LEDES XML Formats Help Make Ebilling and Collaboration Easier - What is XML?

By Jane A. Bennitt, President

LEDES Oversight Committee

Index

Session Goals

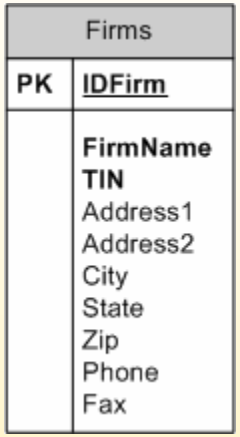
- Learn to visually recognize the components of XML, DTD and XSD documents
- Editors for XML, DTD, XSD and ASCII
- Learn how to read the specification documents for the various LEDES formats
- Review sample files for each format
- Discuss the merits of each of the LEDES formats
- Provide documentation that can be reviewed outside of this session

Background

Flat File and Relational Databases

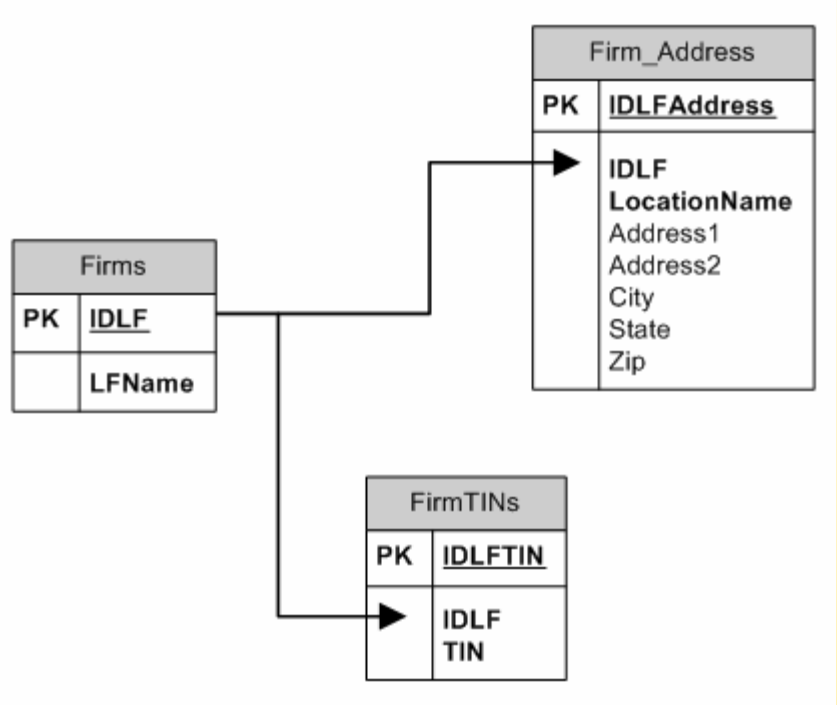
Flat File Databases

- Provides a consistent data string for every record in the table
- Drawbacks:
  - Data may require tweaking to be saved
  - Field data may be repeated from record to record
  - Uses maximum memory to store data
- LEDES 98B is a flat file database



Relational Databases

- Data is stored in related tables
- Table structure represents relationships that exist in data
- Most economical method for storing data
- Data strings need to be assembled to be displayed
- Parent and Child relationships
- LEDES XML formats are relational databases



Index

XML

XML

- Extensible Markup Language
  - Extensible:
    - Can be tailored or customized
    - Allows you to create your own data structures
  - Markup Language:
    - Defines document or data structure and organization
- Recommended by the World Wide Web Consortium on February 10, 1998
- Use with HTML to present data on web pages
  - HTML: how data should be displayed
  - XML: how data is structured and stored

XML

- In XML data is stored separately from its context or presentation
  - XML document contains data
  - “Context or Presentation” stored in a .DTD or .XSD file
- Because XML is software and hardware independent it can be used as a vehicle for data exchange
- XML underlies most of the programs you use today

Index

XML Glossary

- **Attributes:** A property associated with an XML element that is also a named characteristic of the element. Provides additional data about the element, independent of the element contents. Attributes are enclosed in single or double quotes.
- **Declarations:** Control the type of data that tags can contain.
- **Element:** A section of a document defined by start and end tags, including any associated content. The associated content is the data being presented.
  - A **root element** is the base element that contains all the other elements in an XML document
  - An **empty element** has no value
  - A **regular element** contains a value and has opening and closing tags

XML Glossary

- **Entity:** An alias for a block of text. Certain entity references are predefined by XML (< > & ‘ “) and others can be self-defined. Entities can contain text, non-ASCII characters (@ € © ® ™), graphics, multi-media clips and other non-xml files.
- **Nesting:** An ordering of elements that opens and closes a child element before its parent element is closed.
  - A **parent element** contains other elements
  - A **child element** is subordinate to the parent could have a one-to-zero, one-to-one or a one-to-many relationship with its parent
  - A **sibling element** occupies the same level in the hierarchy as the elements before it
- **Node:** The combination of the tag and data.

XML Glossary

- **Parsing:** The process of extracting data from an xml document and storing the data as records in databases.
- **Schema:** Rules that define structure and organization of XML data.
- **Segment:** Refers to the elements that could be considered tables in the XML document. In our documentation the segment names are capitalized and are preceded by the @ symbol.
  - LEDES XML Ebilling Ver 2 segments: FIRM, TAX, CLIENT, INVOICE, MATTER, TAX_SUMMARY, MATTER_DISC_CRED, TAX_MATTER_DISC_CRED, TKSUM, FEE, FEE_ITEM_DISC_CRED, TAX_ITEM_FEE, EXPENSE, EXPENSE_ITEM_DISC_CRED, TAX_ITEM_EXPENSE, ADDRESS_INFO, CONTACT_INFO, EXTEND_HEADER and EXTEND_DATA.
- **Tag:** The markup used to enclose an element's content.

XML Glossary

- **White Space:** Non-printing characters including spaces, tabs, carriage returns, line feeds

- XML documents must be well formed and valid
  - **Well Formed:** The XML document conforms to XML's syntax rules
  - **Valid:** The XML document conforms to the rules for the document specified in the associated DTD or XSD document

Index

Editing XML Documents

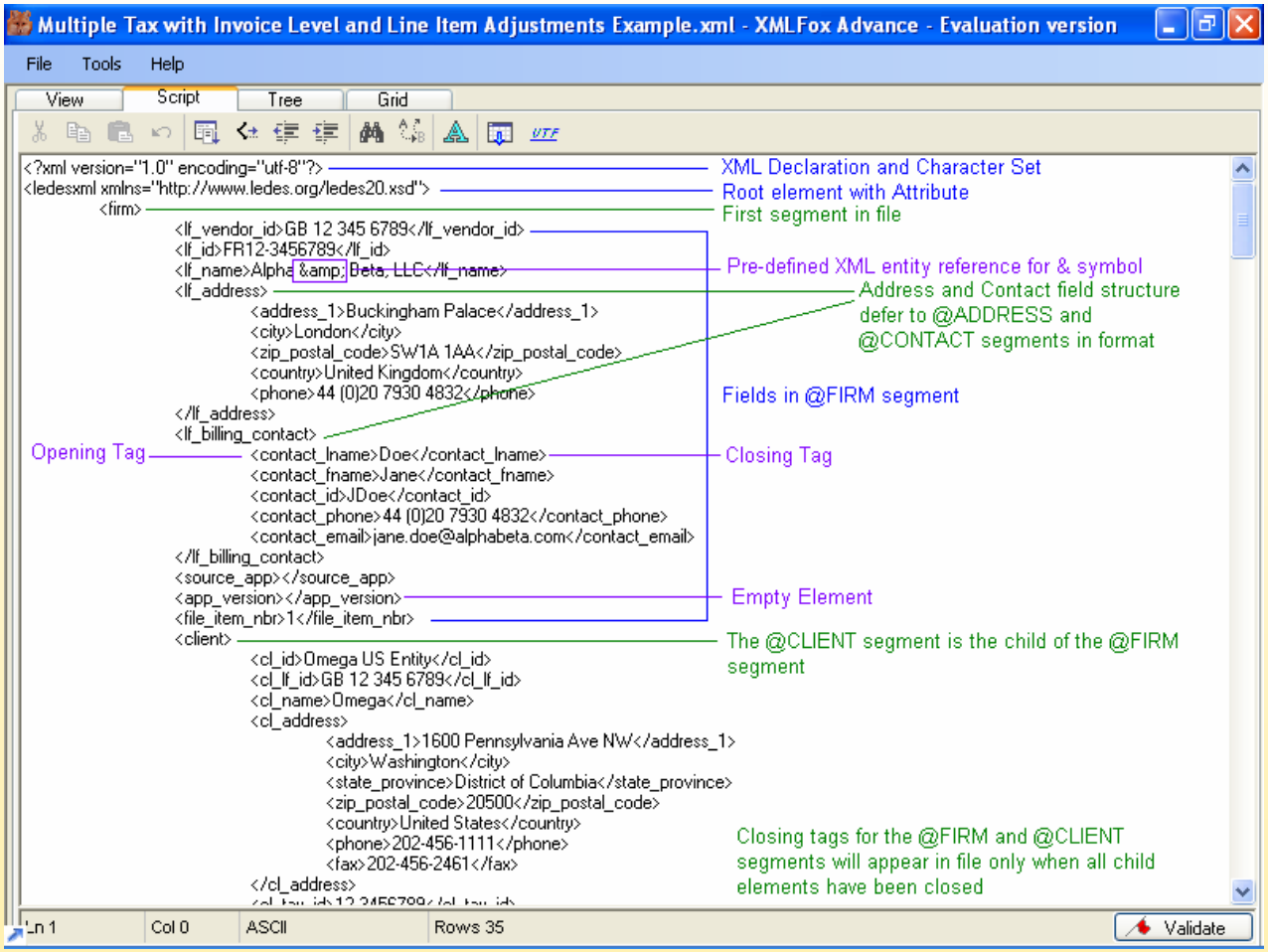
- Can not use Wordpad
  - Does not save a file with .xml extension
- Can edit in Notepad
  - Notepad does not enforce XML rules
- Use an XML Editor to edit and validate XML files

- XML Editors to Purchase
  - XML Spy, $499, available at http://www.altova.com/products/xmlspy/xml_editor.html
  - XMLFox, shareware that can be registered for $44.99, available at http://www.xmlfox.com/

- Free XML Editors
  - Peter's XML File Editor available at http://www.iol.ie/~pxe/index.html
  - Microsoft XML Notepad available at http://www.microsoft.com/downloads/details.aspx?familyid=72d6aa49-787d-4118-ba5f-4f30fe913628&displaylang=en

Index

A Sample XML Ebilling Ver. 2 File



Index

XML Syntax Rules

- The first line in XML document contains an XML declaration which identifies the XML version and character set used
- XML documents must contain a root element that contains all other elements in the document
- XML tags:
  - Can contain letters, number and other characters
  - Can not start with a number or punctuation character
  - Can not contain a space
  - Are case sensitive
  - Can not start with xml, XML or any variation of these 3 consecutive letters
- XML tags are enclosed in angle brackets

XML Syntax Rules

- Opening and closing tags must match. Opening and closing tags look the same except a closing tag has a slash before the element name.

  <lf_name>xxx</lfname> will not work

  <lf_name>xxx</lf_name> works

  - A **regular element**, which contains a value, has opening and closing tags

    <lf_name>**Alpha and Beta, LLC**</lf_name>

  - An **empty element** does not contain a textual value but can contain attributes. Empty elements can have opening and closing tags or can have a single tag with a slash before the angle bracket

    <lf_name></lf_name>

    <lf_name/>

    <lf_name location="Orlando"/>

- White space in your elements is not truncated

  <lf_name>**Alpha and Beta, LLC**</lf_name>

  will appear or be recorded as

  Alpha and Beta, LLC

XML Syntax Rules

- Certain characters can not be used in XML elements and require a pre-defined XML Entity Reference in order for the character to display correctly.

Symbol	Description	XML Entity Reference
<	Less Than	<
>	Greater Than	>
&	Ampersand	&
‘	Apostrophe	'
“	Quotation Mark	"

XML Syntax Rules

- All attribute values must be enclosed in single or double quotation marks
- XML elements can have parent and child relationships just like relational databases
- XML tags must be properly nested
  - “Nesting” is an ordering of elements that opens and closes a child element before its parent element is closed

Index

Defining XML Context

DTDs and XSDs

XML Context

- XML uses a Document Type Definition (DTD) or XML Schema Definition (XSD) to describe the context of the XML data
  - DTD: Document Type Definition. A statement of rules and the attributes (names and values associated with specific elements) allowed in the associated XML document. The DTD governs the order in which the elements and attributes must appear.
  - XSD: XML Schema Definition or XML Schema Language. The successor of the DTD. Like the DTD, the XSD defines the structure of the data in an XML document but, unlike the DTD, it also defines data types and is based on XML syntax.

XML Context

- DTDs and XSDs can be used to validate the content of an XML document
- XML documents can reference the DTD or XSD on which it is based

- You don't need to create the DTD or XSD for your LEDES XML document. You only need to create an XML document that conforms to the DTD or XSD structure set forth by the LOC.
- This document contains information that will enable you to read the DTD and XSD files provided by the LOC for the ebilling, budgeting and timekeeper formats

Index

DTDs

DTDs define the format of the XML data including elements, attributes, entities, parsed character data and character data.

DTD Glossary

- **Attributes:** Provide extra information about elements. Attributes pull data from elements or attributes on which it is based.
- **Character Data:** Data that will not be parsed by a parser, i.e., tags and entity references
- **Elements:** The main building blocks of XML and HTML
- **Entities:** An alias for a block of text. Certain entity references are predefined by XML (< > & ‘ “) and others can be self-defined. Entities can contain text, non-ASCII characters (@ € © ® ™), graphics, multi-media clips and other non-xml files.
  - **Internal Entities:** An alias that looks up a value from within the same DTD document.
  - **External Entities:** An alias that looks up a value from outside of the DTD document.

DTD Glossary

- **Parsed Character Data:** Data that will be parsed by a parser, i.e., text found between the start tag and the end tag of an xml element
- **White Space:** Non-printing characters including spaces, tabs, carriage returns, line feeds
- **XML Prolog:** Contains XML declaration, DOCTYPE declaration, Comments, Processing Instructions and Whitespace

DTD Structure

- DTD files include:
  - XML Prolog
  - Element Declarations
  - Attribute Declarations
  - Entity Declarations

DTD XML Prolog

- DTD files begin with XML Prolog, including:
  - XML Declaration – The version of XML and character set used
  - DOCTYPE declaration – Tells where the specific DTD is located
  - Comments – Notes added by the author. Notes are enclosed in angle brackets, exclamation marks and double dashes.

