DRAFT Legal Matter Specification Standard (LMSS), Rev. 2

Version Revision 2c

Last Update: 2019-06-08

SALI Matter Specification Committee

Adam L. Stock

Contents

1 Overview

The SALI Legal Matter Specification Standard (LMSS) was developed by the Standards for the Advancement of the Legal Industry (SALI) Alliance to provide a standard way for parties to specify, describe and exchange information describing legal services at the matter level throughout a legal matter's lifecycle. The SALI LMSS is designed to function as a worldwide legal matter standard.

1.1 What do we mean by a "legal matter"?

For the purpose of the LMSS, a **Legal Matter** is considered to be any group of activities for the purposes of delivering a legal service to one or more parties. The activities are either a major project with start and end point (e.g. litigation, acquisition of a business, a regulatory filing, etc.) or a grouping of micro projects (e.g. advice).

1.2 What do we mean by the "legal matter lifecycle"?

The legal matter lifecycle means the description of the matter from its conception as a request for services through its inception and execution through closing. Because the LMSS must support a matter that will evolve through its lifecycle, it is designed accommodate the evolution of the matter without jeopardizing the integrity of systems that are depending on data provided at an earlier stage of the matter.

1.3 What are the components of the LMSS?

The SALI LMSS has following components:

- **The legal matter specification** includes: structure, allowed values, and dependencies and supports both matter instances (the description of specific matters), and matter templates (the description of classes of matters).
- **The legal matter application programming interfaces (APIs)** include: the transport API for exchanging data and the UX API for supporting the user interface of applications that use the standard.

2 Design Principles

The SALI LMSS was designed using the following principles:

2.1 Player/Viewer independence

The standard is designed to provide clear guidance and party-independent enumerated values to ensure that matters are encoded the same way independent of the party involved in matter. As an example, the standard supports terms such as “plaintiff” and “defendant” which have the same meaning no matter who is looking at the matter over terms like “client” and “opposing party” which change depending upon a player’s role in the matter.

2.2 Successive refinement and additive coding

Because the LMSS is intended to be used legal matters to describe legal matters that are being executed, the standard implements hierarchical and additive coding. As an example, when a matter begins, the area of law for a matter may not yet be fully determined, for example, we may know that it is a environmental matter, but not what specialty in environmental law. Such a matter would initially be encoded as environmental – ENVT. At a later point it may be determined that the matter is an air quality matter -- AIRQ. This would be encoded as "ENVT-AIRQ" The successive code is added to the previous code and is a further refinement of the previous code. This design ensure that other systems that depend upon the initial restructuring code continue to work.

Furthermore, each code in a code set is guaranteed to be unique, and each code is guaranteed to have a single parent code. This means that if you simply use the code "AIRQ", the full coding path "ENVT-AIRQ" can be derived. In cases where the SALI standard has adopted codes from other standards that don’t adhere to this requirement, the "+" notation is used. For example both US, California and El Salvador, Cardenas use the same ISO code "CA". To distinguish these, the cod is prepended with the parent code followed by a "+". USA ("US"), California ("CA") is coded as "US+CA"; El Salvador ("SV"), Cardenas ("CA") is coded as "CV+CA".

2.3 Standards based

The LMSS is built on and incorporates existing international standards in addition to specifying new codes applicable to the legal services domain.

2.4 Extensible

Since the LMSS cannot anticipate all needs, there is a well-defined way to extend the standard while staying within the design and API functionality. There is a methodology for providing including private structural components and private allowable values that at some point may be submitted and considered for incorporation into to future versions of the LMSS.

2.5 Agile approach

The LMSS is designed using an agile approach wherein the proposed standard drafts are revised and tested by a broad group of stakeholders representing law firms, clients and solution providers to ensure that the standard is practical and appropriate to the specific needs and use cases being addressed.

The process is as follows is described in the table below:

Name	Description	Comment
Draft LMSS 1.0 Rev. 1	First draft revision 1 of the LMSS 1.0 specification	The first version always had the word "Draft" and "Rev. <#>" in its title.
Draft LMSS 1.0 Rev <n>	Nth draft revision 1 of the LMSS 1.0 specification	Foe each successive draft, the revision number is incremented.
LMMS 1.0	Final version of the LMSS 1.0 standard	"Draft" and "Rev. <#>" are removed from the title once the standard is adopted by SALI Alliance members.
Draft LMSS 2.0 Rev 1	First draft revision 1 of the LMSS 2.0 specification	Successive versions of the standard will increment LMSS number.

