



## LEXS 4.0 Quick Start Guide

---

Publish/Discover  
Search/Retrieve  
Subscribe/Notify  
Domain Exchange

---

September 2011

# 1 Table of Contents

- 1 Purpose ..... 4
- 2 Introduction ..... 4
  - 2.1 NIEM Overview..... 5
  - 2.2 ULEX Overview ..... 5
  - 2.3 LEXS Overview ..... 6
    - 2.3.1 Levels of Understanding ..... 7
    - 2.3.2 LEXS Information Environment..... 7
    - 2.3.3 Related and Supporting Technology Standards..... 8
    - 2.3.4 LEXS Supported Operations ..... 8
- 3 Key LEXS Concepts ..... 9
  - 3.1 Message..... 10
  - 3.2 Package..... 10
  - 3.3 Metadata ..... 10
  - 3.4 Digest..... 11
  - 3.5 Structured Payload ..... 11
  - 3.6 Narrative Text..... 11
  - 3.7 Rendering Instructions ..... 12
  - 3.8 Attachment and Attachment Summary ..... 12
  - 3.9 Associations..... 12
  - 3.10 Connections..... 12
  - 3.11 Roles ..... 13
- 4 When and How To Use LEXS ..... 13
  - 4.1 Use LEXS As-Is ..... 14
    - 4.1.1 Extend LEXS Using NIEM ..... 14
  - 4.2 Reuse Other IEPDs..... 14
  - 4.3 Develop a New NIEM IEPD ..... 15
  - 4.4 Use Other Widely Supported Standard..... 15
- 5 LEXS Publish Operations ..... 15
- 6 LEXS Search and Retrieve Operations..... 16
- 7 LEXS Subscribe and Notify Operations..... 18
- 8 LEXS Domain Exchange Operations ..... 21
- 9 Resources ..... 22
  - 9.1 LEXS Distribution Contents..... 22
    - 9.1.1 Schemas ..... 22
    - 9.1.2 High-Level Domain Model Diagrams ..... 23

---

9.1.3	Association Diagrams.....	23
9.1.4	Sequence Diagrams.....	23
9.1.5	Sample XML Instances .....	23
9.1.6	Change log.....	23
9.2	Additional LEXS Resources .....	23
9.2.1	LEXS Community Web Site.....	23
9.2.2	LEXS User Guide .....	23
9.2.3	Component Mapping Workbook .....	23
9.2.4	Code Tables Spreadsheet.....	24
9.2.5	Language bindings .....	24
9.2.6	Sample implementations.....	24
9.2.7	WSCF.blue4NIEM .....	24
9.2.8	ConTesa .....	24
9.2.9	Schema viewers .....	25
9.2.10	WSDL templates.....	25
9.3	NIEM Resources .....	25
9.3.1	NIEM.gov.....	25
9.3.2	SSGT .....	25
9.3.3	NIEM conformance tool.....	25
9.3.4	NIEM codelist generator .....	25
9.3.5	NIEM IEPD tool.....	26
9.3.6	Implementation Guidance for NIEM-Conformant Exchanges.....	26
9.3.7	IEPD Clearinghouse.....	26
9.4	Related Standards .....	26
10	Acronyms .....	27

## 1 Purpose

This Quick Start Guide provides an overview to the Logical Entity eXchange Specifications (LEXS) developed by the Department of Justice and introduces the LEXS 4.0 operations: Publication and Discovery, Search and Retrieval, Subscription and Notification, Domain Exchange. This guide is intended as a starting point for managers or developers who want to gain an understanding of LEXS and its supported operations.

Some readers may be using LEXS to create a new information exchange specification; other users may be implementing an existing LEXS-based information exchange either as a producer of data or a consumer of data. How LEXS is being used will determine which topics in this guide are of the greatest interest.

## 2 Introduction

Logical Entity eXchange Specifications (LEXS) were created to support the primary objectives of the Department of Justice (DOJ) Law Enforcement Information Sharing Program (LEISP). LEXS aims to minimize the impact of the changing requirements and the varied demands of information sharing on the sources and consumers of data. Although LEXS originated from law-enforcement needs, it has been designed for a broad audience and is not limited to use by the law-enforcement community.

LEXS addresses two aspects of information sharing:

- Defines and consistently describes units of information to be shared.
- Defines interfaces and protocols for information distribution:
  - Defines how data producers provide information and allow it to be shared.
  - Defines how data consumers search and retrieve the information.
  - Defines how data consumers subscribe to information and receive notification that the data of interest has been changed, searched, or accessed by someone else.

LEXS provides an extensible framework for consistent packaging of the information, with specific placement and markings for various elements of the shared information. The LEXS specifications shield both data sources and data recipients from the complexity of multiple interfaces and allow for the multipurpose use of information. A data item created by a source can be consumed by multiple recipients who can understand as much, or as little, of the data as necessary.

From a data perspective, LEXS is based on the National Information Exchange Model (NIEM), while from a message perspective it is built on the Universal Lexical Exchange (ULEX) 2.0 framework. LEXS employs the structures, standards, and usage guidelines of NIEM, including NIEM Naming and Design Rules (NDR). It is built in the style of a NIEM Information Exchange Package Documentation (IEPD) although it is not an IEPD itself. An IEPD is a collection of XML schemas and other documentation that represents a specific information exchange. In contrast, LEXS provides a framework that can be used by a number of different communities to create IEPDs that document a number of different exchanges.

## ***2.1 NIEM Overview***

The National Information Exchange Model (NIEM) is a multi-agency initiative to provide the foundation and building blocks for national-level information sharing and data exchange between all levels of government (federal, state, tribal, and local), and with private industry. NIEM standards enable different information systems to share and exchange information regardless of the particular technologies in use.

NIEM can be thought of as a data model and a reference vocabulary from which XML schema-based data components are constructed. These components (which are XML data elements) serve as the basis for information exchanges. In NIEM and in this document, the term “element” refers to a unit of information which may be simple (indivisible) or complex (consist of other elements). In conjunction with the concepts and rules that underlie the NIEM structure, maintain its consistency and govern its use, these NIEM data components can be reused by information practitioners to create an Information Exchange Package Documentation (IEPD). An IEPD is a collection of XML schemas, XML instances, and other documentation and artifacts (e.g., diagrams, spreadsheets) that is the electronic representation of the rules governing an information exchange.

