



developerWorks Россия > Технические материалы > XML > Статьи >

# Начало работы с XPath

## Изучение основ

Это руководство является введением и рассматривает основные аспекты языка XML Path (XPath). Оно предназначено для людей, которые не знают, что такое XPath или хотят освежить свои знания. Если Вы планируете использовать XSLT, вам следует изучить это руководство. Вы узнаете, что такое XPath; синтаксис и семантику языка XPath; как использовать XPath пути, выражения, функции, и каким образом XPath связан с XSLT.

Бертран Портье - инженер по программному обеспечению, работающий в IBM Software Group Services. Он работает с пользователями IBM WebSphere, обеспечивая решения для задач их бизнеса. До этого он работал разработчиком в области Web-сервисов. Вы можете связаться с ним по адресу [portier@uk.ibm.com](mailto:portier@uk.ibm.com).

11.05.2004

## Введение

### Нужен ли мне этот учебник?

Это руководство рассматривает большинство аспектов XML Path Language или XPath. Оно предназначено для людей, которые не знают XPath или хотят освежить свои знания. Если вы собираетесь использовать XSLT, вы должны сначала обратиться к этому учебнику. Вы изучите:

- Что такое XPath
- Синтаксис и семантику языка XPath
- Как использовать маршруты местоположения XPath
- Как использовать выражения XPath
- Как использовать функции XPath
- Как XPath соотносится с XSLT

XPath является стандартом W3C. Этот учебник фокусируется на версии 1.0. За информацией о грядущей XPath 2.0 обратитесь к Заключению.

### Предварительные требования

Этот учебник предполагает базовые знания об XML. Например, вы должны знать, что такое элементы, атрибуты и значения. Если вы не знакомы с этим, я рекомендую вам обратиться к учебнику "Введение в XML", ссылка на который есть в Ресурсах.

Вам также понадобится текстовый редактор и Web-браузер. Хотя это и не является обязательным условием, может оказаться полезным иметь XML-редактор, если в него будет включен вычислитель XPath, что позволит вам проверять используемые здесь выражения XPath.

### Технологии, относящиеся к XML

Ниже перечисляются технологии, упоминаемые в этом учебнике. Пропустите их, если вы уже знакомы с относящимися к XML технологиями. Все эти технологии определяются спецификациями W3C.



Разработайте и разверните ваше следующее приложение на облачной платформе IBM Bluemix.

Начните работу с бесплатной пробной версией

- XML - eXtensible Markup Language. Он является основой для других технологий, перечисленных ниже (и многих других) и он широко используется благодаря своей расширяемости.
- XML Schema, иногда называемая XSD, определяет правила для данных, содержащихся в XML-документах.
- XSL (eXtensible Stylesheet Language) и XSLT (XSL transformations) используются для представления XML-документов в разных форматах - например, в HTML.
- XPath - хмм... это хороший вопрос. :-)

---

## О примерах, используемых в этом учебнике

### Обзор

В этом учебнике вы узнаете об XPath на примере написания презентационного уровня приложения сайта аукциона. Вы будете задавать выражения XPath в таблицах XSLT, используемых для представления XML-документов, реализующих лоты аукциона. Файлы для использования в руководстве:

- XPath/AuctionItemList.xsd - документ XML Schema, который определяет формат данных для лотов аукциона.
- XPath/AuctionItemList.xml - XML\_файл, который содержит список лотов аукциона; это данные для примера.
- XPath/AuctionItemSummary-Base.xsl - таблица стилей XSLT, которая определяет, что будет отображать Web-браузер, когда он загрузит AuctionItemList.xml; она содержит правила представления данных.
- XPath/AuctionItemSummary-Section5.xsl -решение для раздела Маршруты местоположения.
- XPath/AuctionItemSummary-Section6.xsl решение для раздела Выражения.
- XPath/AuctionItemSummary-Section7.xsl - решение для раздела Библиотека функций.

### AuctionItemList.xsd

AuctionItemList.xsd содержит бизнес-правила для лотов аукциона и данных списка лотов, описанных на языке XML Schema:

- Список лотов аукциона имеет только один корневой элемент с именем list типа auctionItemList.
- auctionItemList состоит из нуля или более элементов item типа auctionItem.
- auctionItem состоит из пяти элементов (bidIncrement, currentPrice типа price, endOfAuction, description и sellerId) и одной группы атрибутов типа itemAttributes.
- price является положительным двухзначным десятичным значением и должно иметь связанный с ним атрибут currency типа customCurrency.
- customCurrency должен иметь одно из значений: USD, GBP или EUR.

Группа itemAttributes должна содержать один строковый атрибут type, один строковый атрибут id и один булевский атрибут private, который по умолчанию является false. Атрибут type должен иметь одно из значений: Unknown, Traditional, BidOnly, FixedPrice или IndividualOffer. Если вы хотите узнать больше об XML Schema, см. ссылки на статьи и учебники на developerWorks в Ресурсы.

