

Чтобы понять XML, полезно понимать идею пометки данных. Люди создавали документы на протяжении многих лет и на протяжении всего времени они их помечали. Например, преподаватели могут постоянно пометить документы учащихся. Учащиеся могут перемещать абзацы, уточнять предложения, исправлять опечатки и так далее. Пометка документа определяет структуру, смысл и внешний вид сведений в документе. Если вы когда-либо использовали функцию "Отслеживание изменений" в Microsoft Office Word, то использовали компьютеризированную форму пометки.

В мире информационных технологий термин "пометка" превратился в термин "разметка". При разметке используются коды, называемые тегами (или иногда токенами), для определения структуры, визуального оформления и — в случае XML — смысла данных.

Текст этой статьи в формате HTML является хорошим примером применения компьютерной разметки. Если в Microsoft Internet Explorer щелкнуть эту страницу правой кнопкой мыши и выбрать команду **Просмотр HTML-кода**, вы увидите читаемый текст и теги HTML, например `<p>` и `<h2>`. В HTML- и XML-документах теги легко распознать, поскольку они заключены в угловые скобки. В исходном тексте этой статьи теги HTML выполняют множество функций, например определяют начало и конец каждого абзаца (`<p> ... </p>`) и местоположение рисунков.

Отличительные черты XML

Документы в форматах HTML и XML содержат данные, заключенные в теги, но на этом сходство между двумя языками заканчивается. В формате HTML теги определяют оформление данных — расположение заголовков, начало абзаца и т. д. В формате XML теги определяют структуру и смысл данных — то, чем они являются.

При описании структуры и смысла данных становится возможным их повторное использование несколькими способами. Например, если у вас есть блок данных о продажах, каждый элемент в котором четко определен, то можно загрузить в отчет о продажах только необходимые элементы, а другие данные передать в бухгалтерскую базу данных. Иначе говоря, можно использовать одну систему для генерации данных и пометки их тегами в формате XML, а затем обрабатывать эти данные

Благодаря такой совместимости XML является основой одной из самых популярных технологий обмена данными.

Учитывайте при работе следующее:

- HTML нельзя использовать вместо XML. Однако XML-данные можно заключать в HTML-теги и отображать на веб-страницах.
- Возможности HTML ограничены предопределенным набором тегов, общим для всех пользователей.
- Правила XML разрешают создавать любые теги, требуемые для описания данных и их структуры. Допустим, что вам необходимо хранить и совместно использовать сведения о домашних животных. Для этого можно создать следующий XML-код:

```
<?xml version="1.0"?>
<CAT>
  <NAME>Izzy</NAME>
  <BREED>Siamese</BREED>
  <AGE>6</AGE>
  <ALTERED>yes</ALTERED>
  <DECLAWED>no</DECLAWED>
  <LICENSE>Izz138bod</LICENSE>
  <OWNER>Colin Wilcox</OWNER>
</CAT>
```

Как видно, по тегам XML понятно, какие данные вы просматриваете. Например, ясно, что это данные о коте, и можно легко определить его имя, возраст и т. д. Благодаря возможности создавать теги, определяющие почти любую структуру данных, язык XML является расширяемым.

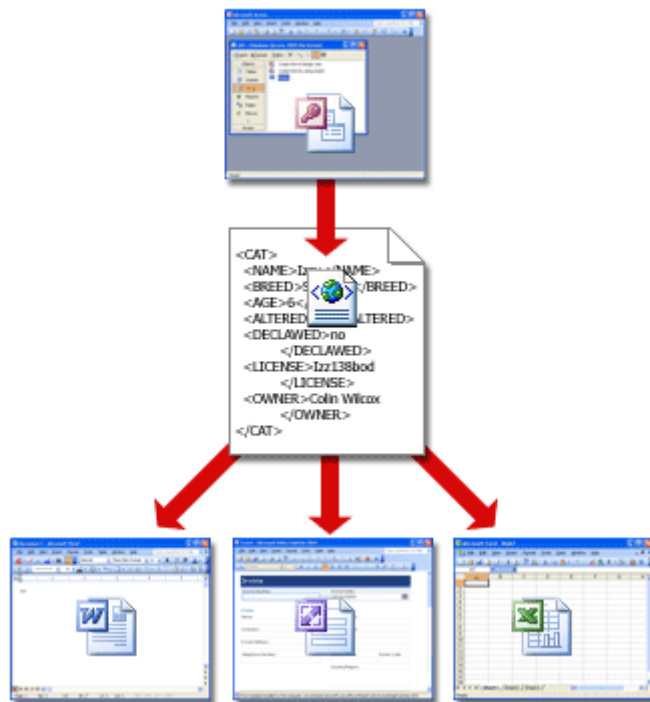
Но не путайте теги в данном примере с тегами в HTML-файле. Например, если приведенный выше текст в формате XML вставить в HTML-файл и открыть его в браузере, то результаты будут выглядеть следующим образом:

Правильно сформированные данные

Вероятно, вы слышали, как кто-то из ИТ-специалистов говорил о "правильно сформированном" XML-файле. Правильно сформированный XML-файл должен соответствовать очень строгим правилам. Если он не соответствует этим правилам, XML не работает. Например, в предыдущем примере каждый открывающий тег имеет соответствующий закрывающий тег, поэтому в данном примере соблюдено одно из правил правильно сформированного XML-файла. Если же удалить из файла какой-либо тег и попытаться открыть его в одной из программ Office, то появится сообщение об ошибке и использовать такой файл будет невозможно.

Правила создания правильно сформированного XML-файла знать необязательно (хотя понять их нетрудно), но следует помнить, что использовать в других приложениях и системах можно лишь правильно сформированные XML-данные. Если XML-файл не открывается, то он, вероятно, неправильно сформирован.

XML не зависит от платформы, и это значит, что любая программа, созданная для использования XML, может читать и обрабатывать XML-данные независимо от оборудования или операционной системы. Например, при применении правильных тегов XML можно использовать программу на настольном компьютере для открытия и обработки данных, полученных с мейнфрейма. И, независимо от того, кто создал XML-данные, с ними данными можно работать в различных приложениях Office. Благодаря своей совместимости XML стал одной из самых популярных технологий обмена данными между базами данных и пользовательскими компьютерами.



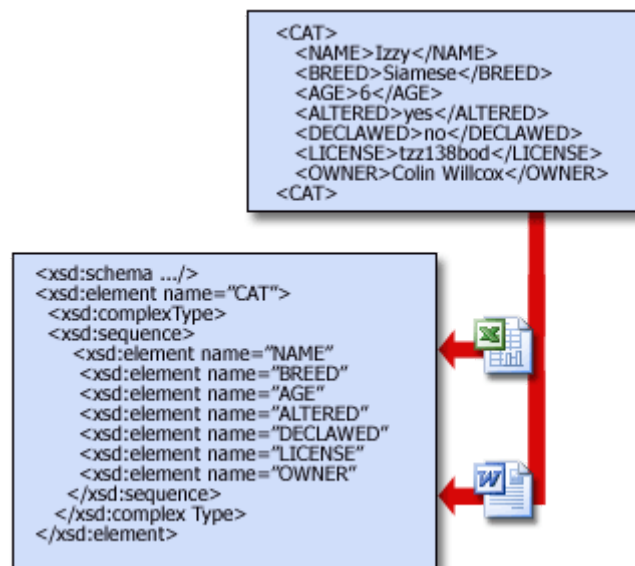
В дополнение к правильно сформированным данным с тегами XML-системы обычно используют два дополнительных компонента: схемы и преобразования. В следующих разделах описывается, как они работают.

Схемы

Не пугайтесь термина "схема". Схема — это просто XML-файл, содержащий правила для содержимого XML-файла данных. Файлы схем обычно имеют расширение XSD, тогда как для файлов данных XML используется расширение XML.

Схемы позволяют программам проверять данные. Они формируют структуру данных и обеспечивают их понятность создателю и другим людям. Например, если пользователь вводит недопустимые данные, например текст в поле даты, программа может предложить ему исправить их. Если данные в

использовать любую программу, поддерживающую XML. Например, как показано на приведенном ниже рисунке, Excel может проверять данные <CAT> на соответствие схеме CAT.