## ***2.2 ULEX Overview***

Universal Lexical Exchange (ULEX) is a family of schemas which define the organizational structures and major components for a message exchange. ULEX defines a set of common, high-level structures for information sharing that can be leveraged by programs to develop standard information sharing schemas for use by their constituent communities or agencies. By defining such high-level structures, ULEX saves communities from each having to design their own unique (and almost certainly incompatible) structures. All ULEX-based standards augment the organizational elements provided by ULEX with well-defined, community-specific structures that allow a common understanding of the data to be shared within the community. However, the utilization of the ULEX framework ensures a level of compatibility between ULEX

implementations due to the common structure and the ability to discover and render documents based on “foreign” implementations.

The ULEX framework includes both abstract and concrete data elements. The LEXS schemas provide implementation of the NIEM-based content for the framework abstracts. Elements that are high level in nature and which are used in all implementations, such as a message time stamp, are included as concrete elements in the ULEX framework schema. Elements that are more oriented to an underlying data model, such as person information, are included as abstract elements in the ULEX framework schema.

ULEX utilizes a generic paradigm for information sharing called the **Data Item**. A Data Item is whatever the source considers a logical unit of information. For example, to an incident based reporting system, a logical unit of information is an incident report that may contain activities, people, places, and things. To a logistics system, a logical unit of information may be details about a single piece of hardware, along with its maintenance history and location. The Data Item concept provides a single, generic container that can be used to encapsulate different types of data needed by various communities. The Data Item can be thought of as a collection of structured entities, attributes of these entities, relationships between these entities and unstructured textual information.

### **2.3 LEXS Overview**

LEXS addresses two major issues found in NIEM-based exchanges. First, LEXS utilizes ULEX to define a message framework, which is something that NIEM does not address since it is exclusively data focused. Second, LEXS defines high-level data objects, known as Business Information Exchange Components (BIEC) in NIEM, that can be used as is and that are common to many NIEM-based exchanges. By utilizing LEXS as the foundation of a NIEM IEPD, programs save time and effort that would normally be required to design their own message frameworks and data contents, and also gain immediate compatibility with other LEXS-based exchanges.

LEXS builds on organizational elements of ULEX and defines a set of high level, commonly understood, structured **base objects** which represent real-world objects such as person, location, or vehicle. For each base object, LEXS specifies NIEM content which provides a wide range of data representation capabilities. LEXS includes a large number of NIEM roles and associations and defines additional roles and associations needed by the LEXS community to provide detailed context for all data. These base objects, roles, and associations provide a great deal of flexibility to communities so each can define their own data items. LEXS groups these base objects and their applicable roles into **entities**. For example, a LEXS person entity includes a person base object and roles such as subject, witness and victim. In this document, the term **object** is used to refer to a base object or a role.

### 2.3.1 Levels of Understanding

ULEX includes a layered mechanism for communities to define entities, roles, associations, structures and elements that are not defined in the Digest. Communities that need additional data not provided in the Digest can supply that data in one or more Structured Payloads. This allows groups or projects who are interested in information sharing to leverage a ULEX-based standard as a base while developing their own specialized schemas targeted to address their own business missions.

This layering concept provides multiple levels of understanding by allowing communities to extend the LEXS Digest with community-specific Structured Payloads. As Figure 1 shows, a community can use as little or as much of the underlying payload and base level as it needs. For example, an incident community needs to exchange information beyond what is available in the Digest and, thus, creates its own Structured Payload schema that builds upon what is in the Digest. Bomb and arson, and cyber communities review the Digest contents along with the incident Structured Payload schema and determine that what the incident community has provided doesn't include everything they need, but that it supplies a good foundation. Rather than creating their own Structured Payload that just builds upon the Digest, each develops an individual Structured Payload schema that builds upon what is in both the incident Structured Payload and Digest. There is no limit to how many layers can exist.

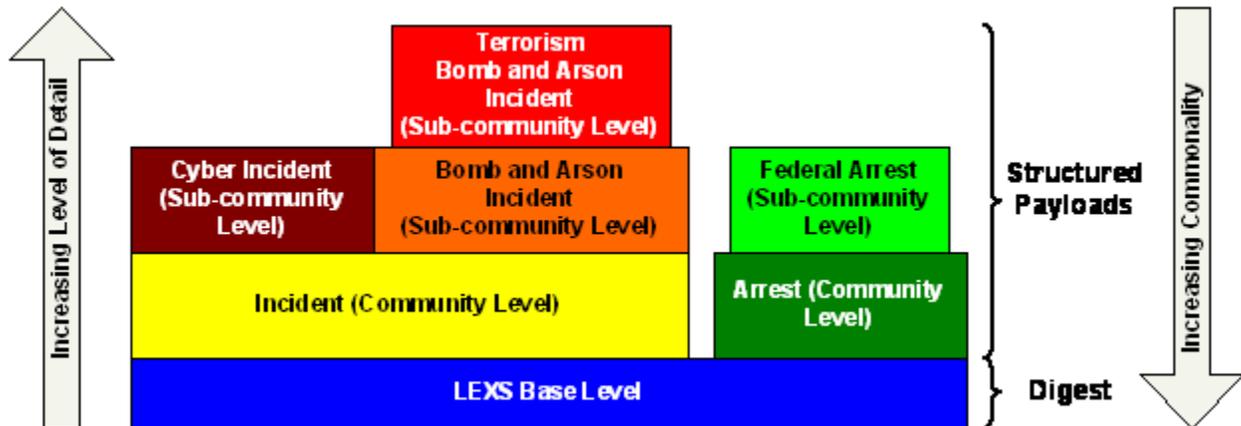


Figure 1. Structured Payload Layers

Each community determines what it does or does not understand or can use from another community's payload. New payloads typically contain greater, more specialized, detail which puts it higher in the stack.

