



**U.S. Department of Justice  
(DOJ)  
LEISP Exchange Specification 3.0  
LEXS 3.0 User Guide**

Revision 9

August 8, 2007

## Change History

Revision	Date	Author	Description of Revision
1	February 19, 2007	Sudhi Umarji	Initial Draft
2	March 19, 2007	Sudhi Umarji	Incorporated Feedback from DOJ Management
3	April 9, 2007	Sudhi Umarji	Added sections for Rendering Instructions and Structured Payload
4	May 29, 2007	Sudhi Umarji	Updated to reflect comments from reviewers
5	July 5, 2007	Sudhi Umarji	Updated to reflect comments from Jeremy Warren
6	July 10, 2007	Sudhi Umarji	Updated to correct formatting and minor edits
7	July 27, 2007	Sudhi Umarji	Added clarification for "lexs:LEXSVersion" and the use of ORI in message and package metadata.
8	August 1, 2007	Sudhi Umarji	Added clarification for data submitters vs. data owners.
9	August 8, 2007	Sudhi Umarji	Corrected version number information for LEXS-SR

## Table of Contents

<b>1. Introduction.....</b>	<b>6</b>
1.1 Background.....	6
1.2 National Information Exchange Model (NIEM).....	6
1.3 LEISP Exchange Specifications (LEXS).....	6
1.4 Partner Systems, Data Sources and Data Consumers.....	8
1.5 Audience.....	10
<b>2. The LEXS Data Model.....</b>	<b>12</b>
2.1 The Data Item and The Package Element.....	12
2.2 The “lexs:Message” Element.....	15
2.3 The “lexs:Submission” Element.....	16
2.4 The Root Element - “lexspd:doPublish”.....	16
2.5 The “lexs:MessageMetadata” Element.....	17
2.6 The “lexs:PackageMetadata” Element.....	21
2.7 The PackageDigest Element.....	24
2.8 Entities.....	25
2.8.1 Examples for EntityPerson.....	26
2.8.2 Example for EntityLocation.....	27
2.8.3 Example for EntityProperty.....	28
2.8.4 Example for EntityOrganization.....	28
2.8.5 Example for EntityDrug.....	28
2.8.6 Example for EntityFirearm.....	29
2.8.7 Example for EntityActivity.....	29
2.9 The “Structures Namespace” in NIEM.....	30
2.10 Roles.....	31
2.11 Augmentation.....	33
2.12 Associations.....	34
2.12.1 Associations Involving Persons.....	36
2.12.2 Associations Involving Organizations.....	38
2.12.3 Associations Involving Locations.....	40
2.12.4 Associations Involving Property.....	41
2.12.5 Associations Involving Activities.....	43
2.12.6 Associations Involving Communications.....	47
2.12.7 Associations Involving Package Structure.....	47
2.13 Attachments.....	47
2.14 Rendering Instructions.....	51
2.14.1 XHTML Stylesheet.....	53
2.14.2 HTML Stylesheet.....	53
2.14.3 PDFStylesheet.....	53
2.14.4 Text Stylesheet.....	53
2.14.5 Rendered Binary.....	53

2.14.6 Narrative.....	53
2.14.7 Other Stylesheet .....	53
2.15 Structured Payload and Levels of Understanding.....	54
2.16 LogicalID and SourceID.....	58
<b>Appendix A. End Notes .....</b>	<b>63</b>

## List of Figures

Figure 1. LEXS-SR and LEXS-PD in OneDOJ .....	9
Figure 2. "Data Submitters" vs. "Data Owners".....	10
Figure 3. LEXS 3.0 Package Structure .....	13
Figure 4. LEXS 3.0 Package-Digest .....	25
Figure 5. Some of the Possible Associations Between Entities.....	35
Figure 6. LEXS Structure for Attachments.....	48
Figure 7. LEXS Structure for Rendering Instructions.....	51
Figure 8. Specialized Systems and/or Communities Leveraging the Structured Payload Element .....	54
Figure 9. Multiple Structured Payload Elements in a Package.....	56

# 1. Introduction

The U.S. Department of Justice (referred to herein as “DOJ”) is transforming the way it shares law enforcement information with its federal, state, local, and tribal law enforcement partners. The vision is to create relationships and methods that allow information to be shared routinely across jurisdictional boundaries to prevent terrorism and to systematically improve the investigation and prosecution of criminal activity. DOJ will achieve its vision by formulating information sharing policies and standard business practices and by creating a unified, DOJ-wide technology architecture that will position DOJ as a committed partner in an information sharing environment of federal, state, local, and tribal law enforcement agencies.

## 1.1 Background

The strategy for DOJ’s transformation is implemented through the DOJ Law Enforcement Information Sharing Program (LEISP). This strategy is the result of a collaborative process involving senior leadership from DOJ component agencies and representatives from across the national law enforcement community. LEISP includes an initiative known as OneDOJ, for sharing DOJ data—from all its components—and is aligned with the Information Sharing Environment (ISE) mandated by the Intelligence Reform and Terrorism Prevention Act of 2004.

## 1.2 National Information Exchange Model (NIEM)

NIEM<sup>1</sup> is an interagency initiative to provide the foundation and building blocks for national-level interoperable information sharing and data exchange. The NIEM project was formally announced at the Global Justice XML Data Model (Global JXDM) Executive Briefing on February 28, 2005. It was initiated as, and continues to be, a joint venture between the U.S. Department of Homeland Security (DHS) and DOJ with outreach to other departments and agencies. The base technology for NIEM is derived from the Global JXDM.

## 1.3 LEISP Exchange Specifications (LEXS)

This document covers the LEISP Exchange Specifications (LEXS, pronounced "lex"), which leverages and reuses work from both LEISP and NIEM. This guide describes the data model, syntax, semantics and also provides usage guidelines for implementers. LEXS was created and is being further developed to support the primary objectives of LEISP and to minimize the impact of the changing requirements and varied demands for information sharing on the sources and consumers of law enforcement data.

LEXS is intended to address two aspects of information sharing:

- define and consistently describe units of information to be shared
- define interfaces and protocols to provide (publish) and request (subscribe, search) such information

LEXS provides an extensible framework for consistent packaging of the information, with specific places and markings for various elements of the shared information. The LEXS specification shields both data sources and data recipients from the complexity of multiple interfaces and allows for the multipurpose use of information: for example, a data item created

by a source can be consumed by multiple recipients that should be able to understand as much or as little of the data item as necessary.

This document describes LEXS 3.0. The different versions of LEXS were released as follows:

- LEXS 1.0 - April 2005
- LEXS 2.0 - February 2006
- LEXS 3.0 - January 2007

LEXS 1.0 was limited to support the sharing of unstructured data (e.g., text, narratives). LEXS 2.0 introduced a structured data model to describe real world objects (e.g., persons, places, locations) and associations between these objects. Since LEXS 2.0 supports the representation of structured data, LEXS 2.0-based systems can support functionality such as link charting / analysis and geo-spatial mapping. LEXS 1.0 is no longer recommended for use because it does not offer support for structured data.

LEXS 3.0 further enhances the specification by extending the LEXS 2.0 data model based on feedback received from LEXS 2.0 implementation experiences and new capabilities introduced in NIEM 1.0. Specifically, LEXS 3.0 introduces:

- additional details for representing some structured elements (e.g., illegal drugs)
- representations for activities (e.g., incidents)
- additional metadata identifying personal contact information relevant to law enforcement personnel associated with the information being shared
- support for rich media attachments (e.g., photos, audio recordings, video footage, PDF files)
- concept of “roles”
- rendering instructions
- a mechanism that allows implementers to define customized (e.g., regional, community, domain, agency specific) structured content that can be carried as payload within a LEXS 3.0 envelope

The mechanism in LEXS 3.0 that supports structured payload will assist implementers who want to develop special structures to meet their own needs (e.g., schemas for incident reports, warrants, case files, booking and incarceration records, criminal histories, suspicious activity reports) while at the same time retaining common definitions of the structured content so that the broader community of law enforcement users can benefit from sharing initiatives.

It is expected that existing implementations of LEXS will eventually migrate to LEXS 3.0 and any new projects being launched will move directly to this version or the subsequent version in the 3.x series.

LEXS 2.0 specifications can be obtained from the LEXS area on the CORE.GOV site at <http://collab.core.gov><sup>2</sup>. Any updates to this document will also be posted at <http://collab.core.gov>. In addition, this site also contains downloadable XML schemas for both LEXS 2.0 and LEXS 3.x.

In supporting the LEISP initiative, DOJ intends to leverage as many of its currently existing or planned information sharing investments as possible. DOJ has identified two investments that will provide capabilities that map well to some of the functions within the LEISP architecture. Deployment of these DOJ investments will facilitate increased timely access to DOJ's law enforcement information. These investments are the Regional Data Exchange (R-DEx) and the National Data Exchange (N-DEx) programs. Each investment will be the catalyst to fulfill a portion of the overall LEISP architecture. R-DEx 6.0 is currently operational, while N-DEx is currently in the planning and development phase.

### ***1.4 Partner Systems, Data Sources and Data Consumers***

Federal law enforcement agencies have a wealth of criminal history and incident information, as do regional (when we use the term regional, we mean state/local/tribal/regional) organizations. When two organizations mutually allow for two-way sharing of law enforcement information, with each party retaining ownership and possession of its information they are partners in law enforcement information sharing and each of the systems is referred to as a **Partner System**.

While LEXS is based on a general model for information sharing, it will be helpful to consider a specific information sharing initiative, to gain a better understanding of the concepts in the more general model. Consider the OneDOJ initiative referred to earlier. As part of OneDOJ, R-DEx is the data repository for full-text and structured, shareable, sensitive but unclassified (SBU), DOJ law enforcement data, and will serve two main functions: providing regional systems with access to DOJ's data and providing DOJ users with access to data in regional systems. Since R-DEx functions as the data repository on behalf of DOJ, DOJ investigative components (e.g., the Bureau of Alcohol, Tobacco, Firearms, and Explosives (ATF); the Drug Enforcement Administration (DEA); the Federal Bureau of Investigation (FBI); the U.S. Marshals Service (USMS); and the Bureau of Prisons (BOP)) publish their sharable data to R-DEx. In this specification 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 (e.g., R-DEx<sup>3</sup>). The data repository that receives and ingests the published information is referred to as a **Data Consumer** (e.g., R-DEx). The data source publishes periodically as determined by operations guidelines. In the long term, these uploads might actually be applied on a real-time basis over a Web Service. (Initially data to be shared may be uploaded in bulk either through FTP or a CD that is provided manually, as in the case of R-DEx.)

In the LEXS model when a user forms a query at a partner system by describing the information he is looking for, he gets back a list of data items which match that query. The query may be executed against the local repository where the user is "logged in" and / or the user's system may forward the search request to a partner system that it can interoperate with. The partner system in turn can execute the same query against its' own repository and return results to the user's system which then displays the information to the user. In the scenario just described, where a query is forwarded to a partner system we refer to the query as **federated**.

Currently, several regional information sharing systems provide tailored solutions for law enforcement, based on regional consensus. These systems provide a wide and varying set of capabilities; this includes capabilities such as phrase-based and concept-based searching of unstructured documents, such as investigative files. R-DEx provides a gateway to these regional



systems to enable searching of unstructured documents and retrieving matching documents based on LEXS. The first such regional system to interface with R-DEx was the Seattle, Washington regional law enforcement information sharing system called Northwest LInX (Law Enforcement Information Exchange). San Diego County's ARJIS (Automated Regional Justice Information System) is another instance of a regional system that interoperates with R-DEx.

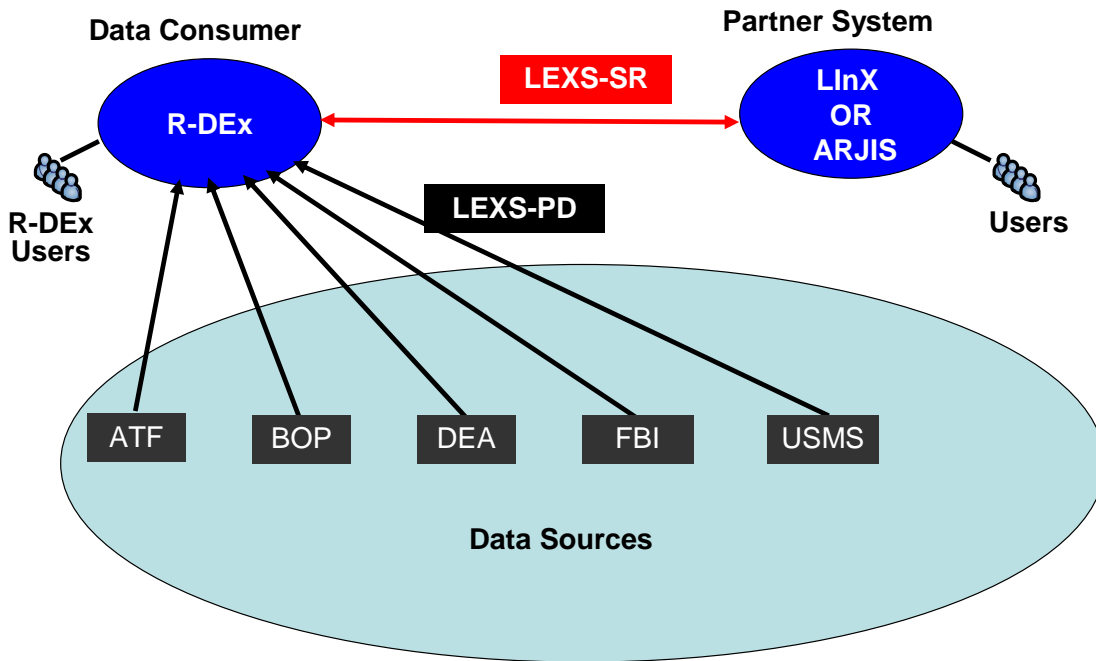


Figure 1. LEXS-SR and LEXS-PD in OneDOJ

The terms data source, partner system or data consumer 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. For illustrative purposes, Figure 1 shows how these terms can be applied to some of the participants in the R-DEx environment. In this context, the term data source refers to systems operated by ATF, BOP, DEA, FBI, and USMS that are currently publishing (uploading) information to R-DEx (data consumer) using LEXS. On the other hand, LInX and ARJIS are not data sources in the context of this definition. Though LInX and ARJIS may operate as data sources or data consumers with respect to other systems, they are not data sources or data consumers for R-DEx. However, both LInX and ARJIS are partner systems in relation to R-DEx since this term refers to a system that interacts with R-DEx by allowing its users to remotely submit queries to R-DEx. When R-DEx responds, the partner system presents the user who submitted the query with results from R-DEx. The same applies when an R-DEx user makes a query against the partner system.

ATF, BOP, DEA, FBI, and USMS are not operating any partner systems with R-DEx in the context of this definition. Currently users from ATF, BOP, DEA, FBI, and USMS access shared information by logging directly into R-DEx.

The term LEXS-PD (Publication and Discovery) applies to the interface between data sources and data consumers, while the term LEXS-SR (Search and Retrieval) applies to the interface between partner systems. See Figure 1. It is possible that in the future a given system is both a data source and a partner system, though no such LEXS-compliant system exists today.

In addition, it must be noted that we have used the term **Data Source** to refer to a system that is operated by an organization (e.g., DOJ investigative component) that publishes information to a data repository. In general, unless explicitly stated otherwise, the assumption is that the data source “owns” the information that it publishes. It is indeed also possible for the data repository to republish information that it has previously received from another source. When a data repository republishes information that it does not own, it is acting as an aggregator and to distinguish this case from one in which a data source only publishes what it owns, we use the term **Data Submitter** to refer to aggregators. We use the term **Data Owner** to refer to the original owner of the information.

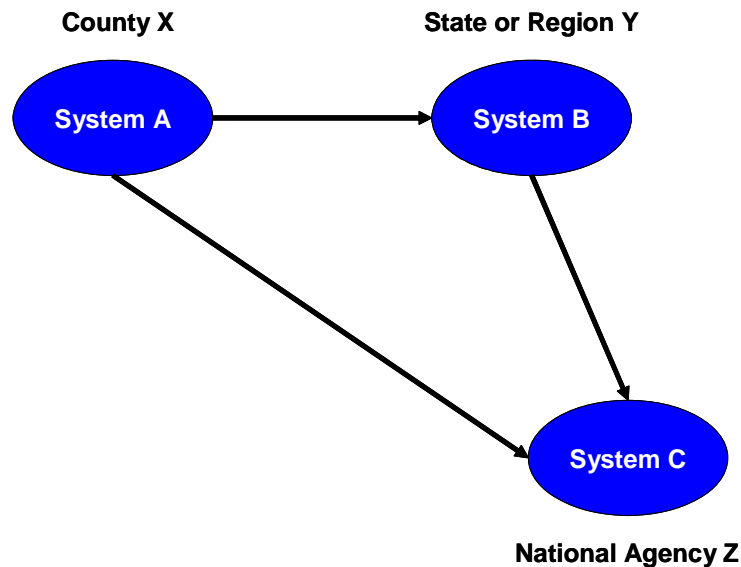


Figure 2. “Data Submitters” vs. “Data Owners”

In Figure 2, system B may be a data submitter to system C for information owned by system A. For a particular piece of information that was published to C, through such a route, A is the data owner and B is the data submitter. In addition, system A may also submit directly to C in which case it is both data owner and data submitter for information published in that operation.

## 1.5 Audience

This document is intended for the technical reader who needs to understand LEXS 3.0. It also presents introductory and context-setting information regarding LEXS 3.0, which may be of interest to business or policy managers involved in law enforcement and government. Currently LEXS-PD 3.0 has been defined and is described in this document. LEXS-SR for LEXS 3.1 is in development and when it is complete, this document will be updated to include the specification

for Search and Retrieval as well. As described earlier, specifications for LEXS-PD 2.0 and LEXS-SR 2.0 are available at <http://collab.core.gov>. Familiarity with those documents will obviously be helpful to the reader of this document, though this document is self-contained and a reading of the LEXS 2.0 specifications is not a pre-requisite to an understanding of this document. The latest schemas corresponding to both LEXS 2.0 as well as LEXS 3.0 are also available at <http://collab.core.gov>. This document has been prepared as a reference to communicate best practices and tips to implementers. Subsequent sections of this document contain XML fragments in an attempt to either explain a concept or communicate a best practice. Wherever specific tag names or sample XML fragments differ from the latest versions of the schema, the schema is the authoritative source.