    <!-- this is my note -->
    - Hyphens or double hyphens can not be included inside a note
  - Processing Instructions – Instructions to the receiving system. Processing instructions are enclosed in angle brackets and question marks

    <? xml:stylesheet … ?>
  - White Space in the data display may also be defined.
- LEDES 2000 DTD does not contain an XML Prolog

DTD Element Declarations

- Element declarations name the elements in the XML document, any children (nested elements) an element may have, their frequency and/or order of appearance

Ledes2000.DTD - Notepad

File Edit Format View Help

```
<!ELEMENT ledesxml (firm, client+)>

  <!ELEMENT firm (lf_tax_id, lf_id?, lf_name, lf_address, lf_remit_address?,
                  lf_billing_contact_lname?, lf_billing_contact_fname?,
                  lf_billing_contact_id?, lf_billing_contact_phone?,
                  lf_billing_contact_fax?, lf_billing_contact_email?,
                  source_app, app_version, extend_header*)>

    <!ELEMENT lf_tax_id (#PCDATA)>
    <!ELEMENT lf_id (#PCDATA)>
    <!ELEMENT lf_name (#PCDATA)>
    <!ELEMENT lf_address (address_info)>
    <!ELEMENT lf_remit_address (address_info)>
    <!ELEMENT lf_billing_contact_lname (#PCDATA)>
    <!ELEMENT lf_billing_contact_fname (#PCDATA)>
    <!ELEMENT lf_billing_contact_id (#PCDATA)>
    <!ELEMENT lf_billing_contact_phone (#PCDATA)>
    <!ELEMENT lf_billing_contact_fax (#PCDATA)>
    <!ELEMENT lf_billing_contact_email (#PCDATA)>
    <!ELEMENT source_app (#PCDATA)>
    <!ELEMENT app_version (#PCDATA)>
```

DTD Element Declarations

Occurrence Indicators

Symbol/Term	Example	Description
, (comma)	<!ELEMENT firm (lf_tax_id, lf_id)>	The child elements *lf_tax_id* and *lf_id* must appear in the order specified
| (pipe)	<!ELEMENT firm (lf_tax_id | lf_id)>	Either child element *lf_tax_id* or *lf_id* can occur inside the parent element *firm*
(No symbol)	<!ELEMENT firm (lf_tax_id)>	The child element *lf_tax_id* must occur once inside the parent element *firm*
+	<!ELEMENT firm (lf_tax_id+)>	The child element *lf_tax_id* must be used one or more times inside the parent element *firm*
*	<!ELEMENT firm (lf_tax_id*)>	The child element *lf_tax_id* may be used zero or more times within the parent element *firm*
?	<!ELEMENT firm (lf_tax_id?)>	The child element *lf_tax_id* may be used once or not at all within the parent element *firm*

DTD Element Declarations

Types of Content

Symbol/Term	Example	Description
ANY	<!ELEMENT firm ANY)>	Allows any type of content in the element *firm*
EMPTY	<!ELEMENT firm EMPTY)>	Element *firm* can not contain any content
	<!ELEMENT firm (#PCDATA)>	Element *firm* contains parsed character data
	<!ELEMENT firm (#PCDATA | lf_id)*>	Element *firm* may contain parsed character data, or a combination of child elements (*lf_id*) and parsed character data
	<!ELEMENT firm (lf_tax_id, lf_id)>	Element *firm* can contain only child elements *lf_tax_id* and *lf_id*

DTD Structure – Element Declarations

DTD File

Ledes2000.DTD - Notepad

File Edit Format View Help

```
<!ELEMENT ledesxml (firm, client+)>

  <!ELEMENT firm (lf_tax_id, lf_id?, lf_name, lf_address, lf_remit_address?,
                  lf_billing_contact_lname?, lf_billing_contact_fname?,
                  lf_billing_contact_id?, lf_billing_contact_phone?,
                  lf_billing_contact_fax?, lf_billing_contact_email?,
                  source_app, app_version, extend_header*)>

    <!ELEMENT lf_tax_id (#PCDATA)>
    <!ELEMENT lf_id (#PCDATA)>
    <!ELEMENT lf_name (#PCDATA)>
    <!ELEMENT lf_address (address_info)>
    <!ELEMENT lf_remit_address (address_info)>
    <!ELEMENT lf_billing_contact_lname (#PCDATA)>
    <!ELEMENT lf_billing_contact_fname (#PCDATA)>
    <!ELEMENT lf_billing_contact_id (#PCDATA)>
    <!ELEMENT lf_billing_contact_phone (#PCDATA)>
    <!ELEMENT lf_billing_contact_fax (#PCDATA)>
    <!ELEMENT lf_billing_contact_email (#PCDATA)>
    <!ELEMENT source_app (#PCDATA)>
    <!ELEMENT app_version (#PCDATA)>
```

XML File With Data

```
<?xml version="1.0"?>
<!DOCTYPE ledesxml SYSTEM "ledes2000.dtd">
<ledesxml>
  <firm>
    <lf_tax_id>33-9876543</lf_tax_id>
    <lf_id>100001</lf_id>
    <lf_name>Bachman, Turner &amp;amp; Associates</lf_name>
    <lf_address>
      <address_info>
        <address_1>101 California street</address_1>
        <city>San Francisco</city>
        <state_province>CA</state_province>
        <zip_postal_code>94101</zip_postal_code>
        <phone>415-123-4567</phone>
        <fax>415-123-4568</fax>
      </address_info>
    </lf_address>
    <lf_remit_address>
      <address_info>
        <address_1>102 California street</address_1>
        <city>San Francisco</city>
        <state_province>CA</state_province>
        <zip_postal_code>94101</zip_postal_code>
        <phone>415-123-4567</phone>
        <fax>415-123-4568</fax>
      </address_info>
    </lf_remit_address>
    <lf_billing_contact_lname>Bachman</lf_billing_contact_lname>
    <lf_billing_contact_fname>Randy</lf_billing_contact_fname>
    <lf_billing_contact_id>cont005</lf_billing_contact_id>
    <lf_billing_contact_phone>415-123-4567</lf_billing_contact_phone>
    <lf_billing_contact_fax>415-123-4568</lf_billing_contact_fax>
    <lf_billing_contact_email>bachman@bachmanturner.com</lf_billing_contact_email>
    <source_app>Ajuger</source_app>
    <app_version>1.0</app_version>
    <extend_header vendor="Examen" app="billview" sequence="0" date="20000301"
      <extend_data>
        <ext_name>LFEXID</ext_name>
        <ext_value>000010</ext_value>
      </extend_data>
    </extend_header>
  </firm>
```

DTD Structure – Attribute Declarations

- Attribute declarations name the element to which the attribute belongs, identifies the name given to the attribute, the attribute data type and specifies the default type value.

  <!ATTLIST elementname attributename datatype defaultvalue>

```
<!ATTLIST extend_header
          vendor CDATA #REQUIRED
          app CDATA #IMPLIED
          sequence CDATA #IMPLIED
          date CDATA #IMPLIED>
```

- The LEDES 2000 DTD contains only one attribute, the @EXTEND_HEADER segment in the file.

DTD Structure - Attribute Declarations

Attribute Data Types

Attribute Data Type	Description
CDATA	Character data
(val1|val2|...)	Must be from the specified list of values
ID	A unique id
IDREF	The id of another element
IDREFS	A list of other ids
ENTITY	An entity (alias for a block of text)
ENTITIES	A list of entities
NMTOKEN	A valid XML name
NMTOKENS	A list of valid XML names
NOTATION	A name of a notation
PCDATA	Parsed character data
xml:	A predefined xml value

DTD Structure - Attribute Declarations

Attribute Default Values

XML Term	Example	Description
#REQUIRED	<!ATTNAME extend_header vendor CDATA #REQUIRED>	The *extend_header* attribute *vendor* is required
#IMPLIED	<!ATTNAME extend_header vendor CDATA #IMPLIED>	The *extend_header* attribute *vendor* is not required
“value”	<!ATTNAME extend_header vendor CDATA “Company”>	The default value for *extend_header* attribute *vendor* is “Company”. While other values can be entered, if none is the default “Company” will be recorded.
#FIXED *“value”*	<!ATTNAME extend_header vendor CDATA #FIXED “Company”>	The fixed default value for *extend_header* attribute *value* is “Company”. No other value can be entered.

DTD Structure - Attribute Declarations

DTD File

```
<!ATTLIST extend_header
          vendor CDATA #REQUIRED
          app CDATA #IMPLIED
          sequence CDATA #IMPLIED
          date CDATA #IMPLIED>
```

XML File With Data

```
<?xml version="1.0"?>
<!DOCTYPE ledesxml SYSTEM "ledes2000.dtd">
<ledesxml>
  <firm>
    <lf_tax_id>33-9876543</lf_tax_id>
    <lf_id>100001</lf_id>
    <lf_name>Bachman, Turner &amp;amp; Associates</lf_name>
    <lf_address>
      <address_info>
        <address_1>101 California street</address_1>
        <city>San Francisco</city>
        <state_province>CA</state_province>
        <zip_postal_code>94101</zip_postal_code>
        <phone>415-123-4567</phone>
        <fax>415-123-4568</fax>
      </address_info>
    </lf_address>
    <lf_remit_address>
      <address_info>
        <address_1>102 California street</address_1>
        <city>San Francisco</city>
        <state_province>CA</state_province>
        <zip_postal_code>94101</zip_postal_code>
        <phone>415-123-4567</phone>
        <fax>415-123-4568</fax>
      </address_info>
    </lf_remit_address>
    <lf_billing_contact_lname>Bachman</lf_billing_contact_lname>
    <lf_billing_contact_fname>Randy</lf_billing_contact_fname>
    <lf_billing_contact_id>cont005</lf_billing_contact_id>
    <lf_billing_contact_phone>415-123-4567</lf_billing_contact_phone>
    <lf_billing_contact_fax>415-123-4568</lf_billing_contact_fax>
    <lf_billing_contact_email>bachman@bachmanturner.com</lf_billing_contact_email>
    <source_app>Ajuger</source_app>
    <app_version>1.0</app_version>
    <extend_header vendor="Examen" app="billview" sequence="0" date="20000301"
      <extend_data>
        <ext_name>LFEXID</ext_name>
        <ext_value>000010</ext_value>
      </extend_data>
    </extend_header>
  </firm>
```

DTD Structure – Entity Declarations

- All self-defined entities must be declared
  - Pre-defined XML entity references do not need to be declared
- Internal Entities: Are defined within the XML document

  <!ENTITY % entityname “replacementtext”>
- External Entities: Point to an external file

  <!ENTITY % entityname “externalfilename”>

- The LEDES 2000 DTD does not contain any entities

To Edit a DTD

- Use an XML Editor that also edits DTD files
  - XML Spy, XMLFox
  - Free editors: Peter's XML Editor; for other free DTD Editors search Google
  - Microsoft XML Editor will not work
- Can use Notepad

LEDES 2000 DTD

Ledes2000.DTD - Notepad

File Edit Format View Help

```
<!ELEMENT ledesxml (firm, client+)>

  <!ELEMENT firm (lf_tax_id, lf_id?, lf_name, lf_address, lf_remit_address?,
                  lf_billing_contact_lname?, lf_billing_contact_fname?,
                  lf_billing_contact_id?, lf_billing_contact_phone?,
                  lf_billing_contact_fax?, lf_billing_contact_email?,
                  source_app, app_version, extend_header*)>

    <!ELEMENT lf_tax_id (#PCDATA)>
    <!ELEMENT lf_id (#PCDATA)>
    <!ELEMENT lf_name (#PCDATA)>
    <!ELEMENT lf_address (address_info)>
    <!ELEMENT lf_remit_address (address_info)>
    <!ELEMENT lf_billing_contact_lname (#PCDATA)>
    <!ELEMENT lf_billing_contact_fname (#PCDATA)>
    <!ELEMENT lf_billing_contact_id (#PCDATA)>
    <!ELEMENT lf_billing_contact_phone (#PCDATA)>
    <!ELEMENT lf_billing_contact_fax (#PCDATA)>
    <!ELEMENT lf_billing_contact_email (#PCDATA)>
    <!ELEMENT source_app (#PCDATA)>
    <!ELEMENT app_version (#PCDATA)>

  <!ELEMENT client (cl_id?, cl_lf_id?, cl_name, cl_address, cl_email?,
                    cl_contact_lname?, cl_contact_fname?,
                    cl_tax_id?, extend_header*, invoice+)>

    <!ELEMENT cl_id (#PCDATA)>
    <!ELEMENT cl_lf_id (#PCDATA)>
    <!ELEMENT cl_name (#PCDATA)>
    <!ELEMENT cl_address (address_info)>
    <!ELEMENT cl_email (#PCDATA)>
    <!ELEMENT cl_contact_lname (#PCDATA)>
    <!ELEMENT cl_contact_fname (#PCDATA)>
    <!ELEMENT cl_tax_id (#PCDATA)>

  <!ELEMENT invoice (inv_id, inv_date, inv_due_date?,
                     inv_currency?, inv_start_date, inv_end_date,
                     inv_desc?, inv_payment_terms?, inv_generic_discount?,
                     inv_total_net_due, extend_header*, matter+)>

    <!ELEMENT inv_id (#PCDATA)>
    <!ELEMENT inv_date (#PCDATA)>
    <!ELEMENT inv_due_date (#PCDATA)>
    <!ELEMENT inv_currency (#PCDATA)>
    <!ELEMENT inv_start_date (#PCDATA)>
    <!ELEMENT inv_end_date (#PCDATA)>
    <!ELEMENT inv_desc (#PCDATA)>
    <!ELEMENT inv_payment_terms (#PCDATA)>
    <!ELEMENT inv_generic_discount (#PCDATA)>
    <!ELEMENT inv_total_net_due (#PCDATA)>

  <!ELEMENT matter (cl_matter_id?, lf_matter_id, matter_name, matter_desc?,
                    lf_managing_contact_lname, lf_managing_contact_fname, lf_contact_id?,
```

Index

XSDs

XSDs define the elements and attributes that can appear in an XML document; elements that child elements, their order and number; whether an element is empty or includes text; data types for elements and attributes; default and fixed values for elements and attributes.

XSD Glossary

- **Attributes:** Provide extra information about elements. Attributes pull data from elements or attributes on which it is based.
  - All attributes are **Simple Attributes** and contain text only.
- **Constraints:** Limitations added to defined data types.
- **Elements:** The main building blocks of XML and HTML
  - **Simple Elements** contain text only, not any other elements or attributes.
  - **Complex Elements** contain other elements and/or attributes. Complex elements can:
    - Be empty elements
    - Contain only other elements
    - Contain only text
    - Contain both other elements and text
- **Schema Declaration or XSD Schema Element**: Identifies the document as XML Schema

XSD Structure

- XSD files include:
  - XML Declaration
  - Schema Declaration
  - Element Declarations
  - Attribute Declarations

XSD: XML and Schema Declarations

- XML Declaration - The version of XML and character set used
- Schema Declaration - Identifies the document as XML Schema and may contain attributes

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:tns="http://www.ledes.org/ledes20.xsd" xmlns="http://www.ledes.org/ledes20.xsd" elementFormDefault="qualified"
targetNamespace="http://www.ledes.org/ledes20.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">LEDES XML EBilling 2.0 Schema</xsd:documentation>
  </xsd:annotation>
```

XSD Element Declarations

- Simple Elements contain text only, not any other elements or attributes

<xs:elementname="country" type="xs:string" />

- Simple Elements can have a Default or Fixed Value
  - A Default Value is assigned to the element when no other value is specified

<xs:elementname="country" type="xs:string" default="United States" />

  - A Fixed Value is assigned to the element, and no other value can be specified

<xs:elementname="country" type="xs:string" fixed="United States" />

XSD Element Declarations

- Complex Elements contain other elements and/or attributes. Complex elements can:
  - Be empty elements
  - Contain only other elements
  - Contain only text
  - Contain both other elements and text

XSD Element Declarations

- There are 44 pre-defined XML data types

Common Data Types	Description
xs:string	String Data
xs:decimal	A Decimal Value
xs:integer	An Integer Value (whole numbers)
xs:boolean	A Boolean Value (true/false: 1/0)
xs:date	A Date Value
xs:time	A Time Value

XML Self-Defined Data Types

- Data Types can be self-defined
- Constraints (restrictions) can be added to Data Types to limit content or require data to match a specific pattern

```
<xsd:simpleType name="ledes_percent2">
 <xsd:restriction base="xsd:decimal">
  <xsd:totalDigits value="3" />
  <xsd:fractionDigits value="2" />
  <xsd:minInclusive value="0" />
  <xsd:maxInclusive value="1" />
 </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ledes_units">
 <xsd:restriction base="xsd:decimal">
  <xsd:totalDigits value="4" />
  <xsd:fractionDigits value="2" />
 </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="inv_payment_terms">
 <xsd:restriction base="xsd:string">
  <xsd:pattern value="[0-9][0-9]/[0-9][0-9]" />
 </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="matter_billing_type">
 <xsd:restriction base="xsd:string">
  <xsd:enumeration value="" />
  <xsd:enumeration value="TM" />
  <xsd:enumeration value="FF" />
  <xsd:enumeration value="CT" />
  <xsd:enumeration value="FS" />
 </xsd:restriction>
```

Other Element or Attribute Constraints

Constraint	Description
enumeration	Defines a list of acceptable values
franctionDigits	Specifies the maximum number of decimal places allowed. Must be equal to or greater than zero
length	Specifies the exact number of characters or list items allowed. Must be equal to or greater than zero
maxExclusive	Specifies the upper bounds for numeric values (the value must be less than this value)
maxInclusive	Specifies the upper bounds for numeric values (the value must be less than or equal to this value)
maxLength	Specifies the maximum number of characters or list items allowed. Must be greater than zero
minExclusive	Specifies the lower bounds for numeric values (the value must be greater than this value)

Other Element or Attribute Constraints

Constraint	Description
minInclusive	Specifies the lower bounds for numeric values (the value must be greater than or equal to this value
minLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero
pattern	Defines the exact sequence of characters that are acceptable
totalDigits	Specifies the exact number of digits allowed. Must be greater than zero
whiteSpace	Specifies how white space (line feeds, tabs, spaces and carriage returns) is handled

XSD Element Declaration

