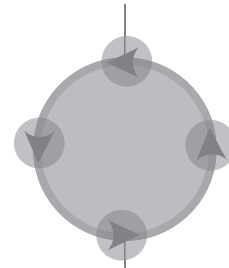


The Power of XML in InDesign

A Designers Guide to XML



The XML Promise

The possibility of a universal data interchange has long been sought after and XML is finally delivering it. It's now being used everywhere and is the basis of all the new middleware Web Services. The next version of Microsoft Office, due in the second half of 2006, will now save its files in the XML format by default. It's also been integrated into Adobe InDesign and Adobe GoLive. Ronald Schmelzer, analyst for ZapThink, an XML research house, says, "XML is so persuasive that it's already everywhere. Eventually, it will even be in dishwashers."

What is XML?

XML stands for "Extensible Markup Language". Although this three letter acronym suggests complexity, XML is simply a way to mark up information in a text file. In other words, it gives you a way to label (or tag) a piece of information (data). These tags then provide a way for the tagged data to be described.

Similar to HTML, XML uses tags, such as "<h1>" and "<p>" to mark up data. However, HTML provides a limited set of predefined tags, whereas XML doesn't have any limits to the type or number of tags you can use. You can even make up your own tags. This is where the word "extensible" comes from. You may be inclined to think that XML is an expanded version of HTML, but it's not. HTML determines what the data should look like. Whereas, XML does not determine how the data looks. Instead, you have full control over how it appears once it is placed into a document.

Learning how to work with XML can be both simple and complicated. However, its many benefits far outweigh any complexities.

Here's what XML can do for you:

- You can easily repurpose information that's already been created. This way, you don't have to recreate it for each new project.
- Work collaboratively with ease. Everyone will be using the same data, making it easier for you to pick up a project that someone else has already completed and reformat it for other projects. For instance, a single XML document can be used over the web or for a printed brochure.
- Information in an XML document is stored in plain-text. Therefore, it can be written with any standard text editor and utilized on any platform.
- XML data can be painlessly shared between previously incompatible applications, making it good for future compatibility. A company that chooses to keep its data in XML can use its data again and again for any project.
- It's an open standard which makes it vendor neutral. Since no one vendor can cause interoperability problems to occur between systems that utilize XML, you can be reassured that your investment in this technology is long-term and safe.
- Overall, you save time and money using XML.

```
<nutrition>
<daily-values>
  <total-fat units="g">65</total-fat>
  <sodium units="mg">2400</sodium>
  <carb units="g">300</carb>
  <fiber units="g">25</fiber>
  <protein units="g">50</protein>
</daily-values>
<food>
  <name>Avocado Dip</name>
  <mfr>Sunnydale</mfr>
  <serving units="g">29</serving>
  <calories total="110" fat="100"/>
  <total-fat>11</total-fat>
  <saturated-fat>3</saturated-fat>
  <cholesterol>5</cholesterol>
  <sodium>210</sodium>
  <carb>2</carb>
  <fiber>0</fiber>
  <protein>1</protein>
  <vitamins>
    <a>0</a>
    <c>0</c>
  </vitamins>
  <minerals>
    <ca>0</ca>
    <fe>0</fe>
  </minerals>
</food>
<food>
  <name>Bagels, New York Style </name>
  <mfr>Thompson</mfr>
  <serving units="g">104</serving>
  <calories total="300" fat="35"/>
  <total-fat>4</total-fat>
  <saturated-fat>1</saturated-fat>
  <cholesterol>0</cholesterol>
  <sodium>510</sodium>
  <carb>54</carb>
  <fiber>3</fiber>
  <protein>11</protein>
  <vitamins>
    <a>0</a>
    <c>0</c>
  </vitamins>
  <minerals>
    <ca>8</ca>
    <fe>20</fe>
  </minerals>
</food>
<nutrition>
```

Figure 1: Sample XML document that contains information for a nutrition directory.

Produced by Gabriel Powell
Metafusion Training, LLC

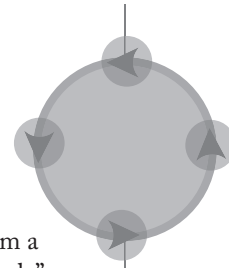
25 NW 23rd Place, Suite 6-122
Portland, OR 97210

gabriel@metafusiontraining.com
www.metafusiontraining.com

Copyright © 2005 by Gabriel Powell

XML Vocabulary

NOTES:



Document Tree

XML files are represented in a structural form called a “tree”. The tree originates from a “root” point and branches out into “leaves”. Wherever a branch occurs, you have a “node”.