As stated before, this document describes technical specifications. Questions regarding legal, privacy and political matters will not be addressed here. Questions about what data can be shared, what data must not be shared and the process for deciding whether a particular data element is sharable or not is outside the scope of this document. Our focus is to understand specifications that describe how information sharing will occur once a decision has been made that certain information must be shared.

## 2. The LEXS Data Model

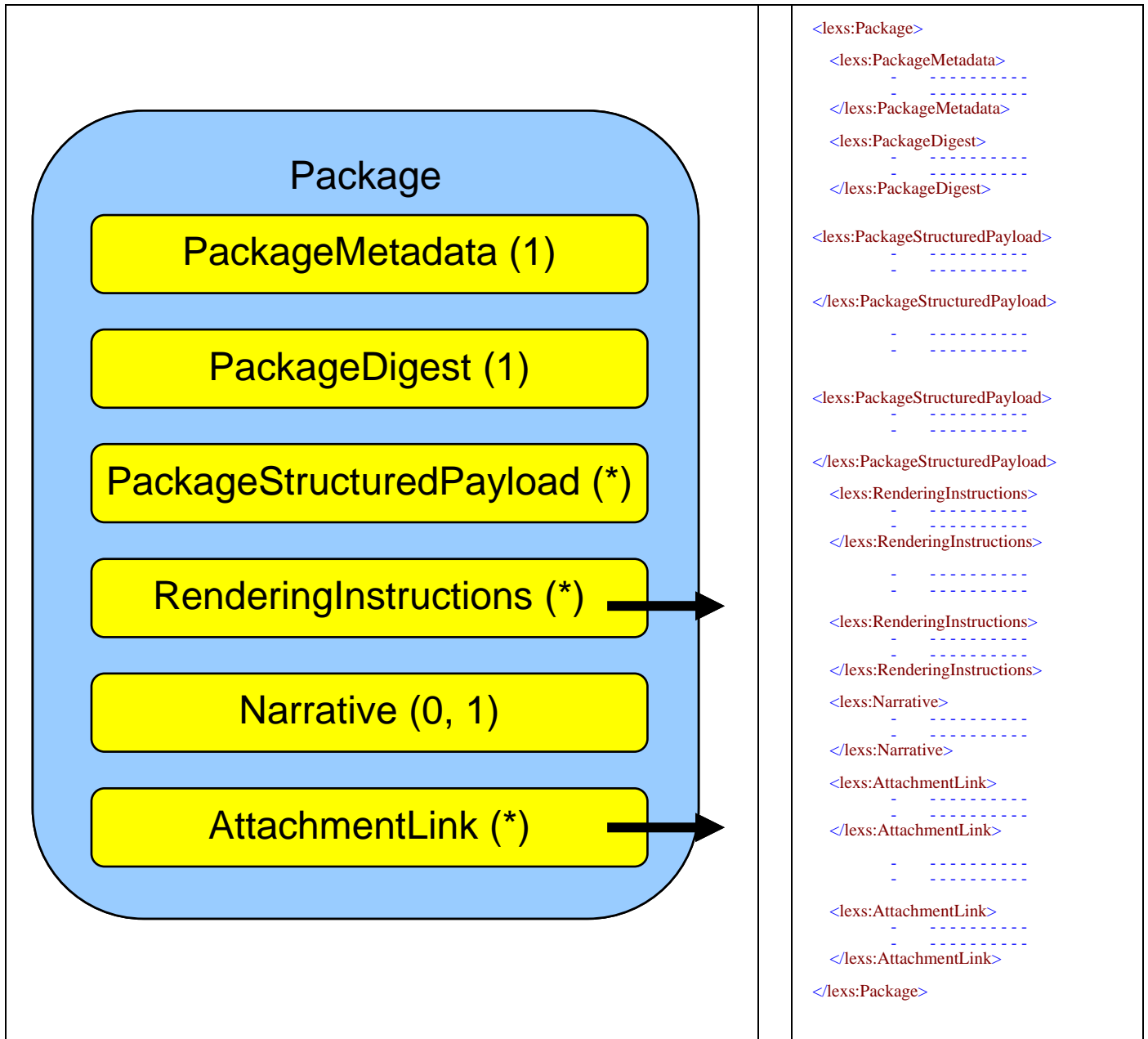
This section describes the building blocks of LEXS 3.0 in terms of the XML elements used in information exchange. The order of discussion is based on a sequence that will facilitate understanding on the part of the reader. We begin by defining the fundamental unit of information sharing: the “data item” and describe its relation to a package.

### *2.1 The Data Item and The Package Element*

Systems supporting law enforcement can be incident-based, event-based, activity-based, case-based, entity-based, or may use some other basis<sup>4</sup> to organize the data they contain in order to support their mission. In reference to these systems, the term **data item** is used in LEXS to denote a collection of related information (both structured and unstructured) that is shared as a unit. Some systems might call this a **record**. Thus, in an incident-based system, a data item may correspond to an incident record. In a case-based system, a data item may correspond to a collection of documents and records comprising a single case. In a system supporting the BOP, where the primary mission relates to prison inmates, a data item corresponds to an inmate record. Whatever the mapping, each data item represents a finite number of **entities** and **attributes**. The data item can be thought of as a collection of entities, attributes of these entities, **relationships** between these entities and **narratives** (unstructured textual information). In the BOP case, one might think of the prison inmate entity as the primary entity due to its correspondence with the data item. Clearly there will also be a number of other entities and associations that make up this data item, such as the entities that represent the telephone numbers that the prison inmate is authorized to call.

Considering the variety of incidents, events, cases, etc., that exist in the context of various data sources in LEXS, these aspects of the business domain are abstracted as data items rather than by introducing distinct structures to represent (each different kind of) them.

The term **package** refers to a standard representation of any data item, used for publication, retrieval and information sharing. In this document we use the term data item to refer to the standard, generic, flexible unit of information sharing in the abstract as it may exist at the source or destination system. For instance, we might use the term data item to refer to a record in a data source. When a data source extracts this data item or record and encodes it in a format that is compliant with the LEXS 3.0 schema we then refer to it as a package. The data source may or may not encode all the information in the data item into the package based on the business rules that it must follow. Conceptually, 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. The data source (owner of information) ultimately decides what information is relevant and should be shared, while the definition of a package in LEXS allows the data source to map such information to a standard format. A LEXS package consists of the package metadata, package digest, and an optional narrative. In addition a package may (optional) also contain, multiple structured payloads (i.e., with a structure defined by LEXS, or some other schema), multiple rendering instructions, and / or multiple attachments.



**Figure 3. LEXS 3.0 Package Structure**

The LEXS Package has the structure shown in Figure 3. This structure indicates that each Package element contains exactly one (number(s) in parenthesis specify cardinality of the element) PackageMetadata element, exactly one PackageDigest element and at most one Narrative element. The asterisk (\*) indicates that the PackageStructuredPayload, AttachmentLink and RenderingInstructions elements can each occur zero or more times in a package. AttachmentLink and RenderingInstructions are references to attachments and rendering instructions respectively.



**Note:** Since attachments and rendering instructions may be shared by multiple data items they are only referenced in the package, but provided as part of the message. The LEXS 3.0 approach for dealing with attachments is discussed in detail in Section 2.13.

The data model in Figure 3 is shown both graphically and as an XML fragment. In describing components of the LEXS 3.0 data model, this document will use graphic models or XML fragments as appropriate for clarity and brevity. Not every element in LEXS 3.0 is described in detail here and the interested reader is referred to the .xsd folders available at <http://collab.core.gov>.



**Note:** XML namespaces are used in the XML fragments shown in Figure 3 and they will also be used in other examples throughout this document. Namespaces have two purposes in XML:

1. To distinguish between elements and attributes from different vocabularies with different meanings that happen to share the same name.
2. To group all the related elements and attributes from a single XML schema together

Namespaces are implemented by attaching a prefix to each element and attribute. Each prefix is mapped to a URI<sup>5</sup> by an *xmlns:prefix* attribute. NIEM uses the term domain to identify a group that has the business requirement, desire, and capability to harmonize NIEM semantics (including structure) for the exchange of data. Some examples of the domains in NIEM are justice, intelligence, immigration, emergency management, international trade and infrastructure protection. NIEM has defined namespaces corresponding to each domain and also to group related elements such as code lists (imported or created from external sources). In addition there are LEXS namespaces that have been defined to group elements defined in LEXS and not in NIEM as follows:

### **NIEM Namespaces used in LEXS 3.0:**

```
xmlns:ansi_d20="http://niem.gov/niem/ansi\_d20/1.0"
xmlns:c="http://niem.gov/niem/common/1.0"
xmlns:can="http://niem.gov/niem/post-canada/1.0"
xmlns:census="http://niem.gov/niem/census/1.0"
xmlns:dod_jcs-pub2.0="http://niem.gov/niem/dod\_jcs-pub2.0-misc/1.0"
xmlns:dod="http://niem.gov/niem/dod\_misc/1.0"
xmlns:em="http://niem.gov/niem/domains/emergencyManagement/1.0"
xmlns:eo-12958="http://niem.gov/niem/eo-12958/1.0"
xmlns:fips_10-4="http://niem.gov/niem/fips\_10-4/1.0"
xmlns:i="http://niem.gov/niem/appinfo/1.0"
xmlns:im="http://niem.gov/niem/domains/immigration/1.0"
xmlns:intel="http://niem.gov/niem/domains/intelligence/1.0"
xmlns:iso_3166="http://niem.gov/niem/iso\_3166/1.0"
```

```
xmlns:iso_639-2b="http://niem.gov/niem/iso_639-2b/1.0"  
xmlns:j="http://niem.gov/niem/domains/justice/1.0"  
xmlns:ncic="http://niem.gov/niem/ncic_2000/1.0"  
xmlns:nibrs="http://niem.gov/niem/nibrs_misc/1.0"  
xmlns:niem-xsd="http://niem.gov/niem/proxy/xsd/1.0"  
xmlns:nonauth="http://niem.gov/niem/nonauthoritative-code/1.0"  
xmlns:s="http://niem.gov/niem/structures/1.0"  
xmlns:scr="http://niem.gov/niem/domains/screening/1.0"  
xmlns:tns="http://usdoj.gov/leisp/lexs/3.0/publishdiscover/ws"  
xmlns:u="http://niem.gov/niem/universal/1.0"  
xmlns:usps="http://niem.gov/niem/usps_states/1.0"  
xmlns:ut_offender="http://niem.gov/niem/ut_offender-tracking-misc/1.0"
```

### **LEXS 3.0 Namespaces:**

```
xmlns:lexs="http://usdoj.gov/leisp/lexs/3.0"  
xmlns:lexspd="http://usdoj.gov/leisp/lexs/3.0/publishdiscover"  
xmlns:lexscodes="http://usdoj.gov/leisp/lexs/3.0/codes"  
xmlns:lexslib="http://usdoj.gov/leisp/lexs/3.0/library"  
xmlns:lexspd="http://usdoj.gov/leisp/lexs/3.0/publishdiscover"
```

## ***2.2 The “lexs:Message” Element<sup>6</sup>***

Data Sources may find it convenient to group packages together in a message when they share certain characteristics in the message metadata or when they can use the same attachments. This is analogous to grouping packages together in a file for an upload. The structure of the message element is shown below.

```
<lexs:Message>  
  
  <!-- A message may include only a single MessageMetadata Element -->  
  <lexs:MessageMetadata> ----- </lexs:MessageMetadata>  
  
  <!-- A message may include one or more packages -->  
  
  <lexs:Package>----- </lexs:Package>  
  <lexs:Package>----- </lexs:Package>  
  <lexs:Package>----- </lexs:Package>  
  <lexs:Package>----- </lexs:Package>  
  <lexs:Package>----- </lexs:Package>  
  
  <!-- A message may include zero or more attachments -->  
  
  <lexs:Attachment>----- </lexs:Attachment>  
  <lexs:Attachment>----- </lexs:Attachment>  
  <lexs:Attachment>----- </lexs:Attachment>  
  
</lexs:Message>
```

Attachments exist outside the Package element but inside the Message element. More than one package can use the same attachment (e.g., fingerprint, mug shot, stylesheet) by using an AttachmentLink. A detailed explanation of this topic is included in Section 2.13.

### 2.3 The “lexs:Submission” Element

Data Sources can use the Submission element as a next higher level of grouping (e.g., call logs in one message and call lists in another message, with both messages in the same Submission). If a message is analogous to grouping packages together in a file, a Submission can be viewed as grouping several files together in a folder. There is no metadata element associated with a Submission element. Exactly one Submission element must be present and the structure of the submission element is shown below:

```
<lexs:Submission>  
  
  <!-- A Submission may include one or more messages -->  
  
  <lexs:Message> ----- <lexs:Message>  
  <lexs:Message> ----- <lexs:Message>  
  <lexs:Message> ----- <lexs:Message>  
  
</lexs:Submission>
```

### 2.4 The Root Element - “lexspd:doPublish”

In LEXS-PD 3.0, “lexspd:doPublish” is the XML document root element and it must contain exactly one submission as shown below:

```
<lexspd:doPublish>  
  
  <lexs:Submission>  
  -----  
  -----  
  </lexs:Submission>  
  
</lexspd:doPublish>
```

“lexspd:doPublish” defines an exchange document formatted in a standard way and, at this time, not bound to any specific communication protocol. For example, “lexspd:doPublish” can be used as the content of a SOAP<sup>7</sup> BODY element in a SOAP message from a data source in the role of a **service provider**<sup>8</sup>.



## 2.5 The “*lexs:MessageMetadata*” Element

This element applies to the Message that contains it. This element must contain the following sub-elements (unless marked as an Optional):

- lexs:LEXSVersion
- lexs:DataSensitivity
- lexs:DataSourceOrganizationAbbreviation
- lexs:DataSourceSystemInformation
- lexs:DataSourceContact
- lexs:MessageDateTime
- lexs:MessageSequenceNumber
- lexs:DomainAttribute (Optional Element)



**Clarification:** The reader will notice that in referring to XML element names, they are sometimes enclosed in quotes as in “*lexs:MessageMetadata*” and sometimes they appear in this document without quotes (as when they are part of a list). There is no hidden meaning ascribed to the use of quotes and their usage is mainly to improve readability.

Each of the above elements are discussed in more detail in the following sections:

### 2.5.1 *lexs:LEXSVersion*

The “*lexs:LEXSVersion*” element announces both the major, minor and patch version of LEXS that the message conforms to. The version numbering format is M.m.P (e.g. 3.2.4), where M is the major version number, m is the minor version number, and P is the patch release. Namespaces do not change between patch releases. Thus, namespaces (see Note on namespaces in Section 2.1) only include the major and minor version number. In general, patch releases should not introduce any structural changes. The intent is for instance documents conforming to different patch releases to validate successfully against a schema with a different patch release number as long as the instance document and schema have the same major and minor release numbers\*.

Example:

```
<lexs:LEXSVersion>3.0.4</lexs:LEXSVersion>
```

### 2.5.2 *lexs:DataSensitivity*

The “*lexs:DataSensitivity*” element contains the sensitivity marking<sup>9</sup> for the message.

Example: For “Sensitive But Unclassified”: `<lexs:DataSensitivity>SBU</lexs:DataSensitivity>`

\* LEXS 3.0.3 and LEXS 3.0.4 are exceptions to this since 3.0.3 instances will not validate against a 3.0.4 schema and vice versa. However, it is expected that applications that can ingest (without validation) a 3.0.3 instance will also be able to ingest 3.0.4 (and vice versa) with some data loss during ingestion. For details, please refer to the LEXS 3.0.4 schema change log.

### 2.5.3 lexs:DataSourceOrganizationAbbreviation

The “lexs:DataSourceOrganizationAbbreviation” element contains the ORI (originating agency identifier) <sup>10</sup> of the organization that owns the data source that is submitting the message (data submitter). If the organization does not have an ORI, then this element contains the abbreviation that is commonly applied to the organization that owns the data submitter.

Example for an organization (e.g. Boston Police Department) that has an ORI:

```
<lexs:DataSourceOrganizationAbbreviation>MA0130100</lexs:DataSourceOrganizationAbbreviation>
```

Example for an organization that does not have an ORI:

```
<lexs:DataSourceOrganizationAbbreviation>BOP</lexs:DataSourceOrganizationAbbreviation>
```

In those cases (e.g. N-DEx program) where the “lexs:DataSourceOrganizationAbbreviation” element contains the ORI of the organization that owns the data submitter, the recommended best practice is to use the “u:OrganizationName” element within “lexs:DataSourceContact” in the “lexs:MessageMetadata” element to communicate a human friendly text string that identifies the owning organization.

### 2.5.4 lexs:DataSourceSystemInformation

The “lexs:DataSourceSystemInformation” element contains the name of the system that is submitting the message.

Example:

```
<lexs:DataSourceSystemInformation>SENTRY</lexs:DataSourceSystemInformation>
```

### 2.5.5 lexs:DataSourceContact

The “lexs:DataSourceContact” element contains contact name and contact information about the person(s) who can be contacted about the message. Typically, the person is the IT manager at the agency who is responsible for supplying the data. The “lexs:PersonName” may be both a split out name and a full name but the “u:PersonSurName” is always required. This element could be useful if a data consumer needs to contact the administrative point of contact at the data source who is responsible for the submission. The contact telephone number is required, while the other kinds of numbers in this element are optional. The “u:OrganizationName” element is optional and when present it designates the Organization that employs the DataSourceContact.

