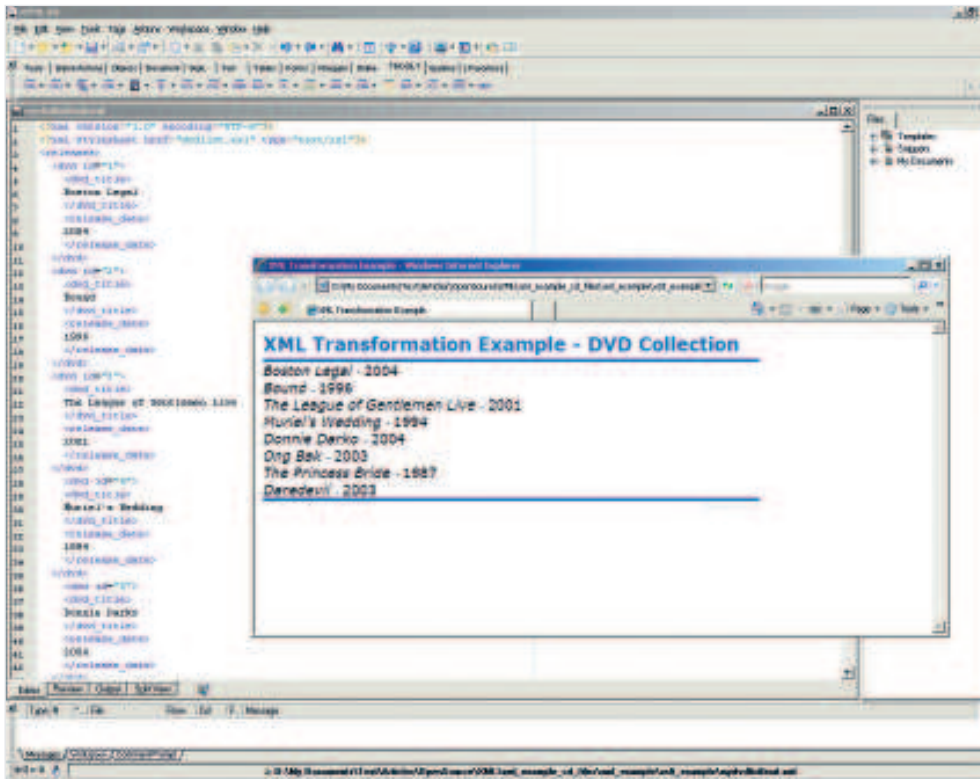


/open_source/publish data dynamically



XSL transformations enable you to style and lay out XML data on-the-fly. This issue, Karl Hodge uses the open source editor HTML-Kit to show you how it's done



What you'll build HTML-Kit is an extendable script-editing tool for HTML, XML, JavaScript, CSS and other languages. The specific tool you'll install makes it easier to create XSLT-powered pages that transform XML data using HTML and CSS rules. You can use this technique to dynamically add data stored in an XML file to pages, without the need for complex server-side scripting

Knowledge needed HTML, CSS, some basic XML

Requires HTML-Kit (from www.htmlkit.com)

Project time 90 minutes

In this tutorial, I'll show you how to incorporate XML marked-up data dynamically into a web page with an XSLT style sheet, using nothing more complex than the open source HTML editor, HTML-Kit.

XSLT stands for Extensible Stylesheet Language Transformations. It's an XML language that lets you add raw data to a page, transforming it into readable HTML. While you can do this with PHP, ASP and other server-side scripting languages, XSLT has one crucial thing going for it: the transformation takes place client-side. There's no need to use a database server; everything is done in XML within your browser.

This is a powerful content-handling technique to add to your toolkit, because once created, you only ever need to update the XML data file to change a page. You can also replace the file entirely, as long as you retain the original filename and dependencies. Many modern database applications enable you to output data in XML format, so you could easily use this method to manage your data in future.

