

# XML Test Format v1.1 Описание

## Оглавление

[1 Статус данного документа](#)

[2 Список изменений](#)

[3 Введение](#)

[4 Как построено данное описание](#)

[4.1 DTD схема.](#)

[4.1.1 Как читать DTD?](#)

[4.1.1.1 Определение элемента](#)

[4.1.1.2 Определение атрибутов](#)

[4.1.1.3 Определение сущностей](#)

[4.2 Описание элементов](#)

[5 Требования к структуре файла](#)

[6 Язык описания теста](#)

[6.1 Элемент test](#)

[6.2 Раздел сведений о тесте](#)

[6.2.1 Элемент name](#)

[6.2.2 Элемент author](#)

[6.2.3 Элемент abstract](#)

[6.3 Раздел секций](#)

[6.3.1 Элемент selector](#)

[6.3.2 Элемент section](#)

[6.3.3 Элемент name](#)

[6.3.4 Элемент allout](#)

[6.3.5 Элемент goodout](#)

[6.3.6 Элемент midlout](#)

[6.3.7 Элемент badout](#)

[6.3.8 Атрибуты списков вопросов](#)

[6.3.9 Пример раздела секций](#)

[6.4 Раздел вопросов](#)

[6.4.1 Элемент questions](#)

[6.4.2 Элемент single](#)

[6.4.3 Элемент check](#)

[6.4.4 Элемент number](#)

[6.4.5 Элемент text](#)

[6.4.6 Элемент box](#)

[6.4.7 Атрибуты элементов вопросов](#)

[6.4.1 Содержимое вопроса](#)

[6.4.2 Элемент answer](#)

[6.4.3 Запись вопросов в стандартном и свободном формате](#)

[6.4.4 Пример раздела вопросов](#)

[6.5 Раздел сообщений](#)

[6.5.1 Элемент messages](#)

[6.5.2 Элемент sectmsg](#)

[6.5.3 Элемент questmsg](#)

[6.5.4 Содержимое сообщения](#)

[6.5.5 Пример раздела сообщений](#)

[7 Элементы форматирования](#)

[7.1 Элемент br](#)

[7.2Элемент img](#)

[7.3Элемент i](#)

[7.4Элемент b](#)

[7.5Элемент u](#)

[7.6Элемент font](#)

[7.7Элемент sub](#)

[7.8Элемент sup](#)

[7.9Элемент p](#)

[7.10Таблицы](#)

[7.10.1Элемент table](#)

[7.10.2Элемент tr](#)

[7.10.3Элемент td](#)

[7.11Специальные символы](#)

[8Приложение 1. Коды цветов](#)

[9Приложение 2. Test.DTD](#)

## **1 Статус данного документа**

Версия описываемого формата XTF: 1.1.

Версия данного документа: 1.1.

Последнюю версию данного документа можно получить по адресу:

<http://veralsoft.com/res.shtml>

Veralsoft не гарантирует, что данный документ свободен от ошибок. О найденных ошибках в этом документе сообщайте, пожалуйста, по адресу [support@veralsoft.com](mailto:support@veralsoft.com).

Veralsoft оставляет за собой право вносить изменения и модифицировать как данный документ, так и сам формат XTF. Все произведенные изменения будут по возможности описаны в новых версиях данного документа.

## **2 Список изменений**

1. В элемент описания секций section добавлен дочерний элемент name, задающий название секции.
2. Изменены правила указания ссылки на DTD схему в описании типа документа.

## **3 Введение**

Формат XTF предназначен для хранения документов тестов, созданных в программном комплексе VeralTest.

Под форматом XTF понимается:

1. Требования к структуре и содержанию файла теста.
2. Требования к языку описания теста.

XTF опирается на общепризнанные открытые стандарты такие как XML и ZIP, и сам, в свою очередь, является открытым. Полное описание формата XTF приведено в данном документе.

## **4 Как построено данное описание**

В первой части документа приведены требования к структуре и содержанию файла, записанного в стандарте XTF.

Во второй части приведены требования к языку описания теста, являющегося подмножеством стандарта XML.

## 4.1 DTD схема.

Язык описания теста задается при помощи DTD (Document Type Definition) схемы. DTD схема применяется для определения структуры XML документа и списка возможных элементов.

Для XTF документа определен файл test.dtd, который содержит DTD схему. Этот файл используется парсером XML для верификации структуры документа.

DTD схема для XTF приведена в [приложении 2](#).

### 4.1.1 Как читать DTD?

Здесь приведены краткие сведения, о формате записи DTD схемы, необходимые для понимания описаний, приведенных в данном документе. Для получения более подробной информации о DTD рекомендуем обратиться к документации "[Школы консорциума W3C](#)".

#### 4.1.1.1 Определение элемента

На языке DTD XML элемент определяется при помощи конструкции:

```
<!ELEMENT element-name (element-content)>
```

Здесь:

- element-name – имя элемента;
- element-content – возможное содержимое элемента.

