

XML. Файлы конфигурации. Реестр

Курс: C# Professional

Средства обучения: Компьютер с установленной Visual Studio 2012

Обзор, цель и назначение урока

Общие принципы использования XML. Работа с [XPath](#). Настройка приложения при помощи файлов конфигурации. Реестр Windows как альтернатива использования файлов конфигурации.

Изучив материал данного занятия, учащийся сможет:

- Самостоятельно выполнять запись и чтение XML файлов
- Выборочно читать или добавлять данные в XML файлы
- Выполнять операции сравнения, форматирования над XML файлами
- Использовать основы работы с XML для создания и редактирования файлов конфигурации приложений
- Работать с реестром Windows, используя средства предоставляемые .NET Framework

Содержание урока

1. Загрузка и чтение из XML файла
2. Применение [XmlTextReader](#)
3. Загрузка содержимого XML по URL
4. Загрузка содержимого XML из строки
5. Поэлементное чтение XML файла
6. Использование [XmlTextWriter](#)
7. Запись комментариев в XML файл
8. Запись пространства имен и префиксов в XML файл
9. Запись в XML файл со специальным форматированием
10. Выборка из XML с помощью [XPath](#). (Запросы [XPath](#))
11. Вычисление min и max с помощью [XPath](#)
12. Создание [XPathDocument](#) из строки
13. Создание файлов конфигурации приложения. Чтение и запись настроек приложения
14. Работа с реестром Windows. Использование реестра для хранения настроек приложения

Резюме

- XML – XML (eXtensible Markup Language) — расширяемый язык разметки (произносится [икс-эм-эль]).
- XML — текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, XHTML).
- XML — это описанная в текстовом формате иерархическая структура, предназначенная для хранения любых данных. Визуально структура может быть представлена как дерево элементов. Элементы XML описываются тегами.
- XML-классы в пространстве имен System.Xml образуют полнофункциональный интегрированный набор классов, позволяющий работать с XML-документами и данными. XML-классы поддерживают синтаксический анализ и запись XML-кода, изменение XML-данных в памяти, проверку данных и преобразование XSLT.
- XML файл состоит из объявления, корневого элемента, комментариев, тегов.
- Первая строка XML-документа называется объявлением XML (англ. XML declaration) — это строка, указывающая версию XML. В версии 1.0 объявление XML может быть

опущено, но, начиная с версии 1.1, оно обязательно. Также здесь может быть указана кодировка символов и наличие внешних зависимостей.

- Корневой элемент - документ имеет только один корневой элемент (англ. root element) (так же иногда называемый элемент документа (англ. document element)). Это означает, что текст или другие данные всего документа должны быть расположены между единственным начальным корневым тегом и соответствующим ему конечным тегом.
- Комментарий - в любом месте дерева может быть размещен элемент-комментарий. XML-комментарии размещаются внутри специального тега, начинающегося с символов `<!--` и заканчивающегося символами `-->`. Два знака дефиса (`--`) внутри комментария присутствовать не могут.
- Теги - остальная часть этого XML-документа, которая состоит из вложенных элементов, некоторые из которых имеют атрибуты и содержимое. Элемент обычно состоит из открывающего и закрывающего тегов, обрамляющих текст и другие элементы. Открывающий тег состоит из имени элемента в угловых скобках, а закрывающий тег состоит из того же имени в угловых скобках, но перед именем ещё добавляется косая черта. Имена элементов, как и имена атрибутов, не могут содержать пробелы, но могут быть на любом языке, поддерживаемом кодировкой XML-документа. Имя может начинаться с буквы, подчёркивания, двоеточия. Остальными символами имени могут быть те же символы, а также цифры, дефис, точка.
- Содержимым элемента (англ. content) называется всё, что расположено между открывающим и закрывающим тегами, включая текст и другие (вложенные) элементы.
- Основные классы для чтения и записи данных XML в пространстве имен `System.Xml`: [XmlReader](#), [XmlWriter](#), [XmlTextReader](#), [XmlTextWriter](#).
- [XmlReader](#) - обеспечивает последовательный доступ только для чтения к потоку XML-данных. Умеет работать только в одном направлении (вперед).
- [XmlWriter](#) - Представляет средство записи, обеспечивающее быстрый прямой способ (без кэширования) создания потоков файлов, содержащих XML-данные.
- Класс [XmlWriter](#) позволяет: проверять допустимость XML-символов и допустимость XML-имен элементов и атрибутов, проверять правильность формата XML-документа, передавать значения, используя типы среды CLR вместо строк, записывать несколько документов в выходной поток, записывать допустимые имена, полные имена и маркеры имен.
- [XmlTextReader](#) - расширяет класс [XmlReader](#). Обеспечивает быстрый однонаправленный потоковый доступ к данным XML.
- [XmlTextWriter](#) - расширяет класс [XmlWriter](#). Обеспечивает быструю однонаправленную генерацию потоков данных XML.
- [XmlNode](#) - предоставляет отдельный узел в XML-документе.
- [XPath](#) (XML Path Language) — язык запросов к элементам XML-документа.
- Пространство имен `System.Xml.XPath` содержит классы, определяющие модель курсора для переходов и изменения элементов сведений XML и предназначено для обеспечения высокой скорости работы.
- [XPathDocument](#) - представляет собой быстрое, доступное только для чтения, хранимое в памяти представление XML-документа с помощью модели данных [XPath](#). Его единственной задачей является создание навигаторов [XPathNavigator](#). Создать класс [XPathDocument](#) можно несколькими способами. Можно передать его конструктору экземпляр [XmlReader](#), имя файла XML-документа или объект, основанный на [Stream](#). Это обеспечивает значительную гибкость, например, можно сначала применить [XmlValidatingReader](#) для проверки правильности XML-документа и затем воспользоваться этим же объектом для создания [XPathDocument](#).
- Класс [XPathNavigator](#) - представляет модель курсора для навигации по данным XML и их редактирования.
- [Registry](#) - этот класс предоставляет набор стандартных корневых разделов, находящихся в реестре компьютеров, работающих под управлением Windows. Реестр является средством хранения сведений о приложениях, пользователях и стандартных системных параметрах. Например, приложения используют реестр для хранения сведений, которые необходимо сохранить после закрытия приложения и к которым необходимо получать доступ при перезагрузке приложения. Например, можно

