Contents

1 Introduction **2**

2 Conventions Used in the User Guide **2**

3 Brief USLM Background **3**

4 Existing Documentation **4**

5 What Has Not Changed **4**

6 Schema Changes **5**

6.1 Changed Schema Document Structure 5
6.2 Added Models 5
6.2.1 Amendment Documents 5
6.2.2 Engrossed Amendment Documents 6
6.2.3 Chamber-specific Metadata 6
6.3 Added Guidance for Usage 6
6.3.1 Identifiers with Multiple Languages 7
6.3.2 Hierarchical Law Levels 7
6.4 Changed Models 7
6.4.1 Content in non-USLM Namespaces 8
6.4.2 Content Elements 9
6.4.3 Layout Element 9
6.4.4 Meta Element 9
6.4.5 Preface Element 9
6.4.6 Editorial Content Element 9
6.4.7 Figure Element 10
6.4.8 Note Elements 10
6.4.9 P Element 10
6.4.10 Page and Line Elements 10
6.4.11 Positioned Note Elements 10
6.4.12 Running Head Elements 10
6.4.13 Recital Element 11
6.4.14 Statement Elements 11
6.4.15 Scaling Brace Markup 11
6.4.16 Schema Scope 11

7 Feedback **12**

Prepared by: Government Publishing Office
Revised: October 8, 2024

1 Introduction

This Review Guide is intended to help users to understand changes in the 2.1 version of the United States Legislative Markup (USLM) schema so that users can provide meaningful feedback about the changes. This guide assumes that the reader is familiar with the 1.0 and 2.0, 2.0.12, and 2.0.17 versions of the USLM schema and is generally knowledgeable about XML schemas in XSD format. For more information about previous versions, see the Existing Documentation section of this document for links to existing documentation.

This guide reflects USLM schema version 2.1.0.

2 Conventions Used in the User Guide

The following conventions are used in the User Guide:

- XML element names are denoted with angled brackets. For example, `<title>` is an XML element.
- Groups of elements are denoted with a word followed by the word 'elements'. For example, 'positioned note elements' includes any of the elements `<footnote>`, `<sidenote>`, `<endnote>`, and `<ear>`.
- XML attribute names are denoted with an "`@`" prefix. For example, `@href`. is an XML attribute.
- Enumerated values are denoted in a fixed width font. For example, `landscape` is an enumeration.
- String values are denoted with double quotes. For example, "`title1-s1`" is a string value.
- A new ***term*** being defined is shown in bold italic.
- A new **element** or **attribute** being defined is shown in bold.

3 Brief USLM Background

The USLM schema was first developed in 2013 by the Office of the Law Revision Counsel of the U.S. House of Representatives (OLRC) in order to produce the United States Code in XML. Since 2013, the OLRC regularly produces a USLM version of the United States Code for download at http://uscode.house.gov/download/download.shtml. The USLM version of the U.S. Code is updated continuously as new laws are enacted.

The original goals of the USLM schema included:

1. *Allow existing titles of the United States Code to be converted into XML.*
2. *Support ongoing maintenance of the United States Code.*
3. *Support the drafting of new positive law codification bills and related materials.*
4. *Provide a flexible foundation to meet future needs of Congress.*
5. *Compatibility with existing legislative documents in other XML formats.*

Building on the "flexible foundation" in goal number four above, the Government Publishing Office (GPO) released the 2.0 update to USLM that extended its use to the following document sets:

- Enrolled Bills and Resolutions
- Public and Private Laws
- Statutes at Large
- Statute Compilations
- Federal Register (FR)
- Code of Federal Regulations (CFR)

The documentation for the USLM used in these document sets is in the 2.0.12 version of the Review Guide.

Further additions were made to USLM 2.0 to enable its use for bills and resolutions in all document stages. This is documented in the 2.0.12 and 2.0.17 versions of the Review Guide.

The changes made to the USLM schema to support its use for amendment documents are breaking changes, hence the 2.1 numbering.

4 Existing Documentation

User documentation for the 1.0 version of the schema can be found at https://github.com/usgpo/uslm/blob/main/USLM-User-Guide.pdf and https://github.com/usgpo/uslm/blob/main/USLM-User-Guide.md.

User documentation for the 2.0 version of the schema, to version 2.0.12, can be found at https://github.com/usgpo/uslm/blob/main/USLM-2_0-Review-Guide-v2_0_12.md and https://github.com/usgpo/uslm/blob/main/USLM-2_0-Review-Guide-v2_0_12.pdf.