```
<xsd:element name="ledesxmlebilling2.0">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element minOccurs="1" maxOccurs="1" name="firm" type="tns:firm" />
   <xsd:element minOccurs="0" maxOccurs="unbounded" name="tax" type="tns:tax" />
  </xsd:sequence>
 </xsd:complexType>
 <xsd:key name="tax_id_key">
  <xsd:selector xpath="tns:tax" />
  <xsd:field xpath="tns:tx_id" />
 </xsd:key>
 <xsd:keyref name="tax_id_ref" refer="tns:tax_id_key">
  <xsd:selector xpath=".//*" />
  <xsd:field xpath="tns:tx_id" />
 </xsd:keyref>
 <xsd:unique name="file_item_nbr_unique">
  <xsd:selector xpath=".//*" />
  <xsd:field xpath="tns:file_item_nbr" />
 </xsd:unique>
</xsd:element>
<xsd:complexType name="firm">
 <xsd:sequence>
  <xsd:element minOccurs="1" maxOccurs="1" name="lf_vendor_id" type="tns:string25" />
  <xsd:element minOccurs="0" maxOccurs="1" name="lf_id" type="tns:string20" />
  <xsd:element minOccurs="1" maxOccurs="1" name="lf_name" type="tns:string60" />
  <xsd:element minOccurs="1" maxOccurs="1" name="lf_address" type="tns:address_info" />
  <xsd:element minOccurs="0" maxOccurs="1" name="lf_remit_address" type="tns:address_info" />
  <xsd:element minOccurs="1" maxOccurs="1" name="lf_billing_contact" type="tns:contact_info" />
  <xsd:element minOccurs="1" maxOccurs="1" name="source_app" type="tns:string25" />
  <xsd:element minOccurs="1" maxOccurs="1" name="app_version" type="tns:string10" />
  <xsd:element minOccurs="0" maxOccurs="1" name="firm_URL" type="tns:string50" />
  <xsd:element minOccurs="1" maxOccurs="1" name="file_item_nbr" type="xsd:nonNegativeInteger" />
  <xsd:element minOccurs="0" maxOccurs="1" name="extend_header" type="tns:extend_header" />
  <xsd:element minOccurs="1" maxOccurs="unbounded" name="client" type="tns:client" />
 </xsd:sequence>
```

XSD Attribute Declarations

- Attributes pull data from elements or attributes on which it is based
- All Attributes are Simple Attributes

<xs:attribute name=”xxx” type=“yyy” />

- Common Attribute Data Types:

Common Attribute Data Types	Description
xs:string	String Data
xs:decimal	A Decimal Value
xs:integer	An Integer Value (whole numbers)
xs:boolean	A Boolean Value (true/false: 1/0)
xs:date	A Date Value
xs:time	A Time Value

XSD Attribute Declarations

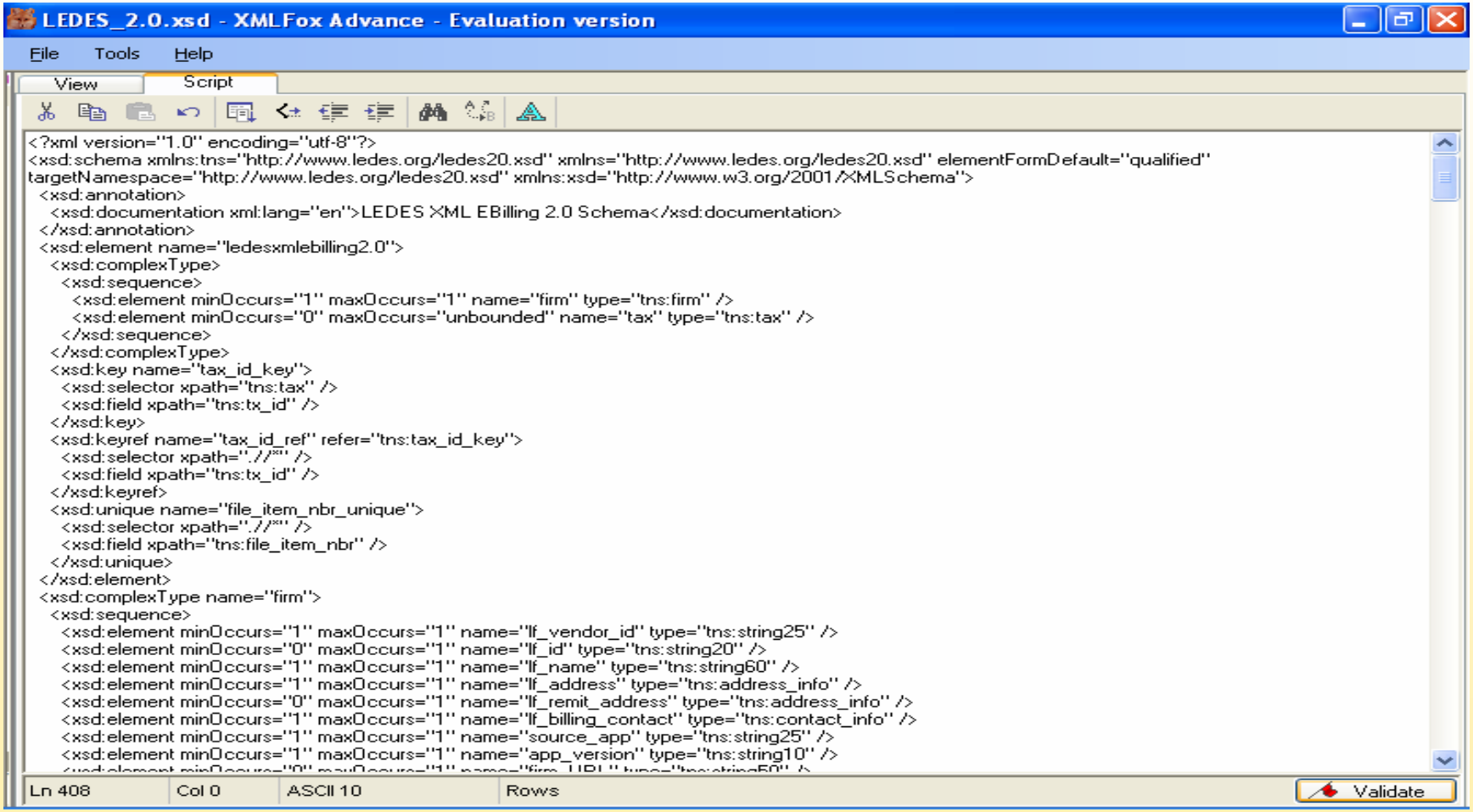
- Since an attribute pulls data from other elements or attributes, it is possible that the data types of the elements or attributes on which it is based could conflict with the attribute's defined data type. In this situation the data will not validate.
- Like elements,
  - You can restrict attribute data types to limit its content or require data to match a specific pattern
  - Attributes can have a Default or Fixed Value
- Attributes are optional by default. To require an Attribute:

```
<xs:attribute name="Country" type="xs:string" use="required" />
```

To Edit an XSD

- Use an XML Editor
  - XML Spy, XMLFox
  - Peter's XML Editor; for other free XSD Editors search Google
  - Microsoft XML Editor
- Can use Notepad

XML Ebilling Ver. 2 XSD



Index

There is More to XML Than Just XML, DTD and XSD Files

XSLTs

- XSLT: Extensible Style Sheet Language Transformation. Provides rules for converting data described by one set of tags into data described by another set of tags.
  - Can be used to move data into database fields or web page content
  - Can be used to sort and filter data

- For LEDES ebilling purposes, it is not necessary to understand XML beyond XML, DTD* and XSD files

* Used in LEDES 2000 only

Index

LEDES Ebilling Formats

Early Ebilling Formats

- 1995: Price Waterhouse, LLP convened a consortium of leading legal industry time and billing and case management system vendors in order to define a standard electronic billing format for use by the legal industry. Two ebilling formats were developed:
  - The ASC X12 EDI standard (the national standard for EDI (Electronic Data Interchange), widely used in several industries). To our knowledge this format has never been used in the legal industry due largely to its enormous syntactic complexity. At least one custom format, however, was based on the ASC X12 format.
  - A delimited ASCII standard. This standard was used, in slightly modified form, by several organizations.
- 1998: PricewaterhouseCoopers reconvened the legal industry consortium using the framework established in 1995 and updated the delimited ASCII standard. The informal LEDES group was created and assumed control of the newly-named LEDES 1998 format.

LEDES 98B History

- 1999: The LEDES 1998B format modified and wholly replaced the ASCII, pipe delimited LEDES 1998 format.
- LEDES 1998B quickly became the accepted standard for ebilling and remains the most widely-used ebilling format today.

- ASCII, pipe delimited format with fixed field lengths
- Flat file database structure
- Allows for some adjustments to invoice:
  - Flat adjustments to fee or expense line items
  - Flat adjustments to fees or expenses at invoice level

LEDES 98B Fields

Field Label	Type	Required/Optional
INVOICE_DATE	Date * 8 YYYYMMDD	Required
INVOICE_NUMBER	Character * 20	Required
CLIENT_ID	Character * 20	Required
LAW_FIRM_MATTER_ID	Character * 20	Required
INVOICE_TOTAL	Currency * 12.4	Required
BILLING_START_DATE	Date * 8 YYYYMMDD	Required
BILLING_END_DATE	Date * 8 YYYYMMDD	Required
INVOICE_DESCRIPTION	Character * 15 KB	Optional
LINE_ITEM_NUMBER	Character * 20	Required
EXP/FEE/INV_ADJ_TYPE	Character * 2	Required
LINE_ITEM_NUMBER_OF_UNITS	Numeric*10.4	Required except if an Invoice Level Adjustment
LINE_ITEM_ADJUSTMENT_AMOUNT	Currency * 10.4	Optional
LINE_ITEM_TOTAL	Currency * 10.4	Required
LINE_ITEM_DATE	Date * 8 YYYYMMDD	Required
LINE_ITEM_TASK_CODE	Character * 20	Required for Task Items
LINE_ITEM_EXPENSE_CODE	Character * 20	Required for Expense Items
LINE_ITEM_ACTIVITY_CODE	Character * 20	Required for Task Items
TIMEKEEPER_ID	Character * 20	Required for Task Items
LINE_ITEM_DESCRIPTION	Character * 15 KB	Required for Task Items
LAW_FIRM_ID	Character * 20	Required
LINE_ITEM_UNIT_COST	Currency * 10.4	Required except if an Invoice Level Adjustment
TIMEKEEPER_NAME	Character * 30	Required for Task Items
TIMEKEEPER_CLASSIFICATION	Character * 10	Required for Task Items
CLIENT_MATTER_ID	Character * 20	Required except if Client does not assign Matter Ids

Issues with LEDES 98B

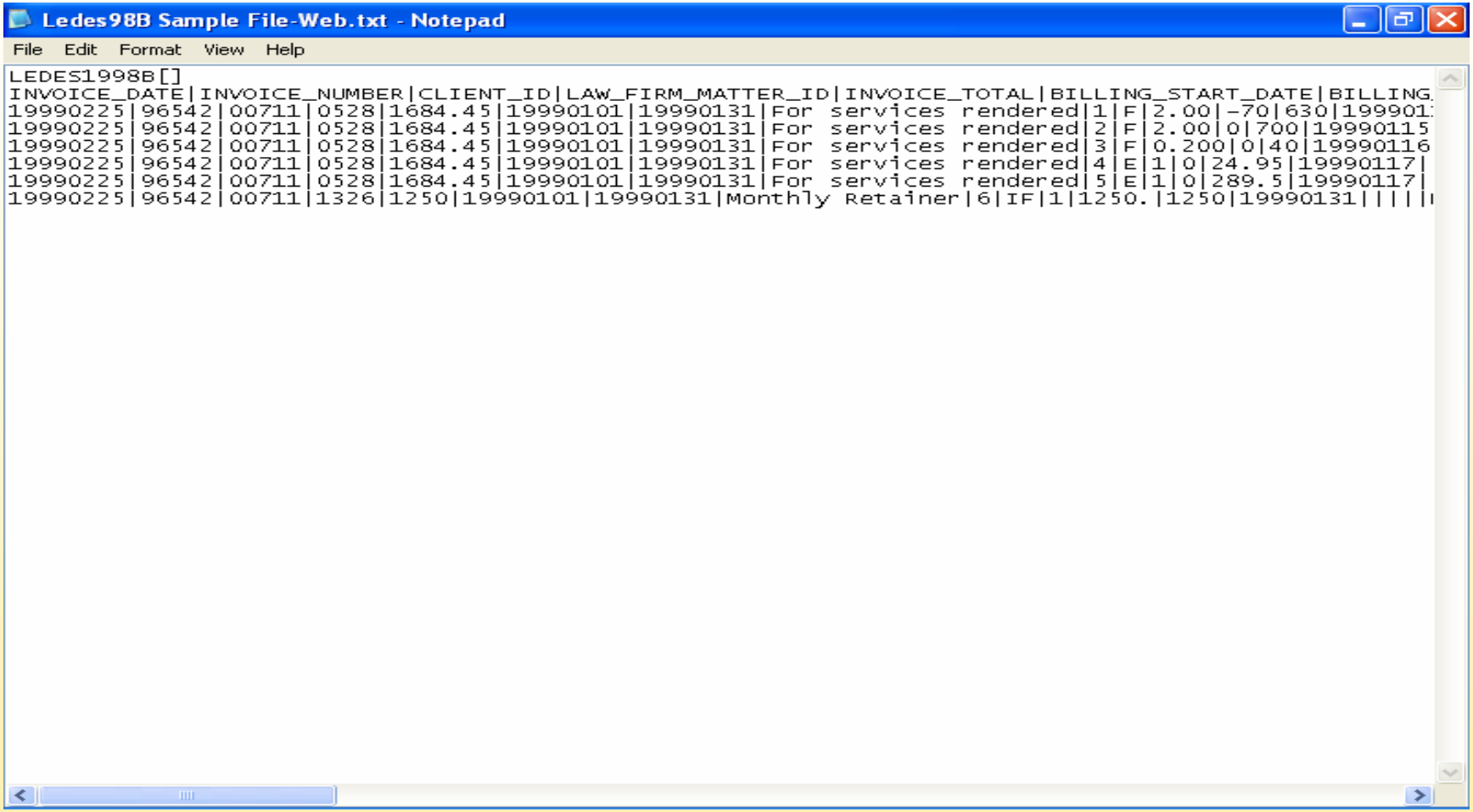
- Format contains only 24 fields
- Does not allow for the submission of
  - Taxes on legal fees
  - Split bills
  - Alternate Fee Arrangements
- Customization of format by clients/inconsistent mapping of field data

INVOICE_DATE
INVOICE_NUMBER
CLIENT_ID
LAW_FIRM_MATTER_ID
INVOICE_TOTAL
BILLING_START_DATE
BILLING_END_DATE
INVOICE_DESCRIPTION
LINE_ITEM_NUMBER
EXP/FEE/INV_ADJ_TYPE
LINE_ITEM_NUMBER_OF_UNITS
LINE_ITEM_ADJUSTMENT_AMOUNT
LINE_ITEM_TOTAL
LINE_ITEM_DATE
LINE_ITEM_TASK_CODE
LINE_ITEM_EXPENSE_CODE
LINE_ITEM_ACTIVITY_CODE
TIMEKEEPER_ID
LINE_ITEM_DESCRIPTION
LAW_FIRM_ID
LINE_ITEM_UNIT_COST
TIMEKEEPER_NAME
TIMEKEEPER_CLASSIFICATION
CLIENT_MATTER_ID

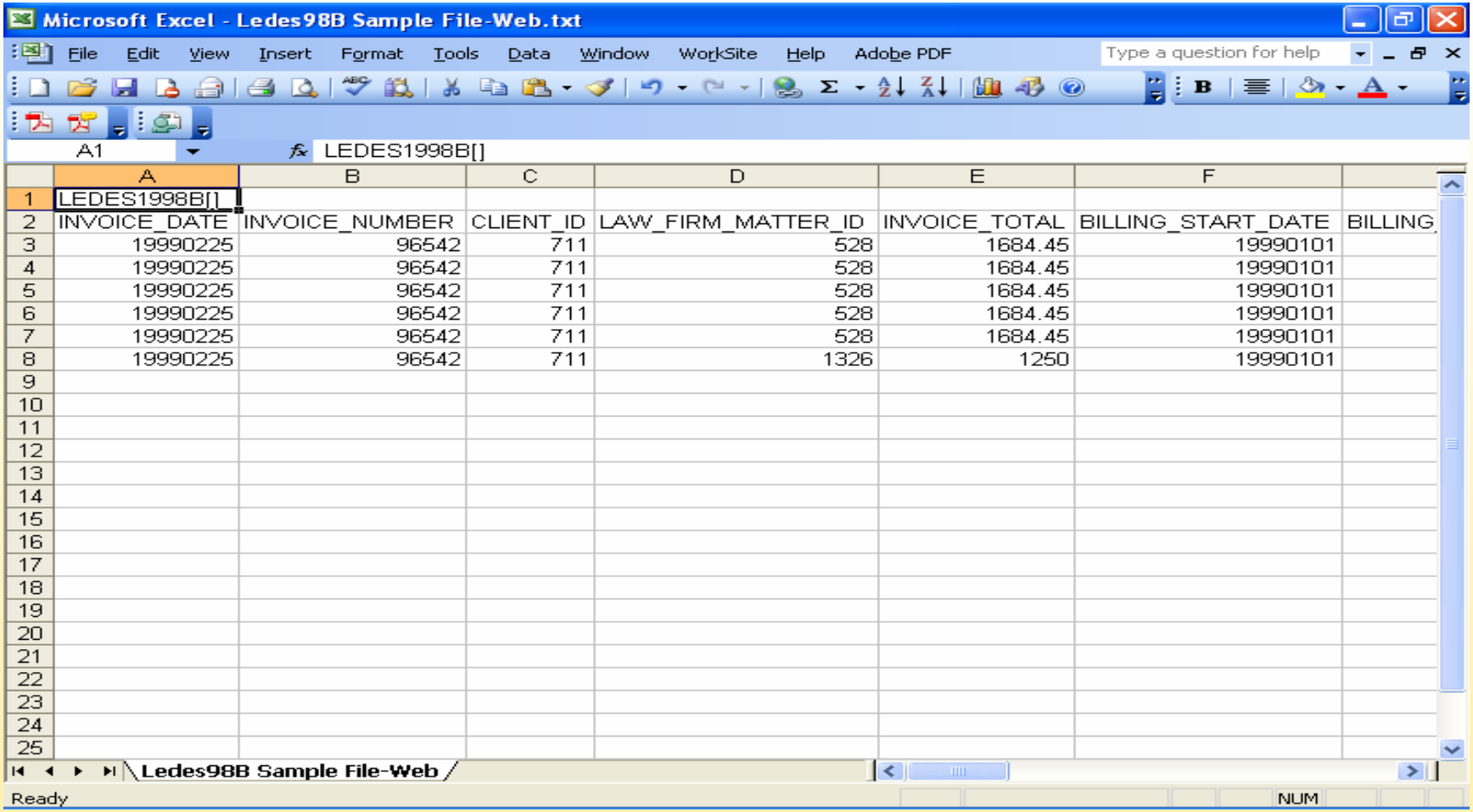
Editing LEDES 98B Files

- Any ASCII editor
- Notepad and Wordpad
  - Under Options set Word Wrap to “No Wrap”
- Open in Excel as a spreadsheet
  - Do not edit and then save in Excel as ASCII text. Commas in field data will throw off the alignment of pipes.
- eBillingHub’s LEDES Toolkit™ available free at http://www.ebillinghub.com/ver2/Solutions_Main.aspx

LEDES 98B File In Notepad

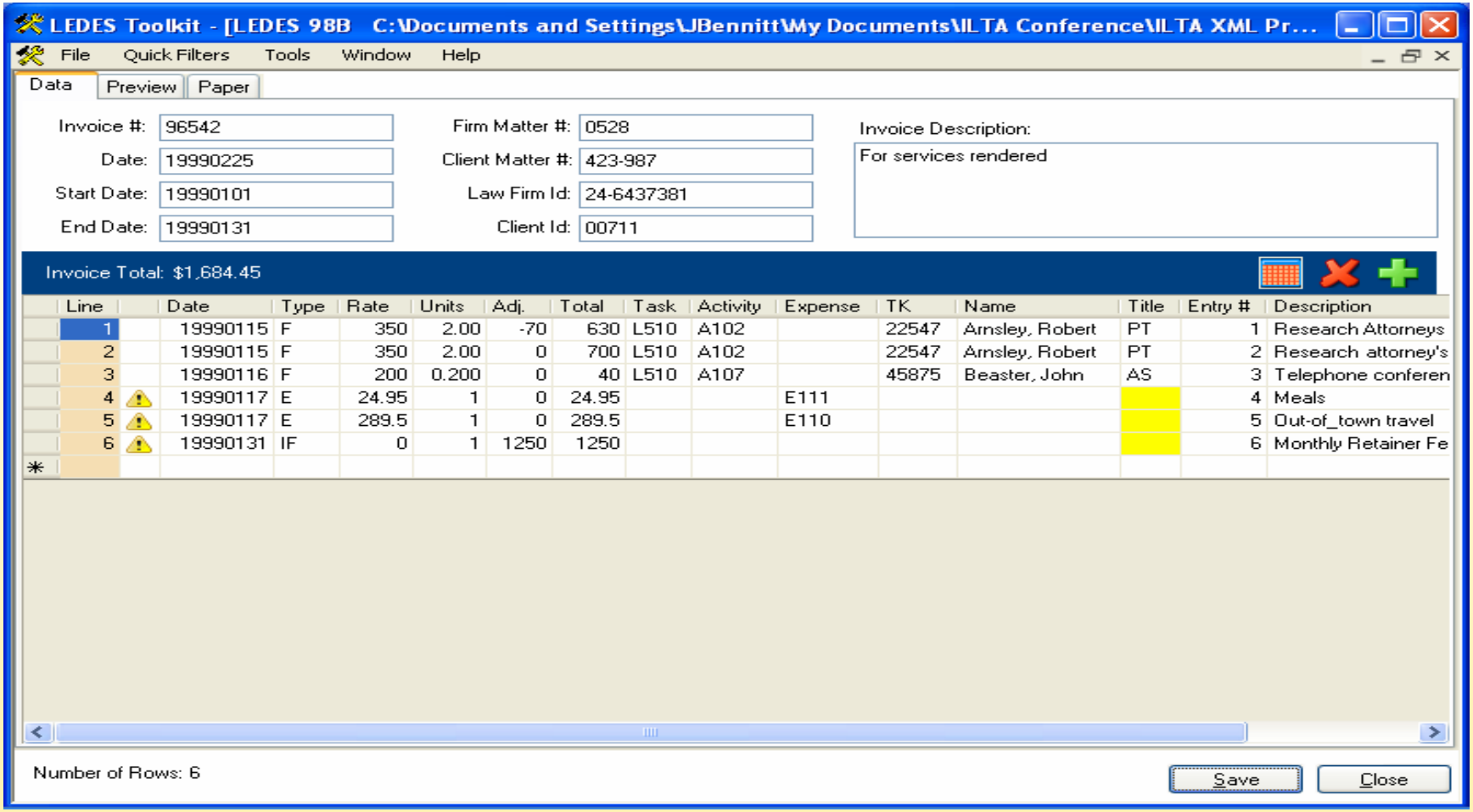


LEDES 98B File In Excel



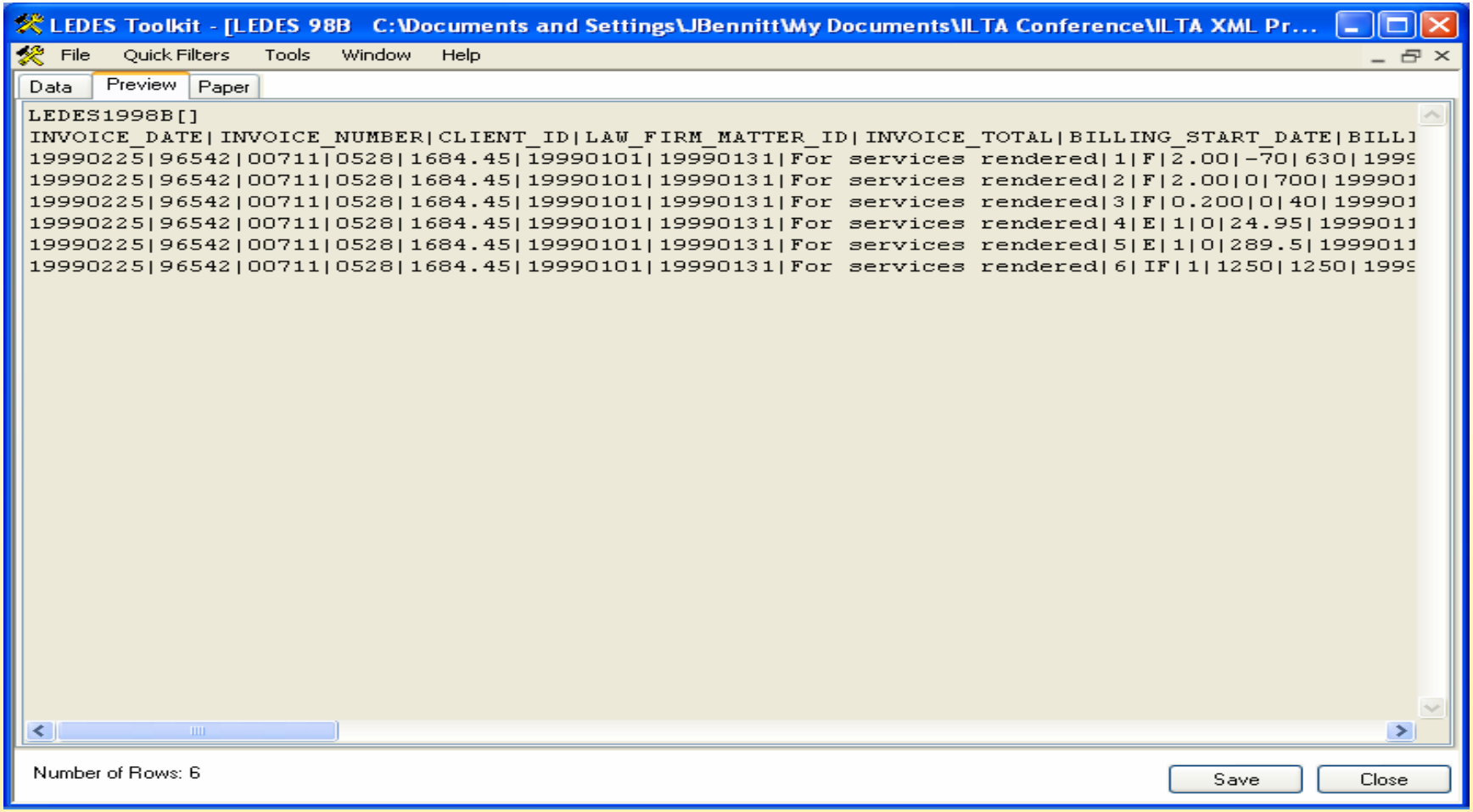
	A	B	C	D	E	F	
1	LEDES1998B[]						
2	INVOICE_DATE	INVOICE_NUMBER	CLIENT_ID	LAW_FIRM_MATTER_ID	INVOICE_TOTAL	BILLING_START_DATE	BILLING
3	19990225	96542	711	528	1684.45	19990101	
4	19990225	96542	711	528	1684.45	19990101	
5	19990225	96542	711	528	1684.45	19990101	
6	19990225	96542	711	528	1684.45	19990101	
7	19990225	96542	711	528	1684.45	19990101	
8	19990225	96542	711	1326	1250	19990101	
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							

LEDES 98B File In LEDES Toolkit™

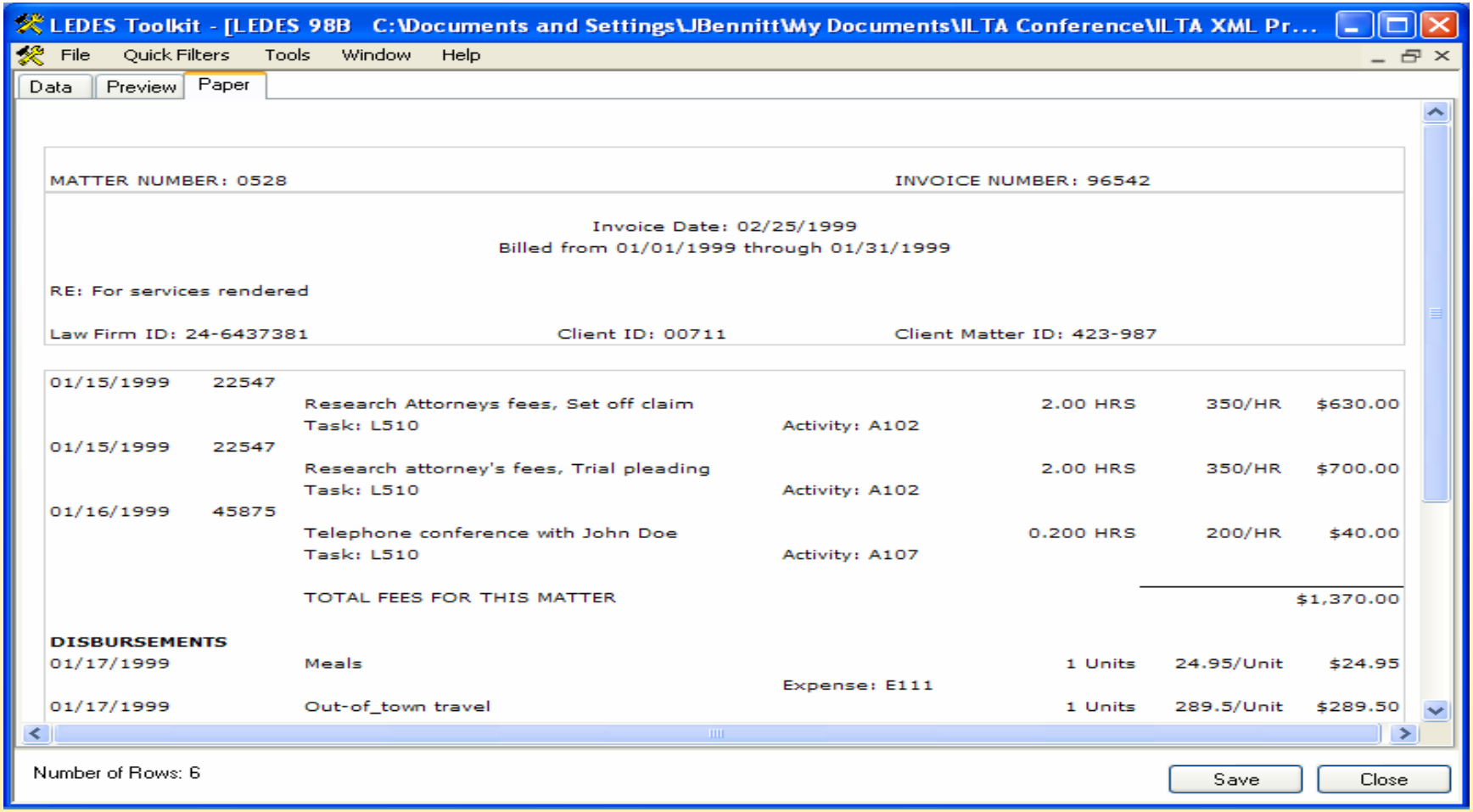


Line		Date	Type	Rate	Units	Adj.	Total	Task	Activity	Expense	TK	Name	Title	Entry #	Description
1		19990115	F	350	2.00	-70	630	L510	A102		22547	Arnsley, Robert	PT	1	Research Attorneys
2		19990115	F	350	2.00	0	700	L510	A102		22547	Arnsley, Robert	PT	2	Research attorney's
3		19990116	F	200	0.200	0	40	L510	A107		45875	Beaster, John	AS	3	Telephone conferen
4	⚠	19990117	E	24.95	1	0	24.95			E111				4	Meals
5	⚠	19990117	E	289.5	1	0	289.5			E110				5	Out-of_town travel