Example:

```
<lexs:DataSourceContact>
  <lexs:PersonName>
    <u:PersonGivenName>James</u:PersonGivenName>
    <u:PersonSurName>West</u:PersonSurName>
  </lexs:PersonName>
  <c:ContactInformation>
    <u:ContactTelephoneNumber>
      <u:TelephoneAreaCodeID>123</u:TelephoneAreaCodeID>
      <u:TelephoneExchangeID>555</u:TelephoneExchangeID>
      <u:TelephoneSubscriberID>1212</u:TelephoneSubscriberID>
      <u:TelephoneSuffixID>x1234</u:TelephoneSuffixID>
    </u:ContactTelephoneNumber>
    <u:ContactMobileTelephoneNumber>
      <!-- Full phone number listed here, although the phone number could be split into parts as done above -->
      <u:TelephoneNumberFullID>123-555-0000</u:TelephoneNumberFullID>
    </u:ContactMobileTelephoneNumber>
    <u:ContactEmailID>
      <u:ID>jim.west@xyz.gov</u:ID>
    </u:ContactEmailID>
  </c:ContactInformation>
  <u:OrganizationName>XYZ Agency – U.S. Department of Justice</u:OrganizationName>
</lexs:DataSourceContact>
```

### 2.5.6 lexs:MessageDateTime

The “lexs:MessageDateTime” element contains a time stamp recording the date/time when the message was produced by the data source.

Example:     <lexs:MessageDateTime>2006-12-01T09:30:47.0Z</lexs:MessageDateTime>

### 2.5.7 lexs:MessageSequenceNumber

The “lexs:MessageSequenceNumber” element contains an increasing sequence number to differentiate between messages that must follow one another. This may be significant for data consumers when processing “INSERT” and “DELETE” instructions with respect to data items (see Section 2.6.6). These numbers cannot be duplicated within a single submission. If the numbers run out (e.g., consider a data source that generates numbers in the range 0 to 99 and then cycles back to 0; the numbers run out when the data source uses the number 99 to label the sequence number for a particular message) and the data source determines that a duplicate number will occur, the data source must begin a new submission (lexs:Submission).

Example:

```
<lexs:MessageSequenceNumber>17</lexs:MessageSequenceNumber>
```

### 2.5.8 lexs:DomainAttribute

The “lexs:DomainAttribute” element allows a data source to provide specific information that might have special meaning to itself and some or all consumers. There is no restriction on the content of these elements other than the requirement that the structure must have a Name/Value Pair and a domain identifier.



**Tip:** The example in this section shows how “REGION” which is an attribute that is specific to “R-DEx” can be carried within the “lexs:DomainAttribute”. In a similar manner, other systems or domains can define attributes that are specific to their own environment.

The domain may be same as “lexs:DataSourceOrganizationAbbreviation”, or may be a specific system/program (e.g., RDEX). The term “domain” as used here has no relationship to the term “domain” as used in NIEM<sup>11</sup> (National Information Exchange Model) or the term “domain” as used by IETF<sup>12</sup> (Internet Engineering Task Force). Data Sources and Data Consumers can define domains as required to meet business needs. Thus, a domain could be defined by a community of organizations and agencies participating in the sharing of Suspicious Activity Reports. A second domain could be defined for all data sources that contribute to R-DEx. A third domain could define all data sources that contribute to N-DEx. A particular data source might choose to belong to all these three domains. In fact this data source could further decide to publish information to three different data consumers, one in each domain. In addition this data source might even decide to publish the identical submission to all three data consumers in LEXS-PD 3.0 in order to simplify implementation and maximize software reuse. In such a case, the data source may include domain defined attributes for each domain that it publishes to.

Example:

```
<lexs:DomainAttribute>  
  <lexs:AttributeName>REGION</lexs:AttributeName>  
  <lexs:AttributeValue>SEA</lexs:AttributeValue>  
  <lexs:Domain>RDEX</lexs:Domain>  
</lexs:DomainAttribute>
```

Data Consumers will only use those domain defined attributes of interest to their mission and can ignore all other domain defined attribute (“lexs:DomainAttribute”) elements in the “lexs:MessageMetadata” element. The domain defined attribute can actually include any type of structured XML content and is not restricted to name/value pairs as shown in the accompanying example (the only usage to date has been in R-DEx which does use name/value pairs).

## 2.6 The “lexs:PackageMetadata” Element

This element applies to the “lexs:Package” that contains it. This element must contain the following sub elements (unless marked as an Optional):

- lexs:DataItemIDText
- lexs:DataItemReferenceIDText
- lexs:DataItemDate (Optional Element)
- lexs:DataItemContact (Optional Element)
- lexs:DataItemOriginatingAgencyIDText
- lexs:DataItemPublishInstruction (Optional Element)
- lexs:DataItemStatus
- lexs:DomainAttribute (Optional Element)

Each of these elements are discussed in more detail in the following sections:

### 2.6.1 lexs:DataItemIDText

The “lexs:DataItemIDText” element contains a unique identifier that is meaningful to the system owning the data item (data source). For example, the content of this element may be a RecordID at the originating data source. The content of “lexs:DataItemIDText” may not be convenient for human users to work with and is meant for use by systems. For example, this ID maybe used by a remote system to request retrieval of a specific data item in a providing system.

Example:

```
<lexs:DataItemIDText>W1nIHs6427964 3074*?.976VywcHg7IG</lexs:DataItemIDText>
```

The content of the “lexs:DataItemIDText” element is used by data consumers to track changes or deletions of data items between any two submissions from a data source. Therefore these identifiers must be persistent and unique to the data source (so that changes / deletions can be tracked).

### 2.6.2 lexs:DataItemReferenceIDText

The “lexs:DataItemReferenceIDText” element contains a human readable and usable mapping of the content of “lexs:DataItemIDText”.

Example:

```
<lexs:DataItemReferenceIDText>42796</lexs:DataItemReferenceIDText>
```



**Tip:** If the content of the “lexs:DataItemIDText” element is also human readable, there is no requirement for the content of “lexs:DataItemReferenceIDText” to be any different from “lexs:DataItemIDText”.

### 2.6.3 lexs:DataItemDate

The “lexs:DataItemDate” element contains a date. It is expected that data source includes the date that the data item was created in this element.

Example:

```
<lexs:DataItemDate>1967-08-13</lexs:DataItemDate>
```

### 2.6.4 lexs:DataItemContact

The “lexs:DataItemContact” element contains contact name and contact information about the person(s) who can be contacted about the data item. While the “lexs:DataSourceContact” element in the message metadata is meant to refer to an IT contact, this element (in package metadata) is generally meant to refer to a mission-related (e.g., law enforcement) contact. The “lexs:DataItemContact” element is optional, and there can be more than one. The “lexs:PersonName” may be both a split out name and a full name and the “u:PersonSurName” is always required. This element could be used to provide “real-time” secure e-mail notifications to designated Points of Contact (POC) when a data item is viewed locally or via a remote retrieval. The contact telephone number is required, while the other kinds of numbers in this element are optional. In the example below, a full telephone number is listed, though the phone number could be split into parts. The “u:OrganizationName” element is optional and when present it designates the Organization that employs the DataItemContact.

Example:

```
<lexs:DataItemContact>  
  <lexs:PersonName>  
    <u:PersonSurName>Smith</u:PersonSurName>  
    <u:PersonFullName>Mary Jones-Smith</u:PersonFullName>  
  </lexs:PersonName>  
  <c:ContactInformation>  
    <u:ContactTelephoneNumber>  
      <u:TelephoneNumberFullID>333-444-5555 x6</u:TelephoneNumberFullID>  
    </u:ContactTelephoneNumber>  
    <u:ContactMobileTelephoneNumber>  
      <u:TelephoneNumberFullID>333-444-6666</u:TelephoneNumberFullID>  
    </u:ContactMobileTelephoneNumber>  
    <u:ContactEmailID>  
      <u:ID>mary.jones.smith@abc.gov</u:ID>  
    </u:ContactEmailID>  
  </c:ContactInformation>  
  <u:OrganizationName>ABC Agency, U.S. DOJ</u:OrganizationName>  
</lexs:DataItemContact>
```



**Tip:** The data item contact may actually not be a person. For instance, this “u:ContactTelephoneNumber” may actually be the phone number for an agency “Call Center” or “Help Desk”. In this case the “u:PersonSurName” could be populated with appropriate descriptive text (e.g., “XYZ Call Center”).

### 2.6.5 lexs:DataItemOriginatingAgencyIDText

The “lexs:DataItemOriginatingAgencyIDText” element contains the name of the agency that originated this data item.

Example:

```
<lexs:DataItemOriginatingAgencyIDText>SDPD</lexs:DataItemOriginatingAgencyIDText>
```



**Tip:** This element in combination with “lexs:DataSourceOrganizationAbbreviation” allows a data source to distinguish between a case when it is merely republishing data from another data source (see Figure 2) vs. a case where the data source is itself the data owner.

For instance<sup>13</sup>, the “lexs:DataSourceOrganizationAbbreviation” identified at the message level might be “ARJIS”, while, at the data item level the “lexs:DataItemOriginatingAgencyIDText” might be “SDPD” for San Diego Police Department. This would imply that SDPD was the original owner who published this data item to ARJIS with ARJIS acting as a data consumer. ARJIS, then acting in a different role as a “submitter” is then republishing the same data item to another system as part of a LEXS-PD 3.0 message.

In those cases (e.g. N-DEx program) where a data source that owns a data item populates the “lexs:DataItemOriginatingAgencyIDText” element with an ORI of the organization that owns the data source, the recommended best practice is to use the “u:OrganizationName” element within “lexs:DataItemContact” in the “lexs:PackageMetadata” element to communicate a human friendly text string that identifies the owning organization.

### 2.6.6 lexs:DataItemPublishInstruction

The content of the “lexs:DataItemPublishInstruction” element is an instruction from the data source indicating how the system receiving the data should process the package. Valid values are “INSERT” or “DELETE”. “DELETE” indicates that the content corresponding to this DataItemID must be deleted. If the value is “DELETE” and that DataItemID does not exist this indicates a protocol logic error. “INSERT” means this package content replaces all content that corresponds to this DataItemID. If the value is “INSERT” and this data item is not already in the system then it must be added.

Example:

```
<lexs:DataItemPublishInstruction>Insert</lexs:DataItemPublishInstruction>
```

### 2.6.7 lexs:DataItemStatus

The “lexs:DataItemStatus” element indicating the status of the data item. For example if this data item corresponds to a closed case, this can be indicated here. The content of this element is free format text and not an enumeration.

Example:

```
<lexs:DataItemStatus>Closed</lexs:DataItemStatus>
```

### 2.6.8 lexs:DomainAttribute

The “lexs:DomainAttribute” element allows a data source to provide specific information that might have special meaning to itself and some or all consumers. There is no restriction on the content of these elements other than the requirement that the structure must have a Name / Value Pair and a domain identifier.

Example:

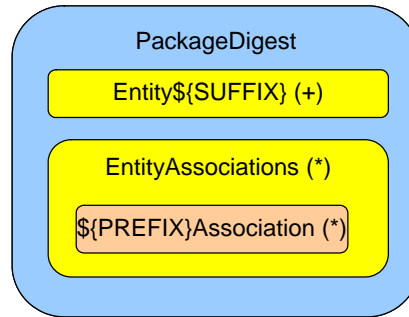
```
<lexs:DomainAttribute>  
  <lexs:AttributeName>DataCurrency</lexs:AttributeName>  
  <lexs:AttributeValue>Archive</lexs:AttributeValue>  
  <lexs:Domain>GBI</lexs:Domain>  
</lexs:DomainAttribute>
```

## 2.7 The PackageDigest Element

The concept of the “digest” is central to the power of LEXS as a framework for information sharing. A LEXS data item may represent any kind of underlying information—a warrant, an incident, a customs manifest. The digest is the common anchor for systems to use to handle this heterogeneous data without having to understand the specific context and meaning of the source. As long as the entities relevant to the “packaged” data item are represented in the PackageDigest, users will be able to discover, link, map, etc. the information within.

The PackageDigest is structured as shown in Figure 4. This element applies to the data item included in the Package and is a collection of the well-defined and commonly-understood structured data objects representing real-world entities (such as person, location, vehicle, etc.) at the very basic level. The objective of the digest is to present the most common characteristics of these real-world entities that can be supported by any (including the least sophisticated) data source or data consumer. The PackageDigest may also contain basic representations of the associations and roles of these entities. Digest-level data objects (structured entities) may be further augmented or described with additional details in the structured or unstructured payload of the package.





**Figure 4. LEXS 3.0 Package-Digest**

The PackageDigest may also contain data objects representing activities described in the payload (e.g., incident, arrest, booking) that allow a producer to define the context for certain data objects or their roles. When there is a need to relate a digest-level entity to a specific message attachment relevant to that entity (e.g., person to a fingerprint, or a telephone to a recording of the conversation), a special type of an association known as “lexs:EntityAttachmentLinkAssociation” is provided for such purposes. This association is used to associate an entity to an element known as “lexs:AttachmentLink”. The “lexs:AttachmentLink” in turn contains a URI<sup>5</sup> reference (contained in the AttachmentURI element) that uniquely identifies the actual electronic attachment (e.g., Word or .PDF file, binary file, image) in the message using the “lexs:Attachment” element. This association, therefore, will allow a data source to create associations between entities and attachments, (e.g., associate a person with an image file containing a fingerprint of that person). A more detailed description of the LEXS approach to attachments is provided in Section 2.13.

Each PackageDigest contains one or more entities (the + symbol signifies one or more) followed by zero or more (the \* symbol signifies zero or more) EntityAssociations elements that can each contain associations in the package. By convention all entities in LEXS 3.0 begin with the string “Entity” followed by a suffix that describes the kind of entity represented (hence the Entity\${SUFFIX} notation). Thus, EntityPerson, EntityLocation and EntityOrganization are some examples of entities.

Most associations in LEXS 3.0 are imported from NIEM 1.0, and as a result they follow the NIEM convention that all associations end with the string “Association” with a prefix that describes the kind of association being represented (hence the \${PREFIX}Association notation). Thus, MarriageAssociation, NeighborAssociation and ResidenceAssociation are all examples of associations. Each EntityAssociations element contains zero or more associations. More details about some of the associations that are supported in LEXS 3.0 are included in Section 2.12.

## **2.8 Entities**

As stated previously, the PackageDigest contains commonly understood structured data objects representing real-world entities (such as person, location, vehicle, etc.) at the very basic level. The objective of the digest is to present the most common characteristics of these real-world entities that can be supported by any (including the least sophisticated) producer or consumer.

LEXS 3.0 supports the following entities:

EntityPerson	EntityAircraft	EntityDrug	EntityEmail
EntityLocation	EntityBoat	EntityExplosive	EntityTelephoneNumber
EntityOrganization	EntityProperty	EntityFirearm	
EntityActivity	EntityVehicle		

This section provides some sample structures for entities.

### 2.8.1 Examples for EntityPerson

A very simple but technically complete example of a person entity is shown below.

```
<lexs:EntityPerson>  
  <lexsdigest:Person>  
    <u:PersonName>  
      <u:PersonGivenName>John</u:PersonGivenName>  
      <u:PersonSurName>Jacobs</u:PersonSurName>  
    </u:PersonName>  
  </lexsdigest:Person>  
</lexs:EntityPerson>
```

Obviously, the example above (XML example for Person “John Jacobs”) is a very simple example and both LEXS and the underlying NIEM structures provide a very rich set of elements that can contain a considerable amount of information relating to a person. Thus, a more detailed person sample may be represented by the following XML fragment:

```
<lexs:EntityPerson>  
  <lexsdigest:Person>  
    <u:PersonName>  
      <u:PersonGivenName>Billy</u:PersonGivenName>  
      <u:PersonMiddleName>Bob</u:PersonMiddleName>  
      <u:PersonSurName>Guy</u:PersonSurName>  
    </u:PersonName>  
    <j:PersonAgeMeasureRange>  
      <u:RangeMaximumMeasure>30</u:RangeMaximumMeasure>  
      <u:RangeMinimumMeasure>25</u:RangeMinimumMeasure>  
    </j:PersonAgeMeasureRange>  
    <u:PersonBirthDate>1972-05-09</u:PersonBirthDate>  
    <c:PersonHeightMeasureRange u:personHeightUnitCode="ncic">  
      <u:RangeMaximumMeasure>600</u:RangeMaximumMeasure>  
      <u:RangeMinimumMeasure>506</u:RangeMinimumMeasure>  
    </c:PersonHeightMeasureRange>  
    <c:PersonSexCode>M</c:PersonSexCode>  
    <u:PersonSSNID>  
      <u:ID>987654321</u:ID>  
    </u:PersonSSNID>  
    <j:PersonWeightMeasureRange u:personWeightUnitCode="lb">  
      <u:RangeMaximumMeasure>250</u:RangeMaximumMeasure>  
      <u:RangeMinimumMeasure>200</u:RangeMinimumMeasure>  
    </j:PersonWeightMeasureRange>  
    <c:PersonAlternateName>  
      <u:PersonFullName>Taz</u:PersonFullName>
```

```
</c:PersonAlternateName>  
<c:PersonEyeColorCode>BRO</c:PersonEyeColorCode>  
<c:PersonHairColorCode>BRO</c:PersonHairColorCode>  
<c:PersonRaceText>White</c:PersonRaceText>  
</lexsdigest:Person>  
</lexs:EntityPerson>
```



**Clarification:** The “lexs:EntityPerson” example for “Billy Bob Guy” clearly has more information than the example for “John Jacobs” in this section. However, we have still not seen how to represent roles that a person may have, associations that may link a person entity with other entities and the augmentation structure that may be used to provide additional information about a person. These topics will be covered in more detail in later sections after we have first had a chance to look at some simple example structures for some of the more common entities.