User documentation for the 2.0 version of the schema after 2.0.12, to version 2.0.17, can be found at https://github.com/usgpo/uslm/blob/main/USLM-2_0-Review-Guide-v2_0_17.md and https://github.com/usgpo/uslm/blob/main/USLM-2_0-Review-Guide-v2_0_17.pdf.

The XSD schema and CSS stylesheets for online viewing can be downloaded at: http://uscode.house.gov/download/resources/schemaandcss.zip and https://github.com/usgpo/uslm. Note that the CSS stylesheet is informational only. It produces a draft view of the documents.

Note: These resources and more are available on GPO's Developers Hub at https://www.govinfo.gov/developers.

5 What Has Not Changed

Version 2.1 of USLM is architecturally an incremental change to the schema. While many new elements have been added and several content models have been extended or modified, the fundamental design of the schema has not changed. The following principles, documented in the 1.0 and 2.0 versions of the User Guide, continue in version 2.1:

- Abstract and Concrete Models
- Inheritance
- Attribute Model
- Core Document Model
- Metadata Model
- Hierarchy Model
- Versioning Model

- Presentation Model
- Relationship to HTML
- Identification Model
- Referencing Model

Many of these models have been extended to accommodate the additional document types and their structures. These extensions are backwards-compatible except in a few cases described below.

6 Schema Changes

6.1 Changed Schema Document Structure

The XSD schema documents were restructured to facilitate schema processing. A single top-level XSD file imports the component schema files for USLM, tables, MathML, and chamber-specific metadata. References to other components internal to the component schemas are made by namespace only, which avoids potential circular reference issues during schema processing.

6.2 Added Models

6.2.1 Amendment Documents

Amendment documents have a structure that is unlike other legislative documents, and require different metadata. For this reason the `<amendment>` document uses new elements that are equivalents to the existing `<meta>`, `<preface>`, and `<main>` elements, namely **`<amendMeta>`**, **`<amendPreface>`**, and **`<amendMain>`**. As in other legislative documents, the `<amendMeta>` element contains the machine-processable metadata, `<amendPreface>` the metadata that is rendered with the document, and `<amendMain>` the main content of the `<amendment>` document.

Within the `<amendMain>` element, the individual amendment instructions are delineated with **`<amendmentInstruction>`** elements, which contain a `<num>` for the amendment number, if present, and `<content>` for the content of the amendment instruction. The lines of content of the amendment instruction are always numbered; the instruction lines may be numbered depending on the value of the new **`@amendmentInstructionLineNumbering`** attribute on the `<amendMain>` element.

An example of an <amendmentInstruction> is

```
<amendmentInstruction><num>3</num><content>On page 4, line 11, add
   at the end of section 2(b) the following new paragraph:
        <amendmentContent styleType="OLC">
            <paragraph>
                <num>(9)</num>
                <content class="inline">Even public health programs
                   do not consistently provide coverage for such
                   treatments.</content>
            </paragraph>
        </amendmentContent>
    </content>
</amendmentInstruction>
```

The **<amendmentContent>** element has the same content model as the <quotedContent> element, but is used for amendments to bills that are not yet law, reported bills, and amendment documents, whereas the <quotedContent> element is used for amending existing law.

6.2.2 Engrossed Amendment Documents

Engrossed amendment documents have the same allowed content model as other amendment documents, but use the **<engrossedAmendment>** element instead of the <amendment> element. The structure usually differs from that used in other amendment documents.

6.2.3 Chamber-specific Metadata

The House and the Senate each have specific metadata that they use in bills and resolutions before they are engrossed, which helps the workflow in each chamber. Since this metadata is specific to the chamber, the USLM 2.1 schema defines only the namespace of the metadata that the chamber can add to the <meta> and <amendMeta> element, and not the schema itself.

6.3 Added Guidance for Usage

These are recommended guidelines. They are not enforced (and in many cases are unenforceable) by the schema but are considered best practice in the set of known documents for which USLM was designed and extended.

6.3.1 Identifiers with Multiple Languages

A language-neutral identifier does not include a language code. To create an identifier for a specific language version of the document, include a language identifier using the three-letter system defined in ISO 639-2, as used by the Library of Congress, using the Bibliographic language code (B) where there is a choice. The accepted list is found at https://www.loc.gov/standards/iso639-2/php/code_list.php For instance, the identifier for section 101 of the English version of Title 51 would be "/us/usc/t51@eng/s101". This should be used in conjunction with the xml:lang attribute with the correct value.

