be available for search and reporting. Ideally the report will provide the reason the files could not be processed. As an example, an exception report might include the following information:

File name, original directory location of the file, reason for exception (failure).

Reasons for file exceptions might include file corruption, encryption, password protected, virus infection, zero byte file, or NIST exclusion.

Conclusion

Although sometimes invisible to the user, the processing stage of e-discovery involves a complicated number of steps as data moves from preservation and collection to analysis and review. Processing software can make the workflow easier to master and automate most of the steps involved. Nonetheless, legal professionals must understand the many functions involved and often make decisions about which steps should be included for a particular data set and/or which options at each step should be chosen.

The end goal of the processing stage is to prepare data, documents, email, files, instant messaging, etc, for the next and arguably most important stage of the process—analysis and review. If the data has not been processed properly, the resulting output may not be searchable, may contain bad metadata or may not even be reviewable. In such a case, the fundamental purpose of the e-discovery process is compromised and the ability to locate relevant information for trial or hearing fails.

EDRM Processing Glossary

Term	Definition
ASCII	American Standard Code for Information Interchange (ASCII) is a plain text character encoding standard where seven- or eight-bit integers correspond to 128 or 256 characters and codes for electronic storage and communication. The eight-bit pairings are often mistakenly referred to as *Extended ASCII*. The 128 characters in 7-bit ASCII encoding correspond to 95 printable characters (a-z, A-Z, 0-9 and punctuation) and 33 non-printable control codes, e.g., carriage return, line feed, tab and bell. The 256 ASCII characters enabled by 8-bit integers are for various purposes, *e.g*., foreign language characters and line drawing symbols.
Bates Numbers	Sequential numeric identifiers imprinted on document pages or assigned to files during the discovery process. Bates Numbers typically include a prefix to identify the producing party or matter as well as a numeric value (*e.g.*, DEF_000000001).
Binary File Signature	Also known as "file header signature," "binary header signature" or "magic number." Typically the first few bytes of data in a file identifies the format of the data contained therein. For example, ZIP compressed files begin with Hex 504B (or the initials PK in ASCII). Most JPG image files begin with Hex FF D8 FF E0.
Case Normalization	Improves search recall by adding information to an index that searches for terms with lowercase characters and identifies its uppercase counterpart and vice versa. For example, a search for Rice will also find instances of RICE and rice.
Character Normalization	Seeks to minimize the impact of variations in alphanumeric characters often overlooked by human beings but posing a challenge to machines. This may include Case Normalization, Diacritical Normalization and Unicode Normalization..

Chain of Custody	The procedures employed to protect and document the acquisition, handling and storage of evidence to demonstrate these activities did not alter or corrupt evidentiary integrity.
Compression	The storage or transmission of data in a reduced size by using technology to eliminate redundancy ("lossless compression") or by removing non-essential details (such as, picture elements in a JPEG or inaudible components of audio). Compression permits more efficient storage, sometimes at the cost of reduced fidelity ("lossy compression"). ZIP, RAR and TAR are common lossless compression formats in eDiscovery.
Container File	A file that holds or transports other files, *e.g.*, compressed container files (.ZIP and .RAR) and email container files (.PST and .MBOX). Container file content is "unpacked" or “exploded” during processing enabling the container file to be suppressed as immaterial once fully extracted.
Corruption	Damage to the integrity of a file that impacts its ability to be processed. File corruption may be caused by, *e.g.,* network transmission errors, software glitches, physical damage to storage media (i.e., bad sectors) or use of an incompatible decoding tool.
Custodian	The individuals or entities who hold, or have the right to control, records and information.
DAT File	A delimited load file used in conjunction with Concordance-formatted productions. A .DAT file includes a header row of field identifiers that corresponds to the data that follows. Each field is separated ("delimited") by a character ("delimiter") that signals the division of fields.

Deduplication	The identification and suppression of identical copies of messages or documents in a data set based upon the items' hash values or other criteria.
DeNIST	The use of hash values to identify, suppress and/or remove commercial software from a data collection. The hash values are maintained by the National Institute of Standards and Technology (NIST) in its National Software Reference Library (NSRL).
Diacritical Normalization	Improves search recall by adding to an index terms with diacritics (*e.g.*, accented characters) so as to locate counterparts without diacritics. For example, a search for "résumé" would also locate instances of resume and vice versa.
DTSearch	A content extraction, indexing and text search tool licensed to and at the heart of several leading e-discovery and computer forensic tools (*e.g.,* Relativity, LAW, Ringtail (now Nuix Discover) and Access Data's FTK).
Elasticsearch	An open source content extraction, indexing and text search tool used by a number of software providers for indexing and keyword search. It is based on the Lucene open source search engine library project.
Encoding	The process of converting electronically stored and transmitted information from one form to another. Character encoding maps alphanumeric characters into numeric values, typically notated as binary or hexadecimal numbers. ASCII and Unicode are examples of character encoding.
Encryption	The process of encoding data to unintelligible ciphertext to prevent access without the proper decryption key (*e.g.*, password).