### 2.3.2 LEXS Information Environment

When two organizations mutually allow for two-way sharing of information, with each party retaining ownership and possession of its information they are partners in information sharing

and each of the systems is referred to as a **partner system**. The term **data source** is used to refer to a system that is operated by an organization (e.g., DOJ investigative component) that publishes information to a data repository. The data repository that receives and ingests the published information is referred to as a **data consumer** (e.g., N-DEx). The data source publishes periodically as determined by operations guidelines. A **service provider** is the system or application that provides access to its data in-place (e.g., LInX or ARJIS).

The terms data source, partner system, data consumer, and service provider do not define system types, but describe roles that systems can have. The same system can have more than one role; hence one or more of these terms may apply to the same system. Figure 2 shows how these terms can be applied to some of the participants in the N-DEx environment.

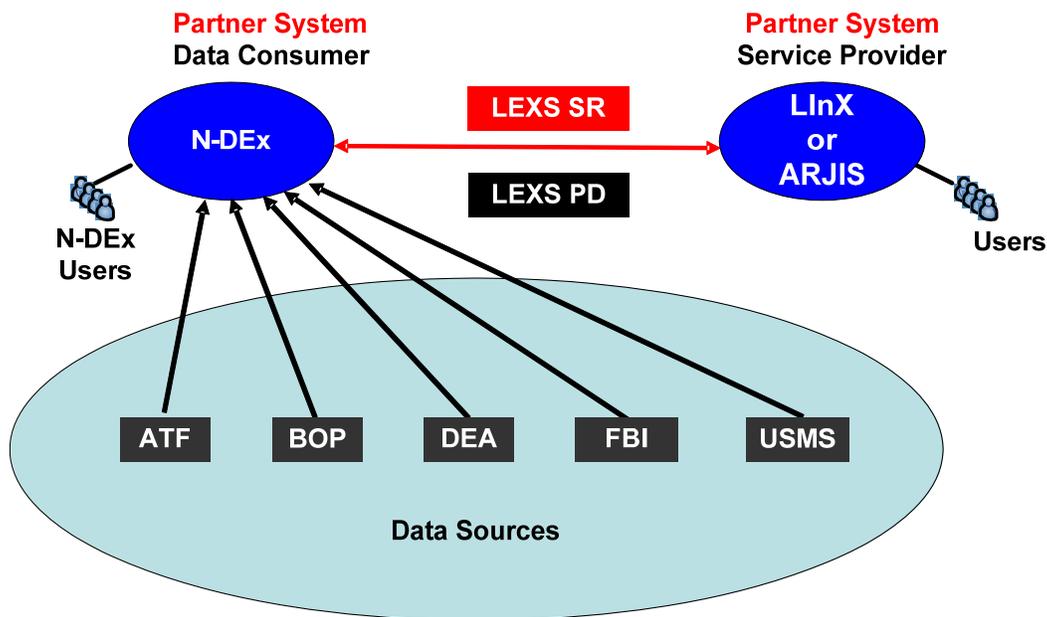


Figure 2. LEXS SR and LEXS PD in N-DEx

### 2.3.3 Related and Supporting Technology Standards

While discussion is beyond the scope of this document, several other standards are leveraged by LEXS. These standards include Security Assertion Markup Language (SAML), WS-Addressing, WS-BaseFaults, WS-BaseNotification, and WS-Topics.

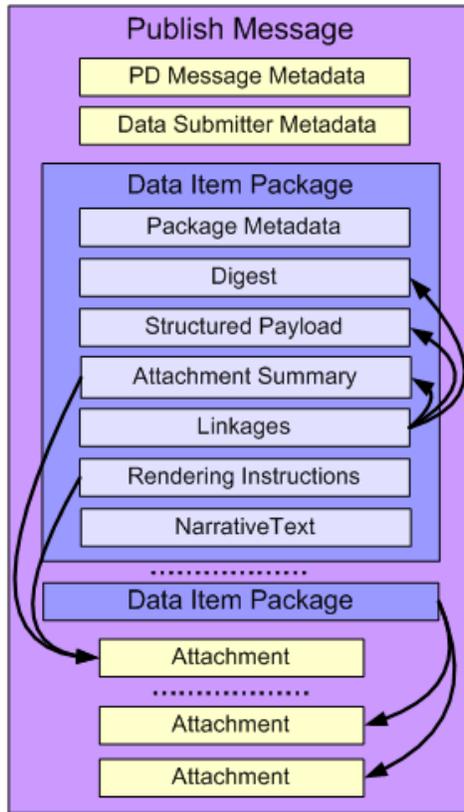
### 2.3.4 LEXS Supported Operations

Four basic categories of operations are defined by separate LEXS schemas, which in turn are supported by shared schemas that provide common definitions of real-world objects and structures:

- LEXS Publication and Discovery (LEXS PD) applies to publishing and updating data from a source to a consumer.
- LEXS Search and Retrieval (LEXS SR) applies to federated query and retrieval of data among partner systems. It is possible for a given system to be both a data source and a partner system.
- LEXS Subscription and Notification (LEXS SN) is for handling subscriptions and notifications that data of interest has been added, changed, or accessed; that someone has performed the same or similar search; or that some situation has occurred such as an Amber Alert.
- LEXS Domain Exchange (LEXS DE) provides a flexible mechanism for exchanging information that is useful in certain specific situations that leverages LEXS constructs but that does not warrant a formal specification within LEXS.

### **3 Key LEXS Concepts**

There are a number of key concepts that underlie the design of LEXS. To understand how to use LEXS it is essential to understand these fundamental building blocks. The message shown in Figure 3 is one of a number of LEXS messages and is used here to illustrate some of these key concepts.



**Figure 3. LEYS Message**

### **3.1 Message**

The “message” is the basic structure that wraps ULEX requests and responses. In reality, ULEX defines a number of different message structures for different purposes, but all message structures contain metadata, and usually additional information specific to the request or response they support.

### **3.2 Package**

“Package” refers to a standard representation of any data item used for information sharing. A package represents a unit of information that is self-contained and able to convey the knowledge from the data source to the data consumer or between partner systems.