To make an identifier for a specified version of a provision or level, an "@" symbol is used. For instance, an identifier for section 101 of Title 51 on 1 February, 2013 would be "/us/usc/t51@2013-02-01/s101", which uses the ISO time reference system. The version specification could also be an arbitrary name or relative date, such as 'pre-1824' or 'v1.1.1'. A version string that is not a date must not begin with a number, to avoid confusion with dates. Version strings must be constructed solely with characters taken from the URI unreserved character set per RFC-3986 2.3.

For compability with AKN, or when including both a language and version specification, the "!" designator may also be used for a language-specific version of a provision instead of the "@" designator, e.g., "/us/stat/53/1561!nor" for the Norwegian version of "/us/stat/53/1561!eng".

When including both a language and a version specification, place the language specification ahead of the version specification. For instance to combine both examples above, the result would be "/us/usc/t51!eng@2013-02-01/s101" or "/us/usc/t51!eng@v1.1.1/s101".

6.3.2 Hierarchical Law Levels

The standard patterns for hierarchical law levels begin with optional num and heading elements in either order. There are two standard patterns for the content that follows those elements:

- a `<content>` element, that does not contain subordinate law levels, or
- an optional `<chapeau>` element, followed by one or more subordinate law levels, followed by an optional `<continuation>` element.

6.4 Changed Models

In parallel with the design work to extend the USLM schema for use with amendment documents, the schema was refactored. The addition of many new elements and attributes

since the development of USLM 1.0 had resulted in the possibility of markup that did not follow the intended model guidelines. In version 2.1 of the schema, the content models of several elements were changed to disallow some document structures that were never intended to be allowed, while allowing for existing and expected document structures. As an example, in the 2.0 schema, the <content> element allows any USLM or any element in any other namespace, in any order, which can lead to documents that do not conform to expected usage. To counteract this, the 2.1 schema no longer allows the <content> element in some elements where it was previously allowed, and selected other elements no longer have the same content model as the <content> element. For example, the positioned note elements no longer allow the <content> element, and the <figure> element no longer has the same content model as the <content> element. Also, the use of elements from other namespaces within the <content> element has been restricted (see Content in non-USLM Namespaces).

As a result, documents that conform to expected modeling patterns will continue to validate according to the 2.1 schema with rare exceptions, but documents that use unexpected modeling patterns and structures may not.

6.4.1 Content in non-USLM Namespaces

USLM 2.0 uses definitions for the content of many elements which allowed elements in any namespace. The 2.0 model has the advantage of allowing discovery of content in other XML namespaces which various stakeholders might wish to include in a USLM document. Unfortunately, it has the corresponding disadvantage of presenting an impossible challenge to anyone required to process USLM content.

USLM 2.1 changes the definitions for these elements so that elements in namespaces other than USLM are not allowed.

There are some exceptions.

The <mathml:math>, <xhtml:img>, and <xhtml:table> elements are permitted in selected elements, such as the content group of elements. Chamber-specific metadata in the chamber-specific namespace is permitted in the <meta> element.

The <foreign> element, which is allowed in some elements, may contain content in a namespace other than the USLM namespace http://schemas.gpo.gov/xml/uslm. Aside from the specific exceptions, this is now the only way to include content in another namespace in a USLM document.

6.4.2 Content Elements

The group of elements with the same expansive content model as the <content> element is smaller in USLM 2.1 than in USLM 2.0. The <figure>, <editorialContent>, <column>, and <p> elements and their substitution groups now have more appropriate content models. The content group of elements in USLM 2.1 comprises the <content>, <text>, <chapeau>, <continuation>, and <component> elements, elements in the quotedContent group, and note elements.

Any USLM element is allowed in content elements, but elements in other namespaces must be contained by a <foreign> element. Exceptions are made for the elements <mathml:math>, <xhtml:img>, and <xhtml:table>, which may be direct children of content elements. See Content in non-USLM Namespaces for more details.

6.4.3 Layout Element

The USLM 2.0 model for a <layout> element allows <header>, <row>, <block>, <content>, and <NoteStructure> elements.