3 Legal Matter Specification (LMSS) Components

The legal matter specification includes the following components:

- structure,
- allowed values,
- dependencies, and,
- extensions.

In addition, the LMSS supports both:

- matter instances (the description of specific matters), and
- matter templates (the description of classes of matters).

3.1 LMSS Structure

The LMSS Structure specifies where descriptive elements of matter are stored and how those elements relate to each other. The structure is comprised of components called “containers.” Containers can be thought of as tables in a relational database. The structure defines structure of each container – its elements – how containers relate to each other, which elements are required and which are optional, and whether the relationship between containers are one-to-one or one-to-many.

The LMSS structure can be expressed as a database schema for storage or an XML or JSON structure for transmission of the matter information.

3.2 LMSS Allowed Values

The LMSS Allowed Values specify the type of information that may be stored in different elements of the containers. Allowed values fall into the following categories:

3.2.1 Enumerated Values

Enumerated values are specifically defined values that are allowed at permitted as values of specific container elements.

The LMSS relies extensively on enumerated values to ensure that common definitions across systems and languages. Some standards are officially incorporated in the standard such as ISO-4217, which provides standard codes standard names codes for currencies.

Enumerated values components are summarized in the table below.

Enumerated value components

Type	Characters	Explanation
Code Set	Text(40)	The name of the code set
Code	Up to 16	A code for the value. Codes are alphanumeric characters excluding the '-' character. They are case insensitive. Codes are typically stored in Text(250) fields because codes may be additive. Separate code may be appended to each other using the "-" character as a separator. The 16 character limit does not include extension prefixes. Codes are of the form: (@[A-Z0-9]+:)?([A-Z0-9.]+ | [A-Z0-9.]+[+][A-Z0-9.]+) '+' has a special meaning to support non-conforming code sets.
Parent Code	Text(250)	The code of the of the parent in a hierarchical coding system. This is null for top level codes.
Short Name	Up to 40	The name of the value intended for use in user interfaces and other applications with limited space.
Name	Up to 100	The full name of the value.
Description	Up to 4000	A description of the value.
Tags	Up to 10	Synonyms or common usage. Tags are used to enhance search
URL	Text(250)	Optional URL used as a reference

Enumerated value codes are combine by adding a "-" (dash character) and the next most specific code as a suffix.

Examples

Code	Concatenated Code	Short Name
NAM	NAM	North America
US	NA-US	United States
US+CA	NA-US-US+CA	California

3.2.2 Text Values

Text values are unstructured natural language descriptions intended to be human readable and informative. Text values are also used to store information not yet

specified by the standard. Text values are intended to be searchable, and subject to standard text operations.

3.2.3 Numeric Values

Numeric values store scalar numeric information. Numeric values are subject to common mathematical operations. Numeric values can be floating point, whole number and boolean.

3.3 Dependencies

Dependencies restrict which structural components and which allowed values may appear in a valid matter instance based on other values in the matter. As an example, if a process is of type “transaction,” the players in that transaction my not be of the type “plaintiff” or “defendant.” These values are restricted to process of type “dispute.”

Dependencies are not part of Draft LMSS 1.0 Rev 2.

> Dependencies have not been implemented in the online version this Draft

3.4 LMSS Encodings

The LMSS can be encoded machine readable formats. JSON is the baseline supported format.

