

Создание тестов без редактора

Введение

Редактор тестов, входящий в состав VeralTest, является довольно удобным и гибким инструментом для создания и редактирования тестов любой сложности. В то же время в ряде ситуаций предпочтительнее было бы составление теста в любом удобном текстовом редакторе, например в блокноте, с разметкой специальными символами вопросов и ответов.

Некоторых пользователей, например, смущает большое количество окон, появляющихся при редактировании теста. У многих уже есть готовые тесты, набранные в текстовых редакторах, например MS Word, или просто отсканированные с бумажных копий, и занесение всей информации в редактор тестов методом copy+paste становится трудоемким, особенно при большом количестве вопросов.

VeralTest предоставляет такую возможность. Все дело в том, что формат файлов, в которых хранятся тесты, основан на языке XML – специальном языке разметки гипертекста. Файлы *.xtf, используемые VeralTest, представляют собой zip архивы, в которых содержится файл content.xml (это непосредственно сам тест) и файлы рисунков, используемых в тесте. Чтобы проверить это, достаточно изменить расширение файла на zip, и распаковать полученный архив.

Краткие сведения о языке XML

XML – это универсальный язык разметки, позволяющий с помощью специальных ключевых слов (тегов) выделять в тексте определенные элементы. Звучит сложно? Ну хорошо, вот простой пример: Рассмотрим один из самых распространенных документов – письмо электронной почты. Какие элементы можно выделить в этом документе?

1. Адрес отправителя; 2. Адрес получателя; 3. Тема письма; 4. Текст письма. На языке XML этот документ можно записать так:

```
<mail>  
<from>Адрес отправителя</from>  
<to>Адрес получателя</to>  
<content>Текст письма</content>  
</mail>
```

Ключевые слова <mail>, </mail>, <from>, </from>, <to>, </to>, <content>, </content> и есть теги, они отделяют одни элементы документа от других. Из нашего примера легко видеть, что каждый тег имеет своего «однофамильца» - отличающегося только символом «/» перед названием. Это открывающий и закрывающий теги. Пара таких тегов называется «Контейнер». Почему контейнер? Потому что, как и любой контейнер, он может хранить что-нибудь внутри себя. Например, контейнер <content></content> содержит внутри себя текст письма, а контейнер <from></from> – адрес отправителя. Особняком в этом примере стоит контейнер <mail></mail> - он хранит в себе все письмо целиком. Про него можно сказать, что он хранит не только текст, но и другие контейнеры.

И так, любой XML документ – это набор контейнеров. Каждый контейнер состоит из открывающего и закрывающего тега. Открывающий тег должен быть записан в виде

<имя_тега>, а закрывающий – в виде </имя_тега>. Внутри контейнера может содержаться текст или другие контейнеры. Видите, как все просто!

Внимательный читатель, конечно же, задаст вопрос: «А какие имена должны быть у тегов?». И вот на этот вопрос я, к сожалению, не смогу ответить. И дело здесь не в том, что я плохо знаком с языком XML, просто язык XML задает лишь правила написания тегов, а имена тегов могут быть любыми. Наш пример можно записать и в таком виде:

```
<почта>  
  <отправитель>Адрес отправителя</отправитель>  
  <получатель>Адрес получателя</получатель>  
  <содержимое>Текст письма</содержимое>  
</почта>
```

С точки зрения XML этот документ имеет такое же право на существование, как и предыдущий.

Но как же тогда узнать, какие теги использовать? Конкретный набор тегов зависит от программы, которая хранит свои данные в XML. Например, почтовые программы могут для хранения и передачи электронных сообщений использовать свои наборы тегов, такие как <from>, <to>, <mail>, а программы для каталогизации музыкальных дисков - <author>, <album> и т.д. Стандарт XML тем и хорош, что позволяет каждой программе использовать свой набор тегов.

Приведенный здесь краткий обзор ни в коем случае не претендует на полноту и строгость описания XML. Этот краткий экскурс нужен лишь для понимания дальнейшего материала, изложенного ниже в данной статье. Если вы хотите узнать больше о XML, его возможностях и сопутствующих технологиях, могу порекомендовать сайт Школы Консорциума W3C (<http://xml.nsu.ru>), содержащий статьи, в которых просто и доходчиво изложены основы языка.