### **3.3 Metadata**

“Metadata,” which is information about the data being shared, exists at several levels and is organized into different structures based on content and usage. For example, the PD Message Metadata is specific to publish and discover messages, but Package Metadata is used in all messages that contain data item packages.

### **3.4 Digest**

The “Digest” is the common denominator for systems to use to handle data without having to understand the specific context and meaning of the source. It contains the most common characteristics of real-world objects that can be supported by any data source or data consumer. Digest-level data objects may be further augmented or described with additional details in the Structured Payload or the unstructured Narrative Text portion of the package.

The LEXS Digest defines a high level set of LEXS entities, where each contains a basic set of NIEM elements and, in some cases, extensions.

Activity	Firearm	Program
Aircraft	Hash	Prosecution
Arrest	Incident	Sentence
Booking	Instant Messenger	Service Call
Case	Intangible Item	Substance
Citation	Location	Supervision
Court Activity	Network Address	Tangible Item
Credit Card	Notification	Telephone Number
Document	Offense	Vehicle
Drug	Organization	Vessel
Email	Other	Warrant
Explosive	Person	

Elements, objects, or even entire entities can be ignored if not implemented by data consumers.

### **3.5 Structured Payload**

The “Structured Payload” provides an extension mechanism for different user communities to extend the Digest to incorporate richer data sets without compromising compatibility across applications. Each Structured Payload is based on schemas created by communities outside of LEXS and may also define a new entity where the LEXS Digest does not provide a foundation. Structured payloads can be ignored if not recognized, understood, or implemented by data consumers.

### **3.6 Narrative Text**

The “Narrative Text” structure provides a container for unstructured data associated with the data item. The ULEX provision for unstructured data allows for the inclusion of data sources that are text-based or elements from any data source that are free-form in nature and cannot easily be represented as attribute-value data components.

### **3.7 Rendering Instructions**

“Rendering Instructions” are used to display the information in a package in a specific viewing or output format for human users. Rendering instructions come in two general forms:

- instructions (i.e., XSL) for converting the XML data in the package into a suitable display format
- a pre-rendered version (frequently in HTML) of the data item (e.g., an attached image or document)

### **3.8 Attachment and Attachment Summary**

An “Attachment” is a container that typically consists of binary data (e.g., a mug shot or fingerprint) that is not directly readable by a human, or stylesheets used by Rendering Instructions. Attachments are located at the message level rather than the package level since each may be part of multiple packages (e.g., a message with multiple incident reports that include the same person and his mug shot).

An “Attachment Summary” provides a mechanism to connect an element in a package, such as a person, to an Attachment, such as a mug shot.

### **3.9 Associations**

An “Association” represents a specific relationship between objects and is used when a simple NIEM or LEXS element is not sufficient to specify the relationship clearly and when there are properties of the relationship that are not attributable to the objects being related (e.g., the Marriage association to indicate a marriage between person objects). Most associations in LEXS are imported from NIEM, but the LEXS Digest defines a number of LEXS-specific associations. Structured Payloads can define additional associations if those provided in the Digest are not sufficient.

### **3.10 Connections**

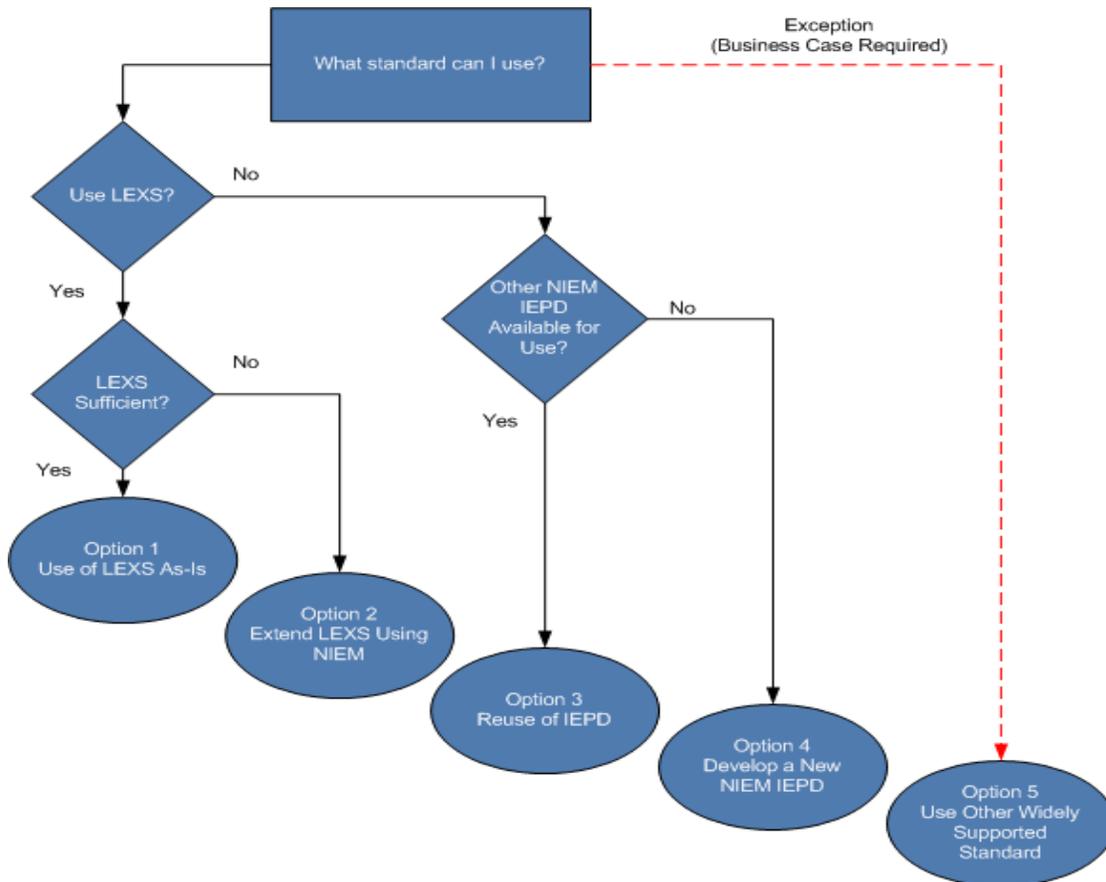
A “Connection” denotes a structural relationship between objects in different sections of a message. For example, the "same as" connection provides a means of identifying that an object in the Digest and an object in a Structured Payload describe the same real-world object. For instance, there may be a person object in the Digest and a person object in a Structured Payload that both describe the same person. Objects in two structured payloads can be similarly connected. There are also a number of connections to link an object to a message attachment. These attachment connections link a Digest or Structured Payload object with an attachment summary, which in turn is linked to a specific attachment.