A document tree consists of a root node, many branch nodes, and some leaf nodes.

Tag

A tag is a label for a piece of data in an XML file and must be enclosed in angle brackets, like “<headline>”. There are two types of tags. The “start” tag marks the beginning of a piece of data while the “end” tag marks the completion of that data.

For example:

```
<headline>The Value of Form and Function</headline>
```

XML Element

An “element” is the building block of an XML file and must begin with a “start” tag and end with a matching “end” tag. Even the case must be matched. Elements can contain both data and other elements in a specified hierarchy. Here’s an example of one element containing five more elements:

```
<card>
  <name>Steve Martin</name>
  <address>4567 SW Glisan</address>
  <city>Portland</city>
  <state>Oregon</state>
  <zip>97210</zip>
</card>
```

Here’s an other example of a structure taken a step further. Notice here, there are elements that contain elements which contain even more elements:

```
<card>
  <name>
    <first>Steve</first>
    <last>Martin</last>
  </name>
  <address>
    <street> 4567 SW Glisan</street>
    <city>Portland</city>
    <state>Oregon</state>
    <zip>97210</zip>
  </address>
</card>
```

Document (root) Element

Also known as the “root element”, the document element is the outermost element in a document. It contains everything except the document prolog (everything at the beginning of the document) and any comments or instructions outside of it. In the example above, <card>, is the document element.

Container Element

A “container element” is an element that contains data or other elements. In other words, it’s a root to its own “subtree” that follows. In the example above, “<name>” and “<address>” are container elements.

Content

Anything in a document that is not markup is considered “content”. Take away the tags, comments, and processing instructions, and what’s left is the content or actual character data.

Attribute

An “attribute” is a variable or term that provides additional information to an element (metadata). For example, you might want to store the last time the XML data was updated. You can do this by adding information to the “start” tag. For example:

```
<name updated= "06/16/05">  
  <first>Steve</first>  
  <last>Martin</last>  
</name>
```

Comment

Text that is specially marked in the XML document that isn’t interpreted by the “parser” is called a “comment”. XML comments are useful for identifying additional information about a document’s markup. To be ignored by the parser, comments must be surrounded by <!-- and --> delimiters.

Whitespace

Spaces, tabs, and extra paragraph returns are collectively called whitespace. It’s used to make a document more readable to the human eye. If you take out this visual padding, your eyes and brain will become fatigued quickly. In previous versions of InDesign, it was necessary to remove any whitespace in an XML file before you imported it. This is because InDesign had no way of ignoring the extra spaces. With InDesign CS2, this is no longer a problem. On import, you get to choose whether or not the whitespace is included or ignored.

DTDs (Document Type Definition) and Schemas

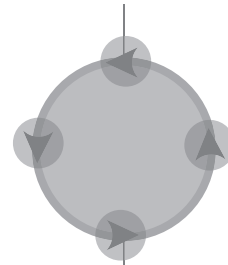
A text file that contains “rules” for an XML file is a “DTD” or “Schema”. It simply declares what elements can be used and in which order they can appear. Since XML can have any number of elements in a document, it can cause chaos when sharing your work with other people. If you use a DTD or Schema to validate the file against an agreed-upon set of tags and data structures, it’s much easier to coordinate a workflow. Although they create restrictions, they make working with other people much easier. DTDs and Schemas are similar, but Schemas have many additional capabilities and are often preferred over DTDs. They are not required by InDesign, but you may need to use them at some point.

Editor

An “editor” is a software program that provides tools for writing and editing simple text files. BBEdit, TextWrangler, and vi (a Unix editor), are among the most popular editors. As a designer, you most likely won’t be writing your own XML data. Most of the time, you’ll get your XML documents from a more automated approach, such as an export from a FileMaker Pro or Access database, or from an InDesign document that you’ve exported as XML.

Parser

A software program that reads XML, validates it, and passes it on for further processing is considered a “parser”. If a document isn’t “well formed”, the parser catches it and reports the problem.



XML in a Publishing Workflow

The following steps outline how XML can be used in a publishing workflow. This workflow allows you to define a template in InDesign and separately develop the content of the document in any standard text editor or database application. Once the template and content are ready, InDesign allows you to merge the two together.

1 Create an InDesign Template

The first step is to create a design that determines how the information should be formatted. Once your design is established, you'll want to use the following tips to create a reliable template that ensures as much automation during the XML import process as possible.