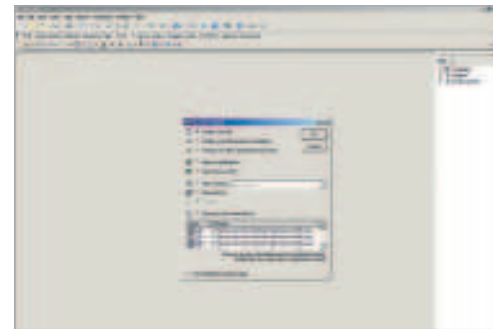


About the author

Name Karl Hodge
Occupation Journalist
Areas of expertise Interface design and development
Clients nerve.com
Who's the fastest: Speedy Gonzales or Road Runner? Road Runner

Expert tip What is XSL?

XSL (Extensible Stylesheet Language) is essentially the XML version of CSS. However, it provides greater controls over data and layout than CSS, with built-in transformation tools and support for styles and HTML. That's the unique selling point of XSL – the ability to transform raw XML data into browser-readable HTML or CSS. XSL is actually a whole family of languages, and in this tutorial, I've brought you into contact with XSLT and XPath. There's also XSL-FO, a comprehensive format for transforming data into other object formats, such as PDF and RTF.



START Launch HTML-Kit Install HTML-Kit and launch it. Dismiss the dialog that asks if you want to check for updates, and start with a new page by selecting **Create a new file** when prompted. Once it has fully launched, click the **Updates** tab. A search box is displayed in the toolbar.



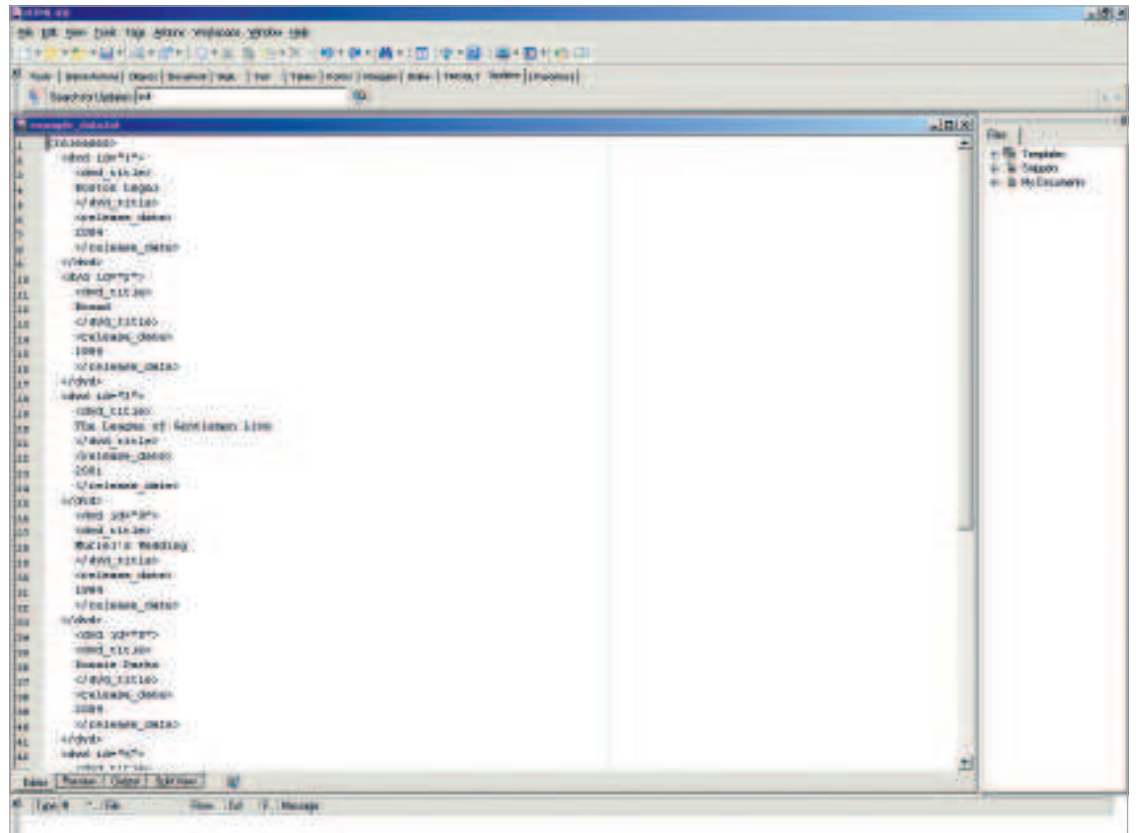
2 Find TMXSLT Type in the acronym XSLT and click the **Search for updates** icon. You're taken to a dialog called **HTML-Kit Setup Plus**. This will take you through the process of adding the plug-in. Dismiss the alert box that asks if you want to go to a search page, then click **Search for updates**.



