Раздел CDATA

Если необходимо включить в XML-документ данные (в качестве содержимого элемента), которые содержат символы '**<**', '**>**', '**&**', ' ' ' и ' " ', чтобы не заменять их на соответствующие определения, можно все эти данные включить в раздел **CDATA**. Раздел **CDATA** начинается со строки "**<[CDATA[**", а заканчивается "**]]>**", при этом между ними эти строки не должны употребляться. Объявить раздел **CDATA** можно, например, так:

```
<data><[CDATA[ 5 < 7 ]]></data>
```

Корректность XML-документа определяют следующие два компонента:

- синтаксическая корректность (well-formed): то есть соблюдение всех синтаксических правил XML;
- действительность (valid): то есть данные соответствуют некоторому набору правил, определённых пользователем; правила определяют структуру и формат данных в XML. Валидность XML документа определяется наличием DTD или XML-схемы XSD и соблюдением правил, которые там приведены.

DTD

Для описания структуры XML-документа используется язык описания DTD (Document Type Definition). В настоящее время DTD практически не используется и повсеместно замещается XSD. DTD может встречаться в достаточно старых приложениях, использующих XML и, как правило, требующих нововведений (upgrade).

DTD определяет, какие теги (элементы) могут использоваться в XML-документе, как эти элементы связаны между собой (например, указывать на то, что элемент **<student>** включает дочерние элементы **<name>**, **<telephone>** и **<address>**), какие атрибуты имеет тот или иной элемент.

Это позволяет наложить четкие ограничения на совокупность используемых элементов, их структуру, вложенность.

Наличие DTD для XML-документа не является обязательным, поскольку возможна обработка XML и без наличия DTD, однако в этом случае отсутствует средство контроля действительности (validness) XML-документа, то есть правильности построения его структуры.

Чтобы сформировать DTD, можно создать либо отдельный файл и описать в нем структуру документа, либо включить DTD-описание непосредственно в документ XML.

В первом случае в документ XML помещается ссылка на файл DTD:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<! DOCTYPE students SYSTEM "students.dtd">
```

Во втором случае описание элемента помещается в XML-документ:

```
<?xml version="1.0" ?>
<! DOCTYPE student [
<!ELEMENT student (name, telephone, address)>
<!--
далее идет описание элементов name, telephone, address
-->
]>
```

Описание элемента

Элемент в DTD описывается с помощью дескриптора **!ELEMENT**, в котором указывается название элемента и его содержимое. Так, если нужно определить элемент **<student>**, у которого есть дочерние элементы **<name>**, **<telephone>** и **<address>**, можно сделать это следующим образом:

```
<!ELEMENT name (#PCDATA)>
<!ELEMENT telephone (#PCDATA)>
<!ELEMENT address (country, city, street)>
```

В данном случае были определены три элемента: **name, telephone** и **address** и описано их содержимое с помощью маркера **PCDATA**. Это говорит о том, что элементы могут содержать любую информацию, с которой может работать программа-анализатор (**PCDATA** – parsed character data). Есть также маркеры **EMPTY** – элемент пуст и **ANY** – содержимое специально не описывается.

При описании элемента **<student>**, было указано, что он состоит из дочерних элементов **<name>**, **<telephone>** и **<address>**. Можно расширить это описание с помощью символов '**+**'(один или много), '*****'(0 или много), '**?**'(0 или 1), используемых для указания количества вхождений элементов. Так, например,

```
<!ELEMENT student (name, telephone, address)>
```

означает, что элемент **student** содержит один и только один элемент **name**, **telephone** и **address**. Если существует несколько вариантов содержимого элементов, то используется символ '|' (или). Например:

```
<!ELEMENT student (#PCDATA | body)>
```

В данном случае элемент **student** может содержать либо дочерний элемент **body**, либо **PCDATA**.

Описание атрибутов

Атрибуты элементов описываются с помощью дескриптора **!ATTLIST**, внутри которого задаются имя атрибута, тип значения, дополнительные параметры и имеется следующий синтаксис:

```
<!ATTLIST название_елемента название_атрибута тип_атрибута
значение_по_умолчанию >
```

Например:

```
<!ATTLIST student
   login ID #REQUIRED
   faculty CDATA #REQUIRED>
```

В данном случае у элемента **<student>** определяются два атрибута: **login**, **faculty**. Существует несколько возможных значений атрибута, это:

CDATA – значением атрибута является любая последовательность символов;

ID – определяет уникальный идентификатор элемента в документе;

IDREF (**IDREFS**) – значением атрибута будет идентификатор (список идентификаторов), определенный в документе;

ENTITY (**ENTITIES**) – содержит имя внешней сущности (несколько имен, разделенных запятыми);

NMTOKEN (**NMTOKENS**) – слово (несколько слов, разделенных пробелами).

Опционально можно задать значение по умолчанию для каждого атрибута. Значения по умолчанию могут быть следующими:

#REQUIRED – означает, что атрибут должен присутствовать в элементе;

#IMPLIED – означает, что атрибут может отсутствовать, и если указано значение по умолчанию, то анализатор подставит его.

#FIXED – означает, что атрибут может принимать лишь одно значение, то, которое указано в DTD.

defaultValue – значение по умолчанию, устанавливаемое парсером при отсутствии атрибута. Если атрибут имеет параметр **#FIXED**, то за ним должно следовать **defaultValue**.

Если в документе атрибуту не будет присвоено никакого значения, то его значение будет равно заданному в DTD. Значение атрибута всегда должно указываться в кавычках.

Определение сущности

Сущность (entity) представляет собой некоторое определение, чье содержимое может быть повторно использовано в документе. Описывается сущность с помощью дескриптора **!ENTITY**:

```
<!ENTITY company 'Sun Microsystems'>
<sender>&company;</sender>
```

Программа-анализатор, которая будет обрабатывать файл, автоматически подставит значение `Sun Microsystems` вместо **&company**.

Для повторного использования содержимого внутри описания DTD используются параметрические (параметризованные) сущности.

```
   <!ENTITY   % elementGroup "firstName, lastName,gender,
address, phone">
   <!ELEMENT employee (%elementGroup;)>
   <!ELEMENT contact (%elementGroup)>
```

В XML включены внутренние определения для символов. Кроме этого, есть внешние определения, которые позволяют включать содержимое внешнего файла:

```
<!ENTITY logotype SYSTEM "/image.gif" NDATA GIF87A>
```

Файл DTD для документа **students.xml** будет иметь вид:

```
<?xml version='1.0' encoding='UTF-8'?>
<!ELEMENT students (student)+>
<!ELEMENT student (name, telephone, address)>
<!ATTLIST student
   login ID #REQUIRED
   faculty CDATA #REQUIRED
>
<!ELEMENT name (#PCDATA)>
<!ELEMENT telephone (#PCDATA)>
<!ELEMENT address (country, city, street)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT street (#PCDATA)>
```