XML
for the Non Technically Minded

Stephen Foley

XML Presentation Online

Available as a PDF from

www.livenote.com/info/xml1.pdf

My email address

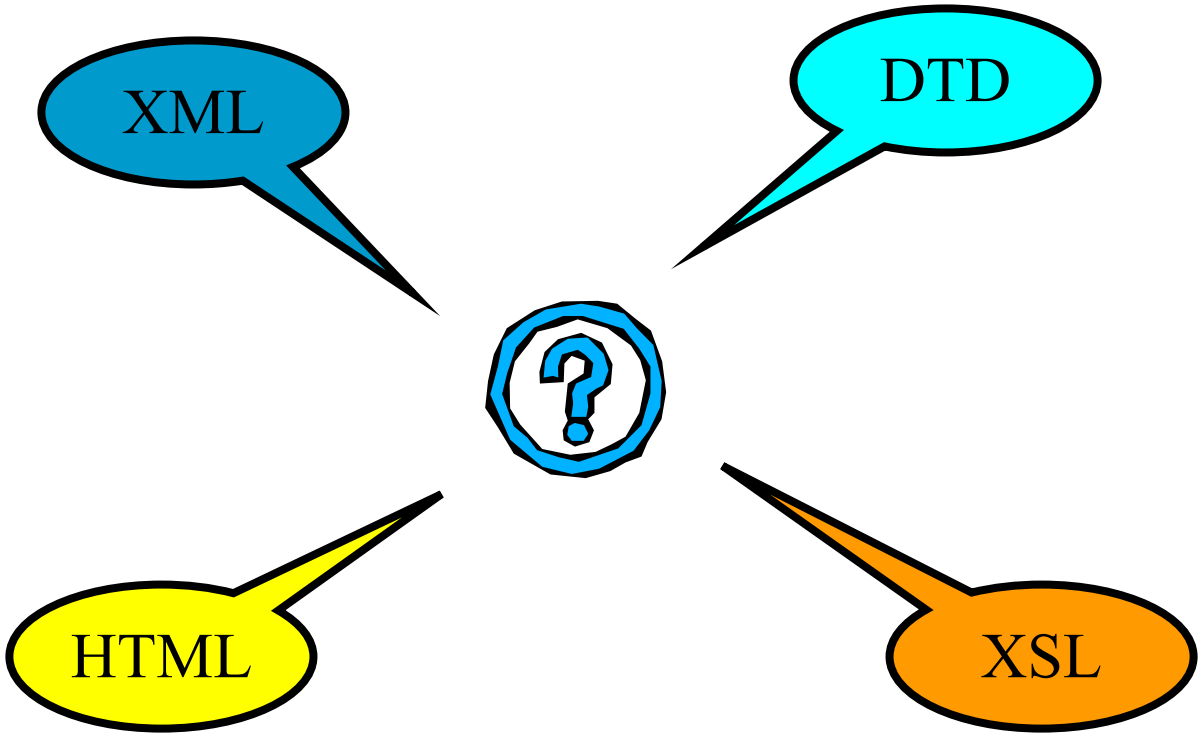
stephenf@livenote.com

Phone 03 9641 2260

XML
Why do you need to be involved ?

- XML provides a mechanism for technical people to solve standardisation problems between applications
- If non technical people do not become involved the chances are that the technology could end up driving the requirements - not the requirements driving the technology
- The focus of this session is to put XML in perspective and hopefully support you to participate in future decisions which may require the use of XML

XML – Demystifying



XML
Open Standard for Data Exchange

- Why has this become so widely adopted ?
- Previously ASCII delimited files or proprietary databases have been used to do data exchange
- By looking at an ASCII example, a proprietary database example and an XML example hopefully the virtues of XML can be understood.