Если элемент пуст (не имеет содержимого), то вместо (element-content) указывается ключевое слово EMPTY :

```
<!ELEMENT element-name EMPTY>
```

Если элемент содержит только символьные данные, то вместо (element-content) указывается ключевое слово (#PCDATA) :

```
<!ELEMENT element-name (#PCDATA)>
```

Если элемент содержит дочерние элементы, то вместо (element-content) должен быть приведен список дочерних элементов в скобках через запятую :

```
<!ELEMENT element-name (child-element1,child-element2,...,child-elementN)>
```

Здесь child-element – имя соответствующего дочернего элемента.

В приведенном выше примере каждый из описанных дочерних элементов должен обязательно присутствовать в содержимом элемента element-name в единственном экземпляре, при этом порядок дочерних элементов должен в точности соответствовать порядку их перечисления в круглых скобках.

Для изменения порядка вхождения дочерних элементов в родительский элемент применяют символьные модификаторы, которые записывают после имени дочернего

элемента:

"+" - дочерний элемент должен появляться внутри родительского элемента минимум один раз;

"?" - дочерний элемент может появляться внутри родительского элемента 0 или 1 раз;

"\*" - дочерний элемент может появляться внутри родительского элемента произвольное число раз.

Пример:

```
<!ELEMENT base (child1+, child2?, child3*)>
```

В данном примере элемент base должен содержать хотя бы один элемент child1, затем может содержать один элемент child2, затем – произвольное число элементов child3.

Если список дочерних элементов разделяется символом "|", то это означает вхождение элементов на условиях ИЛИ:

```
<!ELEMENT base (child1| child2)>
```

В данном примере элемент base должен содержать либо элемент child1, либо элемент child2.

Допускается комбинировать разделители:

```
<!ELEMENT base (child1,child2,(child3 | child4))>
```

В данном примере элемент base должен содержать элементы child1, child2 и один из двух элементов: child3 или child4.

Допускается определение смешанного содержимого для элементов:

```
<!ELEMENT base (#PCDATA|child1|child2|child3)*>
```

Здесь элемент base может содержать символьные данные и дочерние элементы child1, child2, child3 в любом количестве и порядке.

#### 4.1.1.2 Определение атрибутов

Список атрибутов для элемента на языке DTD задается при помощи конструкции:

```
<!ATTLIST element-name attribute-name attribute-type default-value>
```

Здесь:

- element-name – имя элемента, для которого задан список атрибутов;
- attribute-name – имя атрибута;
- attribute-type – тип значения атрибута;
- default-value – значение по умолчанию или условия использования атрибута.

В данном документе используются следующие типы значений атрибутов:

- CDATA – символьные данные;
- (en1,en2...,enN) – заданный список возможных значений.

Условия использования атрибута задаются при помощи ключевых слов

- #IMPLIED – Атрибут может быть опущен при описании элемента;
- #REQUIRED – Атрибут обязательно должен присутствовать в элементе.

Пример:

```
<!ATTLIST base
  att1 CDATA "0"
  att2 (y/n) "y"
  att3 CDATA #REQUIRED >
```

Здесь для элемента base определены три атрибута. Первый атрибут имеет имя att1, значением могут быть символьные данные, значение по умолчанию – "0". Второй атрибут - att2, значениями могут быть символы "y" или "n", значение по умолчанию – "y". Третий атрибут – att3, значением могут быть символьные данные, значение по умолчанию отсутствует, атрибут должен обязательно присутствовать при описании элемента.

#### 4.1.1.3 Определение сущностей

Сущность – это часть текста в описании DTD, которая задается в виде переменной.

Сущности сокращают размер DTD схемы, позволяя выделять часто используемые куски описания в отдельные блоки и использовать ссылки на эти блоки в остальных частях схемы.

Сущность определяется при помощи конструкции:

```
<!ENTITY % entity-name "entity-value">
```

Здесь:

- entity-name – имя сущности;
- entity-value – кусок текста, представляемый сущностью.

Ссылка на сущность определяется при помощи конструкции:

```
%entity-name;
```

Пример:

Пусть у элементов element1 и element2 используется одинаковый набор атрибутов:

```
<!ATTLIST element1
  att1 CDATA "0"
  att2 CDATA "0" >
```

Тогда, для сокращения размера DTD схемы, можно вынести описание атрибутов в отдельную сущность:

```
<!ENTITY % et "
  att1 CDATA '0'
  att2 CDATA '0'
">
```

Теперь список атрибутов для каждого элемента можно задать в виде:

```
<!ATTLIST element1 %et;>
```

## 4.2 Описание элементов

В данном документе при описании элементов XML приняты следующие правила:

Для каждого элемента указывается его назначение.

Для каждого элемента приводится его DTD определение, которое устанавливает:

1. Имя элемента
2. Содержимое элемента
3. Набор атрибутов элемента

Поскольку DTD схема не позволяет точно определить тип значений для всех атрибутов, то описание элементов, имеющих атрибуты, дополняется таблицей, детально описывающей назначение и возможные типы значений для каждого атрибута.