Схемы могут быть сложными, и в данной статье невозможно объяснить, как их создавать. (Кроме того, скорее всего, в вашей организации есть ИТ-специалисты, которые знают, как это делать.) Однако полезно знать, как выглядят схемы. Следующая схема определяет правила для набора тегов <CAT> ... </CAT>:

```
<xsd:element name="CAT">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="NAME" type="xsd:string"/>
      <xsd:element name="BREED" type="xsd:string"/>
      <xsd:element name="AGE" type="xsd:positiveInteger"/>
      <xsd:element name="ALTERED" type="xsd:boolean"/>
      <xsd:element name="DECLAWED" type="xsd:boolean"/>
      <xsd:element name="LICENSE" type="xsd:string"/>
      <xsd:element name="OWNER" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

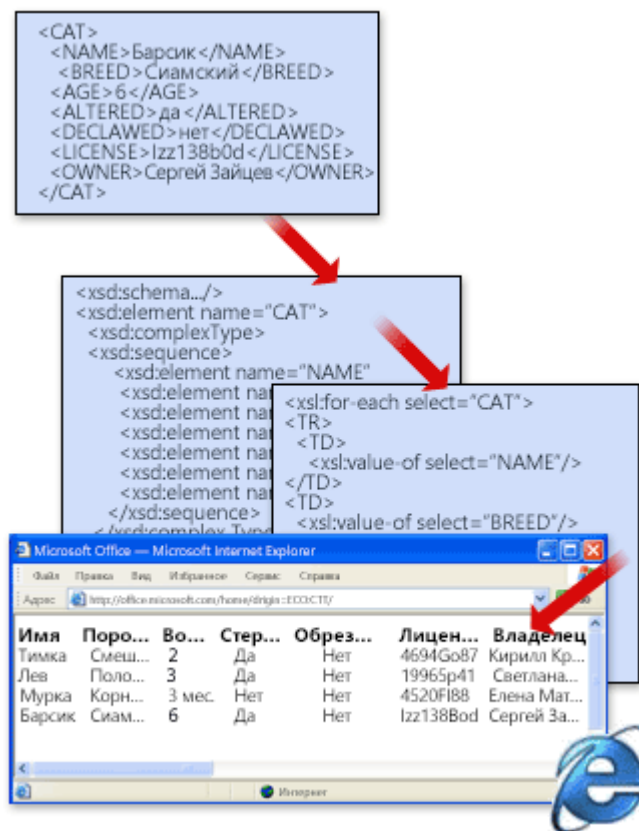
- Строковые элементы в приведенном примере схемы называются объявлениями. Если бы требовались дополнительные сведения о животном, например его цвет или особые признаки, то специалисты отдела ИТ добавили бы к схеме соответствующие объявления. Систему XML можно изменять по мере развития потребностей бизнеса.
- Объявления являются мощным средством управления структурой данных. Например, объявление `<xsd:sequence>` означает, что теги, такие как `<NAME>` и `<BREED>`, должны следовать в указанном выше порядке. С помощью объявлений можно также проверять типы данных, вводимых пользователем. Например, приведенная выше схема требует ввода положительного целого числа для возраста кота и логических значений (TRUE или FALSE) для тегов ALTERED и DECLAWED.
- Если данные в XML-файле соответствуют правилам схемы, то такие данные называют допустимыми. Процесс контроля соответствия XML-файла данным правилам схемы называют (достаточно логично) проверкой. Большим преимуществом использования схем является возможность предотвратить с их помощью повреждение данных. Схемы также облегчают поиск поврежденных данных, поскольку при возникновении такой проблемы обработка XML-файла останавливается.

Преобразования

Как говорилось выше, XML также позволяет эффективно использовать и повторно использовать данные. Механизм повторного использования данных называется преобразованием XSLT (или просто преобразованием).

Вы (или ваш ИТ-отдел) можете также использовать преобразования для обмена данными между серверными системами, например между базами данных. Предположим, что в базе данных А данные о продажах хранятся в таблице, удобной для отдела продаж. В базе данных Б хранятся данные о доходах и расходах в таблице, специально разработанной для бухгалтерии. База данных Б может использовать преобразование, чтобы принять данные от базы данных А и поместить их в соответствующие таблицы.

схемы, а затем передается любым пригодным способом для преобразования. В этом случае преобразование размещает данные в таблице на веб-странице.



В следующем примере кода показан один из способов написания преобразования. Она загружает данные <CAT> в таблицу на веб-странице. В этом примере суть не в том, чтобы показать, как написать преобразование, а в том, чтобы показать одну форму, которую может принять преобразование.

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0">
<TABLE>
  <TR>
    <TH>Name</TH>
    <TH>Breed</TH>
    <TH>Age</TH>
    <TH>Altered</TH>
```

```

<TH>Owner</TH>
</TR>
<xsl:for-each select="CAT">
<TR ALIGN="LEFT" VALIGN="TOP">
  <TD>
    <xsl:value-of select="NAME"/>
  </TD>
  <TD>
    <xsl:value-of select="BREED"/>
  </TD>
  <TD>
    <xsl:value-of select="AGE"/>
  </TD>
  <TD>
    <xsl:value-of select="ALTERED"/>
  </TD>
  <TD>
    <xsl:value-of select="DECLAWED"/>
  </TD>
  <TD>
    <xsl:value-of select="LICENSE"/>
  </TD>
  <TD>
    <xsl:value-of select="OWNER"/>
  </TD>
</TR>
</xsl:for-each>
</TABLE>