### AuctionItemList.xml

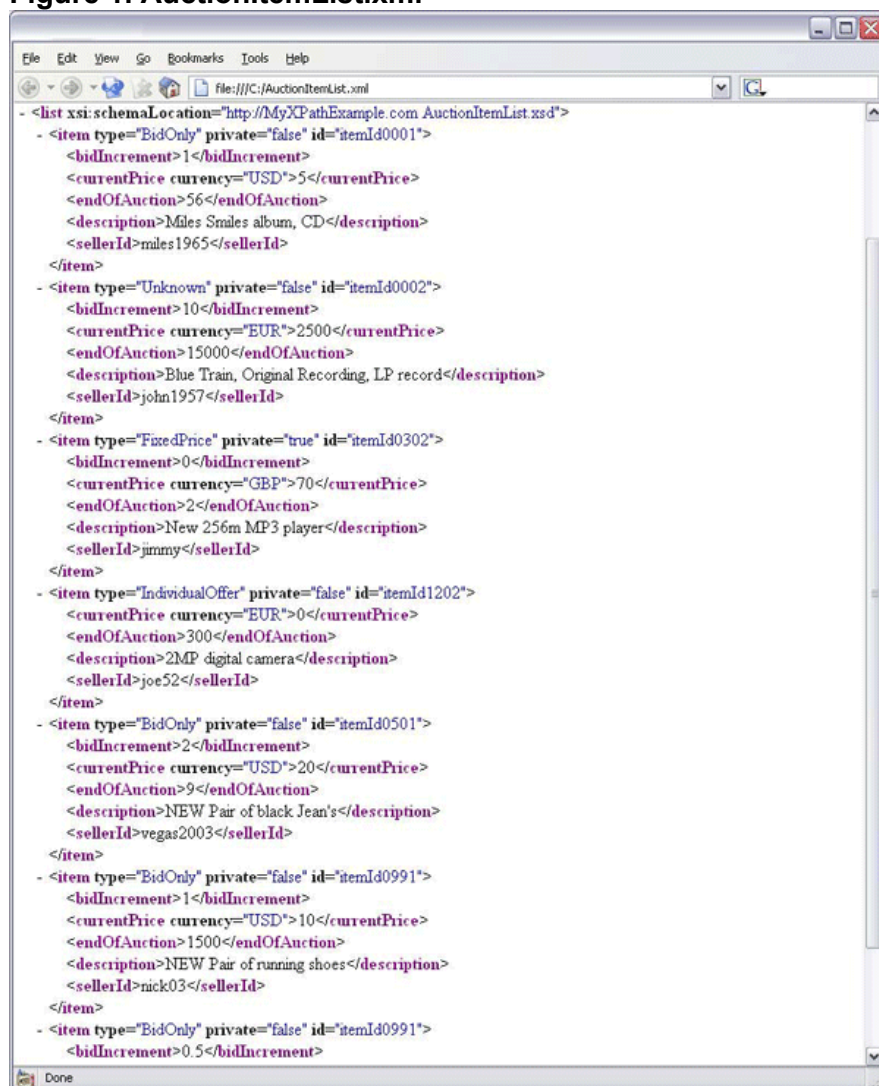
AuctionItemList.xml соответствует XML schema, определенной в defined in AuctionItemList.xsd, и содержит список типа auctionItemList. Этот список содержит семь составляющих. Атрибут xsi:schemaLocation корневого элемента списка показывает, of the list root element indicates that this XML-документ должен соответствовать схеме AuctionItemList.xsd.

Это обеспечивает формат данных, а как же насчет представления? Как вы определяете, какая таблица стилей XSLT используется для отображения этого XML-документа в Web-браузере? Это определяется второй строкой XML-документа:

```
<?xml-stylesheet type="text/xsl" href="AuctionItemSummary-Base.xsl"?>
```

Здесь я устанавливаю, что должна использоваться таблица стилей AuctionItemSummary-Base.xsl. Сами данные должны быть выбраны так, что может быть продемонстрировано применение Xpath, чтобы показать определенные свойства данных. Если с XML-документом AuctionItemList.xml не связывается таблица стилей, то Web-браузер просто показывает содержимое XML, оно выгладит так:

**Figure 1. AuctionItemList.xml**



## AuctionItemSummary-Base.xsl

AuctionItemSummary-Base.xsl является таблицей стилей XSLT, которая определяет правила, применяемые процессором XSLT для отображения XML-документов AuctionItemList. Она использует выражения XPath для нахождения информации в XML-документе и отображения ее в таблице HTML. Я буду более подробно рассматривать использование XPath в XSLT в разделе Обзор Xpath. Здесь я кратко опишу содержимое AuctionItemSummary-Base.xsl. Она определяет шаблоны, которые активируются при обработке XML-документов. Какой шаблон активируется,

зависит от выражения XPath, записанного в атрибуте match этого элемента template. Например, следующие фрагменты, взятые из AuctionItemSummary-Base.xsl, являются выражениями XPath:

- "/"
- "list"
- "item"


Информация, которая отображается, когда шаблон активируется, определяется его атрибутами value-of элемента select. Значения этих атрибутов также являются выражениями XPath. Например:

- "description"
- "currentPrice"
- "@type"
- "@private"

В каждом разделе (Маршруты местоположения, Выражения, Библиотека функций), вы будете модифицировать AuctionItemSummary-Base.xsl, чтобы отображать информацию разными способами.