Напоследок приведу еще несколько сведений. В Языке XML теги и контейнеры принято называть элементами. Элементы могут содержать атрибуты, которые записываются внутри угловых скобок после имени элемента. Атрибуты содержат некоторую дополнительную информацию об элементе. Например, почтовая программа может хранить дату каждого сообщения следующим образом:

```
<mail date="31.12.2007">  
  <from>Адрес отправителя</from>  
  <to>Адрес получателя</to>  
  <content>Текст письма</content>  
</mail>
```

Атрибут записывается в виде имя="значение". Значение атрибута всегда должно помещаться в кавычки.

И еще: Любой XML документ должен начинаться с такой строки:

```
<?xml version="номер_версии"?>
```

Закрывающий тег здесь не нужен.

Что нам понадобится

Для составления теста на языке xml подойдет действительно любой текстовый редактор. Можете писать даже в блокноте, если он Вам нравится. В то же время существует великое множество специализированных XML редакторов, в том числе и бесплатных. Специализированные XML редакторы удобны тем, что они выделяют при помощи цвета XML элементы и проверяют синтаксическую корректность редактируемого XML кода. В качестве примера такого редактора можно привести бесплатную программу WM Help XMLPad, которую можно загрузить по адресу <http://wmhelp.com/download.htm>

Создаем простой тест

Рассмотрим создание простого теста. Наш тест будет состоять из двух вопросов, вопрос с единственным выбором ответа и вопрос с множественным выбором ответа:

Вопрос №1.

Кто автор произведения «Евгений Онегин»?

- a) Лермонтов
- b) Пушкин
- c) Грибоедов

Вопрос №2.

Какие из этих городов находятся в Российской Федерации?

- a) Кишинев
- b) Ярославль
- c) Минск
- d) Нижний Новгород

Теперь рассмотрим, как этот тест должен быть записан при помощи XML.

```
<?xml version="1.0" encoding="windows-1251"?>
<!DOCTYPE test SYSTEM "test.dtd">
<test>
  <questions>
    <single>
      Кто автор произведения «Евгений Онегин»?
      <answer rating="0"> Лермонтов</answer>
      <answer rating="1"> Пушкин</answer>
      <answer rating="0"> Грибоедов</answer>
    </single>
    <check>
      Какие из этих городов находятся в Российской Федерации?
      <answer rating="0"> Кишинев</answer>
      <answer rating="1"> Ярославль</answer>
      <answer rating="0"> Минск</answer>
      <answer rating="1"> Нижний Новгород</answer>
    </check>
  </questions>
</test>
```

Посмотрим, из каких элементов состоит написанный нами текст.

Первая строчка – это стандартный заголовок XML документа. Мы уже говорили о нем выше. Атрибут encoding указывает кодировку, в которой записан текст.

Вторая строчка задает ссылку на файл, в котором хранятся правила для составления тестов на языке XML. Вот в нем-то как раз и указано, какие названия элементов могут использоваться и в каком порядке элементы должны следовать.

Первые две строчки используются без изменений в любом тесте, поэтому при создании новых тестов их можно просто скопировать их из этого примера и вставить в начало нового документа.

Далее следует элемент <test> - это корневой элемент, который содержит собственно сам тест.

Внутри элемента <test> находится блок вопросов, который задается элементом <questions>. Кроме блока вопросов в тесте могут содержаться еще секции и сообщения, но об этом мы поговорим в следующий раз.

Внутри блока <questions> находятся непосредственно сами вопросы. В системе VeralTest различают пять типов вопросов, и каждый тип вопроса задается собственным элементом. Перечислим их:

| Тип вопроса | Элемент |
|--------------------------------|---------|
| Вопрос с единичным выбором | single |
| Вопрос с множественным выбором | check |
| Вопрос с вводом числа | number |
| Вопрос с вводом текста | text |
| Вопрос с сопоставлением | box |

В нашем примере присутствуют вопросы двух типов: Вопрос с единичным выбором и Вопрос с множественным выбором, поэтому мы используем элементы <single> и <check>.