ASCII Data Exchange

- Invoice 65
-
- Peters and Peters
- 219 Swanston Street
- Melbourne 3000
-
- 4 Magazines $225.00

- **Tab Delimited File**

INVOICE NO	CUSTOMER NAME	CUSTOMER ADDRESS 1	CUSTOMER ADDRESS 2	DESCRIPTION	TOTAL
65	Peters and Peters	219 Swanston Street	Melbourne 3000	4 Magazines	$225.00
66	Thompson	27 Elizabeth Street	Melbourne 3000	Aug 2001 Newspapers	$125.00

Proprietary DB Exchange

- Invoice 65
-
- Peters and Peters
- 219 Swanston Street
- Melbourne 3000
-
- 4 Magazines $225.00

- **Access Database**

InvoiceTable : Table

Invoice No	Customer Name	Customer Address 1	Customer Address 2	Description	Total
65	Peters and Peters	219 Swanston Street	Melbourne 3000	4 Magzines	$225.00
66	Thompson	27 Elizabeth Street	Melbourne 3000	Aug 2001 Newspapers	$125.00
0					$0.00

Record: 3 of 3

Proprietary Format

- **Access Database**

- **StandardJetDBnb`Ugr@?~1y0cFNj7(Q'`{65eC>3vy[$|*|aR!f_$g'DeFxbT4.0dvYSYYYYYYYYYYYYYYrYsYEYaYdY2YYYjYConnectDatabaseDateCreateDateUpdateFlagsForeignNameIdLvLvEdateFlagsForeignNameIdLvLvExtraLvModuleLvPropNameOwnerParentIdRmtInfoLongRmtInfongRmtInfoShortTypeniYYIdParentIdNameOY7SYYYY2ACMFInheritableObjectIdSIDYObjectIdYObjectIdYSYYYYYYYYAttributeExpressionFlagLvExtraName1Name2ObjectIdOrdernzfedY"OjectIdOrdernzfedY"ObjectIdAttributeSYYYYYYYYccolumngrbiticolumnszColumnszObjecmnszColumnszObject$szReferencedColumn$szReferencedObjectszRelationshipYYYszObjecszObject$szReferencedObjectszRelationshipYv1bN:k&WCtOJmJJMMQkkfJUQkOJmJJMMQkkfJUQkOJmJLJkQkSdi`k`dOo^QkiQ^JmYdbkWYfkiQfdimkkMiYfmkkvkiQ^mJL^QkYbqdYQkYbqdYMQmJL^Q`kvkJMMQkkdL[QMmk`kvkJMQk`kvkdL[QMmk`kvkhoQiYQk`kvkiQ^JmYdbkWYfkJMkWYfkJMMQkk^Jvdom`kvkOLko``JivYbSdokQiOQSYbQOu!"#$%&'()*+/0123456%d_Z7T_YwWcR(@cR(@AccessLayout4MR2KeepLocalT@zz:::::::8@fM(@]&R(@fM(@]&R(@InvoiceTable@FFF:::::::8@,x|M(@,x|M(@DataAccessPages@@@@@@@@@@>,x|M(@x|M(@,x|M(@SysRel.....x|M(@,x|M(@Modules0000000000,x|M(@,x|M(@Scripts0Scripts0000000000.,x|M(@,x|M(@Reports0000000000,x|M(@,x|M(@Forms,,,,,,,,,,*|M(@|M(@UserDefinedA@DDD88888886@|M(@|M(@SummaryInfo@DDD88888886|M(@|M(@MSysA|M(@MSysAccessObjectsDDDDDDDDDDBs|M(@s|M(@MSysRelationshipsDDDDDDDDDDBs|M(@s|Ms|M(@s|M(@MSysQueries88888888886s|M(@s|M(@MSysACEs22222222220 s|M(@s|M(@MSysObj**

XML Data Exchange

- Invoice 65
-
- Peters and Peters
- 219 Swanston Street
- Melbourne 3000
-
- 4 Magazines $225.00

- **XML Data File**

```
<DOCUMENT>
        <INVOICE>
                <INVOICE_NO> 65 </INVOICE_NO>
                <CUSTOMER_NAME>Peters and Peters</CUSTOMER_NAME>
                <CUSTOMER_ADDRESS_1>219 Swanston Street</CUSTOMER_ADDRESS_1>
                <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
                <DESCRIPTION>4 Magazines</DESCRIPTION>
                <TOTAL>$225.00</TOTAL>
        </INVOICE>
        <INVOICE>
                <INVOICE_NO> 66 </INVOICE_NO>
                <CUSTOMER_NAME>Thompson</CUSTOMER_NAME>
                <CUSTOMER_ADDRESS_1>27 Elizabeth Street</CUSTOMER_ADDRESS_1>
                <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
                <DESCRIPTION>Aug 2001 Newspapers</DESCRIPTION>
                <TOTAL>$125.00</TOTAL>
        </INVOICE>
</DOCUMENT>
```