До сих пор вы должны были просматривать файлы в текстовом или XML-редакторе. Теперь вы можете открыть AuctionItemList.xml в вашем Web-браузере, чтобы увидеть выход, сгенерированный процессором XSLT на основе таблицы стилей. Вы должны увидеть что-то вроде:

**Figure 2. The base auction item table**



The screenshot shows a web browser window titled 'Auction Items Summary' with a menu bar (File, Edit, View, Go, Bookmarks, Tools, Help) and an address bar showing 'file:///C:/AuctionItemList.xml'. The main content area displays a table titled 'Current Auction Items' with the following data:

Time remaining	ID	Description	Seller	Current Price	Bid	Type	Private
56	itemId0001	Miles Smiles album, CD	miles1965	USD5	1	BidOnly	false
15000	itemId0002	Blue Train, Original Recording, LP record	john1957	EUR2500	10	Unknown	false
2	itemId0302	New 256m MP3 player	jimmy	GBP70	0	FixedPrice	true
300	itemId1202	2MP digital camera	joe52	EUR0		IndividualOffer	false
9	itemId0501	NEW Pair of black Jean's	vegas2003	USD20	2	BidOnly	false
1500	itemId0991	NEW Pair of running shoes	nick03	USD10	1	BidOnly	false
3500	itemId0991	NEW Pair of running socks	nick03	USD3	0.5	BidOnly	false

## Обзор XPath

### Что такое XPath?

Язык XML Path (XPath) является набором синтаксических и семантических правил для ссылок на части XML-документов. XPath предназначен для использования другими спецификациями, такими как XSL Transformations (XSLT) и XML Pointer Language (XPath). Чтобы помочь вам понять, что такое XPath, я начну с того, что покажу примеры, относящиеся к AuctionItemList.xml.

Выражения XPath идентифицируют набор узлов в XML-документе. Этот набор узлов содержит ноль или более узлов. Например, выражение XPath /list, примененное к AuctionItemList.xml, идентифицирует один-единственный узел - корневой элемент list.

Выражение XPath /list/item идентифицирует все элементы item.

XPath использует нотацию с прямой наклонной чертой (/), аналогичную shell в UNIX. Поэтому XPath может применяться в Uniformed Resource Identifier (URI), таких как URL. Именно отсюда и пришел XPath: использование нотации маршрута, как в URL.

Допустимые выражения XPath могут включать в себя предикаты. Предикаты содержат логические выражения, которые проверяются для каждого узла в контексте набора узлов. Если истина, узел сохраняется в наборе идентифицируемых узлов; иначе, узел отбрасывается. Предикаты полезны для сокращения результирующего набора. Например, следующее выражение XPath идентифицирует только вторую составляющую:

```
/list/item[currentPrice>1000.0]
```

Выражения XPath могут ссылаться на атрибуты так же, как и на элементы в XML-документе. При ссылке на атрибут используется символ @. Например, следующее выражение XPath идентифицирует элементы currentPrice, в которых атрибуты currency содержат значение EUR:

```
/list/item/currentPrice[@currency="EUR"]
```

XPath также предоставляет функции, которые могут быть очень удобными. Я покажу это вам более подробно в разделе Библиотека функций, но немножко об этом скажем здесь. Выражение Xpath, приведенное ниже, идентифицирует элемент description составляющей, которая имеет type - "IndividualOffer" (и имеет значение 2MP digital camera):

```
/list/item[starts-with(@type,"Ind")]/description
```

Проверьте приведенное выше выражение XPath в вашем XML-редакторе: откройте AuctionItemList.xml и введите выражение в вычислитель Xpath, чтобы увидеть, какие узлы выбираются.

Итак, теперь вы посвящены в XPath! До сих пор вы изучили, что XPath является языком для идентификации частей XML-документов. Вы увидели, как выглядит выражение XPath и как оно ссылается на элементы и атрибуты в XML-документах. Я также показал вам, как XPath обеспечивает функции для манипулирования данными. Однако, это только быстрый обзор; я буду обсуждать все эти вопросы более подробно - а также и другие аспекты XPath - в остальных разделах. Например, я исследую пространства имен XPath и специальные символы (такие, как // и \*) и покажу вам, что не все выражения Xpath имеют форму, показанную в вышеприведенных примерах (называемую сокращенными маршрутами местоположения).

## XSLT, XLink и XPointer

XSLT, Xlink и XPointer являются стандартами W3C. XSLT и Xpath вместе с XSL Formatting Object (XSL-FO) составляют семейство спецификаций eXtensible Stylesheet Language (XSL). (Если вы хотите ознакомиться с этими спецификациями, см. Ресурсы.)

Презентация: XSLT экстенсивно использует XPath для проверки соответствия - то есть, проверки того, соответствует ли узел данному шаблону. XSLT задает контекст, используемый XPath. Вы должны понимать Xpath, если вы хотите работать с XSLT. В разделе О примерах, используемых в этом учебнике вы видели, что таблица стилей AuctionItemSummary-Base.xsl содержит выражения XPath. Эти выражения XPath используются XSLT для нахождения в исходном документе элементов, которые соответствуют критериям, а также для отображения информации в результирующий документ. XSLT также использует функции XPath для выполнения арифметических операций и манипуляций со строками.

Связывание: XLink предоставляет обобщение XML концепции гиперссылок HTML. XLink определяет синтаксис для элементов, вставляемых в XML-документы, чтобы связать ресурсы вместе и описать их отношения. Эти связи могут быть однонаправленными, как гиперссылки HTML, или более сложными. XLink использует XPointer для определения местонахождения ресурсов.

Указание: XPointer является расширением XPath, которое обеспечивает адресацию в XML-документах и их внутренних частях. XPointer обобщает представление узлов XPath концепциями местоположений, точек и рангов XPointer. XPointer также задает контекст, используемый при вычислениях XPath и предоставляет дополнительные функции, которых нет в XPath.

XPath существует для всех упомянутых выше спецификаций. Фактически, спецификация XPath явным образом устанавливает, что XPath разработан для использования в XSLT и XPointer.