## 2.8.2 Example for EntityLocation

```
<lexs:EntityLocation>  
  <u:Location>  
    <u:LocationAddress>  
      <u:LocationStreet>  
        <u:StreetFullText>12 Paradise Street</u:StreetFullText>  
      </u:LocationStreet>  
      <u:LocationSecondaryUnitText>Apt. 2234</u:LocationSecondaryUnitText>  
      <u:LocationCityName>Utopia</u:LocationCityName>  
      <u:LocationStateName>MD</u:LocationStateName>  
      <u:LocationPostalCodeID>  
        <u:ID>21899</u:ID>  
      </u:LocationPostalCodeID>  
    </u:LocationAddress>  
    <u:LocationGeographicCoordinate>  
      <u:GeographicCoordinateLatitude>  
        <u:LatitudeDegreeValue>42</u:LatitudeDegreeValue>  
        <u:LatitudeMinuteValue>12</u:LatitudeMinuteValue>  
        <u:LatitudeSecondValue>20</u:LatitudeSecondValue>  
      </u:GeographicCoordinateLatitude>  
      <u:GeographicCoordinateLongitude>  
        <u:LongitudeDegreeValue>42</u:LongitudeDegreeValue>  
        <u:LongitudeMinuteValue>12</u:LongitudeMinuteValue>  
        <u:LongitudeSecondValue>20</u:LongitudeSecondValue>  
      </u:GeographicCoordinateLongitude>  
    </u:LocationGeographicCoordinate>  
  </u:Location>  
</lexs:EntityLocation>
```

### 2.8.3 Example for EntityProperty

```
<lexs:EntityProperty>  
  <c:Property>  
    <u:PropertyDescriptionText>Diamond Ring</u:PropertyDescriptionText>  
    <u:PropertyMakeName>SS Jewelry Company</u:PropertyMakeName>  
    <u:PropertyModelName>Teardrop Shaped Engagement Ring</u:PropertyModelName>  
    <u:PropertyValue>  
      <u:PropertyValueAmount>375</u:PropertyValueAmount>  
    </u:PropertyValue>  
    <c:PropertySerialID>  
      <u:ID>ABC123Z00021</u:ID>  
    </c:PropertySerialID>  
  </c:Property>  
</lexs:EntityProperty>
```

### 2.8.4 Example for EntityOrganization

```
<lexs:EntityOrganization>  
  <c:Organization>  
    <u:OrganizationName>Westside Renegades</u:OrganizationName>  
    <u:OrganizationCategoryText>Gang</u:OrganizationCategoryText>  
  </c:Organization>  
</lexs:EntityOrganization>
```

### 2.8.5 Example for EntityDrug

```
<lexs:EntityDrug>  
  <lexsdigest:Drug>  
    <u:PropertyValue>  
      <u:PropertyValueAmount>375</u:PropertyValueAmount>  
    </u:PropertyValue>  
    <c:DrugQuantityMeasure c:drugUnitCode="NP">5</c:DrugQuantityMeasure>  
  </lexsdigest:Drug>  
</lexs:EntityDrug>
```

## 2.8.6 Example for EntityFirearm

```
<lexs:EntityFirearm>  
  <c:Firearm>  
    <u:PropertyDescriptionText>Weapon Description. Big Gun.</u:PropertyDescriptionText>  
    <c:PropertySerialID>  
      <u:ID>123ABC4578</u:ID>  
    </c:PropertySerialID>  
    <c:FirearmMakeCode>NOM</c:FirearmMakeCode>  
    <c:FirearmCategoryCode>P</c:FirearmCategoryCode>  
    <c:FirearmCategoryDescriptionCode>R</c:FirearmCategoryDescriptionCode>  
    <c:FirearmAutomaticIndicator>true</c:FirearmAutomaticIndicator>  
    <c:FirearmCaliberCode>9</c:FirearmCaliberCode>  
    <c:FirearmFinishCode>LAV</c:FirearmFinishCode>  
  </c:Firearm>  
</lexs:EntityFirearm>
```

## 2.8.7 Example for EntityActivity

```
<lexs:EntityActivity>  
  <u:Activity>  
    <u:ActivityID>  
      <u:ID>Inc123456</u:ID>  
    </u:ActivityID>  
    <u:ActivityCategoryText>Incident</u:ActivityCategoryText>  
    <u:ActivityDescriptionText>  
      The intoxicated subject became angry and slashed the tires of the victim's vehicle.  
    </u:ActivityDescriptionText>  
    <u:ActivityDate>2000-08-23</u:ActivityDate>  
    <u:ActivityTime>14:11:00</u:ActivityTime>  
    <u:ActivityEndDate>2000-08-23</u:ActivityEndDate>  
    <u:ActivityEndTime>14:22:00</u:ActivityEndTime>  
  </u:Activity>  
</lexs:EntityActivity>
```

## 2.9 The “Structures Namespace” in NIEM

Before reviewing examples for roles and associations, it will be helpful to review the structures namespace that NIEM provides containing structures for organizing data. These structures are used to augment XML data. The structures provided are not meant to replace fundamental XML organization methods; they are intended to assist them. The NIEM structures namespace is identified by the URI “<http://niem.gov/niem/structures/1.0>”. The structures namespace is a “utility” namespace, different from other NIEM namespaces that define NIEM content. Both this document and the NIEM schemas refer to this namespace via the prefix “s”.

In LEXS XML instances, one way of expressing relationships between elements is by expressing the data objects as XML elements, and having one element contain other elements. For example, we have seen in the “lexs:EntityPerson” example for “Billy Bob Guy”

```
<lexs:EntityPerson>
  <lexsdigest:Person>
    <u:PersonName>
      <u:PersonGivenName>Billy</u:PersonGivenName>
      <u:PersonMiddleName>Bob</u:PersonMiddleName>
      <u:PersonSurName>Guy</u:PersonSurName>
    </u:PersonName>
    .....
    <u:PersonSSNID>
      <u:ID>987654321</u:ID>
    </u:PersonSSNID>
    .....
  </lexsdigest:Person>
</lexs:EntityPerson>
```

In this example there is an implicit relationship between the outer element “lexsdigest:Person” (the “containing” element, a.k.a. the parent element) and the inner elements (the “contained” elements, a.k.a. the child elements) such as “u:PersonName” or “u:PersonSSNID”. Here the expression of relationships is made by containment and this pattern is typically used in relating an entity to its attributes. (The entity in the example is a person with name “Billy Bob Guy” and SSN 987654321).

Expression of all relationships via element containment is not always desirable or possible. Situations that cause problems include circular relationships and repeated relationships. For example, let’s say “Object1 has a relationship to Object2” and “Object3 has a relationship to Object2”. Expressed via containment, this would result in a duplicate of Object2. A method that solves this problem is the use of references. In a programming language like C or assembler, a pointer would be used. In Java, a reference value might be used. The method defined by the XML standard<sup>14</sup> is the use of ID and IDREF. An IDREF refers to an ID. However, instead of using ID and IDREF directly, within a NIEM-conformant schema, a reference element is defined to be of type “s:ReferenceType”. Any element of this type can be used as a reference element. In addition, the schema for the structures namespace defines the following attributes that are frequently used with NIEM conformant elements:

- `<attribute name="id" type="ID"/>`
- `<attribute name="linkMetadata" type="IDREFS"/>`
- `<attribute name="metadata" type="IDREFS"/>`
- `<attribute name="ref" type="IDREF"/>`

The following sections will provide examples that use these attributes.

## 2.10 Roles

A role is a specific kind of entity that represents a particular context or activity for another entity.

The only role supported in LEXS-PD 3.0 for firearm, explosive, aircraft, vehicle, boat and drug entities is Weapon. The only roles supported in LEXS-PD 3.0 for property entities are Resource and Weapon.

The following roles are supported in LEXS-PD 3.0 for Person:

Resource	MissingPerson
ArrestOfficial	RegisteredOffender
BookingEmployee	RegisteredSexOffender
BookingReleaseCorrectionsAnalyst	Subject
BookingSearchOfficial	ArrestSubject
BookingTelephoneCallSupervisingOfficial	BailSubject
BookingTransportOfficial	BookingSubject
CitationIssuingOfficial	ChargeSubject
CourtOrderServiceOfficialEnforcement	CitationSubject
CourtOrderServiceOfficialJudicial	ConvictionSubject
CustodyTransferReceivingEnforcementOfficial	CourtOrderDesignatedSubject
CustodyTransferReleasingEnforcementOfficial	CustodyTransferSubject
EnforcementOfficial	ForceSubject
IncidentAssistingOfficial	IncidentSubject
IncidentReportingOfficial	SentenceSubject
IncidentResponseOfficial	SupervisionSubject
IncidentSupervisingOfficial	Suspect
PropertySeizureSeizingEnforcementOfficial	VerdictSubject
ServiceCallDispatchedOfficial	ChargeVictim
SupervisionOfficial	ForceVictim
VisitationSupervisingOfficialEnforcement	IncidentVictim
AppellateCaseDecisionIssuingJudge	CaseWitness
AppellateCaseNoticeProsecutingAttorney	IncidentWitness
Attorney	MissingPersonLastSeenWitness
CourtEventJudge	OtherInvolvedPerson
CourtOrderIssuingJudicialOfficial	Informant
Judge	Inmate
JudicialOfficial	Parolee
SeverityLevelAssignedJudge	Prisoner
VerdictIssuingJudge	Probationer
VisitationSupervisingOfficialJudicial	ProtectedParty
CaseJuror	

The following roles are supported in LEXS-PD 3.0 for Organization:

VehicleBrander	CourtOrderDesignatedSubject
CriminalOrganization	ForceSubject
ChargeVictim	IncidentSubject
ForceVictim	SentenceSubject
IncidentVictim	Subject
CitationSubject	Suspect
ChargeSubject	SupervisionSubject
ConvictionSubject	VerdictSubject

A role may be specific to time, incident, or employment. For example, if “Billy Bob Guy” picks up an object and hits “John Jacobs” with it, the object will take on the role of a weapon, and “Billy Bob Guy” will take on the role of a "subject", and “John Jacobs” may take on the role of "victim". For example, a weapon may be a role of an object, and may have a user of the weapon, an activity in which it is involved, and a description of how the weapon was used. Any entity may take multiple roles in a package. For example, a single person may take the role of "arresting officer", "victim", and "witness". Consider the EntityFirearm described previously as:

```
<lexs:EntityFirearm>  
  <c:Firearm>  
    <u:PropertyDescriptionText>Weapon Description. Big Gun.</u:PropertyDescriptionText>  
    <c:PropertySerialID>  
      <u:ID>123ABC4578</u:ID>  
    </c:PropertySerialID>  
    <c:FirearmMakeCode>NOM</c:FirearmMakeCode>  
    <c:FirearmCategoryCode>P</c:FirearmCategoryCode>  
    <c:FirearmCategoryDescriptionCode>R</c:FirearmCategoryDescriptionCode>  
    <c:FirearmAutomaticIndicator>true</c:FirearmAutomaticIndicator>  
    <c:FirearmCaliberCode>9</c:FirearmCaliberCode>  
    <c:FirearmFinishCode>LAV</c:FirearmFinishCode>  
  </c:Firearm>  
</lexs:EntityFirearm>
```

The same firearm used as a weapon could be represented in the structure:

```
<lexs:EntityFirearm>  
  <c:Firearm s:id="Frm2">  
    <u:PropertyDescriptionText>Weapon Description. Big Gun.</u:PropertyDescriptionText>  
    <c:PropertySerialID>  
      <u:ID>123ABC4578</u:ID>  
    </c:PropertySerialID>  
    <c:FirearmMakeCode>NOM</c:FirearmMakeCode>  
    <c:FirearmCategoryCode>P</c:FirearmCategoryCode>  
    <c:FirearmCategoryDescriptionCode>R</c:FirearmCategoryDescriptionCode>  
    <c:FirearmAutomaticIndicator>true</c:FirearmAutomaticIndicator>  
    <c:FirearmCaliberCode>9</c:FirearmCaliberCode>  
    <c:FirearmFinishCode>LAV</c:FirearmFinishCode>  
  </c:Firearm>  
  <c:Weapon>  
    <u:RoleOfPropertyReference s:ref="Frm2"/>  
  </c:Weapon>  
</lexs:EntityFirearm>
```



In the above example, “u:RoleOfPropertyReference” is of “s:ReferenceType” and points to the “s:id” attribute of “c:Firearm” that has a value of “Frm2”.

The following structure shows how additional roles can be represented for “Billy Bob Guy”:

```
<lexs:EntityPerson>
  <lexsdigest:Person s:id="Sub2">
    <u:PersonName>
      <u:PersonGivenName>Billy</u:PersonGivenName>
      <u:PersonMiddleName>Bob</u:PersonMiddleName>
      <u:PersonSurName>Guy</u:PersonSurName>
    </u:PersonName>
    <u:PersonSSNID>
      <u:ID>987654321</u:ID>
    </u:PersonSSNID>
  </lexsdigest:Person>
  <j:ArrestSubject>
    <u:RoleOfPropertyReference s:ref="Sub2"/>
  </j:ArrestSubject>
  <j:IncidentSubject>
    <u:RoleOfPropertyReference s:ref="Sub2"/>
  </j:IncidentSubject>
</lexs:EntityPerson>

<lexs:EntityFirearm>
  <c:Firearm s:id="Frm2">
    <u:PropertyDescriptionText>Weapon Description. Big Gun.</u:PropertyDescriptionText>
    <c:PropertySerialID>
      <u:ID>123ABC4578</u:ID>
    </c:PropertySerialID>
    <c:FirearmMakeCode>NOM</c:FirearmMakeCode>
    <c:FirearmCaliberCode>9</c:FirearmCaliberCode>
    <c:FirearmFinishCode>LAV</c:FirearmFinishCode>
  </c:Firearm>
  <c:Weapon>
    <u:RoleOfPropertyReference s:ref="Frm2"/>
    <c:WeaponUserReference s:ref="Sub2"/>
  </c:Weapon>
</lexs:EntityFirearm>
```

A “c:WeaponUserReference” to “Billy Bob Guy” is included in the Weapon role through the “Sub2” pointer. Additional roles for “Billy Bob Guy” as a “j:ArrestSubject” and a “j:IncidentSubject” have also been added to the structure.

## 2.11 Augmentation

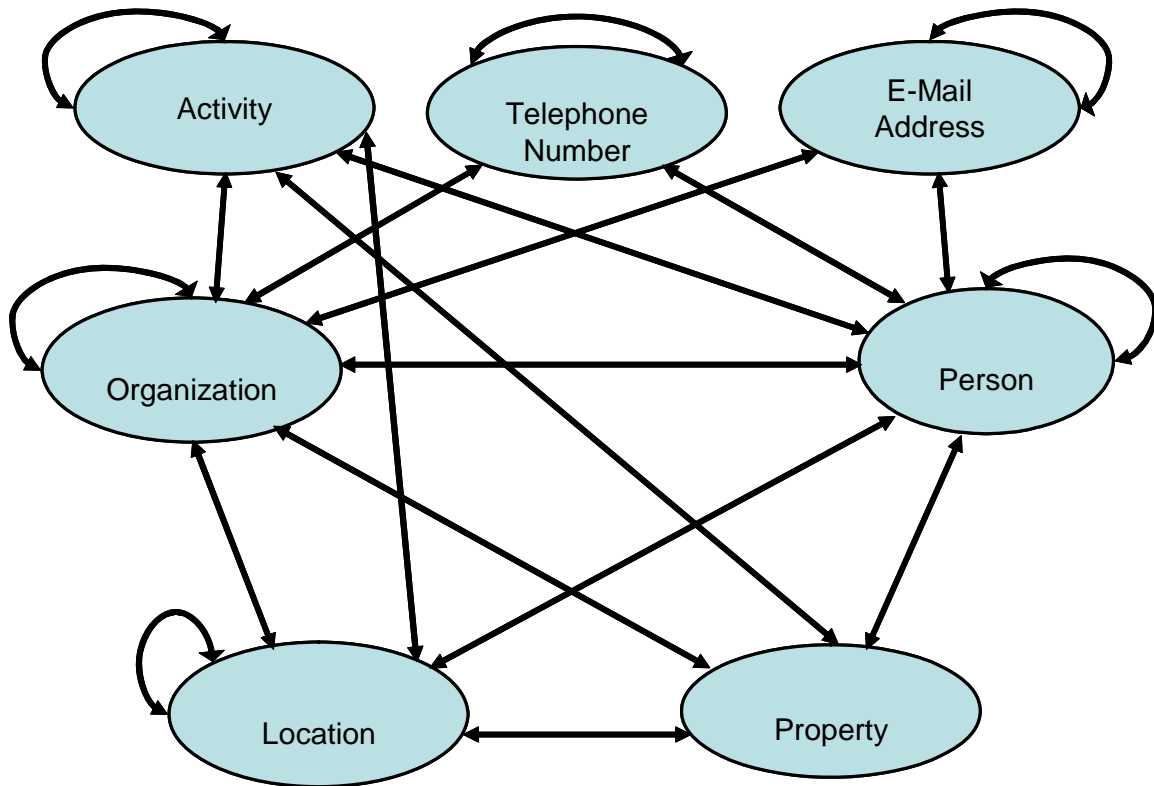
LEXS makes use of the Augmentation mechanism in NIEM. Augmentation allows an author to add new attributes to existing entities, without creating a new entity through an inheritance mechanism, thus maximizing reuse (inheritance is appropriate when there is a need to describe a new kind of entity, not simply add certain attributes as required by the context). The term “Base Type” applies to the type that needs to be enhanced by adding attributes. The base type may

come from a NIEM core namespace or other NIEM-conformant namespaces. The term “Augmentation Data” applies to the information to be added to the base type. Thus, augmentation of an entity is the addition of domain- or model-specific information about a type. Augmentations may be provided by domains or NIEM-conformant application data models. For example, we may need “justice-domain” data about a person. This is different than creating a new kind of person. In the real world, a person for whom justice-related data exists is not a different type of person than one that has intel-related data about them. It is most likely that a person will have both intel-related and justice-related data about them. The example below shows the use of “j:PersonAugmentation” to include “j:DriverLicense” and the use of “lexsdigest:PersonAugmentation” to include “lexsdigest:PersonUSMSFugitiveID” for “Billy Guy”. Here, the driver license structure was added as an augmentation to c:Person by the justice domain and the USMS Fugitive ID structure was added an augmentation to j:Person by LEXS 3.0 in the lexsdigest namespace.

```
<lexs:EntityPerson>
  <lexsdigest:Person s:id="Sub1">
    <u:PersonName>
      <u:PersonGivenName>Billy</u:PersonGivenName>
      <u:PersonSurName>Guy</u:PersonSurName>
    </u:PersonName>
    <j:PersonAgeMeasureRange>
      <u:RangeMaximumMeasure>30</u:RangeMaximumMeasure>
      <u:RangeMinimumMeasure>25</u:RangeMinimumMeasure>
    </j:PersonAgeMeasureRange>
    <c:PersonRaceText>White</c:PersonRaceText>
    <j:PersonAugmentation>
      <j:DriverLicense>
        <j:DriverAuthorizationID>
          <u:ID>E654321</u:ID>
          <u:IDEffectiveDate>1999-06-01</u:IDEffectiveDate>
          <u:IDExpirationDate>2005-06-01</u:IDExpirationDate>
          <j:IDJurisdictionNCICLISCode>WV</j:IDJurisdictionNCICLISCode>
        </j:DriverAuthorizationID>
      </j:DriverLicense>
    </j:PersonAugmentation>
    <lexsdigest:PersonAugmentation xsi:nil="false">
      <lexsdigest:PersonUSMSFugitiveID xsi:nil="false">
        <u:ID>81636-209</u:ID>
      </lexsdigest:PersonUSMSFugitiveID>
    </lexsdigest:PersonAugmentation>
  </lexsdigest:Person>
</lexs:EntityPerson>
```

## 2.12 Associations

In addition to entities and attributes, it is useful to define associations to represent the relationship between entities. In both LEXS and NIEM, these are called association objects, and the corresponding XML types are association types. The associations can be graphically represented as edges between entities at the vertices. Figure 5 is a logical representation of some (**but not all**) of the associations in LEXS 3.0.