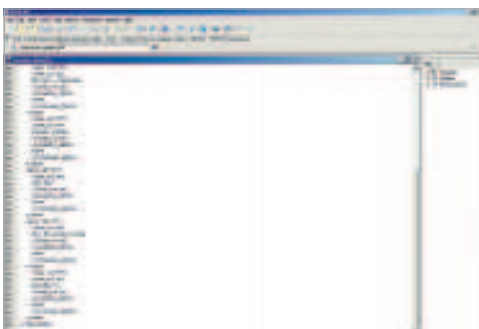
3 Install the plug-in HTML-Kit connects to an online plug-in repository, highlighting tools that have XSLT in their title or description. Among the results returned is tmXSLT. This is a plug-in that makes it easy to add XSL and transformation tags to both XML documents and XSL style sheets. Click Install.



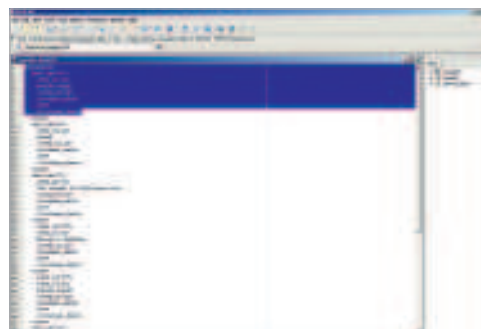
4 Restart HTML-Kit Once the details for the plug-in have been retrieved, click the big **Install tmXSLT** button. You'll be prompted to restart HTML-Kit at the end of the process. When HTML-Kit has relaunched, you'll see a new **TMXSLT** tab on the toolbar. You can start using that almost straight away.



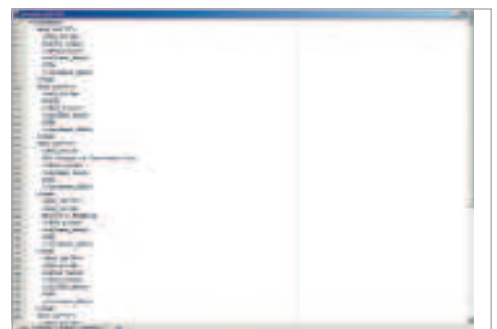
5 Create an XML file You'll be using an XSL style sheet to transform some XML data, so you need some data to start with. Copy the folder 'xml_example' from the CD to your local machine, then open the file 'example_data.txt' in HTML-Kit. This contains a set of data for you to work with, laid out with XML-compliant tags.



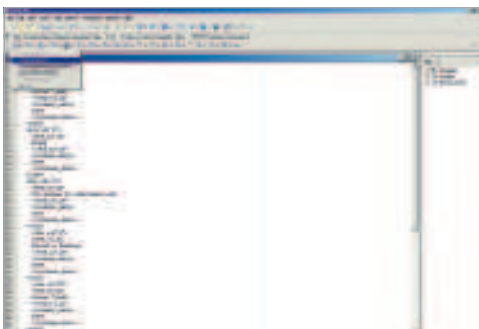
6 Analyse XML structure XML isn't so much a language as a protocol for creating your own markup. In this case, I've created a set of tags for marking up a list of DVDs from my collection. It's a simple list, using intuitive names for the markup. Let's take a quick look at the structure.



