

## Reflection

### Goal

For Project II – XML Database, I chose to mark up my media library. The intended audience for this project is my wife and me. The purpose for this project is to collate and display pertinent information about the books, DVDs, and CDs we own. By having a marked-up library of all our media, we will have a record of everything we own, and the library will allow us to identify gaps in our collection, plan future purchases, and establish what we own.

### Syntax

The basis for my XML database is the syntax, or tags, that surrounds my data. This task provided several challenges that had to be overcome. Since I had three separate media types – DVD, books, and CDs – I had to ensure the tags I created were able to cross all three mediums while still being specific enough to allow different data to be displayed in the same columns. For example, books have authors, CDs have vocalists, and DVDs have directors. I needed a tag that could encompass all three while still being part of the same column. So, to meet this challenge I established the tags `<keypersonnel>` and `<person>` to set the foundation for each key person. With this parent-child foundation, I developed the grandchild `<role>` that allowed me to incorporate a multitude of different roles within the same node. By using this structure, I provided enough flexibility to remain general while still ensuring a specific output for each entry.

The majority of my other tags are self-explanatory, although I experimented with different syntax and levels of granularity before establishing a clear syntax. One example of the level of granularity I needed is seen in my `<extra-info>` node. Similar to `<keypersonnel>`, `<extra-info>` also had item-specific information that I wanted displayed within the same column. To let me display this information together, I created further child and grandchild nodes to contain this information as the specific “extra-info,” so for CD extra info the syntax reads `<cd-extra>`. Within this node are child tags for either a numerical value (e.g. book length or movie length) or a character value (e.g. movie rating). This syntax will be very important for the XSLT transformation to display my data in an HTML table.

### XSLT Transformation

The XSLT transformation process takes XML tags and converts them to HTML for display. Each XML tag is converted to HTML via XSLT and this is achieved through selecting one XML value then choosing (or displaying) another XML value. This process is looped for each entry, thereby building the final table entry by entry. For most pieces of information in my XML database file, this process is straightforward and requires little explanation. There are some entries, however, that deserve closer investigation.

## Reflection

For two nodes - `<summary>` and `<extra-info>` the information to be displayed is item-dependent. For example, the summary for CDs is the track listing, while for DVDs and books the summary is a synopsis of what the movie or book is about. Both pieces of information are discrete and need to be displayed differently: `<tracklist>` needs an ordered list when displayed while `<synopsis>` needs a paragraph style when displayed. XSLT has a function to make this happen: the `<xsl:if>` function. Using the `<xsl:if>` function means that the software displays information following the outcome of an either/or decision. If the `<xsl:if>` test is one XML tag, display that information. If the test results in another XML tag, display another piece of information. Through using this test on two nodes I am able to code for specific results depending on the `<mediatype>` I want displayed. The `<xsl:if>` function provides greater flexibility than the standard `<xsl:for-each>` loops that are the foundation of XSLT transformations.

## Styling

My stylesheet is quite basic, although two styles are worth highlighting. I used a scripting language to shade alternate rows, and then went one step further by coding a “row-hover” item to help the viewer see which item they have selected. This action means that when the cursor is dragged over a row, the background color of the row and the font color changes to a different, determined color – one that contrasts with the other table colors.

## Future Ideas

This XML database project is a work in progress. Future ideas to improve the usability of this table is to fully incorporate the HTML page into a webpage, and also use XSLT to transform the data into a .csv, or spreadsheet format for further manipulation. Other ideas include establishing thumbnails of the different entries to provide a different method of scanning the information – introducing a visual cue to assist the viewer to find a specific entry more efficiently.

This project has been most useful in learning, and then implementing, the principles of XML, XSLT, CSS, and Javascript. The brief introduction this project provides helps me realize the powerful tools available for technical communicators, and how these tools can be used in a professional environment.