### 3.11 Roles

“Role” is a particular context or activity for a base object. The role object represents data specific to the role and contains a reference to the base object. For example, weapon is a role of an item. A weapon object has a reference to the item that is the weapon (the base object) plus additional information specific to weapons, such as how the weapon was used.

## 4 When and How To Use LEXS

Figure 4 shows how an organization may make a decision whether to use LEXS to implement a NIEM-compliant exchange when the need for such an exchange is identified. There are five options to implement information exchanges between systems when new systems are developed or existing systems are enhanced. A system may require several IEPDs to meet its business requirements, and thus may need to leverage more than one option described in the decision flow.



**Figure 4. IEPD development decision tree**

The decision flow begins with the validation that a NIEM-conformant exchange is required. Unless there is a compelling business reason not to use NIEM, organizations should consider

implementing NIEM-conformant information exchanges between systems when new systems are being developed or existing systems are enhanced.

The decision flow then continues to help decision makers target the most suitable option to enable organizations to achieve a high level of interoperability in the implementation of NIEM-conformant information exchanges. More detailed descriptions of the five options are provided in the following sections.

#### **4.1 Use LEXS As-Is**

LEXS defines a standard set of high level entities, roles, associations, and other capabilities which are NIEM-conformant and are commonly used across multiple organizations. Developers should try and locate as many of their data requirements within the Digest since the Digest portion of the IEPD is consistent across all LEXS-based IEPDs.

The primary reasons for using LEXS As-Is:

- During initial development there is uncertainty regarding how the data may be utilized in the future. LEXS allows the data to be easily reused.
- Data can be made readily available for discovery.
- Attachments are available for structured and unstructured data.

#### **4.1 Extend LEXS Using NIEM**

LEXS provides an extension mechanism to easily include additional structures within base LEXS. This is accomplished by specifying one or more Structured Payloads that carry additional information beyond that provided through the LEXS Digest. This feature allows LEXS to support very specific exchange requirements while maintaining a level of compatibility across all applications that understand the base LEXS. Additionally, LEXS can also include a non-NIEM payload if required.

Note that you may be able to reuse and extend, if necessary, a Structured Payload from an existing LEXS-based IEPD.

#### **4.2 Reuse Other IEPDs**

The Information Exchange Clearinghouse describes a number of IEPDs that have been built by federal, state, local and tribal agencies and are available for re-use. This option should be considered if the previous options are not sufficient, based on business requirements of the exchange. This option saves development time and cost, but some customization of the schemas may be necessary to meet all business requirements; however, an exchange will not need to be developed completely from scratch. Some schema customization may be necessary to meet all

business requirements. However, an exchange will not need to be developed completely from scratch.

### **4.3 Develop a New NIEM IEPD**

If an existing IEPD is not available for use, then a new IEPD should be developed following the NIEM IEPD development process and using the NIEM model and tools. This option should be selected only after a thorough evaluation of the previous options has been completed.

### **4.4 Use Other Widely Supported Standard**

A NIEM-based exchange should not be implemented if another widely supported standard already exists that is better suited for a specific exchange type.

## **5 LEXS Publish Operations**

LEXS PD supports the action of publishing sharable information to a data repository. LEXS PD provides the structures required to represent the data and metadata associated with publishing data. The data repository receives and records the published information, generally, for the purpose of analysis, making it easier, faster, and less expensive to share data. In publishing data, a source system can submit one or multiple data items in a single, one-way action to a data consumer. LEXS also provides a “publish with acknowledgment” option where a source system can request a response to acknowledge the receipt of the publish message.

LEXS 4.0 supported one-way and two-way publish operations are show in the table below:

<b>Request</b>	<b>Response</b>
Publish, no response expected	<none>
Publish, response requested	Publish Acknowledgment

**Publish, no response expected** is a one-way operation which is used to submit information to a data consumer when no response or acknowledgment is expected. LEXS allows multiple publish messages in a single publish operation.

**Publish, response requested** is used to submit information to a data consumer when an acknowledgment is requested. This operation contains a single publish message.

**Publish Acknowledgment** is the response acknowledging that the publish message was received.

Note that LEXS recommends data consumers operate under a best effort approach. For PD this means that a data consumer should accept as much of the submission as possible. For example, if a submission contains multiple data items and one of them is faulty, the consumer should process the remaining data items.

## **6 LEXS Search and Retrieve Operations**

LEXS SR supports searching for and retrieving shared information from a data repository. LEXS SR provides a number of requests and corresponding responses that support data search and retrieval. SR requests fall into two broad categories, data requests and service provider requests. LEXS data requests provide the mechanisms for retrieving data and metadata from partner systems. LEXS service provider requests ask for system information about partner systems including capabilities, data owners, and availability.

In LEXS, search and retrieval of data is a multi-step process. A search request uses information that broadly identifies the target being sought and results in a response that returns possible candidates for the user to examine further. In general, a search query is followed by a request for a specific data item, rendered data item, or attachment. For instance, a search request for information about a person might be issued using a first and last name. Usually this is not enough information to specifically identify a single person. As a result of the search, the queried systems respond with information related to any person having the specified first and last name. The intent is that the user can narrow the search by reviewing the search response and then request more detailed information on a specific data item or a limited set of data items.