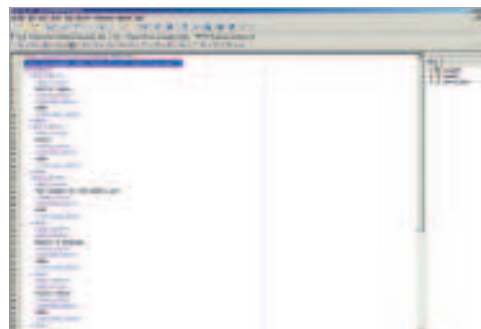
7 Invented language I started with a container: `<releases>`. XML is hierarchical, so the first tag encapsulates the entire code. The second tag, `<dvd>`, creates a single item within it (a DVD listing). Then we have two elements nested between the `<dvd>` tags – the DVD title and release date.



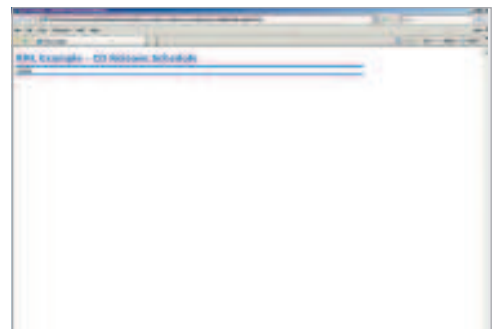
8 Use schema I invented a simple XML language to describe a way to catalogue a DVD release schedule. You don't have to declare the structure (or schema) of this language in your document, but it's good practice to use existing document types (DTD) or schemata if they're available.




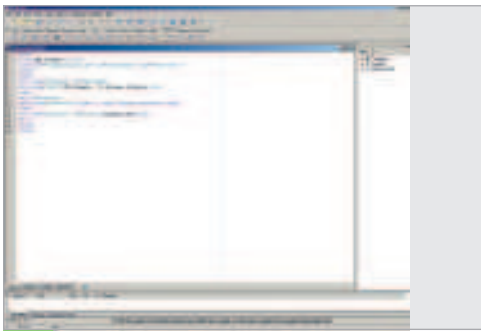
9 Convert to XML Using tmXSLT, you'll turn this list into browser-compliant XML. Place your cursor at the top of the list and click the **TMXSLT** tab. Click the first icon (XSLT XML) in the row. Select `<?XML version?>` from the drop-down list, which inserts an XML declaration at the top of the data.



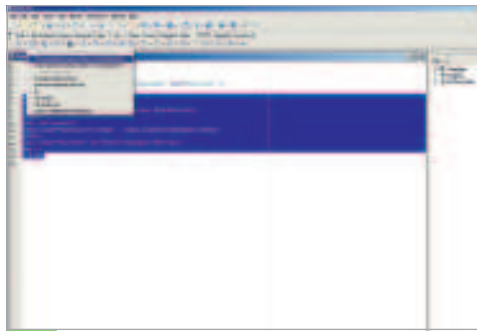
10 Add style links Place the cursor below the last entry and click the **XSLT XML** icon, selecting `<?XMLstylesheet type href?>`. A style sheet link is placed near the top of your code. Type `dvdlist.xsl` between the empty quote marks; this will point to an XSL style sheet, which you'll be making shortly.



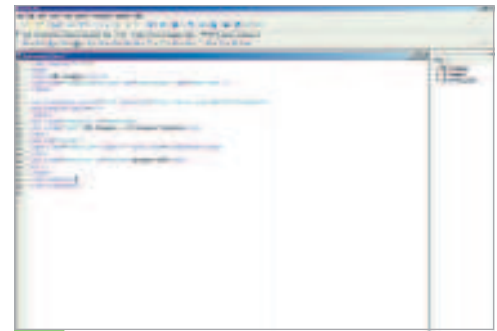
11 Preview the layout Save this as 'mydvdlist.xml' in the example folder. Preview it in a browser, and you'll just get the source code. It's time to make our style sheet. Open 'dvdexample1.html' and preview this – you'll see a basic HTML layout powered by an external CSS style sheet. 