Tips for creating a successful template:

- Place and arrange any static content that will be shared throughout the document. This includes images, text, and other objects (see Figure 2).
- Create placeholder frames that will contain any variable data.
- Create paragraph, character, and object styles for every repeating element. They'll define how the content will look.
- Don't use extra paragraph returns to define space before and after paragraphs. Instead use the Space Before and Space After options in the paragraph style. This allows the imported XML data to flow consistently into your template.
- Connect any text frames that require text to flow between them. This way, you won't have to do any threading after you've already imported the data.
- Pay close attention to the width of your text frames. Make sure they are only as wide as they need to be.
- Apply any necessary Text Wraps. This way, the imported text automatically wraps, saving you from extra work.

2 Load XML Tags

Once you've created a reliable template, you'll need to load XML tags from an XML file or another InDesign file (see Figure 3). It doesn't have to be a file that contains the actual data you'll be using. Just make sure it contains all of the tags that you expect to have in the actual XML data you plan on importing. If you don't have an XML file that contains the tags you need, you can create your own from scratch right in the InDesign file. You'll just have to make sure that the XML tags you're importing precisely match the tags you've defined.

3 Apply XML Tags

Now that your tags are available in your template document (see Figure 4), you can begin applying them to frames and individual paragraphs. This is where you get to decide where the data should appear once it's imported. There are a few methods to choose from when it comes to tagging objects.

Applying tags manually:

- You can use the Tags palette or the Context menu to manually apply a tag to selected text or a frame.

NOTES:

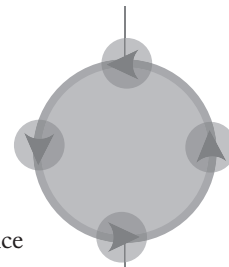


Figure 2: An example of a business card template that contains static and placeholder frames.

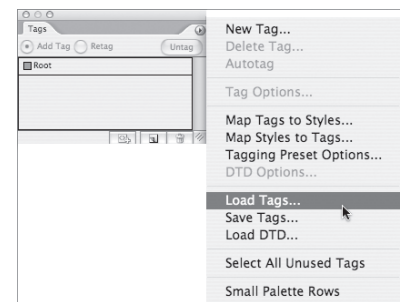


Figure 3: To import XML tags, choose Load Tags from the Tags palette menu.

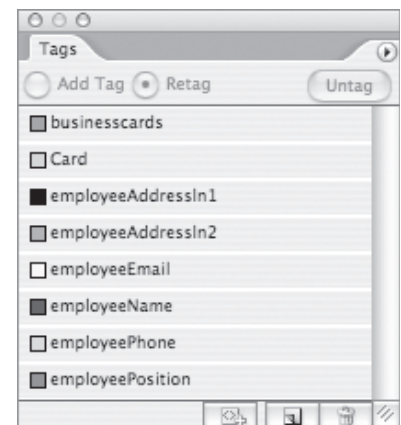


Figure 4: Loaded XML tags are color coded so you can visually see where and what tags have been applied. Double-click a tag to change its color.

- You only need to apply tags to each item as it first appears in the template. For instance, if you're creating a business card template, you only need to apply the tags to the first card. Then, simply duplicate the card and the tags are duplicated too.
- Apply “root” tags to empty placeholder frames first, then apply “subroot” tags to text within each frame. To tag a frame, select it and choose the tag from the Tags palette or Context menu that you want to apply. To tag text within a frame, first choose the Add Tag option in the Tags palette and apply the tag. You can also choose Tag Text from the Context menu.
- Be careful not to select Paragraph Symbols when applying tags to selected text (see Figure 6). It helps to choose Show Hidden Characters from the Type menu in order to see where they're located. If you tag Paragraph Symbols, the formatting won't be remembered when the content is replaced by imported XML data.

Mapping Styles to XML Tags:

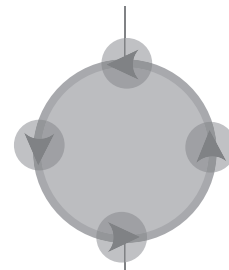
- If you've already applied paragraph and character styles throughout your template, mapping Styles to XML tags can be an efficient way to apply tags.
- With this method, you create a correspondence between styles and tags. This way, InDesign will tag each item that has its corresponding style already applied to it.
- When mapping paragraph styles to tags, InDesign includes the Paragraph Symbol as it applies tags to paragraphs. If you later choose to ignore whitespace in the XML file as it's imported, this will force the text to flow in as a string and force it into one paragraph, preventing the paragraph styles from being applied as you might expect. To overcome this, you can choose from two methods.