Extending our Example

- Invoice 65
-
- Peters and Peters
- 219 Swanston Street
- Melbourne 3000
-
- Total $225.00
-

- Magazine Subscription A $75.00
- Magazine Subscription B $75.00
- Magazine Subscription C $50.00
- Magazine Subscription D $25.00

ASCII Data Exchange 2

- **Tab Delimited File**

- File 1

INVOICE NO	CUSTOMER NAME	CUSTOMER ADDRESS 1	CUSTOMER ADDRESS 2	TOTAL
65	Peters and Peters	219 Swanston Street	Melbourne 3000	$225.00
66	Thompson	27 Elizabeth Street	Melbourne 3000	$125.00

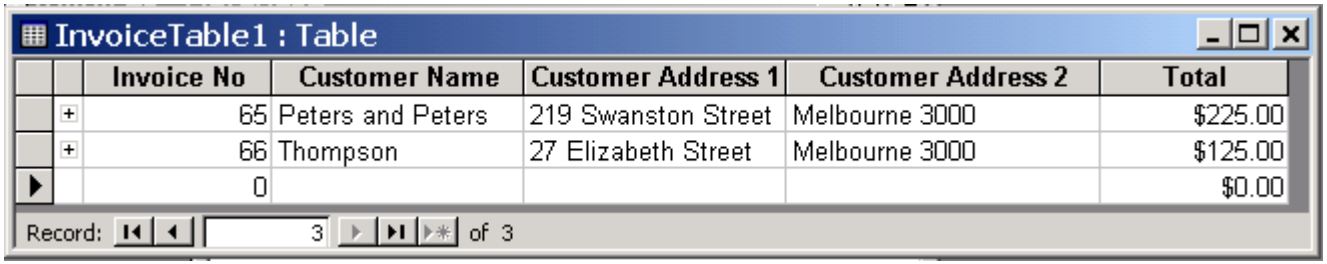
- File 2

INVOICE NO	DESCRIPTION	AMOUNT
65	Magazine Subscription A	$75.00
65	Magazine Subscription B	$75.00
65	Magazine Subscription C	$50.00
65	Magazine Subscription D	$25.00
66	August 2001 Week 1 Newspapers	$25.00
66	August 2001 Week 2 Newspapers	$30.00
66	August 2001 Week 3 Newspapers	$20.00
66	August 2001 Week 4 Newspapers	$50.00

Proprietary DB Exchange 2

Access Database

- Table 1



InvoiceTable1 : Table

Invoice No	Customer Name	Customer Address 1	Customer Address 2	Total
65	Peters and Peters	219 Swanston Street	Melbourne 3000	$225.00
66	Thompson	27 Elizabeth Street	Melbourne 3000	$125.00
0				$0.00

Record: 3 of 3

- Table 2

DetailsTable1 : Table

Invoice No	ItemNo	Description	Amount
65	1	Magazine Subscription A	$75.00
65	2	Magazine Subscription B	$75.00
65	3	Magazine Subscription C	$50.00
65	4	Magazine Subscription D	$25.00
66	1	August 2001 Week 1 Newspapers	$25.00
66	2	August 2001 Week 2 Newspapers	$30.00
66	3	August 2001 Week 3 Newspapers	$20.00
66	4	August 2001 Week 4 Newspapers	$50.00
0	0		$0.00

Record: 9 of 9

XML Data Exchange 2

- **XML Data File**

```
<DOCUMENT>
        <INVOICE>
                <INVOICE_NO> 65 </INVOICE_NO>
                <CUSTOMER_NAME>Peters and Peters</CUSTOMER_NAME>
                <CUSTOMER_ADDRESS_1>219 Swanston Street</CUSTOMER_ADDRESS_1>
                <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
                <TOTAL>$225.00</TOTAL>
                <ITEMS>
                        <DESCRIPTION>
                                <PRODUCT>Magazine Subscription A</PRODUCT>
                                <AMOUNT>$75.00</AMOUNT>
                        </DESCRIPTION>
                        <DESCRIPTION>
                                <PRODUCT>Magazine Subscription B</PRODUCT>
                                <AMOUNT>$75.00</AMOUNT>
                        </DESCRIPTION>
                        <DESCRIPTION>
                                <PRODUCT>Magazine Subscription C</PRODUCT>
                                <AMOUNT>$50.00</AMOUNT>
                        </DESCRIPTION>
                        <DESCRIPTION>
                                <PRODUCT>Magazine Subscription D</PRODUCT>
                                <AMOUNT>$25.00</AMOUNT>
                        </DESCRIPTION>
                </ITEMS>
        </INVOICE>
```