6	⚠	19990131	IF	0	1	1250	1250							6	Monthly Retainer Fe

LEDES 98B File In LEDES Toolkit™



LEDES 98B File In LEDES Toolkit™



Index

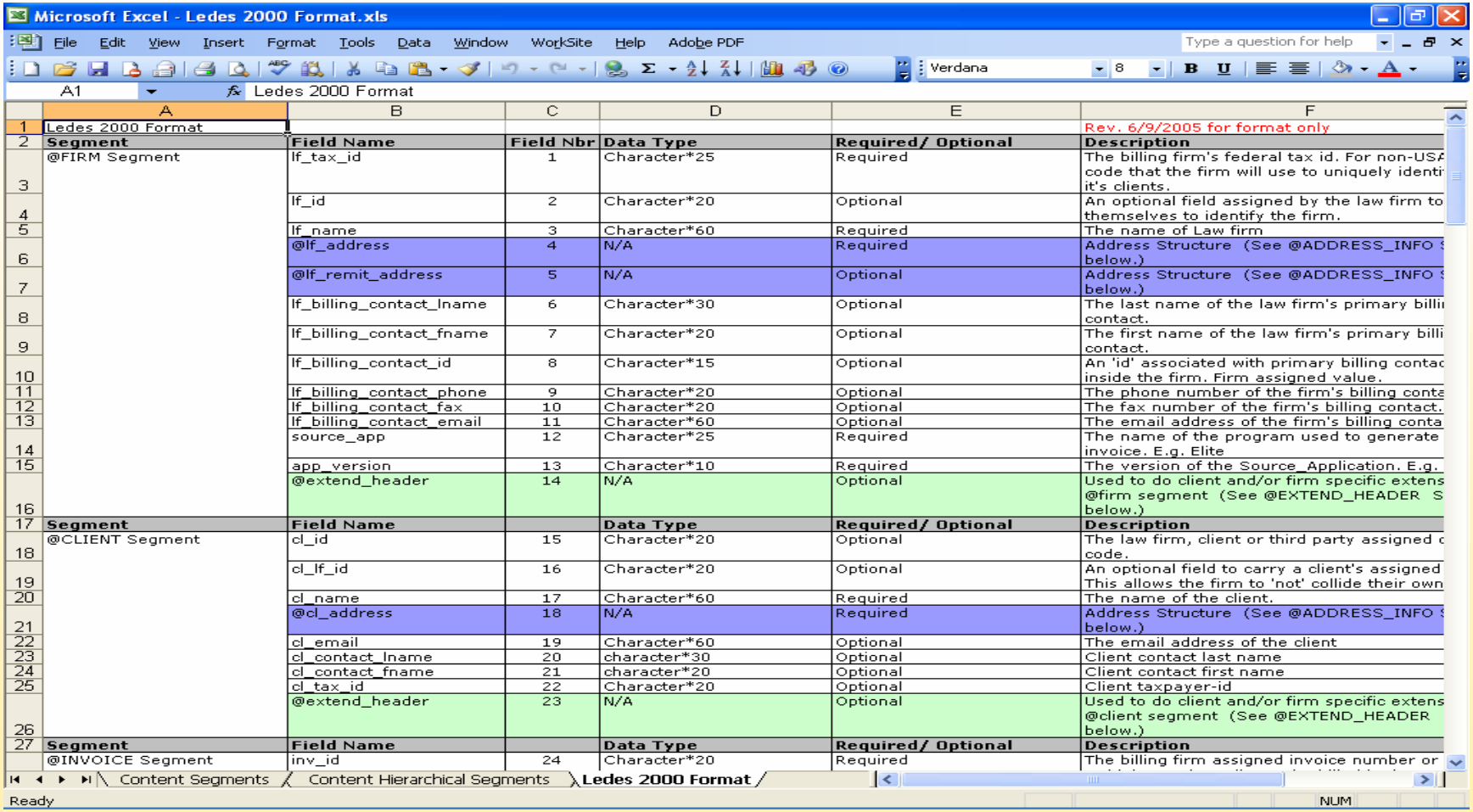
LEDES 2000 History

- 2000: This XML ebilling standard was ratified by the LOC. The LEDES 2000 format contained much more information than the original LEDES 1998B format and more fully accommodated the complex nature of fee arrangements between corporations and law firms.
- 2004 and 2005: Corrections were made to fix math issues.
- 2006: LEDES 2000 was retired with the ratification of the LEDES XML Ebilling Ver. 2 format.

LEDES 2000 Format

- XML based format
- Relational database structure
- Data falls in FIRM, CLIENT, INVOICE, MATTER, TKSUM, FEE, EXPENSE, ADDRESS_INFO, EXTEND_HEADER and EXTEND_DATA segments.
- Contains 125 fields; richer data structure than 98B
- Identifies currency for invoice
- Allows for some adjustments to invoice:
  - Percent or flat adjustments to fee or expense line items
  - Percent shared on fees or expenses at @MATTER level
  - Invoice payment terms at @INVOICE level
  - Generic discount at @INVOICE level

LEDES 2000 Format



Microsoft Excel - Ledes 2000 Format.xls

File Edit View Insert Format Tools Data Window WorkSite Help Adobe PDF

Type a question for help

Verdana 8

A1 Ledes 2000 Format

	A	B	C	D	E	F
1	Ledes 2000 Format					Rev. 6/9/2005 for format only
2	**Segment**	**Field Name**	**Field Nbr**	**Data Type**	**Required/ Optional**	**Description**
3	@FIRM Segment	lf_tax_id	1	Character*25	Required	The billing firm's federal tax id. For non-USA code that the firm will use to uniquely identi it's clients.
4		lf_id	2	Character*20	Optional	An optional field assigned by the law firm to themselves to identify the firm.
5		lf_name	3	Character*60	Required	The name of Law firm
6		@lf_address	4	N/A	Required	Address Structure (See @ADDRESS_INFO below.)
7		@lf_remit_address	5	N/A	Optional	Address Structure (See @ADDRESS_INFO below.)
8		lf_billing_contact_lname	6	Character*30	Optional	The last name of the law firm's primary billi contact.
9		lf_billing_contact_fname	7	Character*20	Optional	The first name of the law firm's primary billi contact.
10		lf_billing_contact_id	8	Character*15	Optional	An 'id' associated with primary billing contac inside the firm. Firm assigned value.
11		lf_billing_contact_phone	9	Character*20	Optional	The phone number of the firm's billing conta
12		lf_billing_contact_fax	10	Character*20	Optional	The fax number of the firm's billing contact.
13		lf_billing_contact_email	11	Character*60	Optional	The email address of the firm's billing conta
14		source_app	12	Character*25	Required	The name of the program used to generate invoice. E.g. Elite
15		app_version	13	Character*10	Required	The version of the Source_Application. E.g.
16		@extend_header	14	N/A	Optional	Used to do client and/or firm specific extens @firm segment (See @EXTEND_HEADER S below.)
17	**Segment**	**Field Name**		**Data Type**	**Required/ Optional**	**Description**
18	@CLIENT Segment	cl_id	15	Character*20	Optional	The law firm, client or third party assigned c code.
19		cl_lf_id	16	Character*20	Optional	An optional field to carry a client's assigned This allows the firm to 'not' collide their own
20		cl_name	17	Character*60	Required	The name of the client.
21		@cl_address	18	N/A	Required	Address Structure (See @ADDRESS_INFO below.)
22		cl_email	19	Character*60	Optional	The email address of the client
23		cl_contact_lname	20	character*30	Optional	Client contact last name
24		cl_contact_fname	21	character*20	Optional	Client contact first name
25		cl_tax_id	22	Character*20	Optional	Client taxpayer-id
26		@extend_header	23	N/A	Optional	Used to do client and/or firm specific extens @client segment (See @EXTEND_HEADER below.)
27	**Segment**	**Field Name**		**Data Type**	**Required/ Optional**	**Description**
	@INVOICE Segment	inv_id	24	Character*20	Required	The billing firm assigned invoice number or

Content Segments | Content Hierarchical Segments | Ledes 2000 Format

Ready NUM

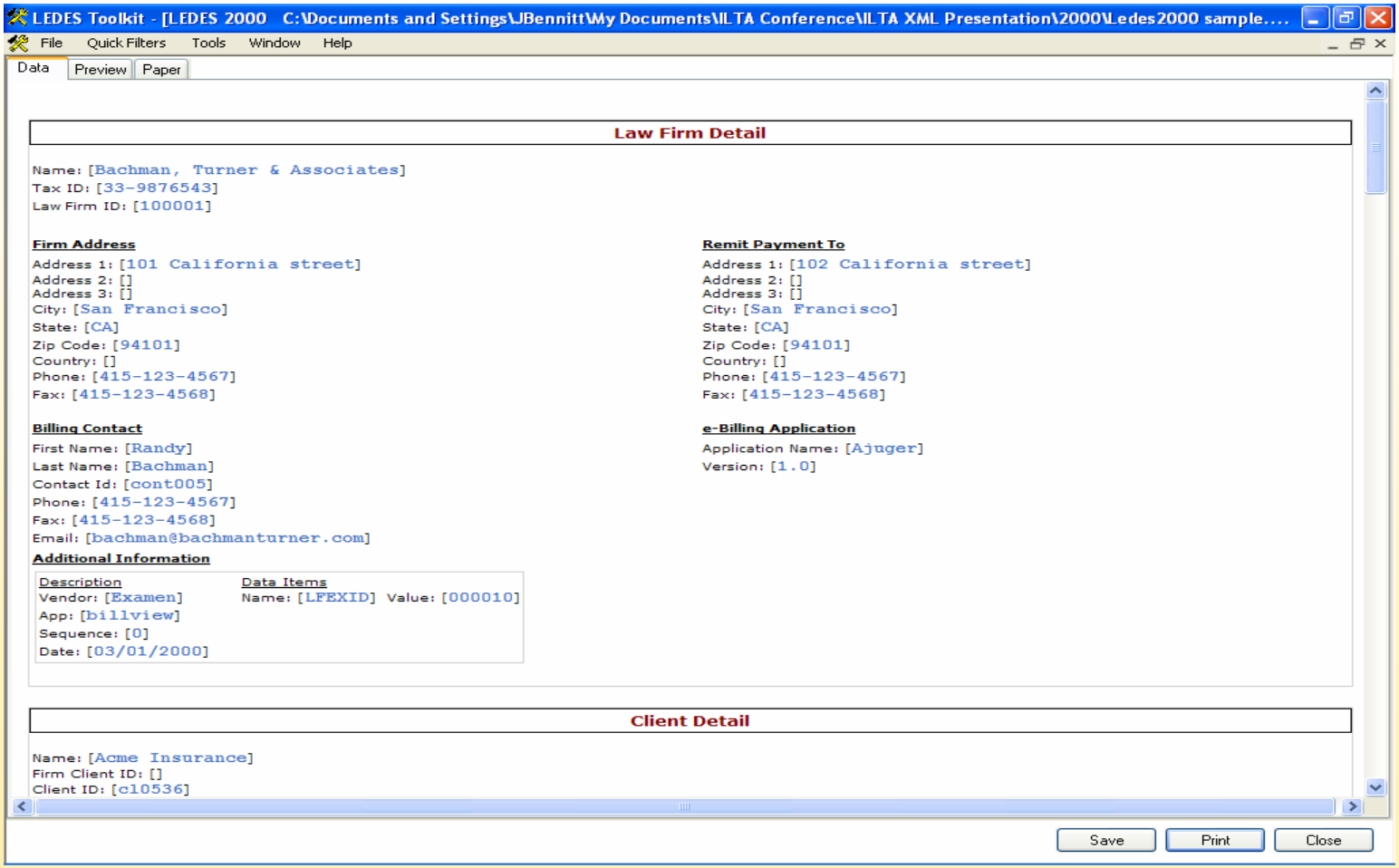
Issues with LEDES 2000

- Contains non-standard ebilling fields
- Inconsistent field definition and data types
- Taxes itemized on line item; leads to math errors
- Issues with math
- Does not allow for itemization if more than one tax applied to line item
- Limited ability to make adjustments
- Does not support Alternate Fee Arrangements
- Requires DTD, considered “old” technology
- Many required elements for international ebilling not supported by base format

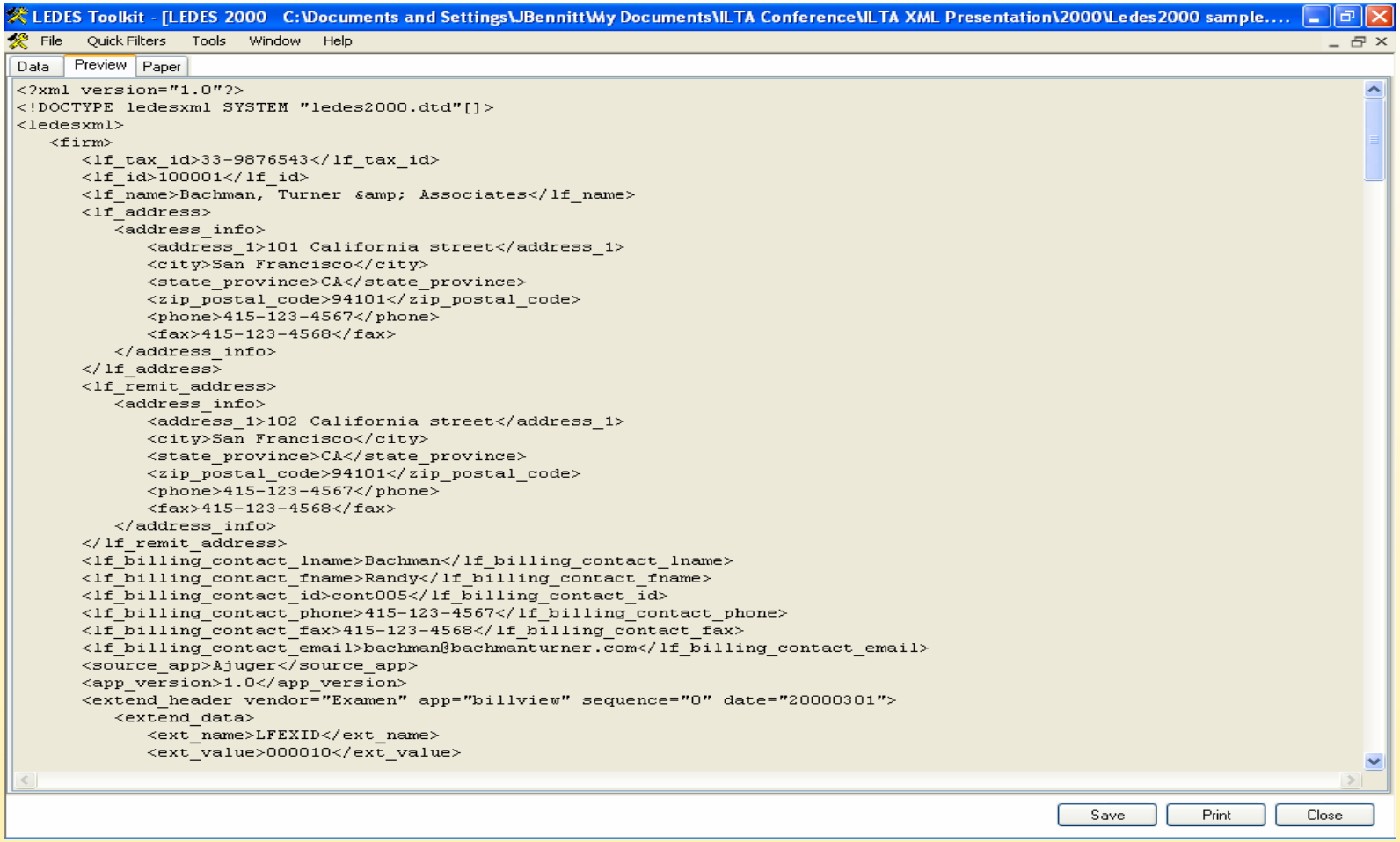
LEDES 2000 Files

- Can use any XML Editor
- Can be viewed using XML parser or eBillingHub's LEDES Toolkit™

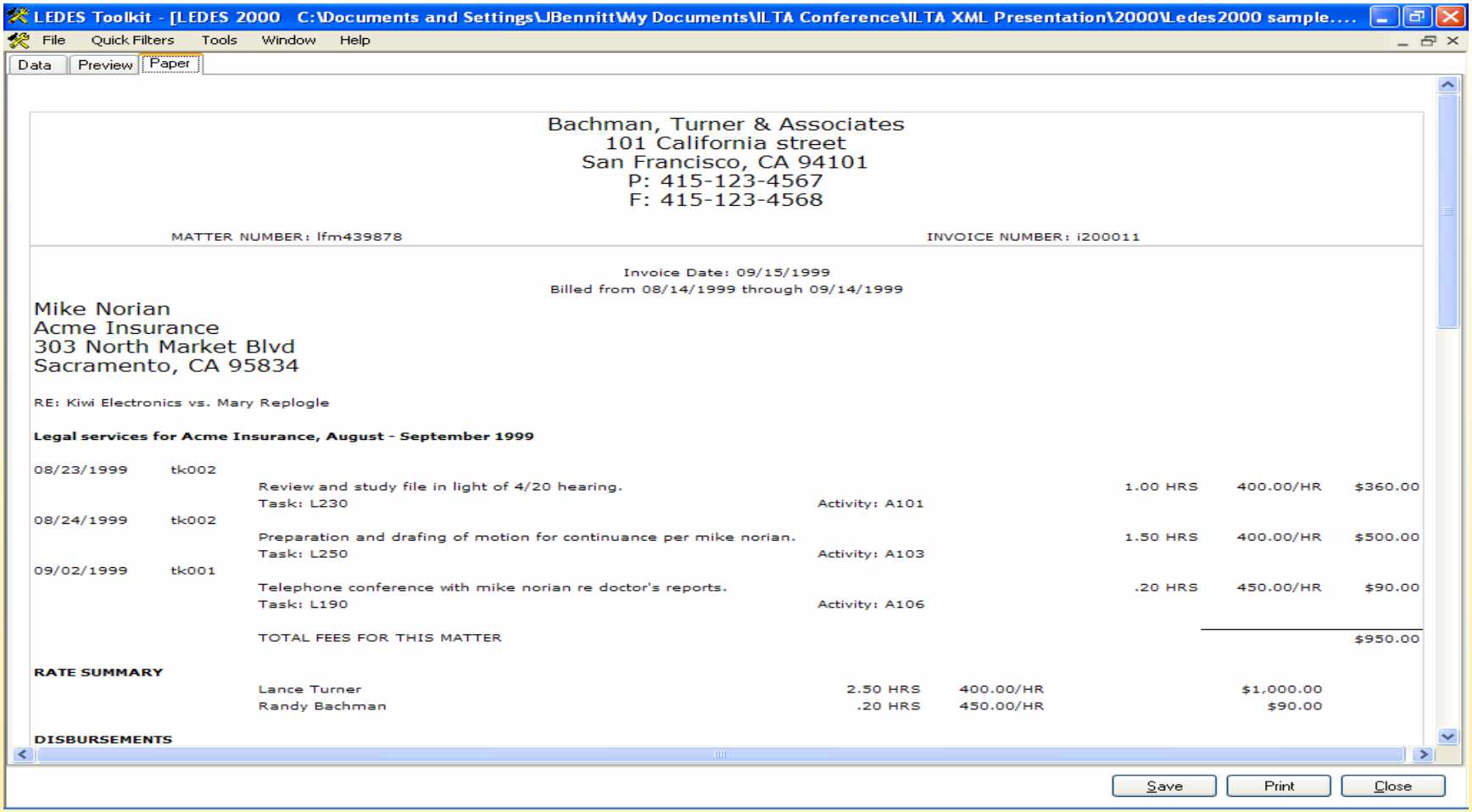
LEDES 2000 File in LEDES Toolkit™



LEDES 2000 File in LEDES Toolkit™



LEDES 2000 File in LEDES Toolkit™

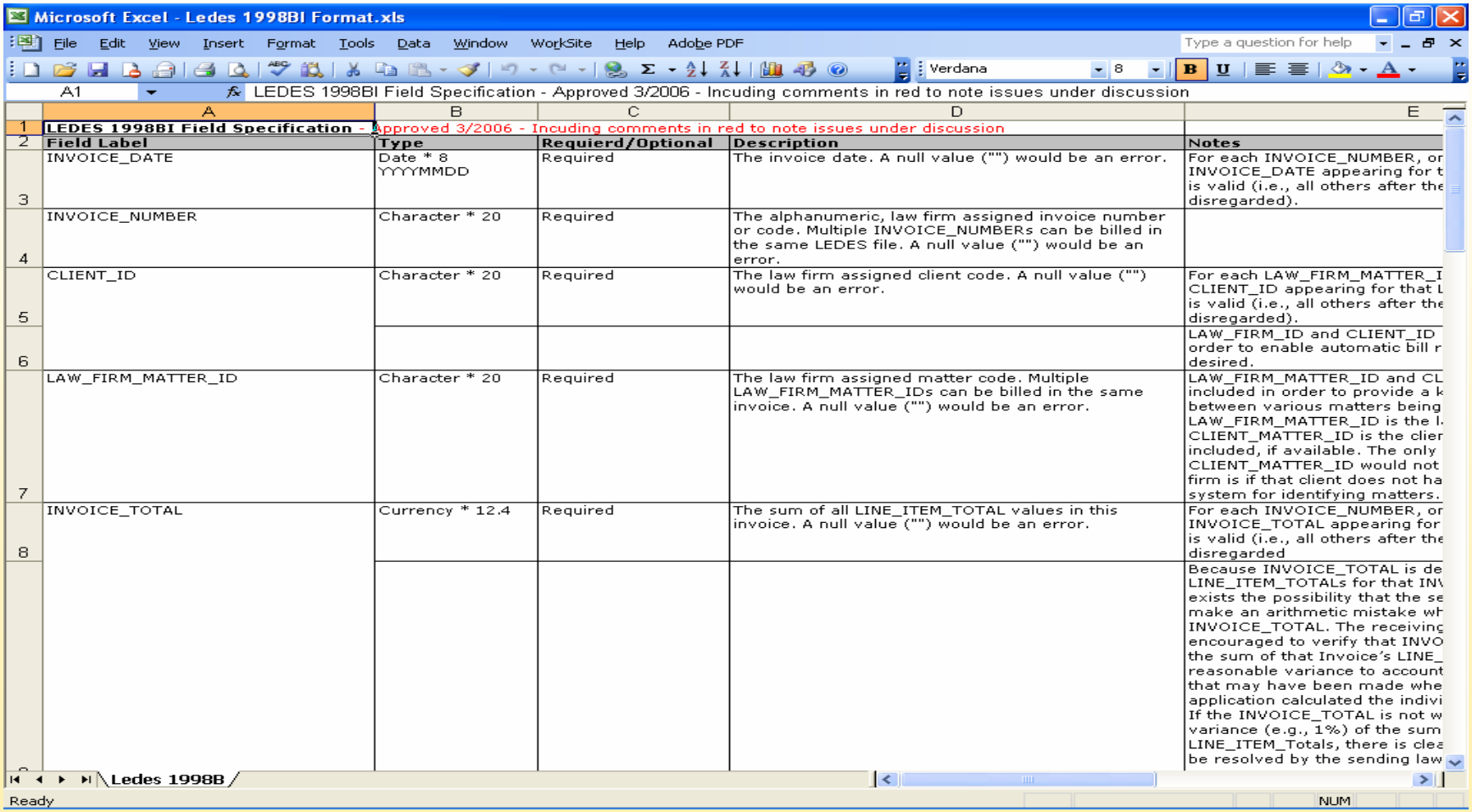


Index

LEDES 98BI History

- 2004: LITIG (Legal IT Innovators Group), a UK-based group proposed an ebilling standard for the United Kingdom's legal service industry based on the LEDES 1998B format.
- 2005: After several meetings and consultation with LITIG, the LOC approved a beta international version of the LEDES 1998B format.
- 2006: The beta format was revised to include additional fields necessary for international ebilling to proceed and was ratified by the LOC membership as a (non-beta) standard.
- It is not intended that the LEDES 1998B-International format will be modified in the future; any international submission issues will be considered through modification of the LEDES XML Ebilling Ver. 2 format only.
- There is currently a request on the table to clarify the definition of two 98BI fields and the addition of one field.

LEDES 98BI Format



Microsoft Excel - Ledes 1998BI Format.xls

