FusionBot Data Extract XML Syntax

Following is a sample XML file with a single entry, providing an overview of the syntax required, followed by an explanation of each field. The order in which the tags, with the exception of your dynamic children tags for the <attributes> node, must be EXACTLY as shown below for the XML to be correctly parsed.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<extract>
 <item>
   <url>http://www.domain.com/item.html</url>
   <title><![CDATA[As defined in HTML <title> tag for page, CANNOT  be blank]]></title>
   <description><![CDATA[As defined in META Description tag, or first 100 characters of BODY]]></desc>
   <body><![CDATA[Entire text on page (NO HTML, text ONLY)]]></body>
   <attributes>
     <pid>1234</pid>
     <clr>Blue</clr>
     <brd>Acme
     <fns>Chrome|Brushed Nickel|Platinum</fns>
     <cat>Lamps;Table|Lighting|Chandelier;3-Way</cat>
     <stk>200</stk>
     <prc>80.00</prc>
     <sku>9876321</sku>
     <lmd>1225411200</lmd>
     <tni>http://www.domain/com/images/1234.jpg</tni>
   </attributes>
 </item>
</extract>
```

Each item to be made searchable in the database is represented in the XML via the **<item></item>** tag.

The **first** entry **MUST** be the absolute **URL** to the items page on the website (**<url>**).

The **second** entry **MUST** be the title of the resultant item, usually set via the HTML **<title>** tag. The 3 text based XML tags, (title, description, and body) should be enclosed in **CDATA tags** to prevent breaking the XML, and there cannot be any HTML tags as part of the text. If HTML exists in the text within your database, a quick function should be written to parse out / remove, before adding the text to your XML extract file.

The **third** entry **MUST** be the items brief description (**<description>**), up to 100 characters, typically defined via the META Description tag. This entry can be left blank if desired. When blank, the description used by FusionBot will be the first 100 characters as set in the body tag (see below).

The **fourth** entry **MUST** be the entire text / body of the HTML page (**<body>**). That is, any text on the physical page that should be made searchable. Again, NO HTML tags may be included in this field.

.

The **fifth** entry **MUST** be the **<attributes>** tag, and this is where much of the magic happens. For each database type field you want to either display, sort by, filter results by, etc, you **MUST** assign a custom **3-letter attribute** value for and place as a sub-node of the <attributes> tag. You can have **n-many** custom attributes, as needed, based on your customer's requirements. The naming convention of these 3-letter attributes is important, as this is how we are able to reference these values for display, sorting, filtering, etc, in your results template. That is, this is how we pull the values back out of your index and into the results set.

Below are the arbitrary attributes provided in the sample XML above explained:

<pid>1234</pid>

In this example, short for 'product ID'. This could have been 'prd, or 'pro', etc, basically any 3 letter combination you choose that is different from any other attribute as a way to readily recognize this field in your extract XML.

Since <pid> was used, when we are then able to access / reference this value in your results template using **$RES_*PID***.

<clr>Blue</clr>

Short for 'color', assigns a color attribute to the applicable item in the database. Again, this could have been named any 3-letter combination you choose, as long as whatever designator you choose is the same for **EVERY** item inserted into your XML that has a color attribute.

<brd>Acme Products</brd>

Short for 'brand', and how we chose to assign a brand to a particular item.

<fns>Chrome|Brushed Nickel|Platinum</fns>

Short for 'finish' in our example. Now, in this case, we are assigning **multiple** finishes to a single item. When assigning multiple values for a particular attribute of an item, each value is **delimited by a pipe (|)**, and thus, a pipe may not be part of the actual value / name.

```
<cat>Lamps;Table|Lighting|Chandelier;3-Way</cat>
```

Short for 'category'. In this example, not only are multiple categories assigned, as delimited by the pipes, in 2 instances, an item has been assigned a **category and sub-category**. You can assign **up to 2 levels of categorization** using FusionBot, so that results can later be 'filtered' (browse) by category / sub-category, if desired.

Two levels are assigned by **delimiting the category / sub-category using a semicolon (;)**, thus, a semicolon also may **NOT** be part of the actual value / category name. It is important to use cosmetically pleasing values assigned to this attribute, since the values assigned will be the values displayed on the results set, if used, including capitalization, etc.

```
<stk>200</stk>
```

In our case, short for 'stock'. Assigning such a value can allow us to display back to visitor the number of units in stock, or allow us to sort the results by quantity in stock, etc.

```
<prc>80.00</prc>
```

In our case, short for 'price'. This should always **contain 2 decimals** and should **NOT** contain any sort of **currency identifier**, i.e. $, so that the values are all numeric and sortable as such.

```
<sku>9876321</sku>
```

Self explanatory, may or may not be necessary / different than <pid>.

Again, all of the above are examples may or may not apply to your specific implementation, but rather, are provided as a guide to assist in understanding the process, and recognizing what values make most sense for adding to your own XML file.

It should be noted that there are a handful of **reserved 3 letter attributes** that you cannot use for your own purposes, as they are already used internally by FusionBot, including the following tags:

LEN - Length of Page (bytes)
LMD - Last Modified Date
TYP - File Type (pdf,rtf,ppt,doc,xls,txt)
CCH - Cache File Position
TNI - Thumbnail Image
TNW - Thumbnail Width
TNH - Thumbnail Height
TNU - Thumbnail Url/Href
TNT - Thumbnail Target
REL - Related/Region

They are referenced above simply as a means to make sure that you do not accidentally choose any of these 3 letter indicators for your own custom attributes. However, there may be cases where you do indeed intend to use them explicitly, such as the examples provided in the sample XML above, and explained below.

<lmd>1225411200</lmd>

Reserved value for '**last modified date**'. Expressed in **time_t**, used to sort results by date added to database / newest to system, if desired. Many times this can also be accomplished by sorting by a product ID, if your Ids are assigned to your items sequentially as they are added to your database.

<tni>http://www.domain/com/images/1234.jpg</tni>

Reserved value for '**thumbnail image url**'. Points us to the **URL of the image to display for your thumbnail**, i.e. how we populate the <img src=""> tag.