XML Data Exchange 2

- **XML Data File (cont)**

```
<INVOICE>
    <INVOICE_NO> 66 </INVOICE_NO>
    <CUSTOMER_NAME>Thompson</CUSTOMER_NAME>
    <CUSTOMER_ADDRESS_1>27 Elizabeth Street</CUSTOMER_ADDRESS_1>
    <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
    <TOTAL>$125.00</TOTAL>
    <ITEMS>
        <DESCRIPTION>
            <PRODUCT> Aug 2001 Week 1 Newspapers</PRODUCT>
            <AMOUNT>$25.00</AMOUNT>
        </DESCRIPTION>
        <DESCRIPTION>
            <PRODUCT>Aug 2001 Week 2 Newspapers </PRODUCT>
            <AMOUNT>$30.00</AMOUNT>
        </DESCRIPTION>
        <DESCRIPTION>
            <PRODUCT>Aug 2001 Week 3 Newspapers </PRODUCT>
            <AMOUNT>$20.00</AMOUNT>
        </DESCRIPTION>
        <DESCRIPTION>
            <PRODUCT>Aug 2001 Week 4 Newspapers </PRODUCT>
            <AMOUNT>$50.00</AMOUNT>
        </DESCRIPTION>
    </ITEMS>
</INVOICE>
</DOCUMENT>
```

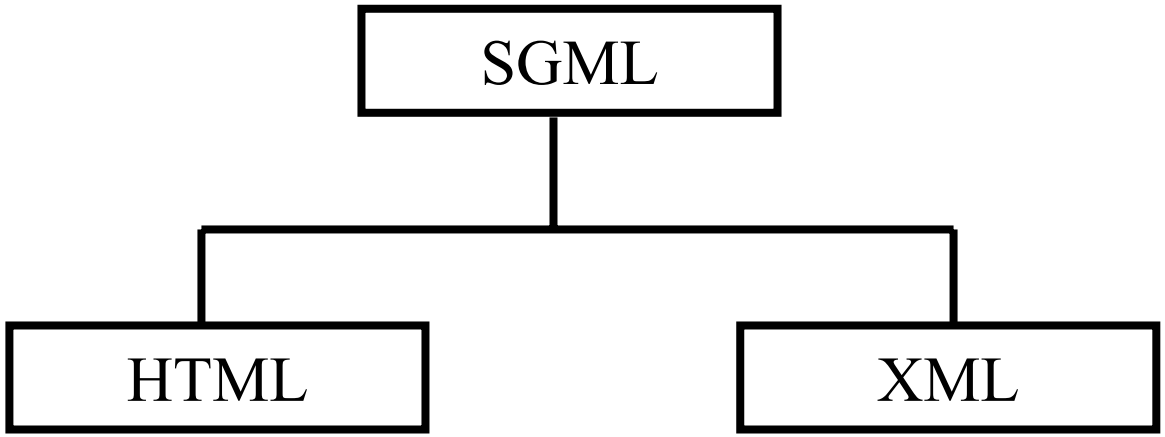
The Virtues of XML

- It is easily readable
- It is extensible
- It is self describing
- And because of these virtues companies have developed tools to exploit XML thereby giving it another huge advantage

XML family

- Syntax Checking Tools (well formed)
- Document Type Definition (DTD)
- Transforming and Formatting Tools (XSL)
- Parsers (DOM and SAX) or build your own

Origins of XML



- SGML Standard Generalized Markup Language
- HTML Hypertext Markup Language, an application of SGML (120 Tags)
- XML Extensible Markup Language, easy to use subset of SGML (lets you create your own tags)