**Figure 5. Some of the Possible Associations Between Entities**

In Figure 5, the loops typically represent associations between two different instances of the same kind of entity. In other words, if person X has a relationship with person Y, then an instance of a person entity representing X will have an association with a different instance of another person entity representing Y. The loop signifies that such a relationship between two instances of a person entity is possible.

The following sections group the various associations in LEXS 3.0 that will be leveraged from NIEM. The NIEM names are used to enable a direct mapping. Not all of the associations listed below are from NIEM. Those that are not yet in NIEM have been included in a LEXS namespace. Also, not all NIEM associations are imported into LEXS 3.0 — just those believed to be of greatest utility to LEXS.

## 2.12.1 Associations Involving Persons

c:AcquaintanceAssociation  
c:ActivityInvolvedPersonAssociation  
c:AuthorityFigureAssociation  
c:BabysitterAssociation  
c:CohabitantAssociation  
c:CoworkerAssociation  
c:DomesticPartnershipAssociation  
c:FamilyAssociation  
c:FriendshipAssociation  
c:GuardianAssociation  
c:LocationNeighboringPersonAssociation  
c:MarriageAssociation  
c:NeighborAssociation  
c:NuclearFamilyAssociation  
c:OrganizationPrincipalOfficialAssociation  
c:PersonalAssociation  
c:PersonAssignedUnitAssociation  
c:PersonCurrentEmploymentAssociation  
c:PersonCurrentLocationAssociation  
c:PersonDetainmentLocationAssociation  
c:PersonEmploymentAssociation  
c:PersonEmploymentLocationAssociation  
c:PersonFormerEmploymentAssociation  
c:PersonGangAssociation  
c:PersonInvolvedInCriminalOrganizationAssociation  
c:PersonInvolvedInDrivingIncidentAssociation  
c:PersonInvolvementInActivityAssociation  
c:PersonKnownPreviousLocationAssociation  
c:PersonLastSeenLocationAssociation  
c:PersonLocationAssociation  
c:PersonOrganizationAffiliationAssociation  
c:PersonPrimaryWorkerAssociation  
c:PersonReferralWorkerAssociation  
c:PersonTemporaryAssignedUnitAssociation  
c:PersonWorkerAssociation  
c:ResidenceAssociation  
c:StrangerAssociation  
j:AccompliceAssociation  
j:ActivityAssistingPersonAssociation  
j:ActivityClearerPersonAssociation  
j:ActivityDispatcherAssociation  
j:ActivityEnforcementOfficialAssociation  
j:ActivityInformationAbstracterPersonAssociation  
j:ActivityInformationApproverAssociation  
j:ActivityInformationOwnerAssociation  
j:ActivityInformationReleaserAssociation  
j:ActivityInformationReporterAssociation  
j:ActivityJudicialOfficialAssociation  
j:ActivityUnknownAffiliateAssociation  
j:IncidentInformantAssociation  
j:IncidentInvestigatorAssociation  
j:IncidentItemObtainerAssociation  
j:IncidentUnknownAssociationPersonAssociation  
lexsdigest:ArrestOfficerAssociation  
lexsdigest:ArrestSubjectAssociation  
lexsdigest:EntityEmailAssociation  
lexsdigest:EntityEmailMessageAssociation  
lexsdigest:IncidentReportingOfficialAssociation  
lexsdigest:IncidentSubjectPersonAssociation  
lexsdigest:IncidentVictimPersonAssociation  
lexsdigest:IncidentWitnessAssociation  
lexsdigest:InmateApprovedTelephoneListAssociation  
lexsdigest:InmateTelephoneCallLogAssociation  
lexsdigest:OffenseSubjectPersonAssociation  
lexsdigest:OffenseVictimPersonAssociation  
lexsdigest:OffenseWitnessAssociation  
lexsdigest:PassengerAssociation  
lexsdigest:PersonArrestLocationAssociation  
lexsdigest:PersonBirthLocationAssociation  
lexsdigest:ServiceCallCallerAssociation  
lexsdigest:ServiceCallDispatcherAssociation  
lexsdigest:ServiceCallOperatorAssociation  
lexsdigest:SubjectVictimAssociation  
lexsdigest:SubjectWitnessAssociation  
lexsdigest:TelephoneCallAssociation  
lexsdigest:VictimWitnessAssociation

Some examples of associations involving people are shown below:

```
<lexs:PackageDigest>
  <!-- Person Arr1 -->
  <lexs:EntityPerson>
    <lexsdigest:Person s:id="Arr1">
      <u:PersonName>
        <u:PersonGivenName>Bubba</u:PersonGivenName>
        <u:PersonSurName>Johnson</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
  </lexs:EntityPerson>
  <!-- Person Vic1 -->
  <lexs:EntityPerson>
    <lexsdigest:Person s:id="Vic1">
      <u:PersonName>
        <u:PersonGivenName>Martha</u:PersonGivenName>
        <u:PersonSurName>Who</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
  </lexs:EntityPerson>
  <!-- Gang Org1 -->
  <lexs:EntityOrganization>
    <c:Organization s:id="Org1">
      <u:OrganizationName>Westside Renegades</u:OrganizationName>
      <u:OrganizationCategoryText>Gang</u:OrganizationCategoryText>
    </c:Organization>
    <j:CriminalOrganization>
      <u:RoleOfOrganizationReference s:ref="Org1"/>
    </j:CriminalOrganization>
  </lexs:EntityOrganization>
  <lexs:EntityAssociations>
    <!-- Marriage Association -->
    <c:MarriageAssociation xsi:nil="false">
      <c:PersonSpouseReference s:ref="Arr1"/>
      <c:PersonSpouseReference s:ref="Vic1"/>
    </c:MarriageAssociation>
    <!-- Person to Gang Association -->
    <c:PersonGangAssociation>
      <u:PersonReference s:ref="Arr1"/>
      <u:OrganizationReference s:ref="Org1"/>
      <c:AssociationIsMembershipIndicator>true</c:AssociationIsMembershipIndicator>
    </c:PersonGangAssociation>
    <!-- Accomplice Association -->
    <j:AccompliceAssociation xsi:nil="false">
      <u:PersonReference s:ref="Arr1"/>
      <j:SubjectReference s:ref="Vic1"/>
    </j:AccompliceAssociation>
  </lexs:EntityAssociations>
</lexs:PackageDigest>
```

## 2.12.2 Associations Involving Organizations

c:ActivityInformationAbstracterOrganizationAssociation	c:OrganizationSubsidiaryAssociation
c:ActivityInformationClearerOrganizationAssociation	c:OrganizationToOrganizationAssociation
c:ActivityInvolvedOrganizationAssociation	c:PersonAssignedUnitAssociation
c:ActivityPrimaryOrganizationAssociation	c:PersonCurrentEmploymentAssociation
c:ActivityReportingOrganizationAssociation	c:PersonEmploymentAssociation
c:ActivityResponsibleOrganizationAssociation	c:PersonFormerEmploymentAssociation
c:ActivitySupervisingOrganizationAssociation	c:PersonGangAssociation
c:LocationContainsOrganizationAssociation	c:PersonInvolvedInCriminalOrganizationAssociation
c:LocationEmergencyServicesAssociation	c:PersonOrganizationAffiliationAssociation
c:LocationOrganizationAssociation	c:PersonTemporaryAssignedUnitAssociation
c:LocationPoliceDepartmentAssociation	lexsdigest:IncidentSubjectOrganizationAssociation
c:OrganizationGangAssociation	lexsdigest:IncidentVictimOrganizationAssociation
c:OrganizationParentAssociation	lexsdigest:OffenseSubjectOrganizationAssociation
c:OrganizationPrincipalOfficialAssociation	lexsdigest:OffenseVictimOrganizationAssociation

### Examples:

```
<lexs:PackageDigest>
  <lexs:EntityPerson>          <!-- Person Arr2 -->
    <lexsdigest:Person s:id="Arr2">
      <u:PersonName>
        <u:PersonGivenName>Bubba</u:PersonGivenName>
        <u:PersonSurName>Johnson</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
  </lexs:EntityPerson>
  <lexs:EntityOrganization>    <!-- Gang Org2 -->
    <c:Organization s:id="Org2">
      <u:OrganizationName>Westside Renegades</u:OrganizationName>
      <u:OrganizationCategoryText>Gang</u:OrganizationCategoryText>
    </c:Organization>
    <j:CriminalOrganization>
      <u:RoleOfOrganizationReference s:ref="Org2"/>
    </j:CriminalOrganization>
  </lexs:EntityOrganization>
  <lexs:EntityOrganization>    <!-- Gang Org3 -->
    <c:Organization s:id="Org3">
      <u:OrganizationName>Snakes</u:OrganizationName>
      <u:OrganizationCategoryText>Gang</u:OrganizationCategoryText>
    </c:Organization>
    <j:CriminalOrganization>
      <u:RoleOfOrganizationReference s:ref="Org3"/>
    </j:CriminalOrganization>
  </lexs:EntityOrganization>
  <lexs:EntityAssociations>
    <!-- Organization to Organization Association -->
    <c:OrganizationSubsidiaryAssociation xsi:nil="false">
      <u:AssociationBeginDate xsi:nil="false">2005-11-18</u:AssociationBeginDate>
      <u:OrganizationReference s:ref="Org2"/>
      <u:AssociatedOrganizationReference s:ref="Org3"/>
    </c:OrganizationSubsidiaryAssociation>
    <!-- Person to Gang Association -->
    <c:PersonGangAssociation>
```

```
<u:PersonReference s:ref="Arr2"/>  
<u:OrganizationReference s:ref="Org2"/>  
<c:AssociationIsMembershipIndicator>true</c:AssociationIsMembershipIndicator>  
</c:PersonGangAssociation>  
</lexs:EntityAssociations>  
</lexs:PackageDigest>
```

## 2.12.3 Associations Involving Locations

c:LocationContainsOrganizationAssociation  
c:LocationEmergencyServicesAssociation  
c:LocationNeighboringPersonAssociation  
c:LocationOrganizationAssociation  
c:LocationPoliceDepartmentAssociation  
c:PersonCurrentLocationAssociation  
c:PersonDetainmentLocationAssociation  
c:PersonEmploymentLocationAssociation  
c:PersonKnownPreviousLocationAssociation

c:PersonLastSeenLocationAssociation  
c:PersonLocationAssociation  
c:PropertyCurrentLocationAssociation  
c:PropertyDispositionLocationAssociation  
c:PropertyLocationAssociation  
c:VehicleGarageLocationAssociation  
lexsdigest:PersonArrestLocationAssociation  
lexsdigest:PersonBirthLocationAssociation  
scr:LocationAssociation

### Examples:

```
<lexs:PackageDigest>
  <!-- Person Arr3 -->
  <lexs:EntityPerson>
    <lexsdigest:Person s:id="Arr3">
      <u:PersonName>
        <u:PersonGivenName>Bubba</u:PersonGivenName>
        <u:PersonSurName>Johnson</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
  </lexs:EntityPerson>
  <!-- Gang Org5 -->
  <lexs:EntityOrganization>
    <c:Organization s:id="Org5">
      <u:OrganizationName>Snakes</u:OrganizationName>
      <u:OrganizationCategoryText>Gang</u:OrganizationCategoryText>
    </c:Organization>
    <j:CriminalOrganization>
      <u:RoleOfOrganizationReference s:ref="Org5"/>
    </j:CriminalOrganization>
  </lexs:EntityOrganization>
  <!--===== Location 1 =====>
  <lexs:EntityLocation>
    <u:Location s:id="Loc1">
      <u:LocationAddress>
        <u:LocationStreet>
          <u:StreetFullText>12 Lucifer Street</u:StreetFullText>
        </u:LocationStreet>
        <u:LocationSecondaryUnitText>Apt. 2234</u:LocationSecondaryUnitText>
        <u:LocationCityName>Inferno</u:LocationCityName>
        <u:LocationStateName>TX</u:LocationStateName>
        <u:LocationPostalCodeID>
          <u:ID>33445</u:ID>
        </u:LocationPostalCodeID>
      </u:LocationAddress>
    </u:Location>
  </lexs:EntityLocation>
  <!--===== Location 2 =====>
  <lexs:EntityLocation>
    <u:Location s:id="Loc2">
      <u:LocationAddress>
```



```
<u:LocationStreet>
  <u:StreetFullText>1 Any St.</u:StreetFullText>
</u:LocationStreet>
<u:LocationSecondaryUnitText>Apt. 16</u:LocationSecondaryUnitText>
<u:LocationCityName>Clay</u:LocationCityName>
<u:LocationStateName>WV</u:LocationStateName>
<u:LocationPostalCodeID>
  <u:ID>12345</u:ID>
</u:LocationPostalCodeID>
</u:LocationAddress>
</u:Location>
</lexs:EntityLocation>
<lexs:EntityAssociations>
  <!-- Organization Mailing Address Association -->
  <c:LocationOrganizationAssociation>
    <u:AssociationBeginDate>1990-10-31</u:AssociationBeginDate>
    <u:CommentText>Mailing Address</u:CommentText>
    <u:LocationReference s:ref="Loc1"/>
    <u:OrganizationReference s:ref="Org5"/>
  </c:LocationOrganizationAssociation>
  <!-- Person to Location Association -->
  <c:PersonLocationAssociation>
    <u:AssociationBeginDate>1999-07-15</u:AssociationBeginDate>
    <u:PersonReference s:ref="Arr3"/>
    <u:LocationReference s:ref="Loc1"/>
  </c:PersonLocationAssociation>
</lexs:EntityAssociations>
</lexs:PackageDigest>
```

## 2.12.4 Associations Involving Property

c:EntityPropertyAssociation	lexsdigest:ArrestInvolvedWeaponAssociation
c:PersonVehicleAssociation	lexsdigest:DriverAssociation
c:PropertyCurrentLocationAssociation	lexsdigest:IncidentInvolvedPropertyAssociation
c:PropertyDispositionLocationAssociation	lexsdigest:IncidentWeaponAssociation
c:PropertyHolderAssociation	lexsdigest:OffenseInvolvedPropertyAssociation
c:PropertyLocationAssociation	lexsdigest:OffenseWeaponAssociation
c:PropertyMoverAssociation	lexsdigest:PassengerAssociation
c:VehicleGarageLocationAssociation	lexsdigest:PropertyLienHolderAssociation
c:VehicleTowerAssociation	

### Examples:

```
<lexs:PackageDigest>
  <!-- Person Arr3 -->
  <lexs:EntityPerson>
    <lexsdigest:Person s:id="Arr3">
      <u:PersonName>
        <u:PersonGivenName>Bubba</u:PersonGivenName>
        <u:PersonSurName>Johnson</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
  </lexs:EntityPerson>
  <!-- Gang Org5 -->
```

```
<lexs:EntityOrganization>
  <c:Organization s:id="Org5">
    <u:OrganizationName>Snakes</u:OrganizationName>
    <u:OrganizationCategoryText>Gang</u:OrganizationCategoryText>
  </c:Organization>
</lexs:EntityOrganization>
<!--===== Property 3 =====-->
<lexs:EntityProperty>
  <c:Property s:id="Prop3">
    <u:PropertyDescriptionText>Cellular Phone</u:PropertyDescriptionText>
    <c:PropertySerialID>
      <u:ID>BO1234RTZ012</u:ID>
    </c:PropertySerialID>
  </c:Property>
</lexs:EntityProperty>
<!--===== Vehicle =====-->
<lexs:EntityVehicle>
  <c:Vehicle s:id="Veh1">
    <u:PropertyDescriptionText>350 GT Convertible</u:PropertyDescriptionText>
    <c:VehicleStyleCode>CV</c:VehicleStyleCode>
    <c:VehicleID>
      <u:ID>1234L7T098356A</u:ID>
    </c:VehicleID>
    <c:VehicleLicensePlateID>
      <u:ID>HOWUDOIN</u:ID>
      <j:IDJurisdictionNCICLISCode>NC</j:IDJurisdictionNCICLISCode>
    </c:VehicleLicensePlateID>
    <c:VehicleModelYearDate>1978</c:VehicleModelYearDate>
    <c:VehicleColorPrimaryCode>RED</c:VehicleColorPrimaryCode>
    <c:VehicleMakeCode>FORD</c:VehicleMakeCode>
    <c:VehicleModelCode>MUS</c:VehicleModelCode>
  </c:Vehicle>
</lexs:EntityVehicle>
<!--===== Firearm Frm1 =====-->
<lexs:EntityFirearm>
  <c:Firearm s:id="Frm1">
    <u:PropertyDescriptionText>Weapon Description. Big Gun.</u:PropertyDescriptionText>
    <c:PropertySerialID>
      <u:ID>123ABC4578</u:ID>
    </c:PropertySerialID>
    <c:FirearmMakeCode>NOM</c:FirearmMakeCode>
    <c:FirearmCaliberCode>9</c:FirearmCaliberCode>
  </c:Firearm>
  <c:Weapon>
    <u:RoleOfPropertyReference s:ref="Frm1"/>
    <c:WeaponUserReference s:ref="Arr3"/>
  </c:Weapon>
</lexs:EntityFirearm>
<lexs:EntityAssociations>
  <!-- Property Lien Holder Association -->
  <c:EntityPropertyAssociation>
    <u:CommentText>Lien Holder</u:CommentText>
    <u:OrganizationReference s:ref="Org5"/>
    <u:PropertyReference s:ref="Veh1"/>
  </c:EntityPropertyAssociation>
  <!-- Property to Person Association -->
```

```
<c:EntityPropertyAssociation>
  <u:AssociationBeginDate>1999-07-20</u:AssociationBeginDate>
  <u:PersonReference s:ref="Arr3"/>
  <u:PropertyReference s:ref="Prop3"/>
</c:EntityPropertyAssociation>
<!-- Property to Person Association -->
<c:EntityPropertyAssociation>
  <u:CommentText xsi:nil="false">Subject arrested with weapon</u:CommentText>
  <u:PersonReference s:ref="Arr3"/>
  <u:PropertyReference s:ref="Frm1"/>
</c:EntityPropertyAssociation>
</lexs:EntityAssociations>
</lexs:PackageDigest>
```