The 2.1 model adds <column>, <page>, and <coverText> and **removes** the <content> element. USLM 2.0 documents which use content elements in layout structures will need to be adjusted to use 2.1 compatible markup.

The <column> element is no longer based on the <content> element and contains a specific list of allowed elements that permits known and expected usage.

6.4.4 Meta Element

The <dcterms> namespace has been removed from the USLM 2.1.0 schema. In consequence, the <dcterms:created> element where it exists must be replaced with the new **<createdDate>** element.

6.4.5 Preface Element

The <preface> element no longer allows the <content> element as a child. The list of allowed elements was expanded to permit known and expected usage.

6.4.6 Editorial Content Element

The <editorialContent> element in current usage contains only text and <toc> elements, and the element model has been restricted to those choices.

6.4.7 Figure Element

The <figure> element is no longer based on the <content> element. It allows the USLM and XHTML <img> elements, as well as the MathML <math> element, the <figCaption>, and <page> elements.

6.4.8 Note Elements

A new note, the **<drafterNote>**, was added to the existing group of note elements. This is used by drafters to make notes while drafting bills. As with other note elements, it is based on the <content> element and thus can contain any USLM element.

6.4.9 P Element

The <p> element is no longer based on the <content> element. It has a long list of specified elements that are allowed, as well as text content.

6.4.10 Page and Line Elements

The <page> and <line> elements are no longer in the group of positioned note elements, as they have a more restricted content model. They allow inline elements and text content.

6.4.11 Positioned Note Elements

These include <footnote>, <sidenote>, <ear>, and <endnote>. These notes are anchored at a specific location in the text, but their contents may float to another position when rendered. The PositionedNoteType is no longer a type of Note, but is a distinct type that does not allow the <content> element. As a consequence, <footnote> and other types of positioned note elements no longer allow the <content> element as a child element. Specific additions to the content model include <xhtml:table>, <list>, and <img>, as well as note elements other than positioned note elements. It is not permitted to nest positioned note elements. (You cannot, for example, put a footnote in an ear or an ear in an endnote.)

6.4.12 Running Head Elements

The three elements <leftRunningHead>, <centerRunningHead>, and <rightRunningHead> are now in their own group, and can contain only text content, with additional attributes to specify rendering. The new **<slugLine>** element is also in this group, although the contents are rendered on the slug (footer) line rather than in the header of a printed page.

6.4.13 Recital Element

The <recital> element in older Statutes at Large volumes is more complex than other types of statement elements, so the content model for <recital> elements was expanded to allow <figure>, <xhtml:table>, <p>, <quotedText>, and <quotedContent>, while excluding the <content> element.

6.4.14 Statement Elements

The statement elements are <docTitle>, <enactingFormula>, <longTitle>, <officialTitle>, <resolvingClause>, <statement>, and <wordsOfIssuance>. The content model for statement elements has been simplified, allowing text content, <marker>, <p>, <page>, note elements, positioned notes, and inline elements.

6.4.15 Scaling Brace Markup

The 2.0 model allows elements in arbitrary namespaces in many places. The 2.1 model does not so allow, and so the following use of MathML markup to represent a scaling brace used in 2.0 content:

```
<column><mo xmlns="http://www.w3.org/1998/Math/MathML"
   stretchy="true">}</mo></column>
```

must become:

```
<column><math xmlns="http://www.w3.org/1998/Math/MathML"><mo
   stretchy="true">}</mo></math></column>
```

since the USLM 2.1 model makes a specific namespace-other-than-USLM exception permitting the inclusion of the <mathml:math> element. Other elements in the MathML namespace must descend from the <mathml:math> element.

6.4.16 Schema Scope

As use of the USLM schemas increases, different stakeholder groups with different objectives and requirements will use USLM to represent and process content. This introduces situations where different content rules are desirable. For example, markup that is acceptable or even necessary and desirable in a pre-introduced bill might not be acceptable in a passed bill. Representing this variation in the schema would introduce substantial overhead from complexity.

The stakeholders have agreed that the more appropriate approach going forward is to maintain the 2.* USLM schemas with only those restrictions which are applicable to all USLM users. Restrictions specific to a particular document set or to a processing mechanism will need to be implemented by mechanisms other than the official USLM XSD schemas.

7 Feedback

To submit feedback, questions, or comments about the USLM 2.1 schema and this Review Guide, please open a GitHub issue at https://github.com/usgpo/uslm/issues.