```

В этом примере показано, как может выглядеть текст одного из типов преобразования, но помните, что вы можете ограничиться четким описанием того, что вам нужно от данных, и это описание может быть сделано на вашем родном языке. Например, вы можете пойти в отдел ИТ и сказать, что необходимо напечатать данные о продажах для конкретных регионов за последние два года, и что эти сведения должны выглядеть так-то и так-то. После этого специалисты отдела могут написать (или изменить) преобразование, чтобы выполнить вашу просьбу.

Корпорация Майкрософт и растущее число других компаний создают преобразования для различных задач, что делает использование XML еще более удобным. В будущем, скорее всего, можно будет

небольшими изменениями. Это означает, что со временем использование XML будет требовать все меньше и меньше затрат.

XML в системе Microsoft Office

Профессиональные выпуски Office обеспечивают всестороннюю поддержку XML. Начиная с Microsoft Office 2007, в Microsoft Office используются форматы файлов на основе XML, например DOCX, XLSX и PPTX. Поскольку XML-данные хранятся в текстовом формате вместо запатентованного двоичного формата, ваши клиенты могут определять собственные схемы и использовать ваши данные разными способами без лицензионных отчислений. Дополнительные сведения о новых форматах см. в сведениях о [форматах Open XML и расширениях имен файлов](#). К другим преимуществам относятся:

- **Меньший размер файлов.** Новый формат использует ZIP и другие технологии сжатия, поэтому размер файла на 75 процентов меньше, чем в двоичных форматах, применяемых в более ранних версиях Office.
- **Более простое восстановление данных и большая безопасность.** Формат XML может быть легко прочитан пользователем, поэтому если файл поврежден, его можно открыть в Блокноте или другой программе для просмотра текста и восстановить хотя бы часть данных. Кроме того, новые файлы более безопасны, потому что они не могут содержать код Visual Basic для приложений (VBA). Если новый формат используется для создания шаблонов, то элементы ActiveX и макросы VBA находятся в отдельном, более безопасном разделе файла. Кроме того, можно удалять личные данные из документов с помощью таких средств, как инспектор документов. Дополнительные сведения об использовании инспектора документов см. в статье [Удаление скрытых и персональных данных при проверке документов](#).

Пока все хорошо, но что делать, если у вас есть данные XML без схемы? У Office программ, которые поддерживают XML, есть свои подходы к работе с данными. Например, Excel выдаст схему, если вы откроете XML-файл, который еще не имеет такой схемы. Excel затем вы можете загрузить эти данные в