Machine Readable JSON

```
{
   "document" : {
      "header" : {
         "lmss version" : "1.02"  //Draft LMSS 1.0 Rev 2
         ,"lmss type" : "INST"  //Instance
         ,"language" : "en-us"
         ,"charset" : "UTF-8"
      }
      ,"matter" : {
         "title" : "Wage and hour class action against XYZ Corp."
         ,"locale" : "NAM-US-US+NY"  //New York
         ,"process" : {
            "title" : "Wage and hour class action against XYZ Corp."
            ,"process type" : "D-CCI"  //Court Proceeding, Civil
            ,"area of law" : "LEMP-WGHR"  //Wage and Hour Law
```

```
            ,"player" : [{
               "name" : "Jane Smith and co-workers"
               ,"player role" : "PLTF"  //Plaintiff
               ,"legal entity" : "GROUP-CLASS"  //Class of Plaintiffs
            }
            ,{
               "name" : "XYZ Corporation, Inc."
               ,"player role" : "DEFT"  //Defendant
               ,"legal entity" : "ENTITY-CORP"  //Corporation
               ,"industry" : "TEC"  //Technology
            }]
         }
      }
   }
}
```

3.5 Extensions to the LMSS Codes

The LMSS is intended to be adaptable to the needs of customers. It is our intent that later versions of the standard will incorporate extensions that have been developed. The LMSS has a well-defined way to extend enumerated codes.

Codes are extended by:

- Defining an extension prefix of the format "@<alphanumeric(6)>". This prefix followed by a colon (":") will be used as a prefix to any extended codes.
- Creating and extension code table that conforms to the specification in section 3.2.1.
- Embedding the extension table or a reference to it in the header.

3.5.1 Extension Example

Law firm AM would like to extend the SALI industry codes to include subcategories of real estate. In this example the "Real Estate" industry type with code "RES" will be extended by adding subtypes of real estate below it.

The prefix code chosen is "@am". This prefix is chosen arbitrarily, but in planned versions of the API, extension keys and be registered and stored to facilitate data exchange.)

The extension table would have this the following data.

Code Set	Code	Parent Code	Full Code	Name
SALI Industries	MULFAM	RES	RES-@am:MULFAM	Multi-Family Residential
SALI Industries	OFFICE	RES	RES-@am:OFFICE	Office
SALI Industries	INDUST	RES	RES-@am:INDUST	Industrial
SALI Industries	RESIDL	RES	RES-@am:RESIDL	Residential
SALI Industries	MIXUSE	RES	RES-@am:MIXUSE	Mixed-Use
SALI Industries	RETAIL	RES	RES-@am:RETAIL	Retail
SALI Industries	DATCTR	RES	RES-@am:DATCTR	Data Center

The following show and example of how the code would appear in an LMSS Structure

```
"player": [{
        "name": "ZZZ Corporation, Inc",
        "legal-entity": "ENTITY-CORP",
        "industry": "RES-@AM:OFFICE"   //Real Estate - Office
   }]
```

4 Structure of LMSS Document

4.1 Overview

The LMSS document is comprised of a Header and one or more Matters. The Header and the Matter(s) are containers Each container has elements that may be of type Text, Numeric, Enumeration or Container. Enumerations are stored in Text(250) fields to accommodate multiple levels of specificity that are represented as concatenated codes. Containers are pointers to other structures.

A well-formed Document is a structure that includes all required fields. (Note that the concept of a well-formed document applies to instance but not templates or queries.) In the tables below, “REQ” indicates whether the element is required (in an LMSS instance but not a query), “MULT” indicates whether multiple values are permitted. A well-formed document is required in the interchange of instances of matters between systems. A well-formed document is not required when the LMSS is used in a query. In those instances, a missing field matches all instances. See LMSS Queries below.

4.2 Document

The Document container is the top-level container in the LMSS instance. The Document must have a Header and one or more Matters.

Document container elements

ELEMENT	REQ	MULT.	TYPE	COMMENTS
Header	Y	N	Container	The header information for the document
Matter	Y	Y	Container	The matters included in the document.

4.3 Document Header

The LMSS document header is describes the version, type, default language and character set needed to for correctly read the LMSS document.

ELEMENT	REQ	MULT.	TYPE	COMMENTS
Title	N	N	Text(250)	An optional title for the document.
Version	Y	N	Float	The version of the LMSS standard being encoded
Type	Y	N	Enumeration: SALI LMSS Type	The type of the document. Supported types include "Instance", "Template", and "Query".

				The default value is "Instance"
Language	N	N	Enumeration: IETF BCP 47	Languages follow the Internet Engineering Task Force recommendation in BCP 47
Charset	N	N	Text	Default value is UTF-8
Extension Link	N	Y	Text: URI	A link to an extension file
Extension	N	Y	Container	One or more extension definitions.
Declaration	N	Y	Container	Declarations are indexes of names that need to be cross referenced in a specification. NameIDs can be used in place of names wherever names may appear.