## 2.12.5 Associations Involving Activities

c:ActivityInformationAbstracterOrganizationAssociation	lexsdigest:ArrestOffenseAssociation
c:ActivityInformationClearerOrganizationAssociation	lexsdigest:ArrestOfficerAssociation
c:ActivityInvolvedOrganizationAssociation	lexsdigest:ArrestSubjectAssociation
c:ActivityInvolvedPersonAssociation	lexsdigest:IncidentArrestAssociation
c:ActivityPrimaryOrganizationAssociation	lexsdigest:IncidentInvolvedPropertyAssociation
c:ActivityReportingOrganizationAssociation	lexsdigest:IncidentLocationAssociation
c:ActivityResponsibleOrganizationAssociation	lexsdigest:IncidentOffenseAssociation
c:ActivitySupervisingOrganizationAssociation	lexsdigest:IncidentReportingOfficialAssociation
c:PersonInvolvedInDrivingIncidentAssociation	lexsdigest:IncidentServiceCallAssociation
c:PersonInvolvementInActivityAssociation	lexsdigest:IncidentSubjectOrganizationAssociation
c:PreviousActivityAssociation	lexsdigest:IncidentSubjectPersonAssociation
c:RelatedActivityAssociation	lexsdigest:IncidentVictimOrganizationAssociation
c:RelatedCaseAssociation	lexsdigest:IncidentVictimPersonAssociation
j:ActivityAssistingPersonAssociation	lexsdigest:IncidentWeaponAssociation
j:ActivityClearerPersonAssociation	lexsdigest:IncidentWitnessAssociation
j:ActivityDispatcherAssociation	lexsdigest:OffenseInvolvedPropertyAssociation
j:ActivityEnforcementOfficialAssociation	lexsdigest:OffenseSubjectOrganizationAssociation
j:ActivityInformationAbstracterPersonAssociation	lexsdigest:OffenseSubjectPersonAssociation
j:ActivityInformationApproverAssociation	lexsdigest:OffenseVictimOrganizationAssociation
j:ActivityInformationOwnerAssociation	lexsdigest:OffenseVictimPersonAssociation
j:ActivityInformationReleaserAssociation	lexsdigest:OffenseWeaponAssociation
j:ActivityInformationReporterAssociation	lexsdigest:OffenseWitnessAssociation
j:ActivityJudicialOfficialAssociation	lexsdigest:ServiceCallCallerAssociation
j:ActivityUnknownAffiliateAssociation	lexsdigest:ServiceCallDispatcherAssociation
j:IncidentInformantAssociation	lexsdigest:ServiceCallOperatorAssociation
j:IncidentInvestigatorAssociation	lexsdigest:SubjectVictimAssociation
j:IncidentItemObtainerAssociation	lexsdigest:SubjectWitnessAssociation
j:IncidentUnknownAssociationPersonAssociation	lexsdigest:VictimWitnessAssociation
lexsdigest:ArrestInvolvedWeaponAssociation	

## Examples:

```
<lexs:PackageDigest>
  <!-- Person Arr3 -->
  <lexs:EntityPerson>
    <lexsdigest:Person s:id="Arr3">
      <u:PersonName>
        <u:PersonGivenName>Bubba</u:PersonGivenName>
        <u:PersonSurName>Johnson</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
  </lexs:EntityPerson>
  <!--===== Person with Witness Role =====-->
  <lexs:EntityPerson>
    <lexsdigest:Person s:id="Wit1">
      <u:PersonName>
        <u:PersonGivenName>Joe</u:PersonGivenName>
        <u:PersonSurName>Schmoe</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
    <j:IncidentWitness>
      <u:RoleOfPersonReference s:ref="Wit1"/>
    </j:IncidentWitness>
  </lexs:EntityPerson>
  <!--===== Organization 2 =====-->
  <lexs:EntityOrganization>
    <c:Organization s:id="Org7">
      <u:OrganizationName>My Agency</u:OrganizationName>
      <u:OrganizationCategoryText>Law Enforcement</u:OrganizationCategoryText>
    </c:Organization>
  </lexs:EntityOrganization>
  <!--===== Person with Enforcement Official and Victim Roles =====-->
  <lexs:EntityPerson>
    <lexsdigest:Person s:id="Vic3">
      <u:PersonName>
        <u:PersonGivenName>Mary</u:PersonGivenName>
        <u:PersonSurName>Ondera</u:PersonSurName>
      </u:PersonName>
    </lexsdigest:Person>
    <j:EnforcementOfficial>
      <u:RoleOfPersonReference s:ref="Vic3"/>
      <j:EnforcementOfficialUnit>
        <u:OrganizationName>Atlanta PD</u:OrganizationName>
        <u:OrganizationCategoryText xsi:nil="true"/>
        <j:EnforcementUnitID>
          <u:ID>AGENCY3700</u:ID>
        </j:EnforcementUnitID>
      </j:EnforcementOfficialUnit>
    </j:EnforcementOfficial>
    <j:IncidentVictim>
      <u:RoleOfPersonReference s:ref="Vic3"/>
    </j:IncidentVictim>
  </lexs:EntityPerson>
  <!--===== Person with Enforcement Official Role =====-->
  <lexs:EntityPerson>
```

```

<lexsdigest:Person s:id="Off1">
  <u:PersonName>
    <u:PersonGivenName>Bob</u:PersonGivenName>
    <u:PersonSurName>Evans</u:PersonSurName>
  </u:PersonName>
</lexsdigest:Person>
<j:EnforcementOfficial>
  <u:RoleOfPersonReference s:ref="Off1"/>
  <j:EnforcementOfficialUnit>
    <u:OrganizationName>Marietta PD</u:OrganizationName>
    <u:OrganizationCategoryText xsi:nil="true"/>
    <j:EnforcementUnitID>
      <u:ID>AGENCY3700</u:ID>
    </j:EnforcementUnitID>
  </j:EnforcementOfficialUnit>
</j:EnforcementOfficial>
</lexs:EntityPerson>
<!-- Gang Org5 -->
<lexs:EntityOrganization>
  <c:Organization s:id="Org5">
    <u:OrganizationName>Snakes</u:OrganizationName>
    <u:OrganizationCategoryText>Gang</u:OrganizationCategoryText>
  </c:Organization>
</lexs:EntityOrganization>
<!--===== Activity Incident =====>
<lexs:EntityActivity>
  <u:Activity s:id="Incident1">
    <u:ActivityID>
      <u:ID>Inc123456</u:ID>
    </u:ActivityID>
    <u:ActivityCategoryText>Incident</u:ActivityCategoryText>
    <u:ActivityDescriptionText>
      The intoxicated subject became angry about a parking space. After shouting profanities, he struck
      the victim and proceeded to puncture the tires of the victim's vehicle.
    </u:ActivityDescriptionText>
    <u:ActivityDate>2000-08-23</u:ActivityDate>
    <u:ActivityTime>14:31:00</u:ActivityTime>
    <u:ActivityEndDate>2000-08-23</u:ActivityEndDate>
    <u:ActivityEndTime>16:22:00</u:ActivityEndTime>
  </u:Activity>
</lexs:EntityActivity>
<!--===== Activity Arrest 1 =====>
<lexs:EntityActivity>
  <u:Activity s:id="Arrest1">
    <u:ActivityCategoryText>Arrest</u:ActivityCategoryText>
    <u:ActivityDate>2001-01-25</u:ActivityDate>
  </u:Activity>
</lexs:EntityActivity>
<!--===== Person with no specific Role =====>
<lexs:EntityPerson>
  <lexsdigest:Person s:id="Per5">
    <u:PersonName>
      <u:PersonGivenName>John</u:PersonGivenName>
      <u:PersonSurName>Jacobs</u:PersonSurName>
    </u:PersonName>
  </lexsdigest:Person>

```

```
</lexs:EntityPerson>
<lexs:EntityAssociations>
  <c:ActivityInvolvedOrganizationAssociation>
    <u:CommentText>Responding Organization</u:CommentText>
    <u:ActivityReference s:ref="Incident1"/>
    <u:OrganizationReference s:ref="Org7"/>
  </c:ActivityInvolvedOrganizationAssociation>
  <!-- Arrestee to Arrest Association -->
  <c:ActivityInvolvedPersonAssociation>
    <u:CommentText>Arrest</u:CommentText>
    <u:ActivityReference s:ref="Arrest1"/>
    <u:PersonReference s:ref="Arr3"/>
  </c:ActivityInvolvedPersonAssociation>
  <!-- Incident Subject Association -->
  <c:ActivityInvolvedPersonAssociation>
    <u:CommentText>Incident Subject</u:CommentText>
    <u:ActivityReference s:ref="Incident1"/>
    <u:PersonReference s:ref="Arr3"/>
  </c:ActivityInvolvedPersonAssociation>
  <!-- Incident Responding Official Association -->
  <c:ActivityInvolvedPersonAssociation>
    <u:CommentText>Responding Official</u:CommentText>
    <u:ActivityReference s:ref="Incident1"/>
    <u:PersonReference s:ref="Off1"/>
  </c:ActivityInvolvedPersonAssociation>
  <!-- Incident Witness Association -->
  <c:ActivityInvolvedPersonAssociation>
    <u:CommentText>Witness</u:CommentText>
    <u:ActivityReference s:ref="Incident1"/>
    <u:PersonReference s:ref="Wit1"/>
  </c:ActivityInvolvedPersonAssociation>
  <!-- Incident Victim Association -->
  <c:ActivityInvolvedPersonAssociation>
    <u:CommentText>Victim</u:CommentText>
    <u:ActivityReference s:ref="Incident1"/>
    <u:PersonReference s:ref="Vic3"/>
  </c:ActivityInvolvedPersonAssociation>
  <!-- Incident Other Involved Person Association -->
  <c:ActivityInvolvedPersonAssociation>
    <u:CommentText>Other Involved Person</u:CommentText>
    <u:ActivityReference s:ref="Incident1"/>
    <u:PersonReference s:ref="Per5"/>
  </c:ActivityInvolvedPersonAssociation>
</lexs:EntityAssociations>
</lexs:PackageDigest>
```

### **2.12.6 Associations Involving Communications**

lexsdigest:EmailMessageAssociation

lexsdigest:EntityEmailAssociation

lexsdigest:EntityTelephoneNumberAssociation

lexsdigest:InmateApprovedTelephoneListAssociation

lexsdigest:InmateTelephoneCallLogAssociation

lexsdigest:TelephoneCallAssociation

### **2.12.7 Associations Involving Package Structure**

The following associations are related to linking elements to other elements or attachments which maybe inside or outside a package and are described in greater detail in sections that cover these topics.

lexs:EntityAttachmentLinkAssociation

lexs:EntityPersonImageAssociation

lexslib:SameAsPayloadAssociation

## **2.13 Attachments**

Figure 6 shows the structure for attachments. The actual attachment is identified by the content of “lexs:AttachmentURI”. In this example it is <http://www.attach.1.jpg>. The “lexs:AttachmentURI” must follow all rules for an URI. The attachment may be submitted as part of a message or may have been sent out of band through another channel. If carried as part of the message, the attachment is contained in the message element. Multiple packages can refer to the same attachment. Each package that refers to the attachment contains a “lexs:AttachmentLink” element inside the “lexs:Package” element. The “lexs:AttachmentLink” is identified by an “s:id” attribute if an “lexs:EntityAttachmentLinkAssociation” will reference this “lexs:AttachmentLink”. The “lexs:EntityAssociations” element in the “lexs:PackageDigest” element for the containing package connects the attachment to the entity that it is associated with through “s:ref” references.

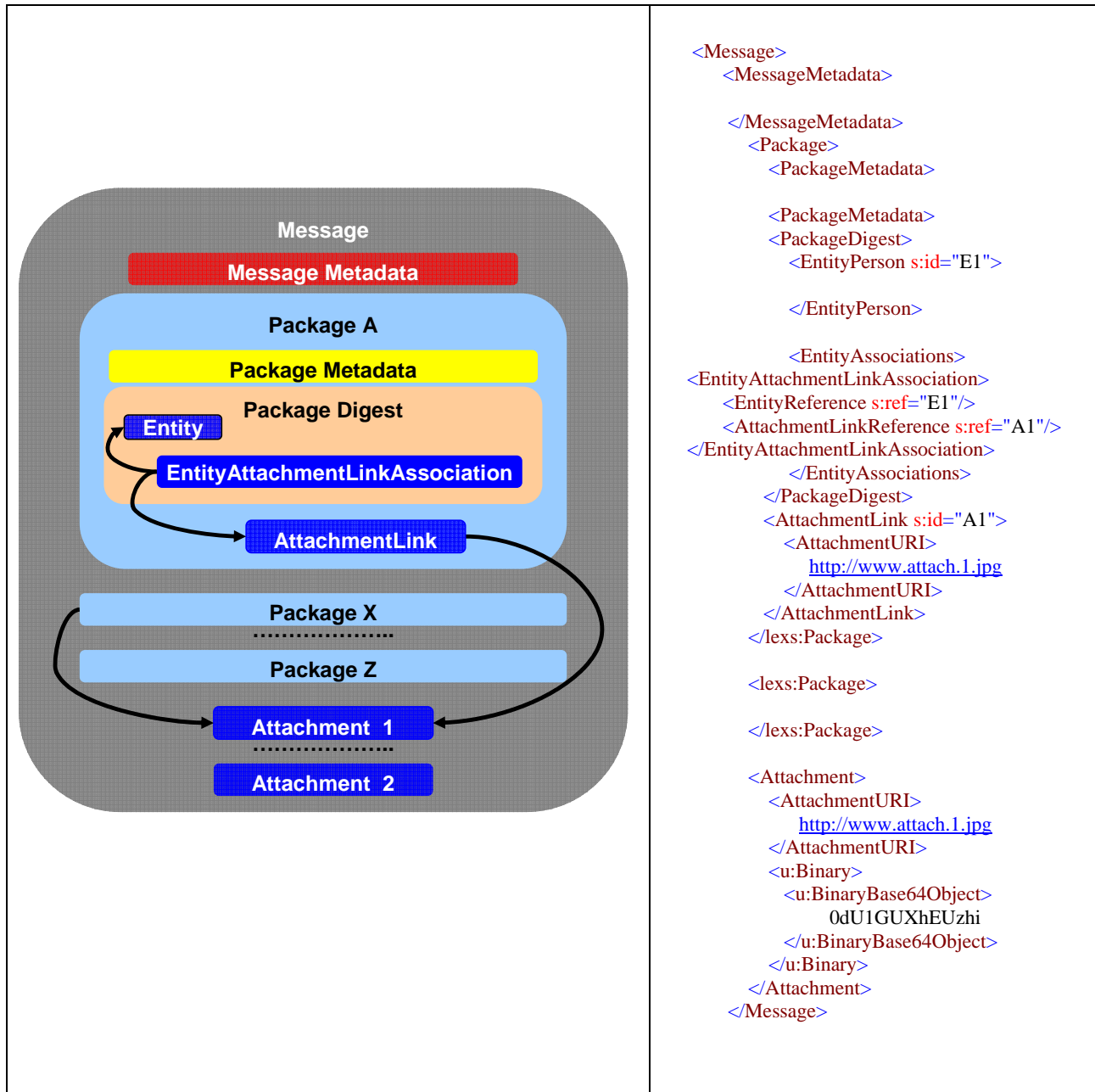


Figure 6. LEXS Structure for Attachments

The detailed structure that corresponds to Figure 6 is shown below:



```
<lexs:Message>
  <lexs:MessageMetadata> ..... </lexs:MessageMetadata>

  <!-- A message may include one or more packages -->
  <lexs:Package>
    <lexs:PackageMetadata> ..... </lexs:PackageMetadata>
    <lexs:PackageDigest>
      <lexs:EntityPerson s:id="E1">
        <lexsdigest:Person>
          <u:PersonName>
            <u:PersonGivenName>Bubba</u:PersonGivenName>
            <u:PersonSurName>Johnson</u:PersonSurName>
          </u:PersonName>
        </lexsdigest:Person>
      </lexs:EntityPerson>
      <!--===== Associations =====-->
      <lexs:EntityAssociations>
        <!-- Attachment for Entity, in this case a fingerprint for a Person -->
        <lexs:EntityAttachmentLinkAssociation>
          <lexs:EntityReference s:ref="E1"/>
          <lexs:AttachmentLinkReference s:ref="A1"/>
        </lexs:EntityAttachmentLinkAssociation>
      </lexs:EntityAssociations>
    </lexs:PackageDigest>
    <!--===== Package Attachment Link for Fingerprint =====-->
    <lexs:AttachmentLink s:id="A1">
      <lexs:AttachmentURI>http://www.attach.1.jpg</lexs:AttachmentURI>
      <lexs:AttachmentDescriptionText>Fingerprint for Bubba Johnson</lexs:AttachmentDescriptionText>
      <lexs:AttachmentViewableIndicator>true</lexs:AttachmentViewableIndicator>
    </lexs:AttachmentLink>
  </lexs:Package>

  <lexs:Package>
    <lexs:PackageMetadata> ..... </lexs:PackageMetadata>
    <lexs:PackageDigest>..... </lexs:PackageDigest>
  </lexs:Package>

  <!--===== Message Attachment containing Fingerprint =====-->
  <lexs:Attachment>
    <lexs:AttachmentURI>http://www.attach.1.jpg</lexs:AttachmentURI>
    <u:Binary>
      <u:BinaryBase64Object>
        0dU1GUXhEUzhihbXA7bmJzcDs8L3hzbDp0ZXh0Pg0KICAgICAgPC94c2w6b3RoZX
        J3aXNIPg0KICAgIDwveHNsOmNob29zZT4NCiAgPC94c2w6dGVtcGxhdGU+DQoN2
      </u:BinaryBase64Object>
      <u:BinaryDescriptionText>Fingerprint for Bubba Johnson</u:BinaryDescriptionText>
      <u:BinaryFormatText>image/jpg</u:BinaryFormatText>
      <u:BinaryFormatCategoryText>MIME</u:BinaryFormatCategoryText>
      <u:BinaryCategoryText>Fingerprint</u:BinaryCategoryText>
    </u:Binary>
  </lexs:Attachment>
</lexs:Message>
```