ESI	Electronically Stored Information (ESI) as defined by Federal Rule of Civil Procedure 34(a)(1)(A), includes "writings, drawings, graphs, charts, photographs, sound recordings, images, and other data or data compilations—stored in any medium from which information can be obtained either directly or, if necessary, after translation by the responding party into a reasonably usable form."
Exception Reporting	This process of identifying items which fail during processing. Exceptions may include encrypted files that cannot be read, corrupt files, files in unrecognized formats or languages, and files that require optical character recognition (OCR) for text extraction.
Family Group	In the context of an email, a transmitting message (parent object) and its attachments (child objects).
File Header Signature	Also known as a "binary header signature," "binary file signature" or "magic number." Typically the first few hex bytes of data in a file identifies the format of the data within the file. For example, ZIP compressed files begin with Hex 504B (or the initials PK in ASCII). Most JPG image files begin with Hex FF D8 FF E0.
Filtering	The process of culling files from a data set based on characteristics such as, file type, date and size. In e-discovery, files are filtered to suppress multiple copies of the same item (deduplication), irrelevant system files (deNISTing), immaterial container files after content extraction and by lexical search (filtering by keywords).
Forensic Image	An exact, verified copy of electronic media. Forensic imaging produces a hash-authenticated, sector-by-sector ("bitstream") copy of electronic media that can be restored for analysis. This process is typically used to preserve active data, unallocated clusters and file slack space.

Hash	A "digital fingerprint" of data or "message digest," generated by a one-way cryptographic algorithm (*e.g.*, MD5, SHA-1, SHA-256) and recorded as a hexadecimal character string, *e.g.* 13bfb1528002a68d94249c4ffb09359f. The potential of two different files having matching hash values is so remote that hash value comparisons serve as effective tools for file authentication, file exclusion (DeNISTing) and data deduplication.
Identification	In e-discovery, the mechanism by which a processing tool determines the structure and encoding of a file based upon the file's header signature and filename extension.
IM	Instant Message (IM) is a form of real-time text communication over the Internet typically expressed in conversation form. IM can involve communications between two people or larger groups, who sometimes communicate in “rooms.”
Image Format	Images initially referred to the output from document scanning but can also refer to files rendered directly from native files. These files are created to emulate a printed page. In e-discovery, the most common image formats are Tagged Image File Format (TIFF), Portable Document Format (PDF) and JPEG. "Rendering" is the processing step where ESI is converted to image formats.
Index	A data structure that improves the speed of search for data retrieval. E-discovery employs full text indexing of processed data to speed search and to reduce storage space.
Ingestion	The act of loading data into an application for processing.
Keyword	Search term used to query an index or database.

Language Detection	Recognition and identification of foreign language content that enables selection of appropriate filters for text extraction, character set selection and diacritical management. Language detection also facilitates assigning foreign language content to native speakers for review.
Load File	An ancillary file used in e-discovery to transmit, system and application metadata, extracted text, Bates numbers and structural information describing the production. Load files accompany folders holding native, text and image files and provide essential information about the files being transmitted.
Lucene	An open source library for content extraction, indexing and text searching used by a number of software providers for indexing and keyword search. Elasticsearch and Solr are based on the Lucene library.
MD5	Message Digest 5 (MD5) is a common cryptographic hash algorithm used for file authentication, file exclusion (DeNISTing) and data deduplication.
Metadata	Data describing the characteristics of other data. File metadata may be System Metadata (*e.g.,* file name, size and date last modified, accessed or created are stored outside the file) or Application Metadata (*e.g.,*last printed date or amount of editing time stored within the file). The term metadata can also include human judgments about a file, e.g. hot or privileged, or information about the file, e.g. from, to, subject, sentdate.
MIME	Multipurpose Internet Mail Extensions (MIME) refers to a two-part, hierarchical method of classification for electronic files. MIME Types (also known as Media Types) classify files within one of ten types: *application, audio, image, message, multipart, text, video, font, example* and *model*. Each type is divided into subtypes with sufficient granularity to describe all common variants within the type. For example, the MIME Type of a PDF file is "application/pdf," a .DOCX file is "application/vnd.openxmlformats-