Вот сводка технологий XML, о которых я говорил до сих пор:

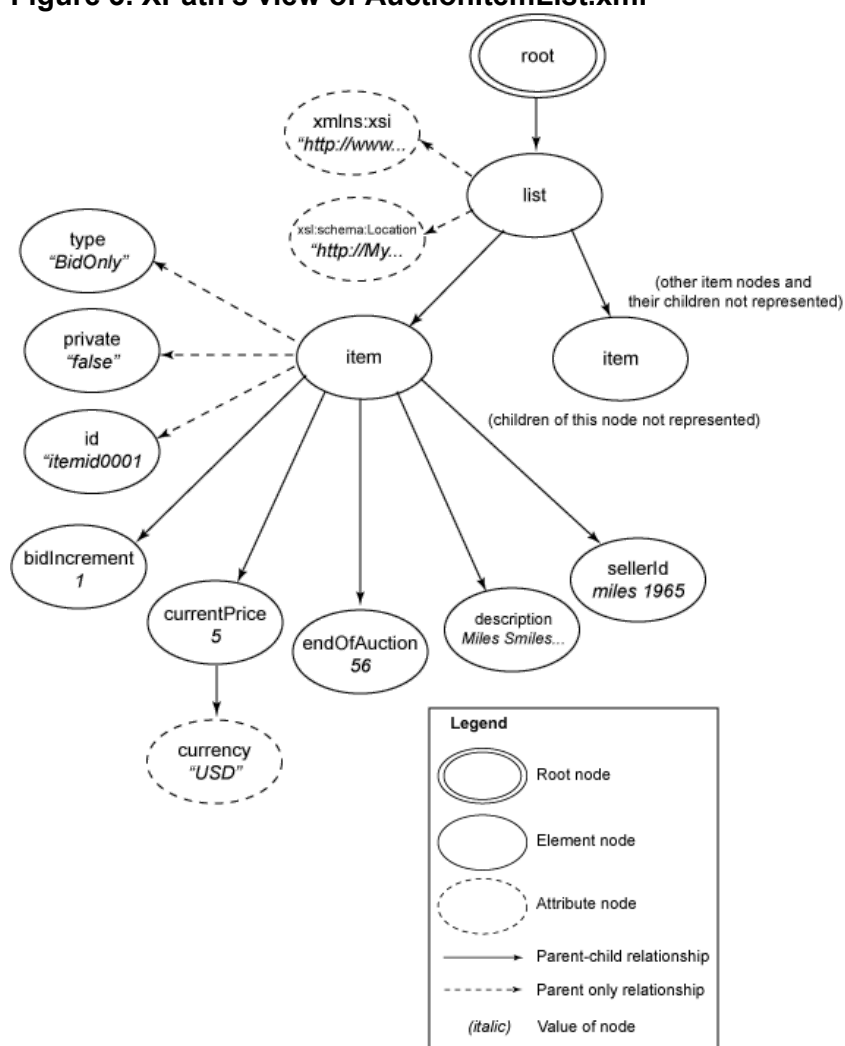
- XML: основа для других технологий (данных)
- XML Schema: правила для формата данных
- XSLT: презентация/проверка соответствия данных
- XLink: связывание
- XPointer и XPath: адресация

## Терминология XPath

### Что такое узел XPath?

XPath выглядит, как XML-документ, как дерево узлов. Узлы могут быть разных типов, таких как узлы элементов или узлы атрибутов. Некоторые типы узлов имеют имена, которые состоят из URI пространства имен XML, возможно не определенного, и локальной части. Например, на рисунке ниже показано представление в XPath AuctionItemList.xml как дерева узлов:

**Figure 3. XPath's view of AuctionItemList.xml**



Специальным типом узла является корневой узел. XML-документ содержит только один такой узел и он является корнем дерева, содержащего весь XML-документ. Заметьте, что корневой узел содержит корневой элемент, а также любые узлы инструкций обработки, объявлений или комментариев, которые появляются до или после корневого элемента. В примере дочерними узлами корневого узла являются:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml-stylesheet type="text/xsl" href="AuctionItemSummary-Base.xsl"?>
```

and