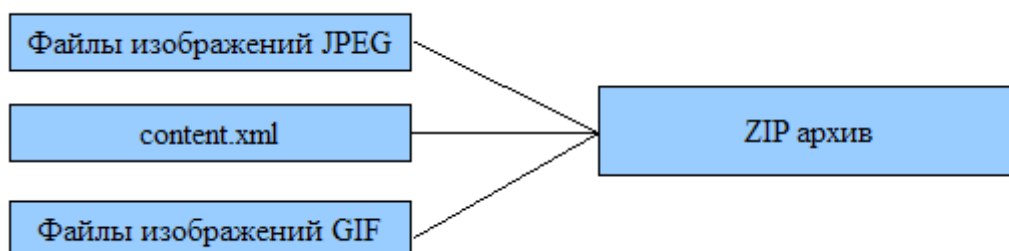
В XTF существуют специфические ограничения на вхождение и взаимное расположение некоторых элементов, которые не могут быть заданы DTD схемой. Эти ограничения обозначены при помощи дополнительных пояснений в описании соответствующих элементов.

## 5 Требования к структуре файла

Файл в формате XTF должен состоять из файла описания теста content.xml и файлов изображений в форматах GIF и JPEG, используемых в тесте.

Все файлы должны быть упакованы в один архив, сжатый по технологии ZIP. Создание каких-либо директорий внутри архива не допускается.

Полученный архив должен иметь расширение \*.xtf.



## 6 Язык описания теста

Файл content.xml представляет собой описание документа теста на языке XML.

Файл должен начинаться со стандартного заголовка XML в виде:

```
<?xml version="1.0" encoding="windows-1251"?>
```

Файл может быть записан в кодировке UTF-8 или ANSI (windows-1251). Соответствующее значение текущей кодировки должно быть указано в атрибуте

encoding заголовка XML документа.

После XML заголовка должен быть указан тип документа: test и имя dtd схемы:

```
<!DOCTYPE test SYSTEM="XTF_v1p1.dtd">
```

В этом примере указано, что содержимое документа соответствует формату XTF версии 1.1. Если требуется составить документ в старом формате XTF 1.0, то в качестве значения параметра SYSTEM нужно указать XTF\_v1p0.dtd или test.dtd

Корневым элементом документа является элемент test, т.е., все описание теста должно начинаться с открывающего тега <test> и заканчиваться закрывающим тегом </test>.

Пример минимального содержимого content.xml:

```
<?xml version="1.0" encoding="windows-1251"?>
<!DOCTYPE test SYSTEM="XTF_v1p1.dtd">
<test>
</test>
```

## 6.1 Элемент test

Является корневым элементом документа теста. Содержит все остальные элементы, описания теста.

```
<!ELEMENT test (name?,author?,abstract?,questions?,selector?,messages?)>
```

Описание теста состоит из следующих частей:

- Раздел сведений о тесте
- Раздел описания вопросов
- Раздел описания секций
- Раздел описания сообщений

## 6.2 Раздел сведений о тесте

Содержит название теста, сведения об авторе документа и текстовое описание документа.

### 6.2.1 Элемент name

Содержит название документа.

```
<!ELEMENT name (#PCDATA)>
```

### 6.2.2 Элемент author

Содержит сведения об авторе документа.

```
<!ELEMENT author (#PCDATA)>
```

### 6.2.3 Элемент *abstract*

Содержит дополнительные сведения о документе.

```
<!ELEMENT abstract (#PCDATA)>
```

## 6.3 Раздел секций

### 6.3.1 Элемент *selector*

Задаёт раздел секций теста. Каждая секция определяется при помощи элемента `<section>`.

```
<!ELEMENT selector (section)+>
```

### 6.3.2 Элемент *section*

Описывает одну секцию теста. Каждая секция может состоять из четырёх списков вопросов, предназначенных для различных уровней сложности.

```
<!ELEMENT section (name?,allout?,goodout?,midlout?,badout?)>  
<!ATTLIST section  
  id CDATA #IMPLIED  
>
```

#### Атрибуты

**id** – задаёт уникальный идентификатор секции в пределах теста. Значение может состоять из букв и цифр, но не должно начинаться с цифры.

### 6.3.3 Элемент *name*

Содержит название секции.

```
<!ELEMENT name (#PCDATA)>
```

### 6.3.4 Элемент *allout*

Определяет список вопросов, которые должны выводиться при любом уровне сложности.

```
<!ELEMENT allout EMPTY>  
<!ATTLIST allout %qlistatt;>
```

### 6.3.5 Элемент *goodout*

Определяет список вопросов, которые должны выводиться при максимальном уровне сложности.

```
<!ELEMENT goodout EMPTY>  
<!ATTLIST goodout %qlistatt;>
```

### 6.3.6 Элемент *midlout*

Определяет список вопросов, которые должны выводиться при среднем уровне сложности.

```
<!ELEMENT midlout EMPTY>  
<!ATTLIST midlout %qlistatt;>
```



### 6.3.7 Элемент *badout*

Определяет список вопросов, должны выводиться при минимальном уровне сложности.

```
<!ELEMENT badout EMPTY>  
<!ATTLIST badout %qlistatt;>
```

### 6.3.8 Атрибуты списков вопросов

```
<!ENTITY % qlistatt "  
  count CDATA #IMPLIED  
  random (y|n) 'y'  
  idref CDATA #REQUIRED  
">
```

**idref** – Определяет список порядковых номеров или идентификаторов вопросов, которые должны быть включены в данный список. Значение задается в виде строки, в которой перечисляются идентификаторы или порядковые номера через пробел.