While Figure 6 and the example shown above illustrate the structure for an attachment (e.g., fingerprint) that is associated with an entity (e.g., suspect), it is not always necessary for an

attachment to be associated with an entity in this manner. When an attachment link (“lexs:AttachmentLink”) is not associated to an entity, then the “lexs:EntityAttachmentLinkAssociation” does not exist and the attachment is (assumed to be) associated in an unspecified way with unspecified parts of the package. While not very semantically helpful to a computer, this still allows the attachment to be displayed to the user in the context of the data item, and its relevance may be obvious to a human. Even when the “lexs:EntityAttachmentLinkAssociation” is present, the attachment is implicitly associated with the package, however the explicit association with an entity provides more precise detail. The structure for such an instance is shown below:

```
<lexs:Message>
  <lexs:MessageMetadata> ..... </lexs:MessageMetadata>

  <!-- A message may include one or more packages -->
  <lexs:Package>
    <lexs:PackageMetadata> ..... </lexs:PackageMetadata>
    <lexs:PackageDigest>
      <lexs:EntityPerson s:id="E1">
        <lexsdigest:Person>
          <u:PersonName>
            <u:PersonGivenName>Bubba</u:PersonGivenName>
            <u:PersonSurName>Johnson</u:PersonSurName>
          </u:PersonName>
        </lexsdigest:Person>
      </lexs:EntityPerson>
    </lexs:PackageDigest>
    <!--===== Package Attachment Link for a spreadsheet =====>
    <lexs:AttachmentLink s:id="A1">
      <lexs:AttachmentURI>http://www.attach3.xls</lexs:AttachmentURI>
      <lexs:AttachmentDescriptionText>Loans Defaulted: B Johnson</lexs:AttachmentDescriptionText>
      <lexs:AttachmentViewableIndicator>true</lexs:AttachmentViewableIndicator>
    </lexs:AttachmentLink>
  </lexs:Package>

  <lexs:Package>
    <lexs:PackageMetadata> ..... </lexs:PackageMetadata>
    <lexs:PackageDigest>..... </lexs:PackageDigest>
  </lexs:Package>

  <!--===== Message Attachment containing spreadsheet =====>
  <lexs:Attachment>
    <lexs:AttachmentURI>http://www.attach3.xls</lexs:AttachmentURI>
    <u:Binary>
      <u:BinaryBase64Object>
        0dU1GUXhEUzhihbXA7bmJzcDs8L3hzbDp0ZXh0Pg0KICAgICAgPC94c2w6b3RoZX
        J3aXNlPg0KICAgIDwveHNsOmNob29zZT4NCiAgPC94c2w6dGVtcGxhdGU+DQoN2
      </u:BinaryBase64Object>
      <u:BinaryDescriptionText> Balance Sheet for XYZ Corp</u:BinaryDescriptionText>
      <u:BinaryFormatText>application/vnd.ms-excel</u:BinaryFormatText>
      <u:BinaryFormatCategoryText>MIME</u:BinaryFormatCategoryText>
      <u:BinaryCategoryText>Spreadsheet</u:BinaryCategoryText>
    </u:Binary>
  </lexs:Attachment>
</lexs:Message>
```

## 2.14 Rendering Instructions

Rendering instructions are used to display the information in a package in a specific viewing or output format for human users. A data source may want to influence the way in which the information it shares is viewed by a user at a consuming system. Using a specific rendering allows a source to more intuitively present data items to users who may not be familiar with the underlying context of the data. This is especially true when the package contains information that is specific to an agency or community (payload) and the data source is knowledgeable about how best to format the content to make it more comprehensible and user friendly. Since LEXS allows for the sharing of heterogeneous data from different sources with different contexts, user interfaces for displaying LEXS data will often need to be very generic, sacrificing intuitiveness.

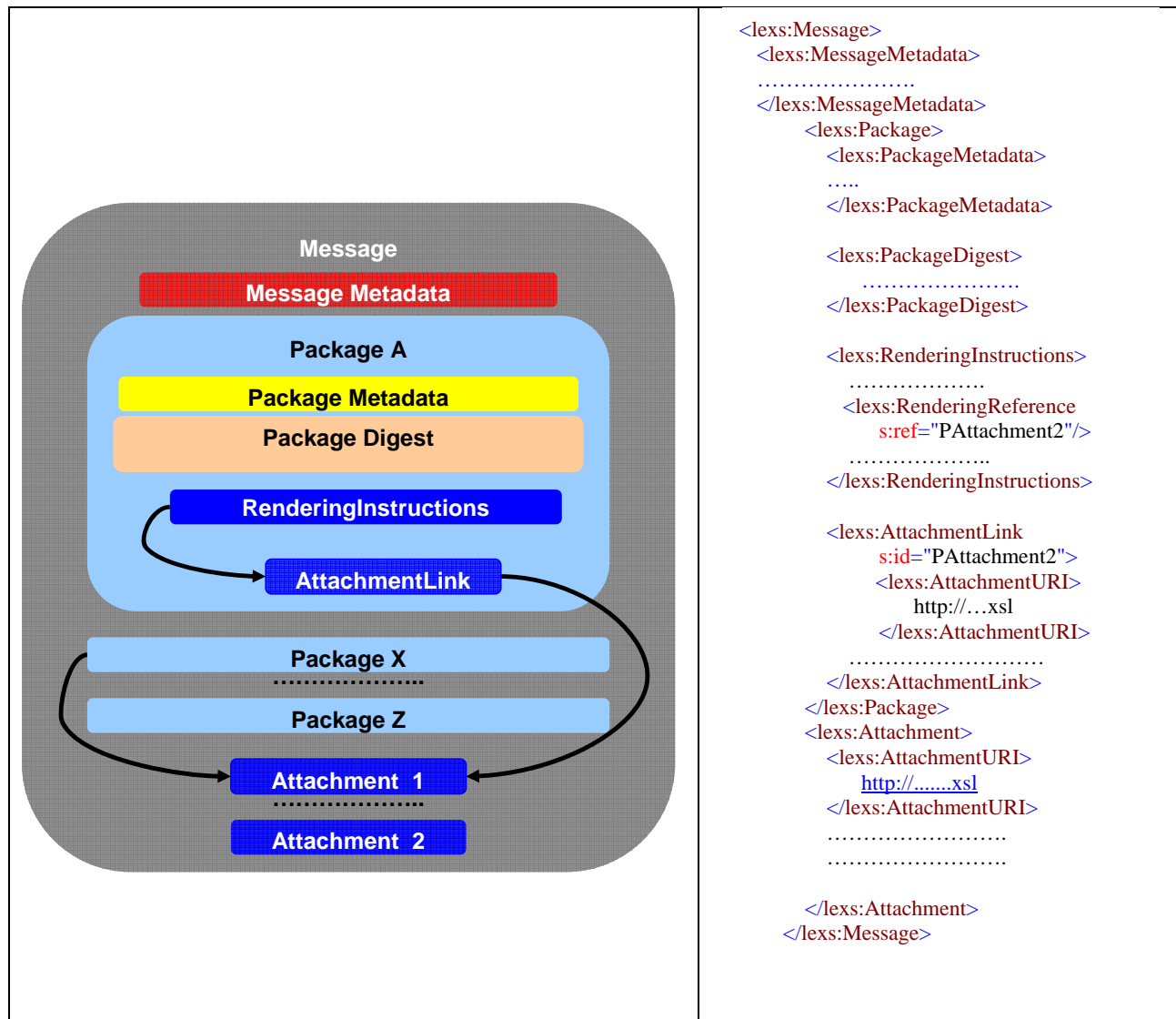


Figure 7. LEXS Structure for Rendering Instructions

Rendering instructions come in two general forms: instructions (i.e., XSL) for converting the XML data in the package into a suitable display format, and instructions to display a pre-rendered version of the data item (i.e., an attached document). The HTML rendering format is

frequently the choice for viewing reports online and other rendering formats, such as Adobe Acrobat Portable Document Format (.pdf) files are ideal for distributing large, page-oriented reports that can be sent to a printer (assuming of course, that printing is authorized) or viewed using an Acrobat Reader.

Figure 7 shows the structure for rendering instructions. The rendering instructions element (`lexs:RenderingInstructions`), is an optional element within the package and contains a pointer (`s:ref`) to a `lexs:AttachmentLink`. (see Section 2.13). The rest of the structure is similar to the structure for an attachment that is not attached to an entity but attached only to the digest. When the content is pre-rendered, the attachment is the pre-rendered document. Rendering instructions can also be communicated by including a stylesheet that applies a transformation to the entire package (including any payload). In this case the stylesheet containing the rendering instructions is identified by the content of the `lexs:AttachmentURI` element. In this example it is `http://www.gtri.org/LEXS/examples/XMLViewer.xsl`. As before, the AttachmentURI must follow all rules for a URI. Like any other attachment, the rendering instructions may be submitted as part of a message or may be sent out-of-band through another channel. If carried as part of the message, the attachment containing the rendering instructions is contained in the message element. Multiple packages can refer to the same rendering instructions. Each package that refers to the attachment containing the rendering instructions contains a `lexs:AttachmentLink` element inside the `lexs:Package` element. The `lexs:AttachmentLink` is identified by an `s:id` attribute.

The detailed structure that corresponds to Figure 7 is shown below:

```
<lexs:Message>
  <lexs:MessageMetadata> ..... </lexs:MessageMetadata>
  <lexs:Package>
    <lexs:PackageMetadata> ..... </lexs:PackageMetadata>
    <lexs:PackageDigest>..... </lexs:PackageDigest>
    <!--===== Rendering Instructions =====-->
    <lexs:RenderingInstructions>
      <lexs:RenderingMethod>HTMLStyleSheet</lexs:RenderingMethod>
      <lexs:RenderingReference s:ref="PAttachment2"/>
      <lexs:RenderingDescription>Pretty prints a message</lexs:RenderingDescription>
    </lexs:RenderingInstructions>
    <!--===== Attachment Link for Stylesheet used for Rendering =====-->
    <lexs:AttachmentLink s:id="PAttachment2">
      <lexs:AttachmentURI>http://www.gtri.org/LEXS/examples/XMLViewer.xsl</lexs:AttachmentURI>
      <lexs:AttachmentDescriptionText>XML Stylesheet for viewing </lexs:AttachmentDescriptionText>
      <lexs:AttachmentViewableIndicator>>false</lexs:AttachmentViewableIndicator>
    </lexs:AttachmentLink>
  </lexs:Package>
  <!--===== Attachment containing Stylesheet =====-->
  <lexs:Attachment>
    <lexs:AttachmentURI>http://www.gtri.org/LEXS/examples/XMLViewer.xsl</lexs:AttachmentURI>
    <u:Binary>
      <u:BinaryBase64Object>UUNBRU1tQ1p0dU1GUXhEUzhi</u:BinaryBase64Object>
      <u:BinaryDescriptionText>XSL Stylesheet for viewing content</u:BinaryDescriptionText>
      <u:BinaryFormatText>application/xml+xslt</u:BinaryFormatText>
      <u:BinaryFormatCategoryText>MIME</u:BinaryFormatCategoryText>
      <u:BinaryCategoryText>Stylesheet</u:BinaryCategoryText>
    </u:Binary>
  </lexs:Attachment>
</lexs:Message>
```

```
</u:Binary>  
</lexs:Attachment>  
</lexs:Message>
```

The following is a list of the rendering methods supported in LEXS-PD 3.0.

### 2.14.1 XHTML Stylesheet

This rendering method refers to an XSL (eXtensible Stylesheet Language) stylesheet provided by a data source that transforms a package from XML into XHTML.

### 2.14.2 HTML Stylesheet

This rendering method refers to an XSL stylesheet provided by a data source that transforms a package from XML into HTML.

### 2.14.3 PDFStylesheet

This rendering method refers to an XSL stylesheet provided by a data source that transforms a package from XML into PDF.

### 2.14.4 Text Stylesheet

This rendering method refers to an XSL stylesheet provided by a data source that transforms a package from XML into text.

### 2.14.5 Rendered Binary

This rendering method refers to an attachment that was pre-rendered into a binary (e.g., a PDF file). No XSL stylesheet is used in this case; the “lexs:RenderingReference” points to the attachment link for the pre-rendered binary attachment.

### 2.14.6 Narrative

This rendering method refers to an attachment that was pre-rendered into unstructured narrative text. The actual content included here may or may not be identical to the content of the “lexs:narrative” element in the package. Again, no XSL stylesheet is used here.

### 2.14.7 Other Stylesheet

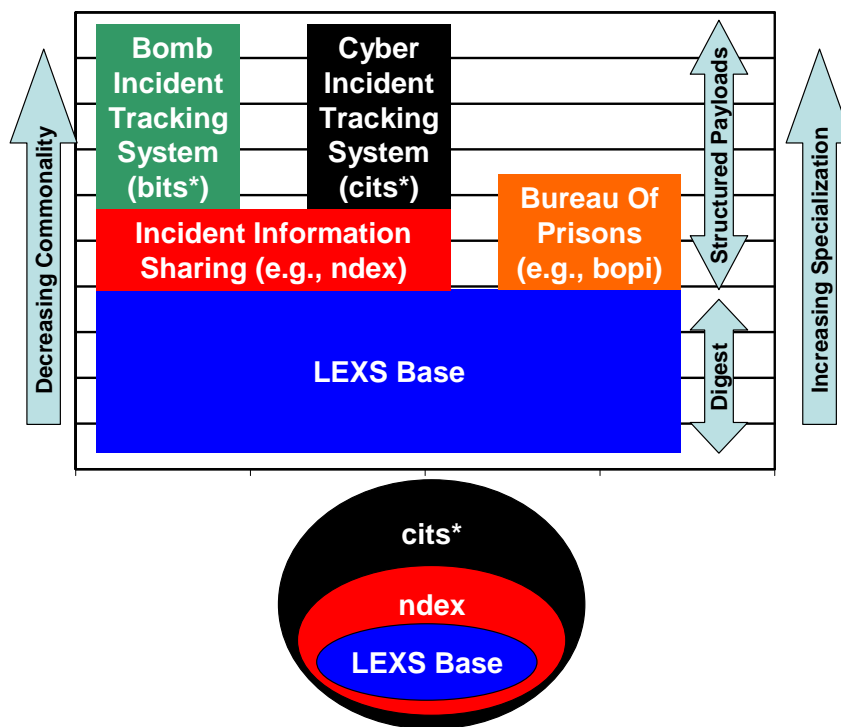
This rendering method refers to a stylesheet provided by a data source that transforms a package from XML into a format not listed. However, the stylesheet itself is expected to use XSLT.



**Tip:** Though consumers may normally ingest a package and store the content of a package within their own database they must also retain the raw XML as received from a data source in order to properly interpret rendering instructions that may be communicated via a stylesheet. In particular, stylesheets may reference content within structured payloads that are not parsed by a particular data consumer.

## 2.15 Structured Payload and Levels of Understanding

LEXS 3.0 provides a model where the “lexs:PackageDigest” allows data sources to encode a well-defined representation of people, places, activities, and things as well as associations among them as described previously in this document. LEXS 3.0 also includes a layered mechanism for programs (e.g., information sharing communities, projects, systems) to define entities, roles, associations, structures and elements that are not defined in the LEXS 3.0 “schema”. Programs that need additional data not provided in the digest can supply that data in one or more Structured Payloads. This allows law enforcement groups or projects who are interested in information sharing to leverage LEXS as a base while developing their own specialized schemas targeted to address their own business missions. For example, N-DEx has defined an Information Exchange Package (IEP) based on LEXS 3.0.