1. You can import the XML with whitespace and perform a Find/Change routine to remove all unnecessary whitespace from the document. When you import XML with whitespace, all paragraphs are maintained and any paragraph styles will be appropriately applied.

2. You can apply character styles instead of paragraph styles. When you map character styles to tags, InDesign only applies the tag to the text that has the character style applied to it. With this method, you can choose to ignore the whitespace while importing the XML file and the content will flow in and style as expected.

Whatever method you choose for applying XML tags, it's crucial that you tag placeholder frames and their contents in the exact same order in which the elements are structured in the XML file you are planning to import. This ensures that the tagged frames will be properly updated with the content that comes from the XML file once it has been imported.

Once you've applied tags throughout a template, it's helpful to view tag markers and tagged frames to visually see what is going to happen to the XML data once it's imported (View > Structure > Show Tag Markers or Show Tagged Frames).



NOTES:

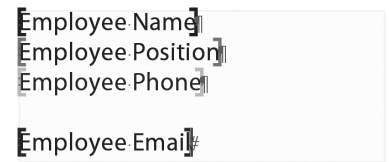


Figure 5: It's easy to see what's been tagged. Color-coded brackets enclose tagged text while tagged frames are outlined and filled with their corresponding tag colors. Notice here that the Paragraph Symbols have not been included in the tag.

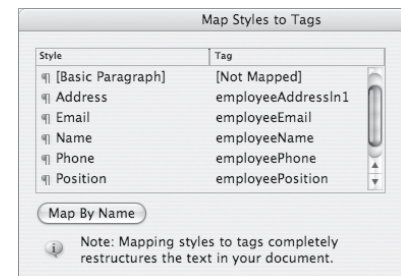


Figure 6: Mapping Styles to Tags makes it easy and efficient to apply tags to text that has paragraph and/or character styles already applied.

4 Import an XML File

Once you've created a template and tagged everything, you're finally ready to import an XML file. Choose Import XML from the File menu or from the Structure pane palette menu. The first time you import XML, the incoming data will automatically appear in any template items that have been tagged with the same element structure. Below is a sample XML file containing information for a business card template. Once it has been imported, the Structure pane allows you to look at it and utilize the data anywhere within your document (see Figure 6). The completed business cards at the bottom show what it looks like once the template cards have been populated with imported XML data.

```
<businesscards>
<Card>
<!-- contents of Suzan Reed -->
  <employeeName>Suzan Reed</employeeName>
  <employeePosition>Director of Marketing</employeePosition>
  <employeeAddressln1>506 SE 72nd Avenue</employeeAddressln1>
  <employeeAddressln2>Portland, Oregon 97215</employeeAddressln2>
  <employeePhone>503-481-5858 direct</employeePhone>
  <employeeEmail>suzan@metafusiontraining.com</employeeEmail>
</Card>
<Card>
<!-- contents of Gabriel Powell -->
  <employeeName>Gabriel Powell</employeeName>
  <employeePosition>Senior Training Director</employeePosition>
  <employeeAddressln1>25 NW 23rd Place, Suite 6-122</employeeAddressln1>
  <employeeAddressln2>Portland, Oregon 97210</employeeAddressln2>
  <employeePhone>503-515-5404 direct</employeePhone>
  <employeeEmail>gabriel@metafusiontraining.com</employeeEmail>
</Card>
<Card>
<!-- contents of Dale Erwing -->
  <employeeName>Dale Erwing</employeeName>
  <employeePosition>Trainer</employeePosition>
  <employeeAddressln1>5131 Buffalo Ave. #20</employeeAddressln1>
  <employeeAddressln2>Sherman Oaks, CA 91423</employeeAddressln2>
  <employeePhone>310-795-8943 cell</employeePhone>
  <employeeEmail>dale@metafusiontraining.com</employeeEmail>
</Card>
<Card>
<!-- contents of Jim Conner -->
  <employeeName>Jim Conner</employeeName>
  <employeePosition>Trainer</employeePosition>
  <employeeAddressln1>25 NW 23rd Place, Suite 6-122</employeeAddressln1>
  <employeeAddressln2>Portland, Oregon 97210</employeeAddressln2>
  <employeePhone>503-515-2376 cell</employeePhone>
  <employeeEmail>jim@metafusiontraining.com</employeeEmail>
</Card>
</businesscards>
```