**random** – Указывает способ выбора вопросов из списка. Значение "n" – вопросы выбираются по-порядку; Значение "y" – вопросы выбираются случайным образом.

**count** – Указывает количество вопросов, которые должны быть выбраны из списка. Значение должно быть положительным целым числом. Значение "0" означает, что должны быть выбраны все вопросы из списка.

### 6.3.9 Пример раздела секций

```
<selector>  
  <section id="a1" allout="5 4 12"/>  
  <section id="a2" allout="3 8 6"/>  
</selector>
```

## 6.4 Раздел вопросов

### 6.4.1 Элемент *questions*

Определяет раздел вопросов теста. Вопрос каждого типа задается при помощи соответствующего элемента.

```
<!ELEMENT questions (single|check|number|text|box)+>
```

### 6.4.2 Элемент *single*

Определяет вопрос с единичным выбором ответа.

```
<!ELEMENT single %Flow;>  
<!ATTLIST single %questatt;>
```

### 6.4.3 Элемент *check*

Определяет вопрос с множественным выбором ответа.

```
<!ELEMENT check %Flow;>  
<!ATTLIST check %questatt;>
```

#### 6.4.4 Элемент *number*

Определяет вопрос с вводом числового ответа.

```
<!ELEMENT number %Flow;>
<!ATTLIST number %questatt;>
```

#### 6.4.5 Элемент *text*

Определяет вопрос с вводом текстового ответа.

```
<!ELEMENT text %Flow;>
<!ATTLIST text %questatt;>
```

#### 6.4.6 Элемент *box*

Определяет вопрос с сопоставлением.

```
<!ELEMENT box %Flow;>
<!ATTLIST box %questatt;>
```

#### 6.4.7 Атрибуты элементов вопросов

```
<!ENTITY % questatt "
  id CDATA #IMPLIED
  random (y|n) 'y'
  format (custom|free) 'custom'
">
```

**id** – Задаёт уникальный идентификатор вопроса в пределах теста. Значение может состоять из букв и цифр, но не должно начинаться с цифры.

**random** – Указывает способ вывода ответов. Значение "n" – ответы выбираются по порядку; Значение "y" – ответы выбираются случайным образом.

**format** – Задаёт формат записи вопроса. Значение "custom" указывает на то, что вопрос записан в стандартном формате; Значение "free" – в свободном порядке.

#### 6.4.1 Содержимое вопроса

Элемент вопроса должен содержать текст вопроса и элементы описания ответов <answer>. Текст вопроса может содержать элементы форматирования (см. [элементы форматирования](#))

#### 6.4.2 Элемент *answer*

Определяет ответ для вопроса.

```
<!ELEMENT answer %Flow;>
<!ATTLIST answer %answatt;>
<!ENTITY % answatt "
  rating CDATA #REQUIRED
  right CDATA #IMPLIED
">
```

**Атрибуты:**

**rating** – Задаёт количество баллов за выбор данного ответа (для вопросов с выбором

ответа) или за правильный ответ (для вопросов с прямым вводом ответа). Значение должно быть целым положительным числом.

**right** – Задает правильный ответ для вопросов с вводом числа или строки. Задает правильный вариант сопоставления для вопросов с сопоставлением. В вопросах с выбором ответа не используется. Значение зависит от типа вопроса.

Для вопроса с числовым вводом ответа значение атрибута **right** задается в виде единичного числа или в виде числового диапазона: число1..число2. Числа могут быть как целыми, так и вещественными, в последнем случае в качестве разделителя целой и дробной части используется "десятичная точка" – символ ".".

Для вопроса с вводом текстового ответа значение атрибута **right** задается в виде строки правильного ответа, в которую могут быть включены управляющие символы:

Символ	Назначение
"\?"	На этом месте в ответе пользователя должен быть один любой символ
"\*"	На этом месте в ответе пользователя может быть любая последовательность символов
"\\"	На этом месте в ответе пользователя должен быть символ "\"

Для вопроса с сопоставлением значение атрибута **right** должно быть строкой, указывающей правильный вариант сопоставления.

### Содержимое

Элемент может содержать текст описания ответа. Текст может содержать элементы форматирования (см. [элементы форматирования](#)).

#### 6.4.3 Запись вопросов в стандартном и свободном формате

Если вопрос записан в стандартном формате (`format="custom"`), то текст вопроса и поля для ввода ответов располагаются отдельно друг от друга и их взаимным расположением управляет программа. Поля для ввода ответов с описаниями ответов могут выводиться как последовательно, так и в случайном порядке, в зависимости от значения атрибута `random`.