```
<list ...>
  <item ...>
    ...
  </item>
  <item ...>
    ...
  </item>
  ...
</list>
```

Не существует типа узла для XML-декларации (такой, как `<?xml version="1.0" encoding="UTF-8"?>`) или Document Type Definition (DTD). Следовательно, на такие сущности нельзя ссылаться в XPath. Элементные узлы представляют каждый элемент в XML-документе. Узлы атрибутов принадлежат элементным узлам и представляют атрибуты в XML-документе. Однако, атрибуты, которые начинаются с `xmlns:`, представляются в XPath с узлами пространств имен. Другие типы узлов включают в себя текстовые узлы, узлы инструкций обработки и узлы комментариев.

Элементные узлы представляют каждый элемент в XML-документе. Узлы атрибутов принадлежат элементным узлам и представляют атрибуты в XML-документе. Однако, атрибуты, которые начинаются с `xmlns:`, представляются в XPath с узлами пространств имен. Другие типы узлов включают в себя текстовые узлы, узлы инструкций обработки и узлы комментариев.

## Вне контекста?

XPath вычисляет выражения относительно контекста. Этот контекст обычно задается приемами, расширяющими XPath, такими, как XSLT и XPointer. Контекст XPath включает в себя узел контекста, размер контекста, позицию контекста и другие данные контекста. С точки зрения контекста, здесь наиболее интересен узел контекста. Если узел контекста является корневым узлом, `list/item` ссылается на семь элементов `item` в `AuctionItemList.xml`. Если узел контекста является другим узлом - например, первым элементом `item` -- `list/item` не ссылается на что-либо в XML-документе.

В XSLT значения атрибутов `select` являются выражениями XPath. Например, в `AuctionItemSummary-Base.xsl` элементы `xsl:apply-templates` и `xsl:value-of` имеют атрибуты `select`, значениями которых (выражениями XPath) являются, помимо прочего, `list`, `item` или `itemId`. В XSLT узлом контекста является текущий вычисляемый узел. Шаблоны XSLT могут быть активированы несколько раз для XML-документа и обеспечивать разные результаты. В `AuctionItemList.xml` первый и второй шаблоны (`match="/"` и `match="list"`, соответственно) активируются один раз, а третий шаблон (`match="item"`) активируются семь раз (по одному разу на каждый элемент `item`). В первый раз, когда активируется шаблон `"item"`, контекстным узлом является первый элемент `item` в XML-документе (`"Miles Smiles album, CD"`) и, например, значением `<xsl:value-of select="@id"/>` является `itemId0001`. Когда шаблон XSLT активируется второй раз, контекстным узлом является второй элемент `item` (`"Coltrane Blue Train" CD`) и значением `<xsl:value-of select="@id"/>` является `itemId0002`. Заметьте, что я использовал `/list/item/@id`, а не `@id` для атрибута `select`, значение элемента `xsl-value-of` должно быть неопределенным.

## Маршруты местоположения

### Что такое маршрут местоположения?

Маршруты местоположения является наиболее полезным и широко используемым свойством XPath. Маршрут местоположения является специализацией выражения XPath (описанного в Выражения). Маршрут местоположения идентифицирует набор узлов XPath относительно его контекста. XPath определяет два синтаксиса: сокращенный синтаксис и несокращенный синтаксис.

В этом учебнике я рассказываю только о сокращенном синтаксисе, потому что он наиболее широко используется; несокращенный синтаксис также является более сложным. Если вы интересуетесь несокращенным синтаксисом, посмотрите спецификацию XPath 1.0 (см. Ресурсы).

Есть два типа маршрутов местоположения: относительный и абсолютный.

Относительный маршрут местоположения - это последовательность местоположений, разделенных /. Например:

```
list/item[currentPrice<20.0]
```

Этот маршрут местоположения состоит из двух шагов установки местоположения: первый, `list`, выбирает набор узлов относительно контекстного узла; второй, `item[currentPrice<20.0]`, выбирает набор узлов в подмножестве, идентифицированном первым шагом, и т.д., если есть еще узлы. Абсолютный маршрут местоположения состоит из /, за которым, возможно, следует относительный маршрут местоположения, здесь / ссылается на корневой узел. Абсолютный маршрут местоположения является относительным маршрутом местоположения, вычисляемым в контексте корневого узла, например:

```
/list/item[currentPrice<20.0]
```

При абсолютном маршруте местоположения (маршруты местоположения, которые начинаются с /) контекстный узел не принимается во внимание, так как маршрут всегда вычисляется от корневого узла.

### Полезный синтаксис

Сокращенный синтаксис предоставляет набор символов (большинство из них мы показывали в Обзор XPath). Я сейчас перечислю наиболее широко используемые символы и дам примеры относительно корневого узла AuctionItemList.xml, - то есть, узлом контекста будет корневой узел AuctionItemList.xml.

@ используется для ссылки на атрибуты. Например, маршрут местоположения @currency идентифицирует атрибут currency. list/item[@private] идентифицирует элементы item с атрибутом private - то есть, все элементы item в AuctionItemList.xml.

\* используется для ссылки на все элементы, которые являются дочерними для узла контекста. @\* используется для ссылки на все атрибуты узла контекста.

[] могут также использоваться для ссылки на определенный элемент в упорядоченной последовательности. Например, list/item[2] ссылается на второй элемент item. На самом деле это - предикат (описывается далее в Предикаты).

// используется для ссылки на все дочерними элементы узла контекста. Например, //item ссылается на все элементы item, а //list/item ссылается на все элементы item, которые имеют родителем list (то есть, все элементы item в примере).

. используется для ссылки на сам узел контекста. Например, . выбирает узел контекста, а ./item ссылается на все элементы item, которые являются дочерними для узла контекста.



.. используется для ссылки на родительский узел узла контекста. Например, ../item ссылается на первый элемент item в контексте первого элемента bidIncrement.

## Предикаты

Предикаты используются в маршрутах местоположения для фильтрации текущего набора узлов. Предикат содержит булево выражение (или выражение, которое может быть преобразовано в булево). Каждый член текущего набора узлов проверяется по булеву выражению и сохраняется, если выражение истинно; иначе он отбрасывается. Предикат заключается в квадратные скобки, []. Посмотрим на следующий маршрут местоположения:

```
list/item/currentPrice[@currency="EUR"]
```

При вычислении все элементы currentPrice в AuctionItemList.xml находятся в выбранном наборе узлов. Затем вычисляется предикат @currency="EUR" и элементы currentPrice, которые не содержат значения EUR, удаляются.

Предикаты могут использовать также операции отношений >, <, >=, <= и !=. Они могут также использовать булевы операторы, как вы увидите в разделе Выражения.

## Лабораторная работа: Маршруты размещения

Теперь, когда я объяснил, что такое маршруты местоположения, вашей задачей будет модифицировать AuctionItemSummary-Base.xsl, чтобы обеспечить следующий вывод: таблицу, содержащую только составляющие, в которых валюта указана в долларах США:

**Рисунок 4. Таблица, содержащая лоты со стоимостью в долларах США**



The screenshot shows a web browser window titled "Auction Items Summary" with a menu bar (File, Edit, View, Go, Bookmarks, Tools, Help) and a toolbar. The address bar shows the file path file:///C:/AuctionItemList.xml. The main content area displays the heading "Auction Items in US dollars" above a table with the following data:

Time remaining	ID	Description	Seller	Current Price	Bid	Type	Private
56	itemId0001	Miles Smiles album, CD	miles1965	5	1	BidOnly	false
9	itemId0501	NEW Pair of black Jean's	vegas2003	20	2	BidOnly	false
1500	itemId0991	NEW Pair of running shoes	nick03	10	1	BidOnly	false
3500	itemId0991	NEW Pair of running socks	nick03	3	0.5	BidOnly	false

Примечание: Вам нужно заменить значение атрибута select в шаблоне list на правильный маршрут местоположения. Используйте апостроф (') внутри строки, заключенной в кавычки ("). Я расскажу об этом подробнее в разделе Выражения.

Решением задачи является AuctionItemSummary-Section5.xsl. Измените вторую строку AuctionItemList.xml, чтобы она ссылалась на AuctionItemSummary-Section5.xsl, и откройте AuctionItemList.xml в вашем Web-браузере.

Маршруты размещения являются подмножеством более общих выражений XPath. Выражение не только ссылается на набор узлов (маршруты местоположения), но может также возвращать булево значение, число или строку.

## Выражения

### Булевы выражения

Булевы выражения могут иметь одно из двух значений: истина или ложь.

XPath определяет операторы and и or. Для and первым вычисляется левый операнд, если он ложь, то выражение возвращает ложь; иначе вычисляется правый операнд и определяет результат

выражения. Для `or` вычисляется левый операнд; если он истина, то выражение возвращает истину; иначе вычисляется правый операнд и определяет результат выражения. Например, булево выражение `type="BidOnly"` вычисляется в истину в контексте второго элемента `item` в `AuctionItemList.xml`.

XPath определяет следующие операторы

- `=` означает "равно"
- `!=` означает "не равно"
- `<` означает "меньше"
- `<=` означает "меньше или равно"
- `>` означает "больше"
- `>=` означает "больше или равно"

Например, булево выражение `bidIncrement != 10` возвращает истину в контексте первого элемента `item` в `AuctionItemList.xml` и ложь в контексте второго элемента `item`.

Оператор `=`, когда он применяется к узлам, проверяет, имеют ли узлы одно и то же значение, а не то, являются ли они одним и тем же узлом. Это может быть использовано при сравнении значений атрибутов. Например, `item[@type = @private]` выбирает составляющие, в которых атрибуты `type` и `private` имеют одно и то же значение.

Когда выражение XPath содержится в XML-документе, должны быть выдержаны правила правильного форматирования XML 1.0, и любые символы `<` или `<=` должны быть представлены, как `<` и `<=`, соответственно. Например, выражение XPath `bidIncrement < 5` является правильным в Xpointer, но должно быть записано, как `bidIncrement < 5`, если оно должно содержаться в документе XSLT.

Если операнды булевского выражения имеют неодинаковый тип (наборы узлов, числа, строки), происходит преобразование. Подробности см. в спецификации XPath 1.0.

## Числа

Числа в XPath - это 64-битные числа двойной точности с плавающей точкой. Числа XPath включают в себя значение "Not-a-Number" NaN, положительную и отрицательную бесконечность и положительный и отрицательный нуль.

Числовые операторы, предоставляемые XPath такие: `+` (сложение), `-` (вычитание), `*` (умножение), `div` (деление) и `mod` (остаток от деления).

Числовые операторы при необходимости преобразуют операнды в числа, как если бы они использовали функцию `number` (описывается в разделе Библиотека функций).

Примечание: Перед оператором вычитания (`-`) должен быть пропуск, поскольку XML допускает символы `"-"` в строках.

Вот несколько примеров числовых выражений XPath:

- `7 + 3` возвращает 10
- `7 - 3` возвращает 4
- `7 * 3` возвращает 21
- `7 div 3` возвращает 2.3333333333333335

▪  $7 \bmod 3$  возвращает 1

Примечание: Звездочка (\*) может интерпретироваться, как символ-джокер или как символ умножения. XPath определяет лексические правила, позволяющий избежать двусмысленности (подробности см. в спецификации XPath 1.0). Но был введен новый оператор для символа деления, div, полтому что прямая косая черта (/) используется для разделения шагов определения местоположения.

## Строки

Строки в XPath являются последовательностями символов, допустимых в XML 1.0 (Unicode) - например, Miles Smiles album, CD.

Строки в XPath заключаются в кавычки (' или "). Если строка XPath содержится в XML-документе и содержит символы кавычек, вы должны использовать один из следующих двух вариантов:

- Используйте ' или ", соответственно. Например, description = 'New 256m "USB" MP3 player'.
- Используйте смвол апострофа ('), если выражение заключено в двойные кавычки (") и наоборот. Например, 'New 356m "USB" MP3 player'.

XPath предоставляет полезные функции для работы со строками, описанные в разделе Библиотека функций.

## Лабораторная работа: Выражения

Теперь, когда я объяснил вам выражения XPath, вашей задачей будет модификация AuctionItemSummary-Base.xsl таким образом, чтобы обеспечить следующий вывод - таблицу, содержащую все составляющие, для которых аукцион закончился в пределах часа:

**Рисунок 5. Таблица, содержащая лоты, торг для которых закончился в пределах часа**

Time remaining	ID	Description	Seller	Current Price	Bid	Type	Private
56 min	itemId0001	Miles Smiles album, CD	miles1965	USD5	1	BidOnly	false
2 min	itemId0302	New 256m MP3 player	jimmy	GBP70	0	FixedPrice	true
9 min	itemId0501	NEW Pair of black Jean's	vegas2003	USD20	2	BidOnly	false

Примечание: endOfAuction является временем (в минутах), остающимся до конца аукциона. Вам нужно модифицировать тот же атрибут select, что и в разделе Маршруты местоположения.

Решение находится в AuctionItemSummary-Section6.xsl. Измените вторую строку AuctionItemList.xml, чтобы она ссылалась на AuctionItemSummary-Section6.xsl, и откройте AuctionItemList.xml в Web-браузере.

## Библиотека функций

### Библиотека функций

XPath определяет набор функций, называемый базовой библиотекой функций . Каждая функция определяется тремя характеристиками:

- Имя функции
- Возвращаемый тип (обязателен, не может быть пустым)

- Типы аргументов (ноль или более, обязательные или опционные)

Вы можете найти полезным применение функций в предикатах или выражениях. Другие спецификации, такие как XSLT, расширяют этот набор функций. Функции подразделяются на четыре группы, которые я обсуждаю в оставшейся части этого раздела:

- [Функции набора узлов](#)
- [Строковые функции](#)
- [Булевы функции](#)
- [Числовые функции](#)

## Функции набора узлов

Функции набора узлов обеспечивают информацию о наборе узлов (одном узле или более). В число полезных функций набора узлов входят:

- `last()` - Возвращает число, называемое размером контекста, которое является числом узлов в данном контексте. Оно отличается от номера последнего узла.
- `position()` - Возвращает число, называемое позицией контекста, которое является позицией текущего узла в наборе (списке) узлов в данном контексте. Например, вы можете проверить, имеете ли вы дело с последним узлом набора, при помощи выражения `position()=last()`.
- `count(node-set)` - Возвращает число узлов в наборе узлов-аргументе. Например, в контексте документа `AuctionItemList.xml`, `count(//item)` возвращает количество элементов `item`, равное 7.
- `id(object)` - Возвращает набор узлов, результат выборки элементов по их уникальному идентификатору, объявленному как тип ID в DTD. Поскольку я не использую DTD в `AuctionItemList.xml`, для нашего примера результат всегда пустой. `Id("ItemId0001")` возвращает пустой результат.

XPath также определяет еще три функции, относящиеся к именам узлов и пространствам имен:

- `local-name()`
- `namespace-uri()`
- `name()`

Подробности см. в разделе 4.1 спецификации XPath 1.0.

## String функции

При помощи строковых функций вы можете манипулировать строками. В число полезных строковых функций входят:

- `string()` - Преобразует аргумент - объект или контекстный узел - в строку. Допустимыми аргументами являются набор узлов, число, булево значение или любой другой тип, но в последнем случае результат преобразования менее предсказуем. Рекомендуется при преобразовании числа в строку для представления пользователю использовать функцию XSLT `format-number` или элемент XSLT `xsl:number`.
- `concat()` - Принимает в качестве аргументов две или более строки и возвращает их конкатенацию. Например, `concat("Original ", "recording ", "Blue Train LP record")` возвращает "Original recording Blue Train LP record".

- `starts-with()` - Возвращает истину, если строка-первый аргумент начинается со строки-второго аргумента; иначе - ложь. Например, `starts-with("Miles Smiles album, CD", "Miles")` возвращает истину.
- `contains()` - Возвращает истину, если строка-первый аргумент содержит строку-второй аргумент; иначе - ложь. Например, `contains("Miles Smiles album, CD", "album")` возвращает истину.

Другими строковыми функциями XPath являются: `substring()`, `substring-before()`, `substring-after()`, `string-length()`, `normalize-space()` и `translate()`. Подробности см. в разделе 4.2 спецификации XPath 1.0.

## Булевы функции

Булевы функции используются для преобразования объекта или строки в истину или ложь или непосредственного получения значений истина или ложь. Булевы функции, доступные в XPath, такие:

- `boolean()` - Возвращает преобразование в булево значение объекта, переданного ей в качестве аргумента, в соответствии со следующими правилами: число является истиной, если оно отлично от нуля или NaN; набор узлов или строка являются истиной, если они не пустые. Другие типы или объекты преобразуются менее предсказуемым способом.
- `not()` - Возвращает истину, если булевское значение, переданное в качестве аргумента, является ложью; иначе - возвращает ложь.