4.3.1 Title

An optional title for the document.

4.3.2 Version

The LMSS version is defined and maintained by SALI. Published version of the standard may be found at Sali.org

4.3.3 Type

Type refers to the type of the Document. Permissible values include:

- Instance – An instance encodes a specific matter. Instances have required containers and elements to be well formed.
- Template – A template encodes default settings of an instance. Parts of templates that are not specified are assumed to accept all values. Templates must adhere to the LMSS container structure, however required elements are relaxed.
- Query – A query encodes a database query following a SQL-like structure. Queries have select, where, order by and limit structures.

4.3.4 Language

Language declares the default language of the LMSS structure. Languages follow the Internet Engineering Task Force recommendation in BCP 47.

4.3.5 Charset

The charset describes the encoding of the characters in the LMSS document. The default encoding is UTF-8.

4.3.6 Extension Link

A link to an extension file that conforms to the LMSS Extension file format.

4.4 Extension

An extension container is an in-document definition of an extension. A document may include an array of these in the header. The scope applies to the document.

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Code Set	Y	N	Text(40)	The code of the code set being extended
Code	Y	N	Text(40)	The extension code. Must be unique in the namespace.
Parent	Y	N	Text(40)	The parent code. This code must be defined before this definition.
Name	Y	N	Text(250)	The short name of the extended code

A sample in-document extension is shown below

```
{
   "document" : {
      "header" : {
         "lmss version" : "1.02"  //Draft LMSS 1.0 Rev 2
         ,"lmss type" : "INST"  //Instance
         ,"language" : "en-us"
         ,"charset" : "UTF-8"
         ,"extension" : [{
            "code set" : "SALI-IND"
            ,"code" : "@MULFAM"
            ,"parent" : "RES"
            ,"name" : "Multi-Family Residential"
         }
         ,{
            "code set" : "SALI-IND"
```

```
            ,"code" : "@OFFICE"
            ,"parent" : "RES"
            ,"name" : "Office"
         }
         ,{
            "code set" : "SALI-IND"
            ,"code" : "@INDUST"
            ,"parent" : "RES"
            ,"name" : "Industrial"
         }
         ,{
            "code set" : "SALI-IND"
            ,"code" : "@RESIDL"
            ,"parent" : "RES"
            ,"name" : "Residential"
         }]
      }
   }
}
```

4.4.1 Code Set

The LMSS Code of the code set. (See 6.1)

4.4.2 Code

The code assigned to the extension. The code will have a "@" prepended to it when used.

4.4.3 Parent

The parent code. The parent code must be an existing standard code in the code set, or a previously defined extension. Examples include "RES" or "@OFFICE".

4.4.4 Name

The short name of extension code.

4.5 Declaration

A declaration may be used to assign and index to an item to ensure accurate cross referencing. Declarations are typically used for legal entities but can be used in place of a Name (See 4.10.1, 4.11.1 and 4.11.2).

A legal person or entity is any human or non-human entity, in other words, any human being, firm, or government agency that is recognized as having privileges and obligations, such as having the ability to enter into contracts, to sue, and to be sued.

Entities are used to maintain referential integrity across LMSS structure. For example, if the matter includes two processes that refer to the same legal entity as in the example, "Review and negotiate project labor agreement for Jane Smith." The matter may be encoded to have two processes – one for review and another of negotiation. But the review and negotiation are the same legal entity. An entity declaration should be used in this instance and the entity NameID should be used in each process to ensure that it is understood that the objects of both process are the same.

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
NameID	Y	N	Text(40)	The NameID of the entity. Must be of the regular expression form \^[A-Za-z0-9_-]{1,39}
Name	Y	N	Text(250)	Name to be inserted wherever the NameID appears

4.5.1 NameID

The ID to be used in place of a name. NameIDs must begin with a caret symbol. (e.g. ^102 or ^EntityXYZ).

4.5.2 Name

The name to be used in place of the NameID.