LEXS 4.0 supported search and retrieve operations are shown in the table below:

<b>Request</b>	<b>Response</b>
Text Search Request	Search Response
Structured Search Request	Search Response
Attachment Search Request	Search Response
Data Item Metadata Search Request	Search Response
Data Item Request	Data Item Response
Rendered Data Item Request	Attachment Response
Data Item With Attachment Request	Data Item With Attachment Response
Attachment Request	Attachment Response
Capabilities Request	Capabilities Response
Data Owners Request	Data Owners Response
Availability Request	Availability Response

LEXS SR provides several types of data requests. A **structured search request** looks for specific elements with certain values, such as a first name of “John” and a last name of “Smith.” A **text search request** is used to search for a value in any context, that is, no element name is specified. Depending on the implementation, the text search could be performed on unstructured data, such as a report, or on structured data, such as a name or narrative element. For example, a text search for the phrase “John Smith” on unstructured data might find a match in a report. A text search on structured data might find “John Smith” in a structured field defined specifically for a person’s full name. An **attachment search request** looks for one or more data items containing an attachment matching the criteria described in an attachment search request. A **data item metadata search request** looks for one or more data items containing metadata elements matching the criteria described in a data item metadata search request.

The **search response** from either a structured search or a text search can contain information about data items and/or attachments, such as mug shots and documents, associated with the search parameters. A **data item request** is used to get details on a specific data item. An

**attachment request** asks for one or more Attachments described in a search response. Either a data item or an attachment request is a follow-up query to a search request and uses the unique identifier supplied by the search response to identify a single target in a single partner system.

LEXS also provides several types of service provider requests and corresponding responses that ask for specific types of information. A **capabilities request** asks about the capabilities of a service provider and the **capabilities response** reports the capabilities, including whether it supports text and/or structured searches and the features of each, whether paging is supported, and what data item categories are available. A **data owners request** allows a data consumer to ask for information about a service provider's data owners. The **data owners response** contains the list of data owners available, including whether the data owner supports structured searches, unstructured searches or both. The **availability request** is used to request an indication as to whether the service provider is available. The **availability response** includes a true/false availability indicator and a text field for additional information that explains or qualifies the indicator, such as the reason the repository is down and when it will be available, anticipated unavailability, or limited access.

A search response may be sparsely populated and data items may contain only elements that are commonly used to help discriminate one data item from another. A data item response should contain all available details and contextual information associated with the requested data items.

Note that LEXS recommends data consumers and providers operate under a best effort approach. For SR this means that a service provider will utilize the search terms and modifiers it can, and gracefully ignore the remainder. For example, if a query specifies a last name of "Smith" and a middle name of "Joe" and the service provider being queried doesn't support searches on middle name, the service provider will search on the last name field and ignore the middle name. Further, the service provider can indicate to the query system that the response was only a 50% match and which fields were included in the search.

## 7 LEXS Subscribe and Notify Operations

LEXS SN includes a number of requests, responses, and notifications that support event-driven interaction between partner systems. The Subscribe and Notify category of operations provides a mechanism for systems to exchange information in a way that is different from Publish or Search and Retrieve discussed above. In general, a system asks to be notified when a situation or an event occurs. The subscription request is acknowledged by a response confirming the subscription and, in the future, when a subscribed to event or situation occurs,

the notifying system will send a notification message to the subscribing system. Subscriptions can also be renewed or cancelled.

A subscription can be requested concerning a specific data item, a set of search terms, or a more general topic. The situation that triggers a notification depends on the subscription. For example, with a Data Item Retrieval Subscription, it is the situation of the data item specified in the subscription being retrieved that causes the notification to be sent. Note that all subscriptions are to future new or modified data; a subscription will not result in notifications for data items that were already present in a partner system at the time of subscription. Each type of request, response, and notification is described below and the structures that support these operations are addressed in section

Subscriptions may be initiated by individual users, by agencies, or by agencies on behalf of users. When an agency subscribes, it may then take responsibility for notifying appropriate users. Agencies may subscribe on behalf of users to save users from managing subscriptions or because the agency makes decisions on what subscriptions are appropriate for specific users.

LEXS 4.0 supported subscribe and notify operations are shown in the table below:

<b>Request</b>	<b>Response</b>	<b>Notification</b>
Data Item Search Subscription	Subscribe Response	Data Item Search Subscription Notification
Data Item Retrieval Subscription	Subscribe Response	Data Item Retrieval Subscription Notification
Identical Search Subscription	Subscribe Response	Search Subscription Notification
Similar Search Subscription	Subscribe Response	Search Subscription Notification
Match Subscription	Subscribe Response	Match Subscription Notification
Topic Subscription	Subscribe Response	Topic Subscription Notification
Renew	Renew Response	
Unsubscribe	Unsubscribe Response	

LEXS SN provides several types of subscriptions and notifications described briefly below. Note that a subscription request is acknowledged by a **subscribe response** confirming the

subscription, if the subscription request is accepted. A fault is issued if a subscription request is rejected. Subscriptions may be rejected due to a malformed request or because the subscription criteria are not valid, for example, if the subscription is for a topic that is not supported. Subscriptions may also be accepted, but with a warning, for example, if a search subscription includes search criteria that are not supported.

A **data item search subscription** allows a partner system to request a subscription which results in a notification when other users perform a search that returns the specified data item. A **data item search subscription notification** is produced if a user sends a search that results in a search response message which contains the data item ID of interest. For example, if an end user specifies the data item ID "123456", a notification will be produced when another user searches on "John Doe", if that search includes data item 123456 in its search response message.

A **data item retrieval subscription** allows a partner system to request a subscription which results in a notification when other users perform an operation that retrieves the specified data item. A **data item retrieval subscription notification** is produced if a user requests a data item specified in the subscription. For example, if an end user specifies the data item ID "123456", a notification will be produced when another user retrieves data item 123456 using a data item request and response.

An **identical search subscription** allows a partner system to request a subscription which results in a notification when other users perform a search that is identical to the query provided in the subscription request. An **identical search subscription notification** is generated every time a user on the notifying system searches on terms that are exactly the same as those specified in the subscription request. For an identical search subscription, the user's search must match all fields and field values specified in the subscription search criteria. For example, if the subscription request specifies "first name is John and last name is Doe", a notification will be produced when another user searches for those exact search terms.