Example 1

- Xmlexample1.xml

```
<?xml version="1.0"?>
<!DOCTYPE DOCUMENT SYSTEM "xmlexamp.dtd">
<DOCUMENT>
        <INVOICE>
                <INVOICE_NO> 65 </INVOICE_NO>
                <CUSTOMER_NAME>Peters and Peters</CUSTOMER_NAME>
                <CUSTOMER_ADDRESS_1>219 Swanston Street</CUSTOMER_ADDRESS_1>
                <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
                <DESCRIPTION>4 Magazines</DESCRIPTION>
                <TOTAL>$225.00</TOTAL>
        </INVOICE>
        <INVOICE>
                <INVOICE_NO> 66 </INVOICE_NO>
                <CUSTOMER_NAME>Thompson</CUSTOMER_NAME>
                <CUSTOMER_ADDRESS_1>27 Elizabeth Street</CUSTOMER_ADDRESS_1>
                <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
                <DESCRIPTION>Aug 2001 Newspapers</DESCRIPTION>
                <TOTAL>$125.00</TOTAL>
        </INVOICE>
</DOCUMENT>
```

DTD

- **Xmlexamp.DTD**

- <!ELEMENT DOCUMENT (INVOICE)*>
- <!ELEMENT INVOICE (INVOICE_NO, CUSTOMER_NAME, CUSTOMER_ADDRESS_1, CUSTOMER_ADDRESS_2, DESCRIPTION, TOTAL)>
- <!ELEMENT INVOICE_NO (#PCDATA)>
- <!ELEMENT CUSTOMER_NAME (#PCDATA)>
- <!ELEMENT CUSTOMER_ADDRESS_1 (#PCDATA)>
- <!ELEMENT CUSTOMER_ADDRESS_2 (#PCDATA)>
- <!ELEMENT DESCRIPTION (#PCDATA)>
- <!ELEMENT TOTAL (#PCDATA)>

Example 2

- Xmlexample2.xml

```
<?xml version="1.0"?>
<!DOCTYPE DOCUMENT SYSTEM "xmlexamp.dtd">
<?xml:stylesheet type="text/xsl" href="xmlexamp.xsl" ?>
<DOCUMENT>
        <INVOICE>
                <INVOICE_NO> 65 </INVOICE_NO>
                <CUSTOMER_NAME>Peters and Peters</CUSTOMER_NAME>
                <CUSTOMER_ADDRESS_1>219 Swanston Street</CUSTOMER_ADDRESS_1>
                <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
                <DESCRIPTION>4 Magazines</DESCRIPTION>
                <TOTAL>$225.00</TOTAL>
        </INVOICE>
        <INVOICE>
                <INVOICE_NO> 66 </INVOICE_NO>
                <CUSTOMER_NAME>Thompson</CUSTOMER_NAME>
                <CUSTOMER_ADDRESS_1>27 Elizabeth Street</CUSTOMER_ADDRESS_1>
                <CUSTOMER_ADDRESS_2>Melbourne 3000</CUSTOMER_ADDRESS_2>
                <DESCRIPTION>Aug 2001 Newspapers</DESCRIPTION>
                <TOTAL>$125.00</TOTAL>
        </INVOICE>
</DOCUMENT>
```

XSL

- **Xmlexamp.xsl**

```
<?xml version="1.0"?>
<HTML xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<BODY STYLE="font-family:Arial, helvetica, sans-serif; font-size:12pt;
 background-color:#EEEEEE">
<H2><CENTER> XML Example using XSL</CENTER></H2>
<table border="1" width="100%">
        <tr>
                <td align="center"><b>Invoice No</b></td>
                <td><b>CUSTOMER_Name</b></td>
                <td><b>ADDRESS_1</b></td>
                <td><b>ADDRESS_2</b></td>
                <td><b>Description</b></td>
                <td align="right"><b>Amount</b></td>
        </tr>
        <xsl:for-each select="DOCUMENT/INVOICE">
        <tr>
                <td align="center"><xsl:value-of select="INVOICE_NO"/></td>
                <td><xsl:value-of select="CUSTOMER_NAME"/></td>
                <td><xsl:value-of select="CUSTOMER_ADDRESS_1"/></td>
                <td><xsl:value-of select="CUSTOMER_ADDRESS_2"/></td>
                <td><xsl:value-of select="DESCRIPTION"/></td>
                <td align="right"><xsl:value-of select="TOTAL"/></td>
        </tr>
        </xsl:for-each>
</table>
</BODY>
</HTML>
```

Putting it Together