NOTES:

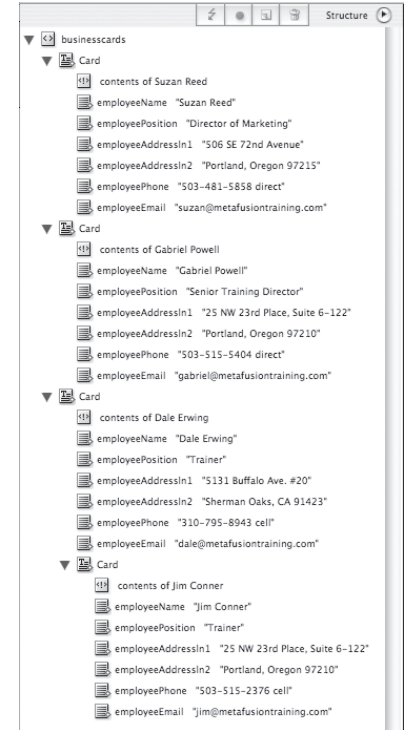


Figure 6: The Structure pane makes it easy to work with elements and their content. Simply drag an element from here to any frame within your document to place its content.

Metafusion TRAINING

Custom Training for Creative Professionals

506 SE 72nd Avenue
Portland, Oregon 97215
503-481-5858 direct

suzan@metafusiontraining.com
www.metafusiontraining.com



Metafusion TRAINING

Custom Training for Creative Professionals

25 NW 23rd Place, Suite 6-122
Portland, Oregon 97210
503-515-5404 direct

gabriel@metafusiontraining.com
www.metafusiontraining.com



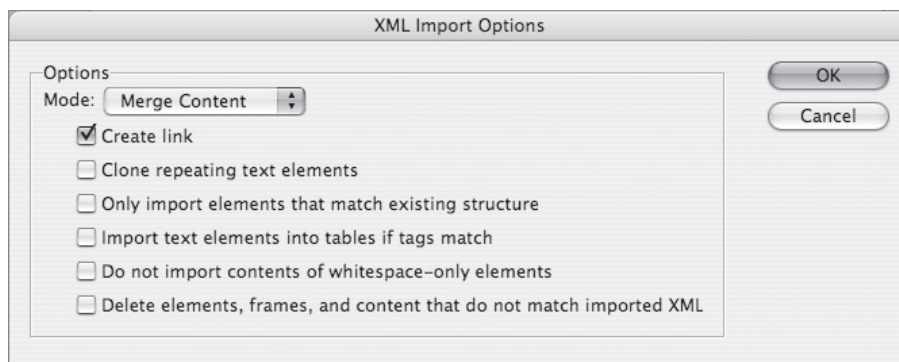
5 XML Import Options

When importing XML, you can choose to append or merge the content into your document. If you choose to append, the new XML content will be added to the document, leaving the existing structure and content unchanged. If you choose to merge, the new XML will replace the existing content and will add new content where it finds no equivalent elements, depending on the options selected.

It's best to merge XML into a document in the following situations:

1. The document contains tagged placeholder frames that you want to fill with the incoming XML file.
2. The document contains XML data that you want to **replace** with the incoming XML file.
3. The document doesn't contain any XML structure, and you want the default root to be replaced with the root of the incoming XML file.

When you choose to import XML files using the Merge Content option, the XML Import Options dialog box has the following options:



Create Link: This option creates a link of the imported XML file upon import. This allows you to store an XML file as a data source. If you update the data source, the InDesign document will reflect those changes. This ensures that you only have to type the data once and repurpose it as many times as needed.

Clone Repeating Text Elements: This option replicates the formatting applied to tagged placeholder text for any repeating content. Use this option when importing content with multiple items that all have the same structure, such as a list of products and prices.

Only Import Elements That Match Existing Structure: This option limits the imported XML content to include only elements that match the tags and structure in the Structure pane.

Import Text Elements Into Tables If Tags Match: This option imports elements into a table if the tags match the tags applied to the placeholder table and its cells. Use this option to place database records into a table, for example, when generating price lists or inventory sheets.

Do Not Import Contents Of Whitespace-only Elements: This option filters out any whitespace (such as a return or tab character) existing in an XML file.

Delete Elements, Frames, and Content That Do Not Match Imported XML: Removes elements from the Structure pane and from the layout that don't have matches in the XML file.