A **similar search subscription** allows a partner system to request a subscription which results in a notification when users perform a search that is similar to the query provided in the subscription request. A **similar search subscription notification** is generated every time a user on the notifying system searches on terms that are similar to those specified in the subscription request. The definition of "similar" is open to interpretation by the notification producer. For example, one system might compare two searches based on a fuzzy comparison of person first name and last name only. Another system might take an exact comparison of identifying numbers into account when comparing two searches.

**Match subscription** allows a partner system to request a subscription which results in a notification when data items that match the query provided in the subscription request are added or modified. A **match subscription notification** is generated every time one or more data items are created or modified that match those search terms specified in the subscription request. For example, if the end user specifies "first name is John and last name is Doe", a notification will be produced whenever a new or modified data item appears that includes the first name John and the last name Doe. If an end user specifies a data item ID from a previous search or retrieve, a notification would be sent if that individual data item is updated. The match query can include any kind of search supported, including a structured query, a text query, or a metadata search.

**Topic subscription** allows a subscriber to request a subscription which results in a notification on the topic or set of topics described in the subscription request. A **topic subscription notification** is generated every time information changes or becomes available which the publisher associates with the topic specified in the subscription request. A Topic is the mechanism used to describe the kind of notification or the underlying situation of interest. Topics are predefined by the partner system and LEXS SR capabilities request is used to learn what topics a partner system has available for subscription.

A **renew request** is sent by the subscriber to renew an existing subscription. A **renew response** is sent to acknowledge the renewal, if the request is accepted. A fault is issued if a renewal request is rejected.

An **unsubscribe request** is sent by the subscriber to terminate an existing subscription. An **unsubscribe response** is sent to acknowledge the action, if the termination request is accepted. A fault is issued if a termination request failed.

## 8 LEXS Domain Exchange Operations

LEXS provides Domain Exchange (DE) for use when LEXS PD, SR or SN do not provide the necessary capabilities required for an information exchange, but when the organizational or data structures of LEXS are beneficial. The information to be exchanged may utilize LEXS data elements or metadata, or may be information that cannot be represented by existing LEXS constructs.

The LEXS Domain Exchange uses the LEXS Domain Attribute element to represent the data being exchanged. The Domain Attribute element is a LEXS extension point which contains name, value, and/or XML block elements that can be used to represent information that is useful to specific service providers or consumers but that is not part of the LEXS specifications. Domain Attribute is included in a number of LEXS objects to provide a place for domain-specific

information. In the case of Domain Exchange, Domain Attribute may contain domain-specific information or may contain LEXS constructs. For example, an organization may use Domain Exchange in a hybrid publish/retrieve process, where a LEXS entity is sent out in a manner similar to a publish operation, but requires a response that is more like an SR response than a PD acknowledgement. In this case, the Domain Exchange operations can be used to send one or more LEXS entities and get one or more LEXS entities back.

LEXS 4.0 supports both one-way and two-way operations as shown in the following table:

Request	Response
Domain Action, no response expected	<none>
Domain Request	Domain Response

**Domain Action, no response expected** is a one-way operation which is used to submit information to a data consumer when no response or acknowledgment is expected.

**Domain Request** is used to submit a request when a response or acknowledgment is expected. Again, Domain Request should be reserved for functionality not directly supported by LEXS.

**Domain Response** is returned in response to a Domain Request.

Note that these operations should be used only to implement functionality not directly supported by the standard LEXS PD, SR, and SN operations.

## 9 Resources

Building LEXS exchanges can be a complicated process. The distribution package contains all the components of the LEXS specifications, but there are a number of other resources available that provide additional information to help in understanding the LEXS specifications and aid in building information exchanges. The contents of the distribution package and associated resources are described below.

### 9.1 LEXS Distribution Contents

#### 9.1.1 Schemas

LEXS consists of a collection of XML schemas that provide a normative definition of the specifications. The schema set includes NIEM subset schemas, LEXS extension schemas, a LEXS library schema, a LEXS code list schema, and a LEXS exchange schema.

### **9.1.2 High-Level Domain Model Diagrams**

High-level domain model diagrams show the interrelationships among the various message components.

### **9.1.3 Association Diagrams**

Association diagrams show the links between objects in the model using NIEM and LEXS associations.

### **9.1.4 Sequence Diagrams**

Sequence diagrams are artifacts that illustrate possible message flow.

### **9.1.5 Sample XML Instances**

Each LEXS operation is illustrated in one or more sample instances. These samples illustrate the organization and data model, the various entities and objects, metadata, and element sequencing.

### **9.1.6 Change log**

The change log documents changes between revisions of the specifications.

## ***9.2 Additional LEXS Resources***

### **9.2.1 LEXS Community Web Site**

The LEXS community web site is available for use by managers and implementers alike. It includes a download section where the LEXS specifications and various documentation artifacts can be accessed. It also includes a community forum where users and implementers can post questions and provide answers to others. The site also has links to other sites, such as ConTesA. The site is located at <http://lexsdev.org>.

### **9.2.2 LEXS User Guide**

The LEXS user guide provides detailed documentation on the LEXS specifications. The document includes a high level overview of the specifications, as well as details on LEXS PD, SR, SN, and DE operations. The guide also documents high level LEXS constructs used in the various operations and includes a detailed list of the various roles, objects, and associations included in the specifications. The latest version of the user guide can be downloaded from the LEXS community web site.

### **9.2.3 Component Mapping Workbook**

The Component Mapping Workbook (CMW) is the LEXS data model represented as a spreadsheet. This multi-page spreadsheet documents the data and metadata elements as well

as associations that are part of the information exchange, including their locations in schema. The CMW also documents the cardinality of all data elements and attributes. The latest version of the workbook can be downloaded from the LEXS community web site.

#### **9.2.4 Code Tables Spreadsheet**

The Code Tables spreadsheet contains all code lists used in the LEXS schemas, regardless of whether they are from NIEM or are specific to LEXS. The latest version of the workbook can be downloaded from the LEXS community web site.

#### **9.2.5 Language bindings**

Java and .NET language bindings are available for implementers to use, or to customize to their specific needs. The latest version of the bindings can be downloaded from the LEXS community web site.