4.6 Matter

The Matter container encapsulates the information for a matter. There may be more than one matter per document. The matter must have a title, a human readable description, a locale and one or more processes. The locale may restrict the types of the processes. For example, certain bankruptcy processes are limited to the US because those processes a defined by US Bankruptcy law.

Matter container elements

ELEMENT	REQ	MULT.	TYPE	COMMENTS
Title	Y	N	Text(250)	The title of the matter.
Locale	Y	N	Enumeration: ISO 3166-2	The location of the matter.
Process	Y	Y	Container	The process or processes for this matter.
Narrative	Y	N	Container	The description of the matter.

4.6.1 Title

A required title for the process.

4.6.2 Locale

A required locale for the matter. The locale is an enumeration.

4.7 Narrative

A narrative holds a group of related matter descriptions. It includes a required type which is an enumerated list, an optional usage tag, and an optional source. Each narrative can have a specified usage so a Matter can have multiple narratives based on the audience. each narrative is intended to capture a logically unique matter interpretation of the matter. Formatting and language variants are accommodated in the description container.

Narrative container elements

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Type	Y	N	Enumeration: SALI Narrative Type	The type of the narrative.
Usage	N	N	Text(250)	Descriptions of the intended audience or other usage of the narrative
Description	Y	Y	Container	One or more descriptions of a narrative.
Source	N	Y	Text(250)	The source of the narrative

4.7.1 Type

The narrative type is an enumerated list intended to capture the sensitivity of the narrative. Values include: public, confidential, private, generic, etc.

4.7.2 Usage

Usage is an optional human readable field that should capture the audience or usage of the narrative. Examples for the same matter might be: "Narrative written for pitches to lenders," "Narrative written for pitches to borrowers," and "Narrative written for generic finance pitches."

4.7.3 Source

The source is an optional element that describe the source of the narrative. You may provide multiple sources. Examples include: "2015 litigation department compensation memo," and "2017 environmental practice Chambers submission."

4.8 Description

The description is an container encapsulates the specific text of a specific narrative. The description has text, a format and a language. If the language is not specified, it is inherited from the matter. For example, if a firm keeps French and English versions of the same narrative, or plain text and HTML formatted versions of the same narrative, these would be accommodated in the description container.

ELEMENT	REQ.	MULT.	TYPE	COMMENTS

Text	Y	N	Text(4000)	The description of the matter.
Format	Y	N	Enumeration of Narrative Formats	Text or HTML
Language	N	N	Enumeration: IETF BCP 47	The language of the description. If omitted, it is assumed to be the same as that of the matter.

4.8.1 Text

The text holds the characters of the description. The format of the text is interpreted based on the description type.

4.8.2 Format

Two formats are supported: Text and HTML.

4.8.3 Language

Language is optionally specified by using BCP 47.

4.9 Process

The Process container describes the process, service or product being delivered.

Every process must have a single process type. Top-level process types are transactions, disputes, regulatory proceedings, bankruptcy/restructurings, and advisory. If we think of the process as a sentence, the process type reprints the “verb.” They describe the action that is being taken.

The players are the legal entities involved in the process. These should be thought of the subject (Joe Smith) and object (Company X) in the sentence “Joe Smith sued Company X for $500,000 for breach of contract.”

The process predicate contains the predecessors or outcomes of the process. In the sentence above, “for $500,000” and “for breach of contract.”

The Area of Law provides context to the process. The area of law should be thought of the primary subject of law for the process — think the class that’s the attorney was in when she learned about the applicable law. The area of law is provided primarily for context as in the examples, “He prosecuted the defendant” and “She prosecuted the patent.” The first is criminal law, the second is intellectual property law.

Process container elements

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Title	N	N	Text(100)	The title of the process.
Description	N	N	Text(4000)	An optional description of the process.
Process Type	Y	N	Enumeration: SALI Process Type	The type of the process. The process should be thought of a verb.
Area of Law	Y	Y	Enumeration: SALI Area of Law	The primary area of law for the process. Area of Law provides context for interpreting other elements of the process.
Player	Y	Y	Container	The players involved in a process.
Process Object	N	Y	Container	The process object hold information specific to each kind of process. See the discussion below.

4.9.1 Title

The optional title of the process. Examples include: "LLC formation in California", and "License of technology to French company.".