сохранять цветовые настройки, положение или размер окна. Для разных пользователей эти сведения могут сохраняться в различных местах реестра.

- Класс [Registry](#) предоставляет доступ к следующим разделам:
 - `CurrentUser` – Сохраняет сведения о пользовательских параметрах.
 - `LocalMachine` – Сохраняет сведения о конфигурации для локального компьютера.
 - `ClassesRoot` – Сохраняет сведения о типах (и классах) и их свойствах.
 - `Users` – Сохраняет сведения о стандартной пользовательской конфигурации.
 - `PerformanceData` – Сохраняет сведения о производительности программных компонентов.
 - `CurrentConfig` – Сохраняет сведения об оборудовании, не являющемся специфическим для пользователя.
 - `DynData` – Сохраняет динамические данные (Считается устаревшим).
- Чтение из реестра осуществляется при помощи метода `GetValue`, узел должен быть предварительно открыт при помощи метода `OpenSubKey`.
- Запись данных в реестр осуществляется при помощи метода `SetValue`, узел должен существовать и быть предварительно открытым при помощи метода `OpenSubKey` с указанием признака «открытие для записи».

Закрепление материала

- Что такое XML?
- Где и для чего используется XML?
- В каких случаях применяют [XmlTextWriter](#), [XmlTextReader](#)?
- Что такое проверка на тип узла и для чего она необходима?
- Как добавить комментарии в Xml файл?
- Какие запросы [XPath](#) вы знаете?
- Что такое файлы конфигурации приложения? Зачем они используются?
- Что такое реестр Windows?
- В каком пространстве имен определены классы для работы с реестром?
- Как совершить запись/чтение из реестра Windows?

Дополнительное задание

Создайте .xml файл, который соответствовал бы следующим требованиям:

- имя файла: `TelephoneBook.xml`
- корневой элемент: `"MyContacts"`
- тег `"Contact"`, и в нем должно быть записано имя контакта и атрибут `"PhoneNumber"` со значением номера телефона.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные акронимы и понятия, рассмотренные на уроке.

Задание 2

Создайте приложение, которое выводит на экран всю информацию об указанном .xml файле.

Задание 3

Из файла `TelephoneBook.xml` (файл должен был быть создан в процессе выполнения дополнительного задания) выведите на экран только номера телефонов.

Задание 4

Используя примеры, рассмотренные на уроке, создайте свое приложение для администратора, которое будет сохранять данные конфигурации в специальном файле или в реестре. Создайте пользовательское приложение, внешним видом которого можно управлять при помощи админ-приложения.

Задание 5

Зайдите на сайт MSDN.

Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

MSDN: Пространство имен System.Xml

<http://msdn.microsoft.com/ru-ru/library/system.xml.aspx>

MSDN: XML-документы и данные

[http://msdn.microsoft.com/ru-ru/library/2bcctyt8\(v=VS.90\).aspx](http://msdn.microsoft.com/ru-ru/library/2bcctyt8(v=VS.90).aspx)

MSDN: Класс XmlReader

<http://msdn.microsoft.com/ru-ru/library/system.xml.xmlreader.aspx>

MSDN: Класс XmlWriter

<http://msdn.microsoft.com/ru-ru/library/5y8188ze.aspx>

MSDN: Справочник по XPath

[http://msdn.microsoft.com/ru-ru/library/ms256115\(v=VS.90\).aspx](http://msdn.microsoft.com/ru-ru/library/ms256115(v=VS.90).aspx)

MSDN: Пространство имен System.Xml.XPath

<http://msdn.microsoft.com/ru-ru/library/system.xml.xpath.aspx>

MSDN: Класс Registry

<http://msdn.microsoft.com/ru-ru/library/microsoft.win32.registry.aspx>

MSDN: Файлы конфигурации

<http://msdn.microsoft.com/ru-ru/library/1xtk877y.aspx>