Если вопрос записан в свободном формате, то поля для ввода ответов отображаются прямо в тексте вопроса, в тех местах, где располагаются элементы `<answer>`. Описания ответов не выводятся. Поля для ввода ответов выводятся всегда в том же порядке, в каком располагаются соответствующие элементы `<answer>`.

#### 6.4.4 Пример раздела вопросов

```
<questions>
```

```
<single random="y" format="custom">  
Автор произведения "Ночной дозор"  
<answer rating="10">  
Сергей Лукьяненко
```

</answer>  
<answer rating="0">  
Дмитрий Глуховский

</answer>  
<answer rating="0">  
Сергей Павлов

</answer>  
</single>

<number random="y" format="custom">  
Какова высота горы Эверест в метрах?  
<answer rating="10" right="8844.8848" />  
</number>

<box random="y" format="custom">  
Сопоставьте классиков и их произведения  
<answer rating="10" right="Толстой">  
Война и мир  
</answer>  
<answer rating="10" right="Лермонтов">  
Герой нашего времени  
</answer>  
<answer rating="10" right="Лермонтов">  
Мцыри  
</answer>  
<answer rating="10" right="Пушкин">  
Евгений Онегин  
</answer>  
</box>

<check random="y" format="custom">  
Какие из этих городов входят в Золотое кольцо России?  
<answer rating="10">  
Суздаль  
</answer>  
<answer rating="10">  
Владимир  
</answer>  
<answer rating="0">  
Москва  
</answer>  
<answer rating="0">  
Воронеж  
</answer>  
<answer rating="10">  
Ярославль  
</answer>  
</check>

<text random="n" format="custom">  
Как звали девочку - главную героиню серии известных детских  
произведений писателя-фантаста Кира Булычева?

```

<answer rating="10" right="Алис\*" />
</text>

<text random="n" format="free">
  <p> Расставьте пропущенные слова в стихотворении: </p>
  <p> И днем и
  <answer rating="10" right="ночью" />
  кот
  <answer rating="10" right="ученый" />
  <br/>
  Все ходит по
  <answer rating="10" right="цепи" />
  кругом...
  </p>
</text>

</questions>

```

## 6.5 Раздел сообщений

### 6.5.1 Элемент *messages*

Определяет раздел сообщений теста. Сообщения могут выводиться после выполнения секций или вопросов теста.

```
<!ELEMENT messages (sectmsg|questmsg)+>
```

### 6.5.2 Элемент *sectmsg*

Задаёт сообщение для секций.

```

<!ELEMENT sectmsg %Flow;>
<!ATTLIST sectmsg
  idrating CDATA 'all'
  idref CDATA 'all'
>

```

#### Атрибуты

**idref** – Задаёт идентификаторы или порядковые номера секций, для которых предназначено сообщение. Значение задается в виде строки, в которой перечисляются идентификаторы или порядковые номера через пробел. Значение "all" означает, что сообщение предназначено для всех секций.

**idrating** – Задаёт результат выполнения секции, по достижению которого должно быть выведено данное сообщение:

- all – любой результат;
- bad – минимальный результат (кол-во набранных баллов за секцию равно нулю);
- middle – средний результат (кол-во набранных баллов за секцию больше нуля, но меньше максимально возможного);
- good – максимальный результат (набрано максимально возможное кол-во баллов).

### 6.5.3 Элемент *questmsg*

Задаёт сообщения для вопросов.

```
<!ELEMENT questmsg %Flow;>
<!ATTLIST questmsg
  idrating CDATA 'all'
  idref    CDATA 'all'
>
```

#### Атрибуты

**idref** – Задаёт идентификаторы или порядковые номера вопросов, для которых предназначено сообщение. Значение задается в виде строки, в которой перечисляются идентификаторы или порядковые номера через пробел. Значение "all" означает, что сообщение предназначено для всех вопросов.

**idrating** – Задаёт результат выполнения вопроса, по достижению которого должно быть выведено данное сообщение:

- all – любой результат;
- bad – минимальный результат (кол-во набранных баллов за вопрос равно нулю);
- middle – средний результат (кол-во набранных баллов за вопрос больше нуля, но меньше максимально возможного);
- good – максимальный результат (набрано максимально возможное кол-во баллов).

### 6.5.4 Содержимое сообщения

Текст с элементами форматирования (см. [элементы форматирования](#)).

### 6.5.5 Пример раздела сообщений

```
<messages>
  <sectmsg idrating="all" idref="1">
    <b>Закончена первая секция</b>
  </sectmsg>
  <questmsg idrating="good" idref="all">
    Отлично!
  </questmsg>
  <questmsg idrating="middle" idref="all">
    Можно лучше...
  </questmsg>
  <questmsg idrating="bad" idref="all">
    Плохо
  </questmsg>
</messages>
```

## 7 Элементы форматирования

Элементы форматирования применяются для оформления текста, вставки изображений и таблиц в текст. Элементы форматирования могут быть использованы в тексте вопроса, описании ответа, тексте сообщения.