4.9.2 Description

An optional field to store additional information about the process.

4.9.3 Process Type

A SALI enumerated value. Top-level process types are transactions, disputes, regulatory proceedings, bankruptcy/restructurings, and advisory. If we think of the process as a sentence, the process type reprints the “verb.” They describe the action that is being taken.

4.9.4 Area of Law

The Area of Law is a SALI enumerated value that provides context to the process. The area of law should be thought of the primary subject of law for the process — think the class that’s the attorney was in when she learned about the applicable law. The area of law is provided primarily for context as in the examples, “He prosecuted the defendant”

and “She prosecuted the patent.” The first is criminal law, the second is intellectual property law. There can be multiple areas of law.

4.10 Player

The players are the subject of the sentence and sometimes the object. Primary players are the primary parties involved in a legal process. The Player container has the following fields:

Player container elements

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Name	Y*	N	Text(250)	The name of the player. This is usually a company name or individual name. May use a Name or a declared ID.
Player Role	N	Y	Enumeration: SALI Player Role	The contextual role that the player has in the process. (e.g. Plaintiff, Licensee, etc.)
Industry	N	N	Enumeration: SALI Industry	The industry that the player is in. This often provides context for how the process is executed.
Legal Entity	Y	N	Enumeration: SALI Legal Entity	The type of legal entity that the player is. A corporation, a partnership, a person, etc.
Governmental Authority	N	N	Enumeration: SALI Governmental Authority	Used when the player is a governmental authority
Counsel	N	Y	Container	The legal representatives of the player

*Either a Name or a NameID is required.

4.10.1 Name

The name is the name of player in human readable form. If a NameID is used, an Name is not required.

4.10.2 Player Role

The Player Role is an optional element that provides context. Examples of roles are Plaintiff, Defendant, Acquiror, etc. .The Roles span both legal roles and functional roles. Play is an enumerated value.

4.10.3 Industry

The industry is an optional field that describes the industry of the player. Industry is an enumerated value.

4.10.4 Legal Entity

Legal entity describes the type of the player. In law, a "legal person" or "legal entity" is any human or non-human entity, in other words, any human being, firm, or government agency that is recognized as having privileges and obligations, such as having the ability to enter into contracts, to sue, and to be sued. Legal Entity is an enumerated value.

4.10.5 Governmental Authority

The enumerated identification of the governmental authority if the player is a governmental authority.

4.11 Counsel

The counsel container includes all of the legal representatives of the player. The fields of the counsel container are described below:

Counsel container elements

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Name	Y*	N	Text(250)	The name of the legal representative. May use a Name or a declared ID.
Firm Name	N*	N	Text(250)	The firms of the legal representative. May use a Name or a declared ID.
Representation Role	Y	N	Enumeration	The role of the legal representative

*Either a Name or a NameID is required.

4.11.1 Name

The name is the name of counsel in human readable form. If a NameID is used, an Name is not required. (NameIDs are indicated by prefixing them with a caret "^" symbol. e.g. "^103".)

4.11.2 Firm Name

The firm name is the name of firm that the counsel is part of in human readable form. If a NameID is used, an Firm Name is not required.

4.11.3 Representation Role

The role of the legal representative. Role is an enumerated value.

4.12 Process Object

The process object encapsulates important elements of the process. A process can have multiple process object. The process object can simply be a summary of key attributes of the overall process. Additional process objects can be attached to describe the preconditions and post conditions of a process. For example, for a merger, there optionally can be individual process objects describing the predecessor entities and resulting entity.

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Description	N	N	Text(4000)	The description the Process Object
Status	N	N	Enumeration	An enumerated status: Open, closed, Canceled
Cross-Border	N	N	Boolean	Is the process cross border?
Filing Date	N	N	Date	The filing date, if appropriate
Term Sheet Date	N	N	Date	The term sheet date, if appropriate
Effective Date	N	N	Date	The effective date of the process
Closing Date	N	N	Date	Closing date, if appropriate
Announce date	N	N	Date	The date process was announced, if appropriate
Asset Location	N	N	Text(4000)	Location of the transaction - esp. for real property
Asset Description	N	N	Text(4000)	Description of the asset such as asset type, asset size, etc.
Monetary Value	N	N	Container	The monetary value of the transaction, dispute or object