Внутри элемента вопроса записан текст вопроса и описания ответов. Каждый ответ представлен элементом <answer>. Атрибут rating элемента <answer> задает количество баллов за выбор этого ответа. Значение атрибута rating может быть любым положительным целым числом.

В нашем примере в первом вопросе правильным ответом является второй (b), а во втором вопросе – второй и четвертый (b,d). Пусть за каждый правильный выбор тестируемому дается один балл. Тогда, при описании первого вопроса мы задаем rating="1" у второго элемента <answer>, а у всех остальных – rating="0". У второго вопроса значение атрибута rating="1" устанавливаем у второго и четвертого элемента <answer>, а у остальных – rating="0".

Таким образом, при правильном ответе на оба вопроса испытуемый получит три балла.

Запишите этот пример в удобном для вас текстовом редакторе. Будьте внимательны, не допускайте ошибок при написании элементов, иначе ничего не получится.

После того, как тест составлен, нужно сохранить его с именем content.xml. Если вы пользуетесь специализированным XML редактором, то достаточно просто нажать на кнопку «Сохранить» и дать название content сохраняемому файлу.

Если вы пользуетесь редактором простого текста, например блокнотом, то выберите пункт меню «Файл/сохранить как», в диалоговом окне сохранения выберите тип файла: «Все файлы» и дайте имя сохраняемому файлу content.xml.

Если вы пользуетесь текстовым процессором типа MS Word, то задача несколько усложняется. Вам сначала надо сохранить файл как обычный текст (*.txt), а затем переименовать его в content.xml

Если вы пользуетесь проводником Windows для операций с файлами, то при переименовании файлов можете столкнуться со следующей трудностью: По умолчанию проводник скрывает расширения файлов, показывая только их имя, например, content вместо content.xml. Чтобы исправить эту ситуацию, выполните следующие действия:

1. Откройте «Мой компьютер».
2. Выберите пункт меню «Сервис/Свойства папки»
3. В открывшемся окне «Свойства папки» перейдите на закладку «Вид»
4. Найдите и снимите галочку у пункта «Скрывать расширения для зарегистрированных типов файлов»
5. Нажмите «Ок».

Следующим шагом будет упаковка полученного файла content.xml в архив zip. Пользователи Windows XP и Vista могут сделать это следующим образом:

1. Откройте папку, в которую вы сохранили файл content.xml.
2. Щелкните левой кнопкой мышки на имени файла и в выпадающем меню выберите пункт «Отправить/Сжатая ZIP папка»
3. Выполнив эти действия, вы получите в той же папке файл архива: content.zip

Пользователям младших версий Windows для упаковки придется использовать сторонние архиваторы: WinZip, WinRar и т.д.

Последний шаг состоит в переименовании полученного zip архива content.zip. Дайте файлу удобное название и расширение: *.xtf, например mytest.xtf

Файл теста готов. Чтобы проверить его работоспособность, откройте полученный файл в редакторе тестов. Если все сделано правильно, то файл откроется, и вы можете просматривать и редактировать вопросы, запускать тест на выполнение. Если при открытии файла появилось сообщение об ошибке, значит, вы допустили неточность при написании кода XML – найдите ошибку в своем коде и повторите операцию еще раз.

И напоследок расскажу об еще одной полезности. Если вы не пользуетесь специализированными XML редакторами, то велика вероятность, что вы допустите ошибку при написании теста. Чтобы можно было быстро найти и исправить эту ошибку еще до упаковки файла zip архив, откройте ваш файл content.xml в веб браузере (например, Internet Explorer или Firefox). Современные браузеры поддерживают стандарт XML, и если ваш файл содержит ошибку, они покажут вам описание этой ошибки и номер строки в вашем файле, где ошибка находится.

Заключение

В этой статье мы рассмотрели возможность создания тестов для системы VeralTest на языке XML без использования встроенного редактора тестов. Конечно, многое осталось за

рамками данной статьи. Впереди нас ждет знакомство с форматированием текста вопросов, вопросы с вводом ответа и сопоставлением, создание секций и сообщений. В общем, продолжение следует...

Соколов А.Ф., veralsoft.com