Ch. 7: Entities and Notations in DTDs

Seoul National University, Internet Database Laboratory

July, 2015
Contents

- Creating a General Entity
- Using General Entities
- Creating an External General Entity
- Using External General Entities
- Creating Entities for Unparsed Content
- Embedding Unparsed Content
- Creating and Using Parameter Entities
- Creating an External Parameter Entity
Creating a General Entity

- **Entity** is a shortcut of the text

- **Internal general entities**
  - Are defined in a DTD and simply represent text

  ![Diagram of entity declaration]

- Entity name must follow the rules for valid XML names

- **Five built-in general entities**
  - `&amp;`, `&lt;`, `&gt;`, `&quot;`, `&apost;`
The first and most interesting fact about the gardens is that there is significant controversy about whether the gardens existed at all.

Regardless of the final outcome, it is interesting to note that the imagination of the poets and ancient historians have created one of the Wonders of the World.

`&wow;` &wow “expands” when XML is parsed

Figure 7.2

The Hanging Gardens of Babylon was built in 600 BC and was destroyed by earthquake in 226 BC.

The first and most interesting fact about the gardens is that there is significant controversy about whether the gardens existed at all.

Regardless of the final outcome, it is interesting to note that the imagination of the poets and ancient historians have created one of the Wonders of the World.

Figure 7.3

`&&` + Entity name + `;`
Using General Entities

- **Use** general entity **after** it has been **defined** in the DTD
  - If not, the parser will return an error

- **Special symbols in entity reference tables of Unicode**
  - E.g.) `&246;` generate the ö symbol
  - They are not general entities
    - Do not need to be declared in the DTD

- **General entities are used in XML document only**
  - Cannot be used in XSLT documents

- **You may use an entity within another entity’s definition**
  - As long as there is no circular reference
Creating an External General Entity

- Can be reused in multiple DTD documents
- Save it in a separate external document
- Create the content for the entity in an external file
- Save the file as text-only using an extension of `.ent`
  - Not required, but most common

```xml
<!ENTITY wow "Wonders of the World">
```

Figure 7.4

```xml
<!ENTITY gardens_story SYSTEM "gardens.ent">
```

Figure 7.5

&garden_story: garden.ent에 들어있는 story tag의 contents
Using External General Entities

- To use external general entities

```xml
<?xml version="1.0" standalone="no"?>
...
<wonder>
  <name language="English">
    Hanging Gardens of Babylon
  </name>
  <location>Al Hillah, Iraq</location>
  <height units="feet">0</height>
  <history>
    <year_built era="BC">600</year_built>
    <year_destroyed era="BC">226</year_destroyed>
    <how_destroyed>earthquake</how_destroyed>
    &gardens_story;
  </history>
...
```

'Tell the XML parser that the document will rely on an external file'

Figure 7.6

Figure 7.7
Creating Entities for Unparsed Content

- **Parsed entities**
  - Entities that contain text
  - The XML parser looks at them and analyzes them

- **Unparsed entities**
  - Anything; plain text, an image file, a video file, a PDF file, or anything else
  - Bypassed by the XML parser
  - Embed non-text or non-XML content into an XML document

- Typical unparsed data: A JPEG image – lighthouse.jpg

Figure 7.8
Creating Entities for Unparsed Content

- To create a notation about the unparsed content

```xml
<!ELEMENT ancient_wonders (wonder*)>
<!ELEMENT wonder (name+, photo)>
<!ELEMENT name (#PCDATA)>
<!ATTLIST name language CDATA #REQUIRED>
<!NOTATION jpg SYSTEM "image/jpeg">
```

To identify the unparsed content

The entity is defined in a separate document

How to process the unparsed content

The name for the external entity

The location of the file

Identifying name

The entity is defined in a separate document

Figure 7.9

Figure 7.10
Creating Entities for Unparsed Content

- “image/jpeg” in `<!NOTATION jpg SYSTEM "image/jpeg">`
- can be
  - a MIME type
  - A URI
  - Anything else