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Non-Monetary Value	N	N	Text(400)	The non-monetary value of the transaction, dispute or object.
Transaction: Consideration	N	Y	Text(4000)	Description of the consideration for the deal - Stock, Cash, Real Estate, etc.
Transaction: Deal Type	N	Y	Text(4000)	Type of deal.
Transaction: Location	N	Y	Text(4000)	Location of the transaction.
Transaction: Legal Entity	N	Y	Enumerated Value: Legal Entity	If a formation or dissolution, the kind of legal entity formed or destroyed.
Regulatory: Authority	N	Y	Enumerated Value: Governmental Authority	Regulator. A public authority or government agency responsible for exercising autonomous authority over some area of human activity in a regulatory or supervisory capacity.
Regulatory: Authority Other	N	Y	Text(250)	Regulatory authority not included in the enumeration values
Dispute: Venue	N	Y	Enumerated Value: Court	The venue of the dispute.
Dispute: Venue Other	N	Y	Text(250)	Dispute venue not fully covered by the enumerated values.
Dispute: Trial Type	N	N	Enumerated Value: Trial Type	The type of the trial.
Dispute: Case Name	N	N	Text(250)	Name of the case.
Dispute: Resolution	N	N	Text(250)	Enumeration TBD
Dispute: Resolution Date	N	N	Date	Date of the dispute resolution
Dispute: Duration (Months)	N	N	Integer	Duration of the dispute in months
Dispute: Multi-Jurisdictional	N	N	Boolean	Set if the dispute is multijurisdictional

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Dispute: Number of Depositions	N	N	Integer	Number of depositions
Dispute: Number of Experts	N	N	Integer	Number of experts

See the table above for descriptions.

4.13 Monetary Value

Monetary Value encapsulates a financial value attribute. It consists of a floating point number and a currency code.

Monetary value container elements

ELEMENT	REQ.	MULT.	TYPE	COMMENTS
Currency	Y	N	Enumeration: Currency	A currency code a defined by ISO 4217
Value	Y	N	Float	A floating point number that represents the value

5 The Legal Matter Application Programming Interfaces (APIs)

LMSS APIs are not supported in Draft LMSS 1.0 Rev 2.

The LMSS supports several types of APIs:

- LMSS Instance
- LMSS Queries
- LMSS UI/Sych API

5.1 LMSS Instance

The Instance API is used to create, import and export matters. It also supports default values and user-defined extensions.

- Helps create conforming matters
- Validates matter structures
- Supports user-defined extensions

- Supports default values/templates

5.2 LMSS Queries

An important part of the LMSS is to support the querying of data stores against different criteria. The standard leverages the LMSS definition — structure, enumerated values, text and numeric values — to specify search criteria. In these instances sparse version of the LMSS are applied to develop search criteria.

5.2.1 LMSS Query WHERE Clauses

In the standard SQL queries clauses SELECT, WHERE, and ORDER BY, the LMSS template can be used to define the WHERE clauses.

For enumerated values that are filled in, the search uses the “starts with” criteria for specifying a search. For text and numeric fields, you can apply standard SQL wildcard matching.

Below are several examples of how this applied.

Using LMSS Templates in Queries

QUERY	STRUCTURE

Select all matters WHERE lease transactions are in New York State	`{`
	`  "document" : {`
	`  "header" : {`
	`  "lmss version" : "1.02"  //Draft LMSS 1.0 Rev 2`
	`  ,"lmss type" : "QRY"  //Query`
	`  ,"language" : "en-us"`
	`  ,"charset" : "UTF-8"`
	`  }`
	`  ,"matter" : {`
	`  "locale" : "NAM-US-US+NY"  //New York`
	`  ,"process" : {`
	`  "process type" : "T-LEA"  //Lease`
	`  }`
	`  }`
	`  }`
	`}`