#### **9.2.6 Sample implementations**

The sample web service implementations are functional web service client and server implementations that can be used as (a) a reference to see how LEXS web services can be implemented, (b) the basis for a production LEXS web service, or (c) a library of reusable components for developing LEXS web services. The samples include language bindings, sample WSDL files, test harnesses, client and server implementations, etc. The latest version of the sample implementations can be downloaded from the LEXS community web site.

#### **9.2.7 WSCF.blue4NIEM**

WSCF.blue4NIEM is a modification of the WSCF.blue (<http://wscfblue.codeplex.com/>) Visual Studio .NET 2008/2010 toolset that facilitates the development of web services using a contract first (specifically, a schema first) approach. WSCF.blue4NIEM has been tailored to process NIEM based exchanges and is available from the LEXS community web site.

#### **9.2.8 ConTesA**

The Conformance Testing Assistant (ConTesA) is a tool that tests a data instance for conformance to business rules and provides visualization of XML instances. ConTesA performs LEXS schema validation and tests for business rules that cannot be represented in XML Schema. Information exchanges can define additional business rules which can be added to ConTesA's rule base. ConTesA is available on-line, as a desktop tool, and as Web components that can be installed in an implementer's Web server. The online version, as well as links to the desktop and Web components, are available at <http://contesa.ittl.gtri.org>.

### 9.2.9 Schema viewers

Schema viewers are not specific to LEXS, but are general tools that apply to a number of different XML schemas, including LEXS, ULEX and NIEM.

Schema Central, a web-based tool to traverse and search a number of XML schemas (including LEXS, ULEX and NIEM), is available at <http://www.schemacentral.com>.

### 9.2.10 WSDL templates

Samples of Web Services Description Language (WSDL) files are provided as examples only. They can be used in production with minor modifications, or can be used as a reference to show how new WSDL files can be created.

## 9.3 NIEM Resources

### 9.3.1 NIEM.gov

This is the web site for the NIEM program that provides access to the NIEM specifications, documentation, news, training information, implementer tools, etc. Current documentation for the NDR and BIEC mentioned above are available on the web site. The site is located at <http://niem.gov>.

### 9.3.2 SSGT

The Subset Generator Tool (SSGT) provides a means for creating a subset of the full NIEM schemas. A user selected properties and types required for a data exchange, and the tool generates a conformant schema subset of the full NIEM schema set. All dependencies are automatically added to ensure the resulting schema subset is valid. The user requirements can be saved and/or reloaded in a wantlist file, allowing for the subset to be modified and regenerated. It is available at <http://tools.niem.gov/niemtools/ssgt/index.iepd>.

### 9.3.3 NIEM conformance tool

The NIEM conformance tool assists developers by automatically identifying potential locations of non-conformance within IEPD artifacts (such as schemas, metadata, catalog, etc.) using the latest published NIEM NDR, and associated specifications. A user uploads an IEPD, a set of schemas in a zip file, or an individual schema. A report is generated outlining any rules that the files violate. The tool is located at <http://tools.niem.gov/conformance/index.html>.

### 9.3.4 NIEM codelist generator

The NIEM codelist generator tool builds a NIEM conformant code list schema from an Excel or CSV file. Multiple code lists can be included in a single code list schema. The tool is located at <http://tools.niem.gov/niemtools/codelist/index.iepd>.

### 9.3.5 NIEM IEPD tool

The NIEM IEPD tool allows a user to upload/enter the artifacts required for an IEPD (schemas and documentation) along with its metadata and assembles it into a package per the NIEM IEPD specification. The tool is located at <http://tools.niem.gov/niemtools/iepdt/index.iepd>.

### 9.3.6 Implementation Guidance for NIEM-Conformant Exchanges

Excerpts from the Implementation Guidance for NIEM-Conformant Exchanges document are presented in section 4, When and How To Use LEXS. The document provides high-level guidance to program managers and architects on implementing NIEM-conformant information exchanges. This document can be downloaded from the LEXS community web site.

### 9.3.7 IEPD Clearinghouse

The Information Exchange Package Documentation (IEPD) Clearinghouse provides information on a variety of IEPDs that have been submitted by individuals and organizations who have implemented the Global Justice XML Data Model (Global JXDM) and the National Information Exchange Model (NIEM). The IEPD Clearinghouse is located at <http://www.it.ojp.gov/framesets/iepd-clearinghouse-noClose.htm>.

## 9.4 Related Standards

Links are provided below to a number of standards that may be important when implementing a LEXS-based information exchange.

- Global Federated Identity and Privilege Management (GFIPM): <http://gfipm.net/>
- SAML (Security Assertion Markup Language): <http://saml.xml.org/saml-specifications>
- Web Services Description Language (WSDL): <http://www.w3.org/TR/wSDL/>
- WS-Addressing: <http://www.w3.org/TR/ws-addr-core/>
- WS-BaseFaults: [http://docs.oasis-open.org/wsr/wsr/ws\\_base\\_faults-1.2-spec-os.pdf](http://docs.oasis-open.org/wsr/wsr/ws_base_faults-1.2-spec-os.pdf)
- WS-BaseNotification: [http://docs.oasis-open.org/wsn/wsn-ws\\_base\\_notification-1.3-spec-os.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf)
- WS-Topics: [http://docs.oasis-open.org/wsn/wsn-ws\\_topics-1.3-spec-os.pdf](http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf)

## 10 Acronyms

ARJIS	Automated Regional Justice Information System
BIEC	Business Information Exchange Components
CMW	Component Mapping Workbook
ConTesA	Conformance Testing Assistant
DHS	Department of Homeland Security
DOJ	Department of Justice
GFIPM	Global Federated Identity and Privilege Management
GJXDM	Global Justice XML Data Model
IEPD	Information Exchange Package Documentation
JRA	Justice Reference Architecture
LEISP	Law Enforcement Information Sharing Program
LEXS	Logical Entity Exchange Specification
LiNX	Law Enforcement Information Exchange
N-DEx	National Data Exchange
NDR	NIEM Naming and Design Rules
NIEM	National Information Exchange Model
OASIS	Organization for the Advancement of Structured Information Standards
SAML	Security Assertion Markup Language
ULEX	Universal Lexical Exchange
WSDL	Web Services Description Language