\*Hypothetical Sub-Communities

**Figure 8. Specialized Systems and/or Communities Leveraging the Structured Payload Element**

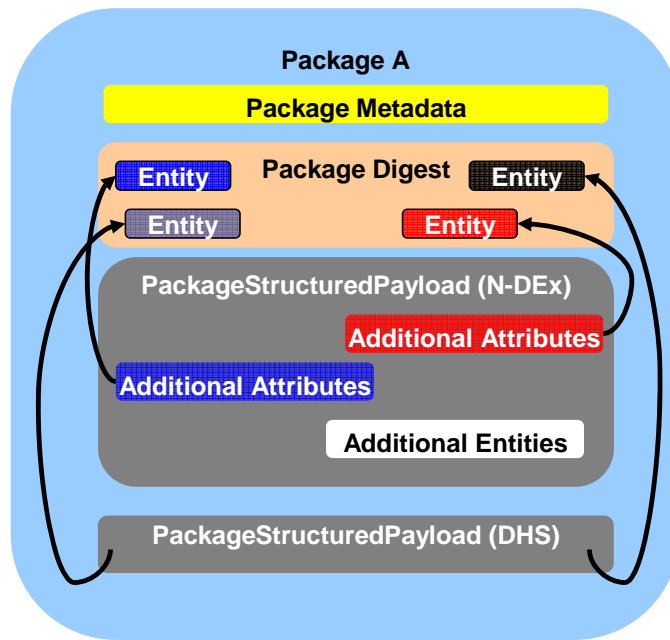
Users of a consuming system might still benefit from the parts of the message not understood or processed by their system if the submission from the data source includes rendering instructions that apply to the structured payload portion. For example, the LEXS 3.0 submission to R-DEX from the Bureau of Prisons (BOP) includes rendering instructions that generate HTML. In R-DEX 6.0 the handling of the rendering instructions is implemented so that R-DEX users can view the data contained in the structured payload portion of the BOP submission even though R-DEX does not understand it and cannot ingest and utilize this content for machine processing. (This description, of course, implies that the BOP data is not pre-rendered and uses an XSLT method.) Though LEXS does not utilize a formal MustUnderstand / MustIgnore processing model with

mustUnderstand and mustIgnore attributes, both data sources and consumers must implement as if the structured payload element was marked up with mustIgnore and the rest of LEXS-PD was marked up with mustUnderstand.

In preparing a submission, a data source can incorporate information about people, places, activities, and things as well as associations related to an incident in the “lexs:PackageDigest” while N-DEX-specific incident information that cannot be represented in the “lexs:PackageDigest” is included in the Structured Payload. This approach allows for multiple levels of understanding as shown in Figure 8. Sub-communities can be formed from existing communities. Thus a system or community that is focused on “Cyber Incidents” can use the N-DEX incident IEP as a base and define their own extensions in a “cits<sup>15</sup> schema” as indicated in Figure 8. A consuming system that encounters structured payload elements (“lexs:PackageStructuredPayload”) in a LEXS-PD instance does not have to understand or process every possible structured payload. Clearly there is a distinction here between recognizing the existence of the structured payload tag (“lexs:PackageStructuredPayload”) and understanding or processing the content of this tag (the actual payload). All LEXS 3.0 compliant systems must recognize this tag and its existence if it occurs, but do not “have” to process the content within the “lexs:PackageStructuredPayload” element. The “processContents” attribute associated with “lexs:PackageStructuredPayloadType” is defined as “skip” in the “lexs schema”. This implies that no validation is necessary; the element must simply be well formed.

Figure 9 shows the placement of structured payload elements in LEXS 3.0. The structured payload element (“lexs:PackageStructuredPayload”), is an optional element within the package and contains a metadata element that identifies the community or system that defined the payload. The rest of the definition within the structure is entirely up to the particular community or system.

In Figure 9 two structured payload elements are shown. The first is for N-DEX and the second is for DHS.



**Figure 9. Multiple Structured Payload Elements in a Package**

Entities and attributes that are already represented in the Package Digest are refined with additional information and attributes within elements inside the N-DEX structured payload. These elements in the N-DEX payload are linked to the corresponding entities in Package Digest through the use of references from the structured payload back to the Package Digest. Similarly, the structured payload element for DHS has its own references that link back to the Package Digest.



```

<lexs:Package>
  <lexs:PackageMetadata> ..... </lexs:PackageMetadata>
  <lexs:PackageDigest>
    <lexs:EntityActivity> <!-- Activity Arrest (in lexs:PackageDigest) =====>
      <u:Activity s:id="Arrest1">
        <u:ActivityCategoryText>Arrest</u:ActivityCategoryText>
        <u:ActivityDescriptionText>
          The driver, Michael Leonard is arrested for Driving under the Influence (DUI).
          Arrest Incident number123015.
        </u:ActivityDescriptionText>
        <u:ActivityDate>1996-09-01</u:ActivityDate>
      </u:Activity>
    </lexs:EntityActivity>
  </lexs:PackageDigest>
  <lexs:PackageStructuredPayload> <!-- N-DEx Data =====>
    <lexs:StructuredPayloadMetadata>
      <lexs:CommunityURI>http://fbi.gov/cjis/N-DEx/1.0</lexs:CommunityURI>
      <lexs:CommunityDescription>N-DEx</lexs:CommunityDescription>
      <lexs:CommunityVersion>1.0</lexs:CommunityVersion>
    </lexs:StructuredPayloadMetadata>
    <ndx:ArrestReport xmlns:ndx="http://fbi.gov/cjis/N-DEx/1.0"
      xmlns:j="http://niem.gov/niem/domains/justice/1.0"
      xmlns:u="http://niem.gov/niem/universal/1.0" >
      <ndx:Arrest> <!-- Arrest Information in ndex namespace =====>
        <j:ArrestAgency>
          <j:OrganizationORIID>
            <u:ID>TN0470100</u:ID>
          </j:OrganizationORIID>
        </j:ArrestAgency>
        <j:ArrestAgencyRecordID>
          <u:ID>123015</u:ID>
        </j:ArrestAgencyRecordID>
        <j:ArrestCategoryCode>O</j:ArrestCategoryCode>
        <ndx:ArrestAugmentation>
          <lexslib:SameAsDigestReference lexslib:ref="Arrest1"/>
          <ndx:ArrestNarrativeDate>1996-09-01</ndx:ArrestNarrativeDate>
        </ndx:ArrestAugmentation>
      </ndx:Arrest>
    </ndx:ArrestReport>
  </lexs:PackageStructuredPayload>
</lexs:Package>

```

The detailed structure that corresponds to Figure 9 is shown above in an example that has been abbreviated for illustrative purposes. In this example, the “lexs:PackageDigest” element contains an activity entity, “lexs:EntityActivity” which represents the arrest of “Michael Leonard” as described by the elements contained within this entity. The data source has also enclosed a “lexs:PackageStructuredPayload” element and the metadata tag contained within this element (“lexs:StructuredPayloadMetadata”) describes this structured payload as being compliant with the schema for the N-DEx program / community. The “ndx:ArrestReport” declares the namespaces and the associated namespace prefixes (e.g., ndex) used and contains an arrest element that uses various containing elements to provide additional details about the arrest in a structured format (e.g., “j:ArrestAgency”, “j:ArrestAgencyRecordID”, “j:ArrestCategoryCode”).

Finally, the “lexslib:SameAsDigestReference” is used to link back (using the reference “Arrest1”) to the “lexs:EntityActivity” that represents the same arrest in the “lexs:PackageDigest”.

## 2.16 LogicalID and SourceID

If two different packages each contain an EntityPerson with an identical value for Social Security Number, these two entities might represent the same real world person. When entities have attributes that match, these entities (and thus data items) are only linked implicitly. Business rules can be used to determine when entities from separate data items can be determined to be referring to the same real-world object—for use in a link analysis chart, etc.

It was not possible to communicate this information, however, to another system before LEXS 3.0, however—the other system had to re-discover this implicit linking for itself. In addition, many source systems actually store entity data in resolved/normalized tables, where the creation of packages actually involved de-normalizing (duplicating) the entity data across multiple packages. The data item paradigm is still needed to provide granularity, but it would be beneficial to not lose the knowledge source systems may have in the exchange of packages. A mechanism for explicit linking of entities is needed (two mechanisms, actually).



**Clarification:** There can be some confusion stemming from the terminology around entities and entity records. Consider an example where there is only one real-world “Billy Bob Guy”, but a data source has two normalized records: Billy Guy (#1) and Billy B. Guy (#2). Now suppose the system has one warrant that mentions BG (#1) as the target and two warrants that mention BG (#1) as an associate; plus one warrant that mentions BBG (#2) as the target and three warrants that mention BBG (#2) as an associate. If this system produces one package for each warrant, then how many Billy *entities* are there? One, if you mean *real-world entities*; two, if you mean *normalized source records*; but seven if you mean *package entities*. As you can see, we have to be careful with our language. In this document (when not otherwise clear from context) we use the simple term “entity” to refer to entities within a package—records/descriptions of entities within the context of a package (i.e., the answer is seven).

An issue with the explicit linking of entities between packages is apparent in the publication model. There may be ambiguity as to whether the source system intends for the linked entities to be treated as distinct for display and update (within their data items), or as the *same* entity (if updated in one package, must be updated everywhere by a consuming system). Thus there are two distinct kinds of explicit linking:

- Where the source system effectively stores a single record for an entity referenced in multiple packages, wishes to have the entity data be shown identically in those multiple data items by the receiver, and wishes to be able to update the entity in all data items without having to send updates for each and every one in multiple packages. Some of the DOJ component data sources are good examples of this.

- Where the source system effectively stores multiple records for an entity (or multiple variants of the information for that entity) and needs/wishes to preserve distinct views of the entity in distinct data items, but somehow knows/believes/thinks that these multiple entity records all refer to the same real world object and would like to make this apparent to the receiving system and its users. In addition, the source system is willing/wants to update the different entity records individually. Systems that perform entity resolution on data contributed from multiple sources are good examples of this.

As a result, two separate linking schemes are provided in the specification: one to handle the case where entity records are normalized into a single master table and an update to that entity means an update to every data item that references that entity (hard-linked entities), and one to handle the case where the entities are linked together somehow, but entity information is stored separately for each data item and may be different in terms of detail or actually conflicting and thus must be updated independently (soft-linked entities).

The first (hard linking) uses the concept of a "source id." All entities with the same source id should be (effectively) collapsed into a single entity record in the receiving system, such that any update of the entity within any single package updates it for each and every data item in which it is referenced. As a corollary, all representations of the entity within a single submission (snapshot) should be identical; otherwise, the results are not deterministic.

The second (soft linking) uses the concept of a "logical id." All entity representations with the same logical id—but distinct or omitted source ids—will be preserved as separate records in the receiving system, and thus the receiving system will be able to show different views of the entity for different data items. Updates are interpreted individually—the entity is only updated in association with the updated data item—no other soft-linked entities are modified.

The following example shows four person entities, *each in a different package*:

Person A		Person B		Person C		Person D	
Name	Billy Guy	Name	Billy B Guy	Name	Billy Guy	Name:	Jon Doe
SS#	087678945	SS#	093426721	SS#	087678945	SS#	
SourceID	Per47	SourceID	Per84	SourceID	Per47	SourceID	Per24
LogicalID	Per73	LogicalID	Per73	LogicalID	Per73	LogicalID	Per94

A and C refer to the same person. Though they are different entities in different packages they originate from a single record in a data source.

A and B are suspected to refer to the same person, however the information is retained at the data source as two distinct records. The data source (or intermediary) has a link between these entities and communicates this information as a “hint” to consumers.

A and D are definitely different entities representing two entirely different people.



**Tip:** Each source system can choose to publish as is appropriate for it—DOJ components will probably use hard linking (SourceID) for person entities (wanted or associates) when referring to the same entity, while R-DEx should probably use soft linking (LogicalID) to communicate the results of its entity resolution to other systems, because it can't normalize (and thus change) the underlying entity records it is given. For source systems, it shouldn't be an extra complication—each should be able to now choose the model that works like it does internally for entities: de-normalized entity records (implicit entity links between data items - old model), linked de-normalized entity records (soft entity links - LogicalID), or fully normalized entity records (hard entity links - SourceID).



**Tip:** It is slightly more complicated for consuming systems, but with some flexibility. For one thing, any consuming system can choose to ignore logical ids—they will not be able to provide data as rich to users/systems, but it will not impact the integrity of updates coming from the sources. The LogicalID is essentially just “additional info” that systems may or may not use to aid entity resolution. All PD-consuming systems, however, would have to support SourceIDs or else data items would be out of sync when shared entities were updated. This might not be as complicated as one might think, however, as most PD-consumers would probably put each entity type into its own relational table, so they would just have to manage the insert/update logic for entities to handle SourceID as a unique key.

Hard links, soft links, and implicit links are all between entities in *distinct* packages. Packages *contain* entities. It is about tying the BBG entity in package #1 to the BBG entity in package #2.

When all of the BBG entities are hard-linked together, then they are by definition always identical—updated in sync and displayed identically for all data items. But the entities in the package are not first-class objects—they do not exist separately from the packages and they are not published or updated separately. Mechanically, you still update data items by providing a complete updated picture of the data item; it is just that other data items may be implicitly updated as well if they contain a hard-linked entity.

LEXS 3.0 uses the Metadata construct in NIEM to implement linking for SourceID and LogicalID. This technique provides a general method for applying metadata and additional content to existing entities. A block of metadata is created to contain the SourceID and/or LogicalID and this Metadata block is contained in the “lexs:Entity\${SUFFIX}” element. A specific “lexsdigest:\${SUFFIX}” element states what metadata applies to it using the “s:metadata” attribute described previously. Thus the following structure represents that Billy Guy (Person A) and Billy B Guy (Person B) are suspected to be the same person:

```
<!-- Person A-->
<lexs:EntityPerson>
  <lexsdigest:Metadata s:id="MSub2">
    <u:SourceIDText>Per47</u:SourceIDText>
    <lexsdigest:LogicalIDText>Per73</lexsdigest:LogicalIDText>
  </lexsdigest:Metadata>
  <lexsdigest:Person s:id="A" s:metadata="MSub2">
    <u:PersonName>
      <u:PersonGivenName>Billy</u:PersonGivenName>
      <u:PersonSurName>Guy</u:PersonSurName>
    </u:PersonName>
    <u:PersonSSNID>
      <u:ID>097678945</u:ID>
    </u:PersonSSNID>
  </lexsdigest:Person>
</lexs:EntityPerson>
<!-- Person B-->
<lexs:EntityPerson>
  <lexsdigest:Metadata s:id="MSub1">
    <u:SourceIDText>Per84</u:SourceIDText>
    <lexsdigest:LogicalIDText>Per73</lexsdigest:LogicalIDText>
  </lexsdigest:Metadata>
  <lexsdigest:Person s:id="B" s:metadata="MSub1">
    <u:PersonName>
      <u:PersonGivenName>Billy</u:PersonGivenName>
      <u:PersonMiddleName>B</u:PersonMiddleName>
      <u:PersonSurName>Guy</u:PersonSurName>
    </u:PersonName>
    <u:PersonSSNID>
      <u:ID>087678945</u:ID>
    </u:PersonSSNID>
    <c:PersonAlternateName xsi:nil="false">
      <u:PersonFullName>Tommy Jones</u:PersonFullName>
    </c:PersonAlternateName>
  </lexsdigest:Person>
</lexs:EntityPerson>
```



**Tip:** The core syntax is as follows:

- The “lexsdigest:Person” element refers to its metadata (Msub1 or MSub2)
- The reference uses the attribute “s:metadata”
- The reference is to the element with “s:id” = (Msub1 or MSub2)
- The Metadata is a separate element
- The separate element is called “lexsdigest:Metadata”
- The metadata element has the id Msub1 or MSub2.
- The ID is conveyed with the attribute s:id
- The “lexsdigest:Metadata” element contains an element for SourceID
- The “lexsdigest:Metadata” element contains an element for LogicalID
- The SourceID element is “u:SourceIDText”
- The LogicalID element is “lexsdigest:LogicalIDText”
- The LogicalID element is the same whereas the SourceID element is different
- Since the two entities are not identical the SourceIDs could not be the same

## Appendix A. End Notes

The following end notes are referenced in the text.

---

<sup>1</sup> <http://www.niem.gov/>

<sup>2</sup> Navigate to the LEXS folder in Community Explorer at <http://collab.core.gov>

<sup>3</sup> For brevity we will describe illustrative examples in the context of R-DEx, rather than both R-DEx and N-DEx.

<sup>4</sup> It should be noted that there is no established or official taxonomy for these terms, though various organizations may have published glossaries, describing what they mean by event, incident, activity, case, etc. Sometimes, different systems might use the same term to abstract different aspects of the real world. Conversely, different implementations or organizations might indeed use different terms to refer to identical abstractions of reality.

<sup>5</sup> A Uniform Resource Identifier (URI) provides a simple and extensible means for identifying a resource. This specification of URI syntax and semantics is derived from concepts introduced by the World Wide Web global information initiative, whose use of these identifiers dates from 1990 and is described in "Universal Resource Identifiers in WWW" [RFC1630 @ <http://www.ietf.org/rfc/rfc1630.txt> ].

<sup>6</sup> In this document, when we use the word element we mean to refer to the logical structure known as “**element**” as defined by the XML Specification. Elements are the basic building blocks of XML, and may be thought of as containers. For additional information please refer to <http://www.w3.org/TR/REC-xml/>.

<sup>7</sup> See <http://www.w3.org/TR/soap12-part0/> for a definition of SOAP and SOAP BODY.

<sup>8</sup> Here we use the term **service provider** to refer to the data source as a “**provider agent**”. See <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#wordonspr> for a discussion on **service provider** and **provider agent**.

<sup>9</sup> Currently the sensitivity marking is specified not as an enumeration but as text.

<sup>10</sup> An originating agency identifier code (ORI) is permanently assigned by the Federal Bureau of Investigation (FBI) to a bureau/office that must access information in a CJIS (Criminal Justice Information Services) system.

<sup>11</sup> <http://www.niem.gov/index.php>

<sup>12</sup> <http://www.ietf.org/>

<sup>13</sup> This is a hypothetical example and should not lead to any conclusion that ARJIS republishes data from SDPD. In fact, today, ARJIS does not publish any data in LEXS-PD 3.0 format.

<sup>14</sup> <http://www.w3.org/TR/REC-xml/>

<sup>15</sup> It must be noted that both “cits” and “bits” are hypothetical communities and are shown here only for the purpose of illustration.