File Edit View Insert Format Tools Data Window WorkSite Help Adobe PDF

A1: LEDES 1998BI Field Specification - Approved 3/2006 - Incuding comments in red to note issues under discussion

	A	B	C	D	E
1	LEDES 1998BI Field Specification - Approved 3/2006 - Incuding comments in red to note issues under discussion				
2	Field Label	Type	Requierd/Optional	Description	Notes
3	INVOICE_DATE	Date * 8 YYYYMMDD	Required	The invoice date. A null value ("") would be an error.	For each INVOICE_NUMBER, or INVOICE_DATE appearing for t is valid (i.e., all others after the disregarded).
4	INVOICE_NUMBER	Character * 20	Required	The alphanumeric, law firm assigned invoice number or code. Multiple INVOICE_NUMBERs can be billed in the same LEDES file. A null value ("") would be an error.	
5	CLIENT_ID	Character * 20	Required	The law firm assigned client code. A null value ("") would be an error.	For each LAW_FIRM_MATTER_I CLIENT_ID appearing for that L is valid (i.e., all others after the disregarded).
6					LAW_FIRM_ID and CLIENT_ID order to enable automatic bill r desired.
7	LAW_FIRM_MATTER_ID	Character * 20	Required	The law firm assigned matter code. Multiple LAW_FIRM_MATTER_IDs can be billed in the same invoice. A null value ("") would be an error.	LAW_FIRM_MATTER_ID and CL included in order to provide a k between various matters being LAW_FIRM_MATTER_ID is the l CLIENT_MATTER_ID is the clier included, if available. The only CLIENT_MATTER_ID would not firm is if that client does not ha system for identifying matters.
8	INVOICE_TOTAL	Currency * 12.4	Required	The sum of all LINE_ITEM_TOTAL values in this invoice. A null value ("") would be an error.	For each INVOICE_NUMBER, or INVOICE_TOTAL appearing for is valid (i.e., all others after the disregarded
					Because INVOICE_TOTAL is de LINE_ITEM_TOTALs for that INV exists the possibility that the se make an arithmetic mistake wh INVOICE_TOTAL. The receiving encouraged to verify that INVO the sum of that Invoice's LINE_ reasonable variance to account that may have been made whe application calculated the indivi If the INVOICE_TOTAL is not w variance (e.g., 1%) of the sum LINE_ITEM_Totals, there is clea be resolved by the sending law

Ledes 1998B

Ready NUM

Editing LEDES 98BI Files

- Any ASCII editor
- Notepad and Wordpad
  - Under View/Options, set Word Wrap to “No Wrap”
- Open in Excel as a spreadsheet
  - Do not edit and then save in Excel as ASCII text. Commas in field data will throw off the alignment of pipes

LEDES 98BI

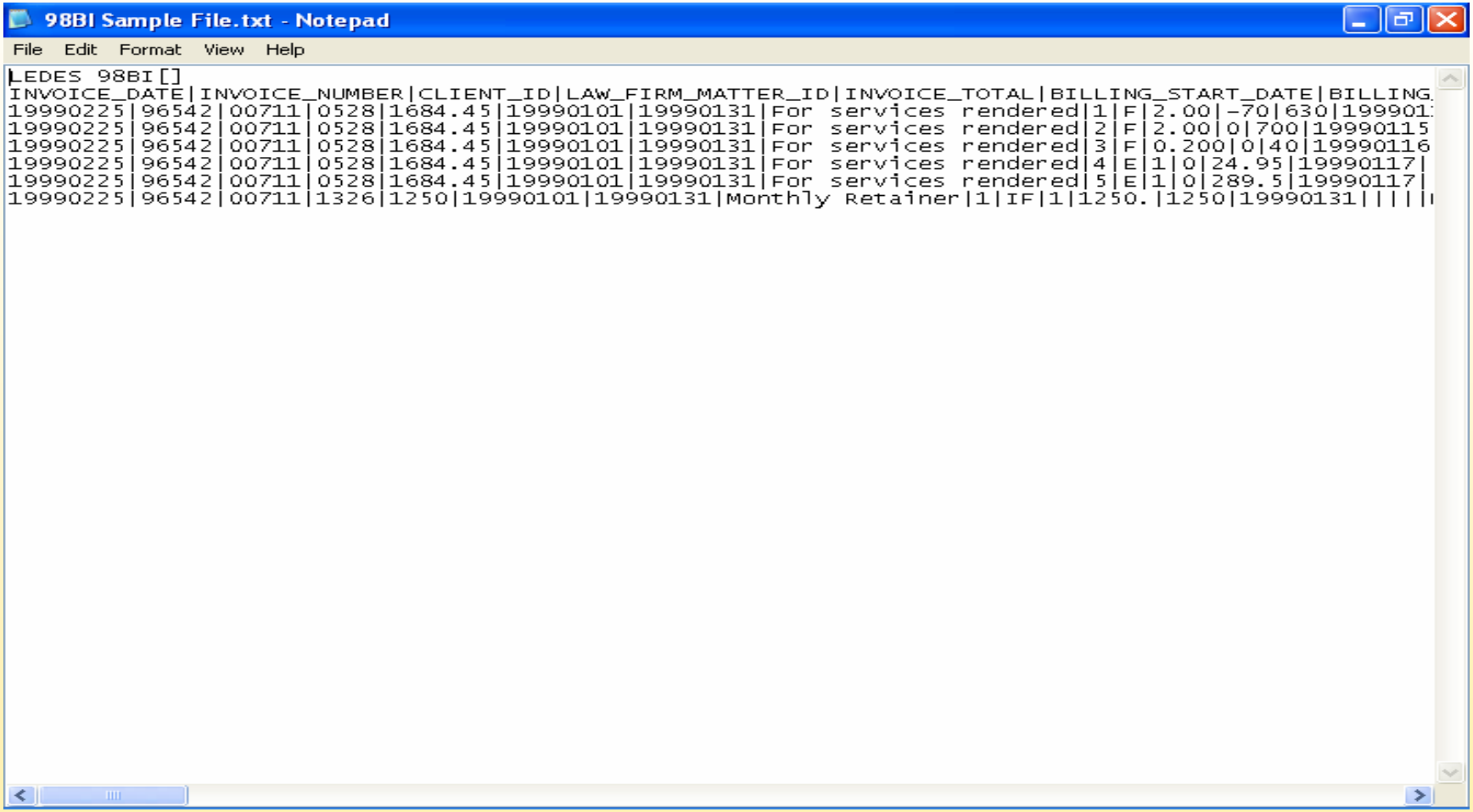
Benefits

- ASCII format is easier to develop and learn
- Includes field for total Tax charged on invoice
- Includes field for Purchase Order Number

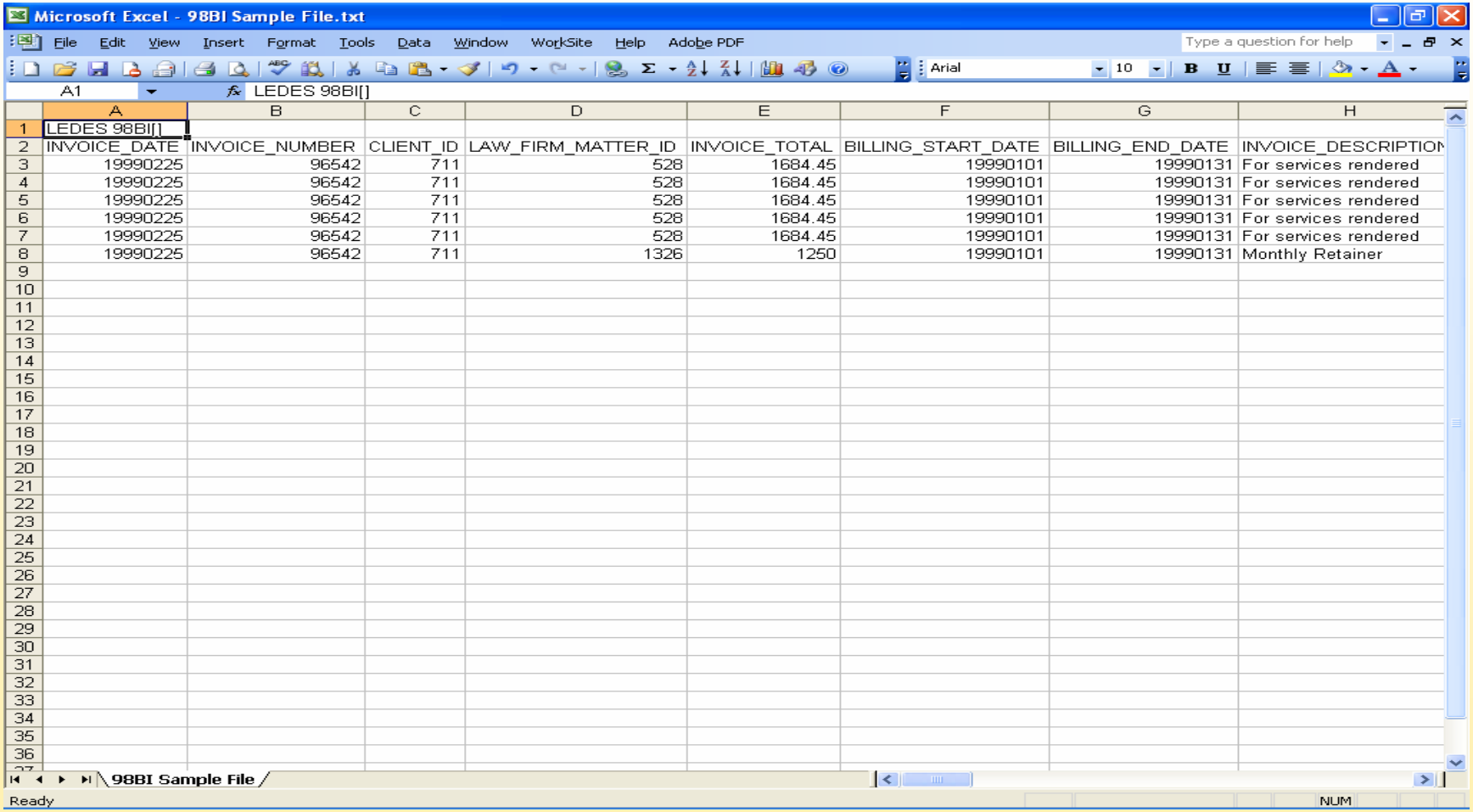
Limitations

- Does not itemize taxes if more than one tax is assessed against invoice
- Does not allow for the submission of
  - Split bills
  - Alternate Fee Arrangements

LEDES 98BI File in Notepad



LEDES 98BI File in Excel



	A	B	C	D	E	F	G	H
1	LEDES 98BI[]							
2	INVOICE_DATE	INVOICE_NUMBER	CLIENT_ID	LAW_FIRM_MATTER_ID	INVOICE_TOTAL	BILLING_START_DATE	BILLING_END_DATE	INVOICE_DESCRIPTION
3	19990225	96542	711	528	1684.45	19990101	19990131	For services rendered
4	19990225	96542	711	528	1684.45	19990101	19990131	For services rendered
5	19990225	96542	711	528	1684.45	19990101	19990131	For services rendered
6	19990225	96542	711	528	1684.45	19990101	19990131	For services rendered
7	19990225	96542	711	528	1684.45	19990101	19990131	For services rendered
8	19990225	96542	711	1326	1250	19990101	19990131	Monthly Retainer

Index

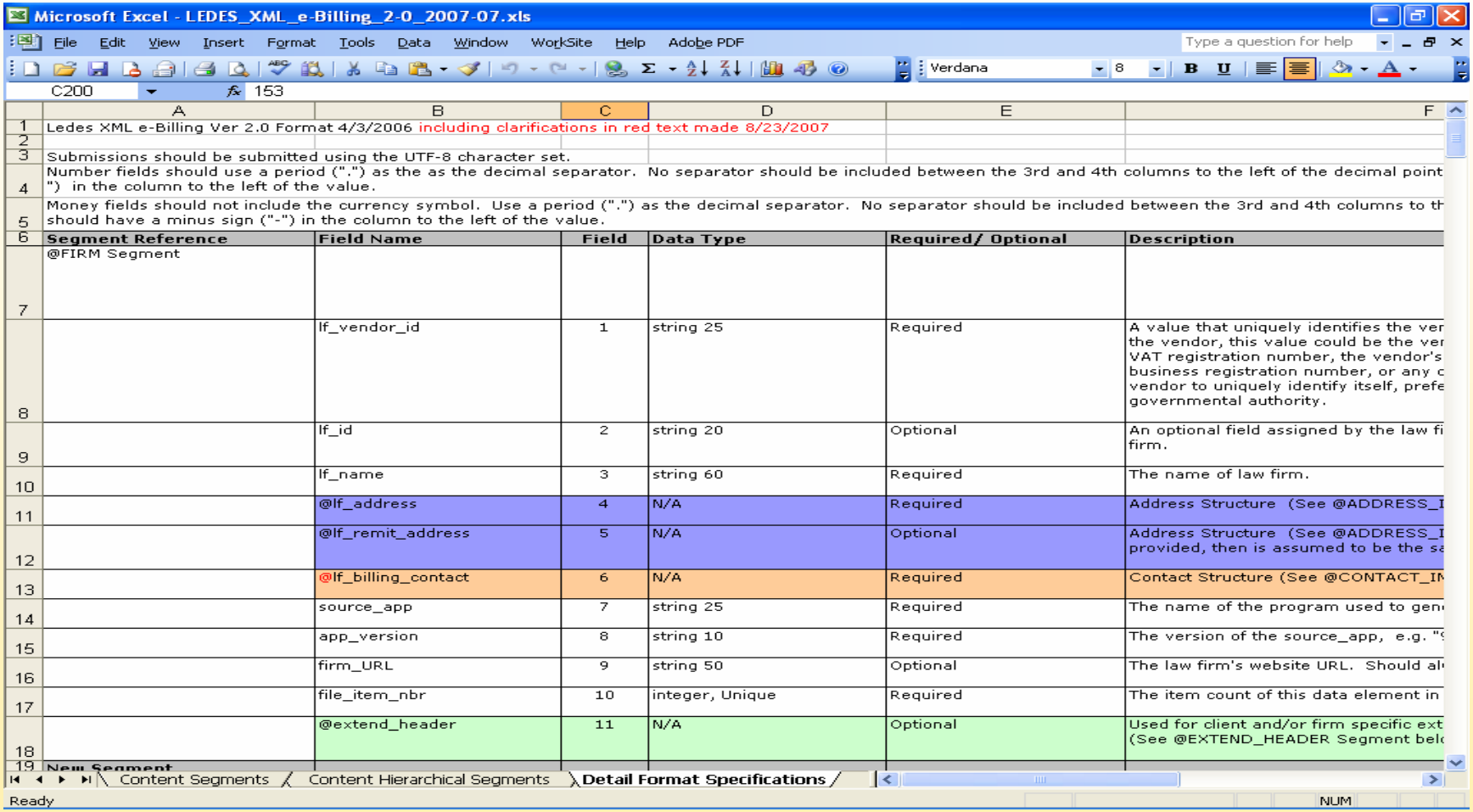
LEDES XML Ebilling Ver. 2 History

- 2006: LEDES XML Ebilling Ver. 2 format was ratified by the LOC membership as an update to and replacement for the LEDES 2000 XML format.
- A team from the Ebilling Subcommittee is currently considering revisions to this format, mostly concerned with additional requirements for international ebilling.

LEDES XML Ebilling Ver. 2 Format

- XML based format
- Relational database structure
- Data falls in FIRM, TAX, CLIENT, INVOICE, MATTER, TAX_SUMMARY, MATTER_DISC_CRED, TAX_MATTER_DISC_CRED, TKSUM, FEE, FEE_ITEM_DISC_CRED, TAX_ITEM_FEE, EXPENSE, EXPENSE_ITEM_DISC_CRED, TAX_ITEM_EXPENSE, ADDRESS_INFO, CONTACT_INFO, EXTEND_HEADER and EXTEND_DATA segments.
- Contains 153 fields; richer data structure accommodates complex transactions and more accurate calculations

LEDES XML Ebilling Ver. 2 Format



Microsoft Excel - LEDES_XML_e-Billing_2-0_2007-07.xls

File Edit View Insert Format Tools Data Window WorkSite Help Adobe PDF

Type a question for help

Verdana 8

C200 153

	A	B	C	D	E	F
1	Ledes XML e-Billing Ver 2.0 Format 4/3/2006 including clarifications in red text made 8/23/2007					
2						
3	Submissions should be submitted using the UTF-8 character set.					
4	Number fields should use a period (".") as the as the decimal separator. No separator should be included between the 3rd and 4th columns to the left of the decimal point ") in the column to the left of the value.					
5	Money fields should not include the currency symbol. Use a period (".") as the decimal separator. No separator should be included between the 3rd and 4th columns to th should have a minus sign ("-") in the column to the left of the value.					
6	**Segment Reference**	**Field Name**	**Field**	**Data Type**	**Required/ Optional**	**Description**
7	@FIRM Segment					
8		lf_vendor_id	1	string 25	Required	A value that uniquely identifies the ver the vendor, this value could be the ver VAT registration number, the vendor's business registration number, or any c vendor to uniquely identify itself, prefe governmental authority.
9		lf_id	2	string 20	Optional	An optional field assigned by the law fi firm.