The sample list of MIME type:

- .bm image/bmp
- .bmp image/bmp
- .bmp image/x-windows-bmp
- .boo application/book
- .book application/book
- .boz application/x-bzip2
- .bsh application/x-bsh
- .bz application/x-bzip
- .bz2 application/x-bzip2
- .c text/plain
- .c text/x-c
- .c++ text/plain
- .cat application/vnd.ms-pki.seccat
- .cc text/plain
- .cc text/x-c
- .ccad application/clariscad
- .cco application/x-cocoa
Embedding Unparsed Content using Tag Element

- Through a special **ENTITY** attribute type for unparsed content
- Declare the **source** attribute that will contain the reference to the unparsed entity

```
<!ELEMENT ancient_wonders (wonder*)>
<!ELEMENT wonder (name+, photo)>
<!ELEMENT name (#PCDATA)>
<!ATTLIST name language CDATA #REQUIRED>
<!NOTATION jpg SYSTEM "image/jpeg">
<!ENTITY lighthouse_pic SYSTEM "lighthouse.jpg" NDATA jpg>
<!ELEMENT photo EMPTY>
<!ATTLIST photo source ENTITY #REQUIRED>
```

Figure 7.11
Embedding Unparsed Content in XML

- In an XML document

```xml
<?xml version="1.0" standalone="no"?>
...
<ancient_wonders>
  <wonder>
    <name language="English">Lighthouse of Alexandria</name>
    <name language="Greek">ο Φάρος της Αλεξάνδρειας</name>
    <photo source="lighthouse_pic"/>
  </wonder>
</ancient_wonders>
```

- Tips
  - Current browsers **cannot view/play/display** the unparsed entity
  - Instead of unparsed entities, you could set an element’s value to a URL
Creating and Using Parameter Entities

- Parameter entities are entities for the DTD
- To create a parameter entity

**Figure 7.14**

```xml
<!ENTITY % p "(#PCDATA)">
<!ELEMENT ancient_wonders (wonder+)>
<!ELEMENT wonder (name+, location, height, history, main_image, source*)>
<!ELEMENT name %p;>
<!ATTLIST name language CDATA #REQUIRED>
<!ELEMENT location %p;>
<!ELEMENT height %p;>
<!ATTLIST height units CDATA #REQUIRED>
<!ELEMENT history (year_built, year_destroyed?, how_destroyed?, story)>>
<!ELEMENT year_built %p;>
<!ATTLIST year_built era (BC | AD) #REQUIRED>
<!ELEMENT year_destroyed %p;>
<!ATTLIST year_destroyed era (BC | AD) #REQUIRED>
<!ELEMENT how_destroyed %p;>
<!ELEMENT story (#PCDATA | para)*)>
<!ELEMENT para EMPTY>
<!ELEMENT main_image EMPTY>
<!ATTLIST main_image file CDATA #REQUIRED h CDATA #REQUIRED w CDATA #REQUIRED>
<!ELEMENT source EMPTY>
<!ATTLIST source sectionid CDATA #REQUIRED newspaperid CDATA #REQUIRED>
```

**Figure 7.15**

The entity is a parameter entity

Entity name

The shortcut text that will appear when you use the entity in your DTD

%’ + entity name +’;’
Creating an External Parameter Entity

- To create an external parameter entity
  - Create the entity’s content in an external file
  - Save it as text only using an .ent extension

```
<!ELEMENT main_image EMPTY>
<!ATTLIST main_image
  file CDATA #REQUIRED
  w CDATA #REQUIRED
  h CDATA #REQUIRED>
```

Figure 7.16

- To define an external parameter entity

```
<!ENTITY % full_pic SYSTEM "pic.ent">
```

Figure 7.17

This is for a parameter entity

Entity name

The entity is defined externally

Location of the file
Creating an External Parameter Entity

To use external parameter entity

- To use external parameter entity
  - If you are using an internal DTD
    - Then in the XML document containing the DTD
      - Add `standalone="no"`
        - This tells the XML parser that the document will rely on an external file