- `true()` и `false()` - Возвращают истину и ложь соответственно. Эти функции полезны потому, что истина и ложь выглядят в XPath как нормальные строки, а не как булевские значения истины и лжи.
- `lang()` - Возвращает истину, если язык контекстного узла тот же или является подязыком заданного строкового аргумента; иначе - ложь. Язык контекстного узла определяется значением атрибута `xml:lang`. Например, `lang("en")` вернет ложь на любом узле дерева для `AuctionItemList.xml`, потому что атрибут `xml:lang`.

## Числовые функции

Числовые функции XPath возвращают числа. Они такие:

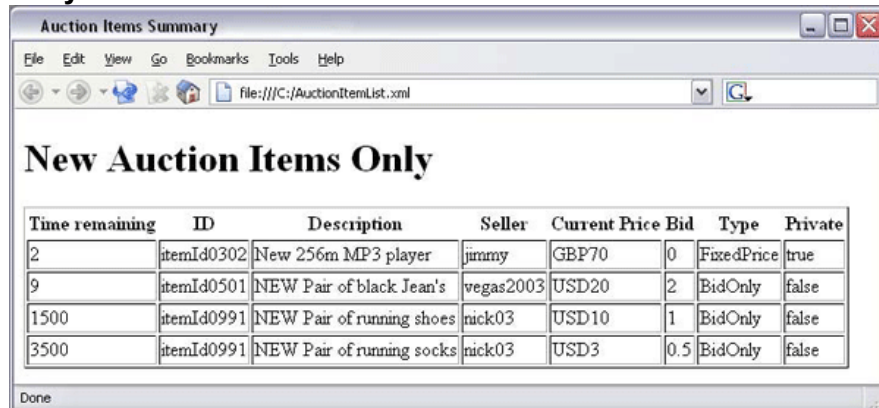
- `number()` - Преобразует необязательный объект-аргумент (или контекстный узел, если аргумент не задан) в число в соответствии со следующими правилами:
  - Булева истина преобразуется в 1, ложь - в 0.
  - Строка преобразуется в соответствующее число.
  - Набор узлов сначала преобразуется в строку, а затем строка преобразуется в число.Другие типы или объекты преобразуются менее предсказуемым образом. Например, `number("250")` возвращает 250, а `number("miles1965")` возвращает NaN.
- `sum()` - Возвращает сумму всех узлов в наборе узлов-аргументе после того, как к ним будет применена функция `number()`.
- `floor()` - Возвращает наибольшее целое, не большее числового аргумента. Например, `floor(2.75)` возвращает 2.
- `ceiling()` - Возвращает наименьшее целое, не меньшее числового аргумента. Например, `ceiling(2.75)` возвращает 3.

- `round()` - Возвращает целое число, которое является округлением числового аргумента. Например, `round(2.75)` возвращает 3.

## Лабораторная работа: Библиотека функций

Теперь, когда я объяснил функции XPath, вашей задачей будет модифицировать `AuctionItemSummary-Base.xml`, чтобы получить следующий вывод - таблицу, содержащую только новые аукционные лоты:

Рисунок 6. Только новые лоты



Time remaining	ID	Description	Seller	Current Price	Bid	Type	Private
2	itemId0302	New 256m MP3 player	jimmy	GBP70	0	FixedPrice	true
9	itemId0501	NEW Pair of black Jean's	vegas2003	USD20	2	BidOnly	false
1500	itemId0991	NEW Pair of running shoes	rick03	USD10	1	BidOnly	false
3500	itemId0991	NEW Pair of running socks	rick03	USD3	0.5	BidOnly	false

Примечание: Такие составляющие будут содержать строку `New` или `NEW` в их описании. Вам нужно модифицировать тот же атрибут `select`, что и в Маршруты местоположения и Выражения. Решение для этого находится в `AuctionItemSummary-Section7.xml`. Измените вторую строку `AuctionItemList.xml`, чтобы она указывала на `AuctionItemSummary-Section7.xml` и откройте `AuctionItemList.xml` в вашем Web-браузере.

## Заключение

### XPath 2.0

XPath 2.0 является надмножеством XPath 1.0 и сейчас является Рабочим Проектом W3C. Две рабочие группы W3C работают над версией XPath 2.0: рабочая группа XML Query и рабочая группа XSL. XPath 2.0 должен быть более мощным и более надежным, поскольку он поддерживает расширенный набор типов данных. Это потому, что значения XPath 2.0 используют типы данных XML Schema, а не просто строки, числа и булевские значения. XPath 2.0 является обратно-совместимым, так что выражения 1.0 ведут себя точно так же в 2.0, кроме исключений, перечисленных в спецификации. Дополнительную информацию о XPath 2.0 см. в Ресурсах.

### Резюме

В этом учебнике вы узнали, что XPath является языком, который используется для адресации частей XML-документов. Вы увидели, как XPath соотносится с другими технологиями XML, такими как XSLT и XPointer. Вы увидели, что такое выражения XPath, включая особый случай выражений, называемых маршрутами местоположения. Вы также получили обзор библиотеки функций XPath и новых свойств грядущей XPath 2.0.

## Ресурсы

### Научиться

- Оригинал руководства [Get started with XPath](#).
- Прочитайте [XML Path Language \(XPath\) version 1.0 Recommendation](#)
- Взгляните на [XPath 2.0 specification \(draft\)](#).



### Bluemix

Узнайте больше информации о платформе IBM Bluemix, создавайте приложения, используя готовые решения!



### developerWorks Premium

Эксклюзивные инструменты для построения вашего приложения. Узнать больше.

**Библиотека документов**

Более трех тысяч статей, обзоров, руководств и других полезных материалов.

- Узнайте о новых свойствах XPath 2.0 в статье на developerWorks "[XML for Data: What's new in XPath 2.0?](#)" (September 2002).
- Освежите свое знание основ XML по учебнику Дага Тидвэла "[Введение в XML](#)" (*developerWorks*, Август 2002).
- Узнайте больше о трансформации XML-документов - прочитайте [XSL Transformations \(XSLT\) 1.0 specification](#).
- Просмотрите [W3C XML Schema Recommendation](#).
- Попробуйте [XML Linking Language \(XLink\)](#) для вставки элементов в XML-документы, чтобы создавать и описывать связи между ресурсами.
- Рассмотрите [XML Pointer Language \(XPointer\)](#), расширение XPath, которое обеспечивает адресацию в XML-документах и их частях.
- Найдите дополнительные ресурсы по XML на [developerWorks XML](#).
- Посмотрите, как вы можете стать [Сертифицированным IBM разработчиком в XML и относящихся к нему технологиях](#).
- Будьте в курсе послений [семинаров и учебных курсов developerWorks](#).

**Получить продукты и технологии**

- Создайте свой следующий проект при помощи [пробного программного обеспечения IBM](#), которое можно загрузить прямо с developerWorks.

**Обсудить**

- [Примите участие в обсуждении материала на форуме](#).
-