Элементы форматирования в XTF позаимствованы из языка XHTML и совместимы с ним, в то же время многие конструкции элементов значительно упрощены и адаптированы для использования в тесте.

Условно все элементы форматирования в XTF можно разделить на несколько групп:

- Элементы оформления текста: <font>, <i>, <b>, <u>;
- Элементы, влияющие на смещение текста: <sub>, <sup>;
- Элементы разделения текста: <p>, <br>;
- Элемент вставки изображений: <img>;
- Элемент вставки таблиц: <table>.

```
<!ENTITY % TextAlign "align (left|center|right) #IMPLIED">
<!ENTITY % ImgAlign "(left|right|texttop|absmiddle|baseline|absbottom|bottom|
middle|top)">
<!ENTITY % Length "CDATA">
<!ENTITY % URI "CDATA">
<!ENTITY % Text "CDATA">
<!ENTITY % Pixels "CDATA">
<!ENTITY % Color "CDATA">
<!ENTITY % Character "CDATA">
<!ENTITY % Number "CDATA">
<!ENTITY % MultiLength "CDATA">
<!ENTITY % special "br | img ">
<!ENTITY % fontstyle "i | b | u |font">
<!ENTITY % phrase "sub | sup">
<!ENTITY % inline "answer|%special; | %fontstyle; | %phrase;">
<!ENTITY % Inline "(#PCDATA | %inline;)*">
<!ENTITY % Flow "(#PCDATA | p | table | %inline;)*">
```

Примечание. Использование элемента <answer> допускается только при описании текста вопроса.

## 7.1 Элемент br

Принудительный разрыв строки.

```
<!ELEMENT br EMPTY>
```

## 7.2 Элемент img

Вставляет изображение в текст.

```
<!ELEMENT img EMPTY>
<!ATTLIST img
  src      %URI;      #REQUIRED
  alt      %Text;     #IMPLIED
  height   %Length;   #IMPLIED
  width    %Length;   #IMPLIED
  align    %ImgAlign; #IMPLIED
  hspace   %Pixels;   #IMPLIED
  vspace   %Pixels;   #IMPLIED
>
```

## Атрибуты

**src** – Указывает имя файла изображения. В отличие от XHTML значение может быть только именем файла без указания пути или URL.

**alt** – Указывает альтернативный текст для изображения.

**height** – Высота изображения в процентах или пикселах. По умолчанию берется исходная высота изображения.

**width** – Ширина изображения в процентах или пикселах. По умолчанию берется исходная ширина изображения.

**align** – Задаёт выравнивание изображения относительно текста. Значения:

- **left** – Изображение выравнивается по левому краю. Текст обтекает изображение справа.
- **Right** – Изображение выравнивается по правому краю. Текст обтекает изображение слева.
- **texttop, top** – Верхняя кромка изображения находится вровень с верхней кромкой текстовой строки.
- **absmiddle** – Центр изображения находится вровень с центром текстовой строки.
- **baseline, bottom** – Нижняя кромка изображения находится вровень с базовой линией текстовой строки. Значение по-умолчанию.
- **Absbottom** – Нижняя кромка изображения находится вровень с нижней кромкой текстовой строки.
- **middle** – Центр изображения находится вровень с базовой линией текстовой строки.

**hspace** – Горизонтальный отступ от текста в пикселах.

**vspace** – Вертикальный отступ от текста в пикселах.

## 7.3 Элемент i

Выделяет текст курсивом.

```
<!ELEMENT i %Inline;>
```

## 7.4 Элемент b

Выделяет текст полужирным шрифтом.

```
<!ELEMENT b %Inline;>
```

## 7.5 Элемент u

Подчеркивает текст

```
<!ELEMENT u %Inline;>
```



## 7.6 Элемент font

Задаёт тип, размер и цвет шрифта.

```
<!ELEMENT font %Flow;>
<!ATTLIST font
  size      CDATA      #IMPLIED
  color     %Color;    #IMPLIED
  face      CDATA      #IMPLIED
>
```

### Атрибуты

**face** – Имя шрифта.

**size** – Размер шрифта. В XTF применяются условные размеры шрифтов. Значением может быть число в интервале от 1 до 7. Точный размер шрифта определяется в зависимости от настроек браузера или настроек печати при формировании бумажной версии теста.

**color** – Цвет текста (см. [приложение 1](#)).

## 7.7 Элемент sub

Выводит подстрочный текст.

```
<!ELEMENT sub %Inline;>
```

## 7.8 Элемент sup

Выводит надстрочный текст.

```
<!ELEMENT sup %Inline;>
```

## 7.9 Элемент p

Задаёт абзац текста.

```
<!ELEMENT p %Inline;>
<!ATTLIST p
  %TextAlign;
>
```

### Атрибуты

**align** – Выравнивание текста в абзаце. Возможные значения:

- left – Текст выравнивается по левому краю. Это значение по умолчанию;
- center – Текст выравнивается по центру;
- right – Текст выравнивается по правому краю.