Select all matters WHERE McEvoy & Edwards represented the buyer in real estate transactions	{
	"document" : {
	"header" : {
	"lmss version" : "1.02" //Draft LMSS 1.0 Rev 2
	,"lmss type" : "QRY" //Query
	,"language" : "en-us"
	,"charset" : "UTF-8"
	}
	,"matter" : {
	"process" : {
	"process type" : "T" //Transaction
	,"area of law" : "REAL" //Real Property Law
	,"player" : {
	"player role" : "BUYR" //Buyer
	,"counsel" : {
	"firm name" : "McEvoy & Edwards%"
	,"representation role" : "COUN" //Counsel/Attorney
	}
	}
	}
	}
	}
	}

Select all civil court matters where Wells Fargo was the defendant in an employment class action.	{
	"document" : {
	"header" : {
	"lmss version" : "1.02" //Draft LMSS 1.0 Rev 2
	,"lmss type" : "QRY" //Query
	,"language" : "en-us"
	,"charset" : "UTF-8"
	}
	,"matter" : {
	"process" : {
	"area of law" : "LEMP" //Labor and Employment Law
	,"player" : [{
	"name" : "Wells Fargo%"
	,"player role" : "DEFT" //Defendant
	}
	,{
	"player role" : "PLTF" //Plaintiff
	,"legal entity" : "GROUP-CLASS" //Class of Plaintiffs
	}]
	}
	}
	}
	}

Select all matters where there is an enforcement action involving the U.S. National Labor Relations Board	{
	"document" : {
	"header" : {
	"lmss version" : "1.02" //Draft LMSS 1.0 Rev 2
	,"lmss type" : "QRY" //Query
	,"language" : "en-us"
	,"charset" : "UTF-8"
	}
	,"matter" : {
	"process" : {
	"process type" : "R-ENF" //Enforcement
	,"process object" : {
	"regulatory: authority" : "US-FD-NLRB" //National Labor Relations Board
	}
	}
	}
	}
	}

5.2.2 LMSS Query SELECT Statements

We anticipate the following select specifications as part of the API.

> Not implemented in Draft LMSS 1.0 Rev 2

SELECT TYPE	RETURN TYPES	COMMENTS
Aggregate values	Returns single values for aggregate functions such as: COUNT, COUNT DISTINCT AVERAGE, SUM	
Scalar values	Return lists of named values	
Partial Structures	Return lists of structures with descriptions and narratives stripped out	
Full Structures	Full lists of full structures	
Comparison Operators	>, >=, <, <=, <>, LIKE, NOT, IN	Need to support wildcard characters

5.3 LMSS UI/Synch API

Not implemented in Draft LMSS 1.0 Rev 2

The LMSS has an API designed to help application providers drive keep the applications updated with the most current versions of the standard. The API supports returning structures code information.

REQUEST	RETURN TYPES	COMMENTS
List Supported Enumerations	Returns list of enumerations.	
Get Enumeration List	Returns key value pairs of a given enumeration to drive dropdown and autocomplete user interface elements. Supports simple filtering by level, by "contains" and by "starts with"	

6 Code Sets

6.1 Code Set Types

The following code sets are used by Draft LMSS 1.0 Rev 2.

Code Set	Code	Description	
SALI Areas of Law	SALI-AOL:2	SALI Area of Law/Practice	View
SALI Court	SALI-COURT	Codes for courts. Currently limited to U.S. courts.	View
SALI Currency (ISO 4217)	ISO-4217	List of world currencies	View
SALI Format	SALI-FMT	The format of a description	View
SALI Governmental Body	SALI-GOVT	Government codes. Currently U.S. federal only	View
SALI Industry	SALI-IND	A default set of codes provided to define industries. Industry codes are not officially part of the SALI Legal Matter standard.	View
SALI Legal Entity	SALI-LEGENT:2	Specifies the kind of legal entity of a party to a legal matter	View

SALI LMSS Type	SALI-LMST	The types of LMSS documents	View
SALI Location	SALI-ISO31662	World locations to the state/province level	View
SALI Matter Narrative	SALI-MATNAR	Codes used to identify the type of narrative.	View
SALI Player Role	SALI-PROLE	The types of player roles that can be applied to a matter.	View
SALI Process	SALI-PROC	Codes used to define a legal service process	View
SALI Process Status	SALI-PROCSTAT	Codes for the status of a process	View
SALI Representation Role	SALI-RROLE	The types of representation roles for a player.	View
SALI Trial Type	SALI-TRITYP	The type of trial format	View