10		lf_name	3	string 60	Required	The name of law firm.
11		@lf_address	4	N/A	Required	Address Structure (See @ADDRESS_I
12		@lf_remit_address	5	N/A	Optional	Address Structure (See @ADDRESS_I provided, then is assumed to be the sa
13		@lf_billing_contact	6	N/A	Required	Contact Structure (See @CONTACT_IN
14		source_app	7	string 25	Required	The name of the program used to gen
15		app_version	8	string 10	Required	The version of the source_app, e.g. "9
16		firm_URL	9	string 50	Optional	The law firm's website URL. Should al
17		file_item_nbr	10	integer, Unique	Required	The item count of this data element in
18		@extend_header	11	N/A	Optional	Used for client and/or firm specific ext (See @EXTEND_HEADER Segment belo
19	New Segment					

Content Segments | Content Hierarchical Segments | Detail Format Specifications

Ready NUM

XML Ebilling Ver. 2 Benefits

- Changes how the math is calculated on an invoice
- Includes the ability to itemize complex taxes in such a way that it would be possible to recalculate tax on adjusted invoices
- Allows for the submission of alternate fee arrangements, credits and debits on a matter
- Supports multiple vendor tax identification numbers
- Eliminates non-mainstream ebilling data elements
- Is consistent in the use of terminology
- Includes fields necessary for international ebilling submissions
- Uses XSD
- Is intended to be the only ebilling format subject to revision by the LOC

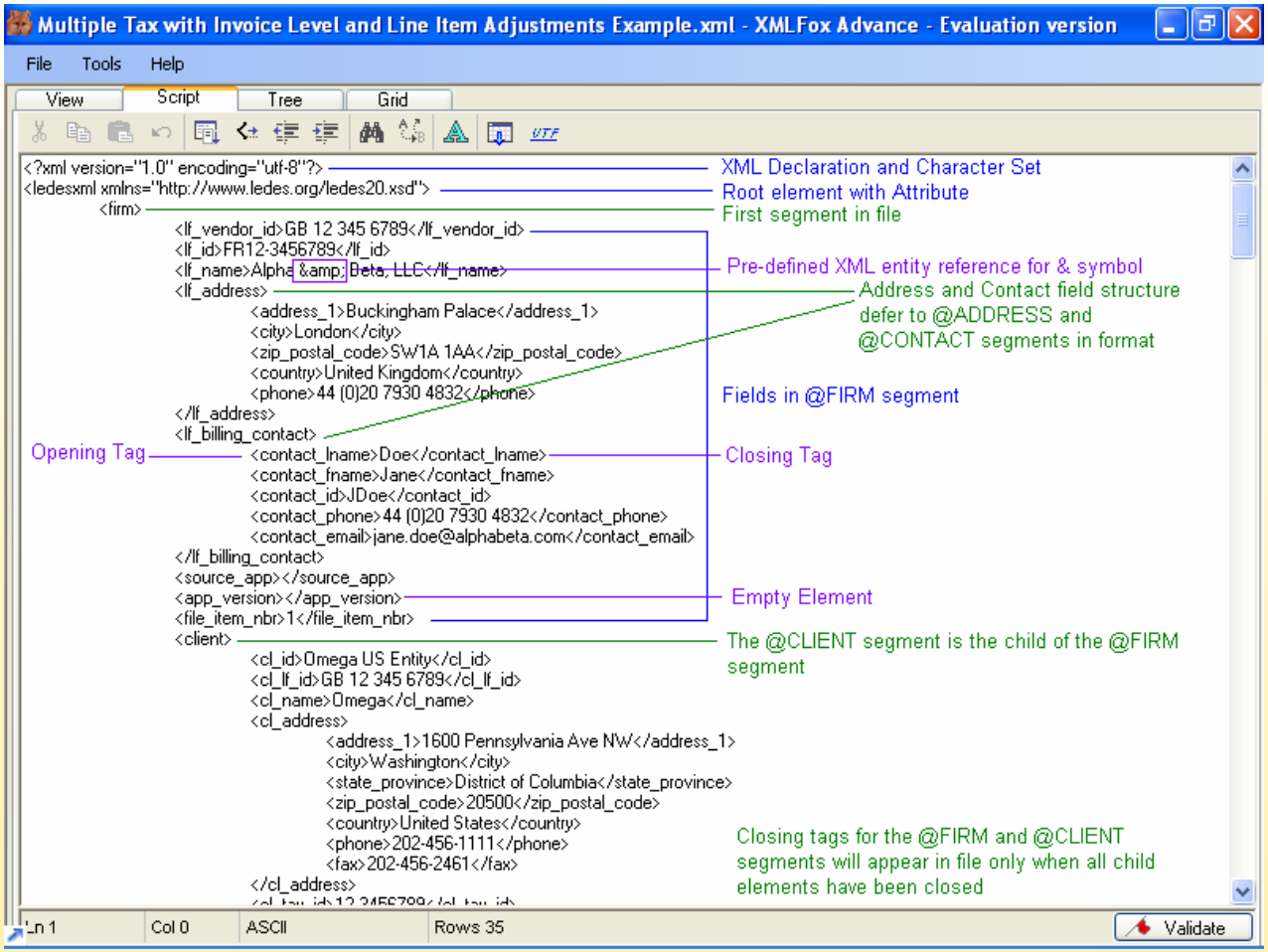
XML Ebilling Ver. 2 Issues

- It is complicated! There's a lot in there.
- Will get more complicated as we add fields to accommodate statutory requirements for international ebilling
- Some math issues currently pending revision
- Mandates use of UTF8 character set; will be problematic for some international users

Editing XML Ebilling Ver. 2 Files

- Can use any XML Editor
- Can be viewed using XML parser

Editing XML Ebilling Ver. 2 Sample File



Index

LEDES XML Budgeting Format

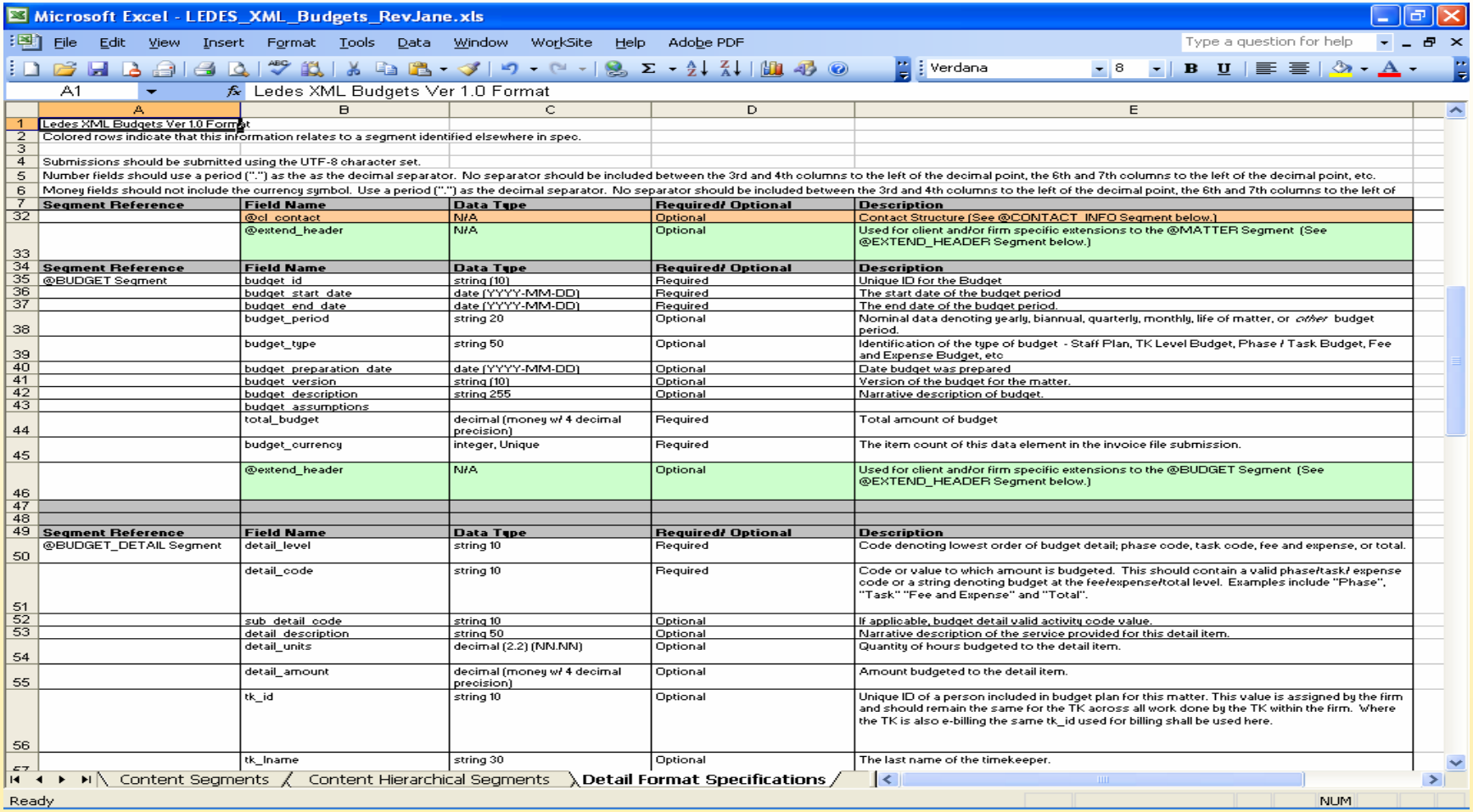
LEDES Budgeting History

- 2006: A budgeting standard was approved by the LOC membership. The LEDES XML budget standard accommodates the submission and revision of varying types of legal budgets, such as timekeeper or task and activity based budgets across varying time frames and also matter staffing plans.
- A Budgeting Subcommittee, headed by Cole Morgan and Adam Jaffe, continues to collect comments on the format.

LEDES Budgeting Format

- This standard, in XML, conveys information required for Budget and Case Plans.
- The format includes segments for FIRM, CLIENT, MATTER, BUDGET, BUDGET_DETAIL, BUDGET_TKSUM, ADDRESS, CONTACT, EXTEND_HEADER and EXTEND_DATA
- Allows for submission of budgets for any period (start and end date fields provided) and currency (ISO field provided)
- Budgeting can be for periods defined by the client (provides examples of Phase, Task, Fee and Expense and Total), allows for budgeting of hours and/or amounts and can also be by Timekeeper.

LEDES Budgeting Format Fields



Microsoft Excel - LEDES_XML_Budgets_RevJane.xls

File Edit View Insert Format Tools Data Window WorkSite Help Adobe PDF

Type a question for help

Verdana 8

A1 Ledes XML Budgets Ver 1.0 Format

	A	B	C	D	E
1	Ledes XML Budgets Ver 1.0 Format				
2	Colored rows indicate that this information relates to a segment identified elsewhere in spec.				
3					
4	Submissions should be submitted using the UTF-8 character set.				
5	Number fields should use a period (".") as the as the decimal separator. No separator should be included between the 3rd and 4th columns to the left of the decimal point, the 6th and 7th columns to the left of the decimal point, etc.				
6	Money fields should not include the currency symbol. Use a period (".") as the decimal separator. No separator should be included between the 3rd and 4th columns to the left of the decimal point, the 6th and 7th columns to the left of				
7	**Segment Reference**	**Field Name**	**Data Type**	**Required/ Optional**	**Description**
32		@cl_contact	N/A	Optional	Contact Structure (See @CONTACT_INFO Segment below.)
33		@extend_header	N/A	Optional	Used for client and/or firm specific extensions to the @MATTER Segment (See @EXTEND_HEADER Segment below.)
34	**Segment Reference**	**Field Name**	**Data Type**	**Required/ Optional**	**Description**
35	@BUDGET Segment	budget_id	string (10)	Required	Unique ID for the Budget
36		budget_start_date	date (YYYY-MM-DD)	Required	The start date of the budget period
37		budget_end_date	date (YYYY-MM-DD)	Required	The end date of the budget period.
38		budget_period	string 20	Optional	Nominal data denoting yearly, biannual, quarterly, monthly, life of matter, or *other* budget period.
39		budget_type	string 50	Optional	Identification of the type of budget - Staff Plan, TK Level Budget, Phase / Task Budget, Fee and Expense Budget, etc
40		budget_preparation_date	date (YYYY-MM-DD)	Optional	Date budget was prepared
41		budget_version	string (10)	Optional	Version of the budget for the matter.
42		budget_description	string 255	Optional	Narrative description of budget.
43		budget_assumptions			
44		total_budget	decimal (money w/ 4 decimal precision)	Required	Total amount of budget
45		budget_currency	integer, Unique	Required	The item count of this data element in the invoice file submission.
46		@extend_header	N/A	Optional	Used for client and/or firm specific extensions to the @BUDGET Segment (See @EXTEND_HEADER Segment below.)
47					
48					
49	**Segment Reference**	**Field Name**	**Data Type**	**Required/ Optional**	**Description**
50	@BUDGET_DETAIL Segment	detail_level	string 10	Required	Code denoting lowest order of budget detail; phase code, task code, fee and expense, or total.