## 7.10 Таблицы

```
<!ENTITY % TAlign "(left|center|right)">
<!ENTITY % cellhalign
  "align (left|center|right) #IMPLIED
>
```

```
<!ENTITY % cellvalign
"valign (top|middle|bottom) #IMPLIED "
>
```

### 7.10.1 Элемент table

Вставляет таблицу в документ.

```
<!ELEMENT table
  (#PCDATA | tr)*>
<!ATTLIST table
width      %Length;    #IMPLIED
border     %Pixels;    #IMPLIED
cellspacing %Length;    #IMPLIED
cellpadding %Length;    #IMPLIED
align      %TAlign;    #IMPLIED
bgcolor    %Color;     #IMPLIED
>
```

#### Атрибуты

**width** – Задаёт ширину таблицы в пикселах или процентах от рабочей области. По умолчанию ширина таблицы определяется автоматически, исходя из содержимого.

**border** – Задаёт ширину бордюра таблицы в пикселах. Значение "0" означает отсутствие бордюра и является значением по-умолчанию.

**cellspacing** – определяет расстояние в пикселах между границами соседних ячеек.

**cellpadding** – определяет расстояние в пикселах между рамкой ячейки и её содержимым.

**align** – определяет положение таблицы по горизонтали. Возможные значения:

- left – у левого края рабочей области;
- center – в центре рабочей области;
- right – у правого края рабочей области.

**bgcolor** – цвет фона (см. [приложение 1](#)).

### 7.10.2 Элемент tr

Задаёт строку таблицы.

```
<!ELEMENT tr (#PCDATA | td)*>
<!ATTLIST tr
  %cellhalign;
  %cellvalign;
  bgcolor %Color;    #IMPLIED
>
```

#### Атрибуты

**align** – Задаёт горизонтальное выравнивание содержимого всех ячеек строки. Возможные значения:

- left – по левому краю ячейки;
- center – по центру ячейки;

- right – по правому краю ячейки.

**valign** – Задаёт вертикальное выравнивание содержимого всех ячеек строки.

Возможные значения:

- top – по верхнему краю ячейки;
- middle – по средней линии ячейки;
- bottom – по нижнему краю ячейки.

**bgcolor** – цвет фона строки (см. [приложение 1](#)).

### 7.10.3 Элемент *td*

Задаёт ячейку таблицы.

```
<!ELEMENT td %Flow;>
<!ATTLIST td
  rowspan    %Number;    "1"
  colspan    %Number;    "1"
  %cellhalign;
  %cellvalign;
  nowrap    (nowrap)    #IMPLIED
  bgcolor    %Color;     #IMPLIED
  width     %Pixels;    #IMPLIED
  height    %Pixels;    #IMPLIED
>
```

#### Атрибуты

**rowspan** – Количество строк, которое занимает ячейка. Значение по умолчанию – "1".

**colspan** – Количество столбцов, которое занимает ячейка. Значение по умолчанию – "1".

**align, valign** – Выравнивание содержимого ячейки по горизонтали и вертикали (см. элемент *tr*).

**nowrap** – Запрет переноса текста в ячейке. Значение true или nowrap – включить запрет.

**bgcolor** – Цвет фона ячейки (см. [приложение 1](#)).

**width** – Ширина ячейки в пикселах.

**height** – Высота ячейки в пикселах.

## 7.11 Специальные символы

Поскольку стандарт XML резервирует некоторые символы для служебного использования, то для использования данных символов в тексте, необходимо заменять их специальными символьными ссылками:

СИМВОЛ	ССЫЛКА
< - меньше чем	&lt;
> - больше чем	&gt;
& - амперсанд	&amp;

" - двойная кавычка	&quot;
' - апостроф	&apos;

Ссылки всегда начинаются с амперсанда "&" и заканчиваются точкой с запятой ";".

**Пример.**

Вместо

**a<b**

Следует писать:












**a&lt;b**

## 8 Приложение 1. Коды цветов

Значения для атрибутов, устанавливающих цвет, могут быть заданы в виде шестнадцатеричного представления цвета RGB, например, белый цвет может быть записан в виде: #FFFFFF, а красный - #FF0000.

Для шестнадцати базовых цветов можно использовать названия вместо кодов, например для белого цвета - "white", для черного – "black" и т.д.

Список базовых цветов с наименованиями и кодами RGB приведен в таблице:

	black (#000000)		navy (#000080)
	gray (#808080)		blue (#0000FF)
	silver (#C0C0C0)		aqua (#00FFFF)
	white (#FFFFFF)		green (#008000)
	red (#FF0000)		lime (#00FF00)
	fuchsia (#FF00FF)		teal (#008080)
	maroon (#800000)		yellow (#FFFF00)
	purple (#800080)		olive (#808000)