	officedocument.wordprocessingml.document," and a TIFF image file is "image/tiff." The Internet Assigned Numbers Authority (IANA) is a standards organization that registers new types and subtypes in the MIME Type taxonomy.
Media Type	Alternate term for MIME Type, see MIME.
Native Format	In the context of software applications, native format refers to the file format which an application creates and uses by design—generally the default, unprocessed format of a file when collected from the original source, *e.g.,* Microsoft Word stores documents as .DOCX files, their native format.
NSRL	The National Software Reference Library (NSRL) is maintained by the National Institute of Standards and Technology (NIST), an agency of the U.S. Department of Commerce. The data published by the NSRL (principally hash values of commercial software) is used to rapidly identify and eliminate known files, such as operating system and application files.
Noise Words	Common terms purposefully excluded from a searchable index to conserve storage space and improve performance. Also known as "stop words."
Normalization	The process of reformatting data to a standardized form, such as setting the date and time stamp of files to a uniform time zone or converting all content to the same character encoding. Normalization facilitates search and data organization.
OCR or Optical Character Recognition	The use of software to identify alphanumeric characters in static images (*i.e.,* TIFF or PDF files) to facilitate text extraction and electronic search. OCR programs typically create matching text

	files that are used for text search with the accompanying images.
Processing	Encompasses the steps required to extract text and metadata from information items and to build a searchable index. ESI processing tools perform five common functions: (1) decompress, unpack and fully explore (*i.e., recurse)* ingested items; (2) identify and apply templates (filters) to encoded data to parse (interpret) contents and extract text, embedded objects, and metadata; (3) track and hash items processed, enumerate and unitize all items, and track failures; (4) normalize and tokenize text and data and create an index and database of extracted information; and (5) cull data by file type, date, lexical content, hash value, and other criteria.
Recursion	The mechanism by which a processing tool explores, identifies, unpacks and extracts all embedded content from a file, repeating the recursive process as many times as needed to achieve full extraction.
Request for Comment (RFC)	The longstandinginformal circulation of proposed protocols and standards among computer scientists, engineers and others interested in the development of the Internet and other networks. RFCs define the structure of email messages and attachments for transmission via the Internet.
SHA	Secure Hash Algorithm (SHA) (SHA-1, SHA-256) is a family of cryptographic hash algorithms used for file authentication, file exclusion (DeNISTing) and data deduplication.
SMS	Short Message Service (SMS) is a communication protocol that enables mobile devices to exchange text messages up to 160 characters in length.
Solr	An open source content extraction, indexing and text search tool used by a number of software providers for indexing and keyword search. It is based on the Lucene open source search

	engine library project.
Stop Words	Common terms purposefully excluded from a searchable index to conserve storage space and improve performance. Also known as "noise words."
System Files	The program and driver files crucial to the overall function of a computer's operating and file systems. Because system files are not user-created, they may be excluded from a collection of potentially responsive data by deNISTing.
Targeted Collection	A technique used to reduce overcollection of ESI by marshaling potentially responsive data based on data characteristics (such as, file type, date, folder location, keyword search, etc.) as opposed to duplicating the entire contents of a storage device (*e.g.,* by imaging).
Threading	Collection and organization of messaging as a chronologically ordered conversation.
Tika	An open-source toolkit for extracting text and metadata from over one thousand file types, including most encountered in e-discovery. Tika was a subproject of the open-source Apache Lucene project. Lucene is an indexing and searching tool at the core of several commercial e-discovery applications.
Time Zone Normalization	The recasting of time values of ESI--particularly of e-mail collections--to a common temporal baseline, often Coordinated Universal Time (UTC) or another time zone the parties designate.
Tokenization	A method of document parsing that identifies words ("tokens") to be used in a full-text index. Because computers cannot read as humans do but only see sequences of bytes, computers employ programmed tokenization rules to identify character sequences that constitute words and punctuation.

	Western languages typically use spaces and punctuation to identify word (or token) breaks. Because other languages, *e.g.* Chinese, Japanese and Korean, do not use these methods to break characters into words, I tokenization software ensures that words and other tokens are indexed properly for search.
Unicode	An international, multibyte encoding scheme for text, symbols, emoji and control codes. Unicode 13.0 offers 154 encoding schemes or scripts comprising 143,859 characters. Unicode was developed to overcome the limits of the single byte ASCII encoding scheme that lacked the capacity to encode foreign language characters and other symbols needed for international writing and communication. Unicode is now the standard for Western and international text encoding.
Unicode Normalization	Improves search recall by adding information to an index that locates Unicode characters encoded in multiple ways when searching with any counterpart encoding. Linguistically identical characters encoded in Unicode (so-called "canonical equivalents") may be represented by different numeric values by virtue of accented letters having both precomposed (é) and composite references (e + ◌́). Unicode normalization replaces equivalent sequences of characters so that any two texts that are canonically equivalent will be reduced to the same sequence of searchable code called the "normalization form" or "normal form" of the original text.
UTF-8	Unicode Transformation Format (character encoding 8) or UTF-8 is the most widely used Unicode encoding, employing one byte for standard English letters and symbols (making UTF-8 backwards compatible with ASCII), two bytes for additional Latin and Middle Eastern characters, and three bytes for Asian characters. Additional characters may be represented using four bytes.