51		detail_code	string 10	Required	Code or value to which amount is budgeted. This should contain a valid phase/task/ expense code or a string denoting budget at the fee/expense/total level. Examples include "Phase", "Task" "Fee and Expense" and "Total".
52		sub_detail_code	string 10	Optional	If applicable, budget detail valid activity code value.
53		detail_description	string 50	Optional	Narrative description of the service provided for this detail item.
54		detail_units	decimal (2.2) (NN.NN)	Optional	Quantity of hours budgeted to the detail item.
55		detail_amount	decimal (money w/ 4 decimal precision)	Optional	Amount budgeted to the detail item.
56		tk_id	string 10	Optional	Unique ID of a person included in budget plan for this matter. This value is assigned by the firm and should remain the same for the TK across all work done by the TK within the firm. Where the TK is also e-billing the same tk_id used for billing shall be used here.
57		tk_lname	string 30	Optional	The last name of the timekeeper.

Content Segments | Content Hierarchical Segments | Detail Format Specifications

Ready NUM

Editing LEDES XML Budget Files

- Can use any XML Editor
- Can be viewed using XML parser

LEDES Budgeting Format Sample File

Index

LEDES XML Timekeeper Attributes

LEDES Timekeeper Attributes History

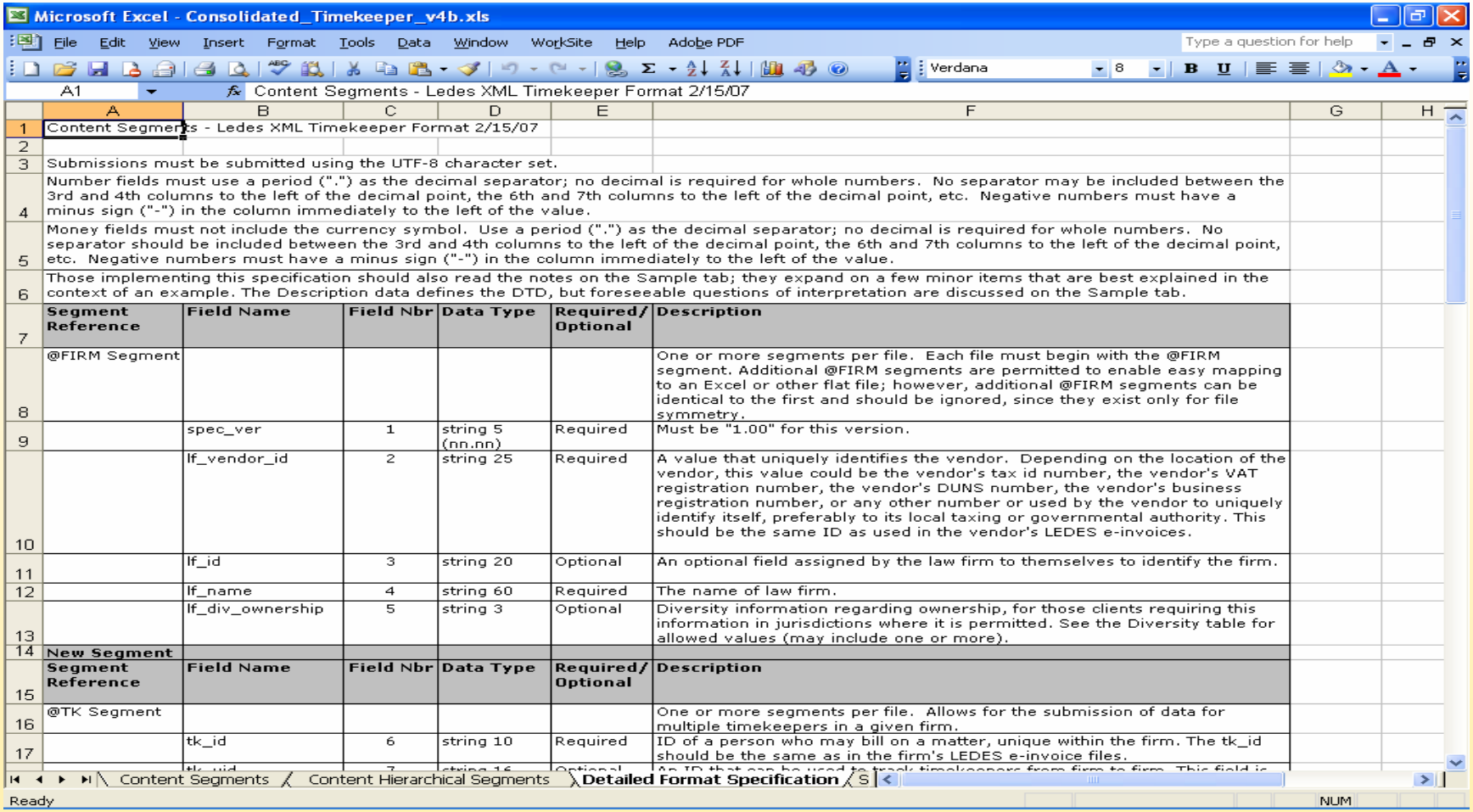
- 2007: A standard for the exchange of timekeeper attribute information was approved by the LOC membership.

- Subcommittee head Steven Levy has moved on to another position within Microsoft. Need volunteer to chair subcommittee in order to collect comments on format.

LEDES Timekeeper Attributes Format

- This standard, in XML, conveys information required internally by the corporate client and SEC information requirements placed on public companies in the US.
- Includes segments for FIRM, TK, RATE, EXTEND_TK and EXTEND_RATE
  - Includes profile information on the timekeeper including the year they were admitted to the bar, timekeeper level, area of expertise, ethnicity and gender.
  - Includes rate information, the time period it applies to, the billing currency, discounts which apply to the billing rate and the client matter number it applies to.

LEDES Timekeeper Attributes Format



Microsoft Excel - Consolidated_Timekeeper_v4b.xls

File Edit View Insert Format Tools Data Window WorkSite Help Adobe PDF

Type a question for help

Verdana 8

A1 Content Segments - Ledes XML Timekeeper Format 2/15/07

	A	B	C	D	E	F	G	H
1	Content Segments - Ledes XML Timekeeper Format 2/15/07							
2								
3	Submissions must be submitted using the UTF-8 character set.							
4	Number fields must use a period (".") as the decimal separator; no decimal is required for whole numbers. No separator may be included between the 3rd and 4th columns to the left of the decimal point, the 6th and 7th columns to the left of the decimal point, etc. Negative numbers must have a minus sign ("-") in the column immediately to the left of the value.							
5	Money fields must not include the currency symbol. Use a period (".") as the decimal separator; no decimal is required for whole numbers. No separator should be included between the 3rd and 4th columns to the left of the decimal point, the 6th and 7th columns to the left of the decimal point, etc. Negative numbers must have a minus sign ("-") in the column immediately to the left of the value.							
6	Those implementing this specification should also read the notes on the Sample tab; they expand on a few minor items that are best explained in the context of an example. The Description data defines the DTD, but foreseeable questions of interpretation are discussed on the Sample tab.							
7	**Segment Reference**	**Field Name**	**Field Nbr**	**Data Type**	**Required/ Optional**	**Description**		
8	@FIRM Segment					One or more segments per file. Each file must begin with the @FIRM segment. Additional @FIRM segments are permitted to enable easy mapping to an Excel or other flat file; however, additional @FIRM segments can be identical to the first and should be ignored, since they exist only for file symmetry.		
9		spec_ver	1	string 5 (nn.nn)	Required	Must be "1.00" for this version.		
10		lf_vendor_id	2	string 25	Required	A value that uniquely identifies the vendor. Depending on the location of the vendor, this value could be the vendor's tax id number, the vendor's VAT registration number, the vendor's DUNS number, the vendor's business registration number, or any other number or used by the vendor to uniquely identify itself, preferably to its local taxing or governmental authority. This should be the same ID as used in the vendor's LEDES e-invoices.		
11		lf_id	3	string 20	Optional	An optional field assigned by the law firm to themselves to identify the firm.		
12		lf_name	4	string 60	Required	The name of law firm.		
13		lf_div_ownership	5	string 3	Optional	Diversity information regarding ownership, for those clients requiring this information in jurisdictions where it is permitted. See the Diversity table for allowed values (may include one or more).		
14	**New Segment**							
15	**Segment Reference**	**Field Name**	**Field Nbr**	**Data Type**	**Required/ Optional**	**Description**		
16	@TK Segment					One or more segments per file. Allows for the submission of data for multiple timekeepers in a given firm.		
17		tk_id	6	string 10	Required	ID of a person who may bill on a matter, unique within the firm. The tk_id should be the same as in the firm's LEDES e-invoice files.		

Content Segments | Content Hierarchical Segments | Detailed Format Specification | S

Ready NUM

Index

Editing LEDES XML Timekeeper Files

- Can use any XML Editor
- Can be viewed using XML parser

LEDES Timekeeper Attributes Sample File

Wrap Up

For additional information on XML, Google:

“XML Tutorial”

“XSD Tutorial”

“DTD Tutorial”