## 9 Приложение 2. Test.DTD

```
<!--XTF DTD Created 01.06.2008-->
<!--MARKUP ELEMENTS-->
<!ENTITY % TextAlign "align (left|center|right) #IMPLIED">
<!ENTITY % ImgAlign "(left|right|texttop|absmiddle|baseline|absbottom|bottom|
middle|top)">
<!ENTITY % Length "CDATA">
<!ENTITY % URI "CDATA">
<!ENTITY % Text "CDATA">
<!ENTITY % Pixels "CDATA">
<!ENTITY % Color "CDATA">
<!ENTITY % Character "CDATA">
<!ENTITY % Number "CDATA">
<!ENTITY % MultiLength "CDATA">
<!ENTITY % special "br | img ">
<!ENTITY % fontstyle "i | b | u |font">
<!ENTITY % phrase "sub | sup">
<!ENTITY % inline "answer|%special; | %fontstyle; | %phrase;">
<!ENTITY % Inline "(#PCDATA | %inline;)*">
<!ENTITY % Flow "(#PCDATA | p | table | %inline;)*">

<!ELEMENT br EMPTY>
<!ELEMENT img EMPTY>
<!ATTLIST img
  src      %URI;      #REQUIRED
  alt      %Text;     #IMPLIED
  height   %Length;   #IMPLIED
  width    %Length;   #IMPLIED
  align    %ImgAlign; #IMPLIED
  border   %Pixels;   #IMPLIED
  hspace   %Pixels;   #IMPLIED
  vspace   %Pixels;   #IMPLIED
  >

<!ELEMENT i %Inline;>
<!ELEMENT b %Inline;>
<!ELEMENT u %Inline;>
<!ELEMENT sub %Inline;>
<!ELEMENT sup %Inline;>

<!ELEMENT font %Flow;>
<!ATTLIST font
  size     CDATA      #IMPLIED
  color    %Color;   #IMPLIED
  face     CDATA      #IMPLIED
  >

<!ELEMENT p %Inline;>
<!ATTLIST p
  %TextAlign;
  >
```

```

<!--HTML TABLES-->
<!ENTITY % TAlign "(left|center|right)">
<!ENTITY % cellhalign
"align (left|center|right) #IMPLIED "
>
<!ENTITY % cellvalign
"valign (top|middle|bottom) #IMPLIED "
>
<!ELEMENT table (#PCDATA | tr)*>
<!ELEMENT tr (#PCDATA | td)*>
<!ELEMENT td %Flow;>
<!ATTLIST table
width %Length; #IMPLIED
border %Pixels; #IMPLIED
cellspacing %Length; #IMPLIED
cellpadding %Length; #IMPLIED
align %TAlign; #IMPLIED
bgcolor %Color; #IMPLIED
>
<!ATTLIST tr
%cellhalign;
%cellvalign;
bgcolor %Color; #IMPLIED
>
<!ATTLIST td
rowspan %Number; "1"
colspan %Number; "1"
%cellhalign;
%cellvalign;
nowrap (nowrap) #IMPLIED
bgcolor %Color; #IMPLIED
width %Pixels; #IMPLIED
height %Pixels; #IMPLIED
>

<!--TEST ELEMENTS-->
<!ENTITY % questatt "
id CDATA #IMPLIED
random (y|n) 'y'
format (custom|free) 'custom'
">
<!ENTITY % qlistatt "
count CDATA #IMPLIED
random (y|n) 'y'
idref CDATA #REQUIRED
">
<!ENTITY % answatt "
rating CDATA #REQUIRED
right CDATA #IMPLIED
">

<!ELEMENT test (name?,author?,abstract?,questions?,selector?,messages?)>

```

```
<!ELEMENT name (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT abstract (#PCDATA)>
<!ELEMENT selector (section)+>
<!ELEMENT section (name?,allout?,goodout?,midlout?,badout?)>
<!ATTLIST section
  id CDATA #IMPLIED
>
<!ELEMENT allout EMPTY>
<!ATTLIST allout %qlistatt;>
<!ELEMENT badout EMPTY>
<!ATTLIST badout %qlistatt;>
<!ELEMENT goodout EMPTY>
<!ATTLIST goodout %qlistatt;>

<!ELEMENT midlout EMPTY>
<!ATTLIST midlout %qlistatt;>
<!ELEMENT design (template)+>
<!ELEMENT template %Flow;>
<!ATTLIST template
  idtype CDATA 'all'
  idref CDATA #IMPLIED
>
<!ELEMENT messages (sectmsg|questmsg)+>
<!ELEMENT sectmsg %Flow;>
<!ATTLIST sectmsg
  idrating CDATA 'all'
  idref CDATA 'all'
>
<!ELEMENT questmsg %Flow;>
<!ATTLIST questmsg
  idrating CDATA 'all'
  idref CDATA 'all'
>
<!ELEMENT questions (single|check|number|text|box)+>
<!ELEMENT single %Flow;>
<!ATTLIST single %questatt;>
<!ELEMENT check %Flow;>
<!ATTLIST check %questatt;>
<!ELEMENT number %Flow;>
<!ATTLIST number %questatt;>
<!ELEMENT text %Flow;>
<!ATTLIST text %questatt;>
<!ELEMENT box %Flow;>
<!ATTLIST box %questatt;>
<!ELEMENT answer %Flow;>
<!ATTLIST answer %answatt;>
```