12 Convert to XSL You're going to convert this HTML document into an XSL style sheet, but you need to remove all HTML declarations from it. Work on 'dvdexample2.html', where I've done this for you. XSL documents are XML documents, so add an XML version declaration at the top of the file.



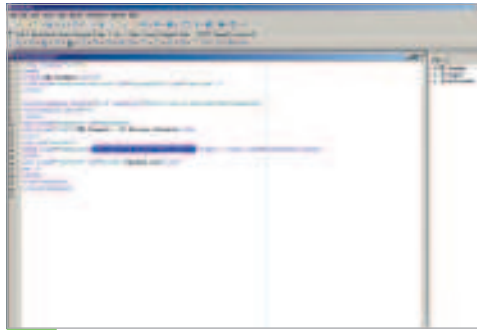
13 Declare the version Directly under that, add an XSL style sheet version declaration so your browser knows it's an XSL stylesheet. Select everything after the XML declaration and click XSLT XSLT. Choose <xsl:stylesheet version="1.0">. Make sure the closing tag is placed after the HTML markup.



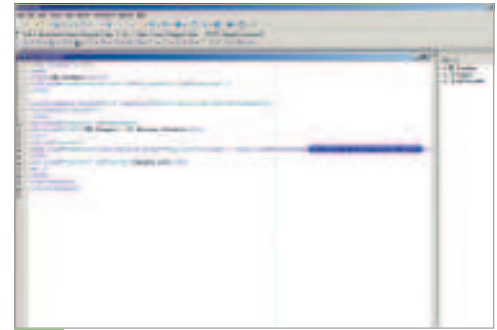
14 Match data There's one more declaration to add, so type <xsl:template match="/">. This tells the XSL document what kind of information you want to pull from the XML file (the forward slash refers to the root of the DOM). Close the tag at the bottom of the document like this: </xsl:template>.

Expert tip XPath techniques

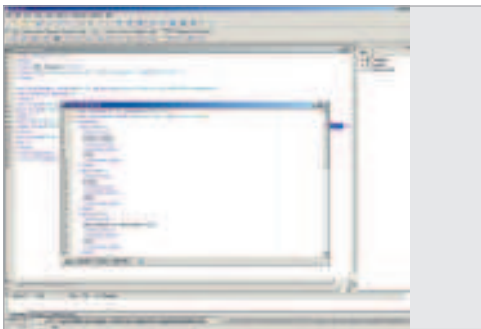
Without knowing it, you've already used XPath techniques in this tutorial. XPath is used to locate specific data within an XML document. The XPath expression we've used in step 14 refers to the root of the Document Object Model, encompassing all available data. The expression we used was <xsl:template match="/">. We could have had <xsl:template match="dvd/dvd_title"> – matching only the dvd_title entries in our data. That's an absolute path, but you can also specify relative paths like this: <xsl:template match="dvd_title">. It's an easy way to narrow down target data.



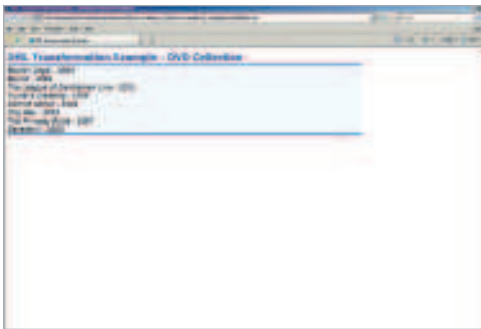
15 Add content Bring in content from the XML file – I've marked where it will go using tags. Look for tags and place your cursor between them. Open XSLT ABC on the toolbar and choose <xsl:value-of select="/>. Type dvd_title between the empty quote marks.



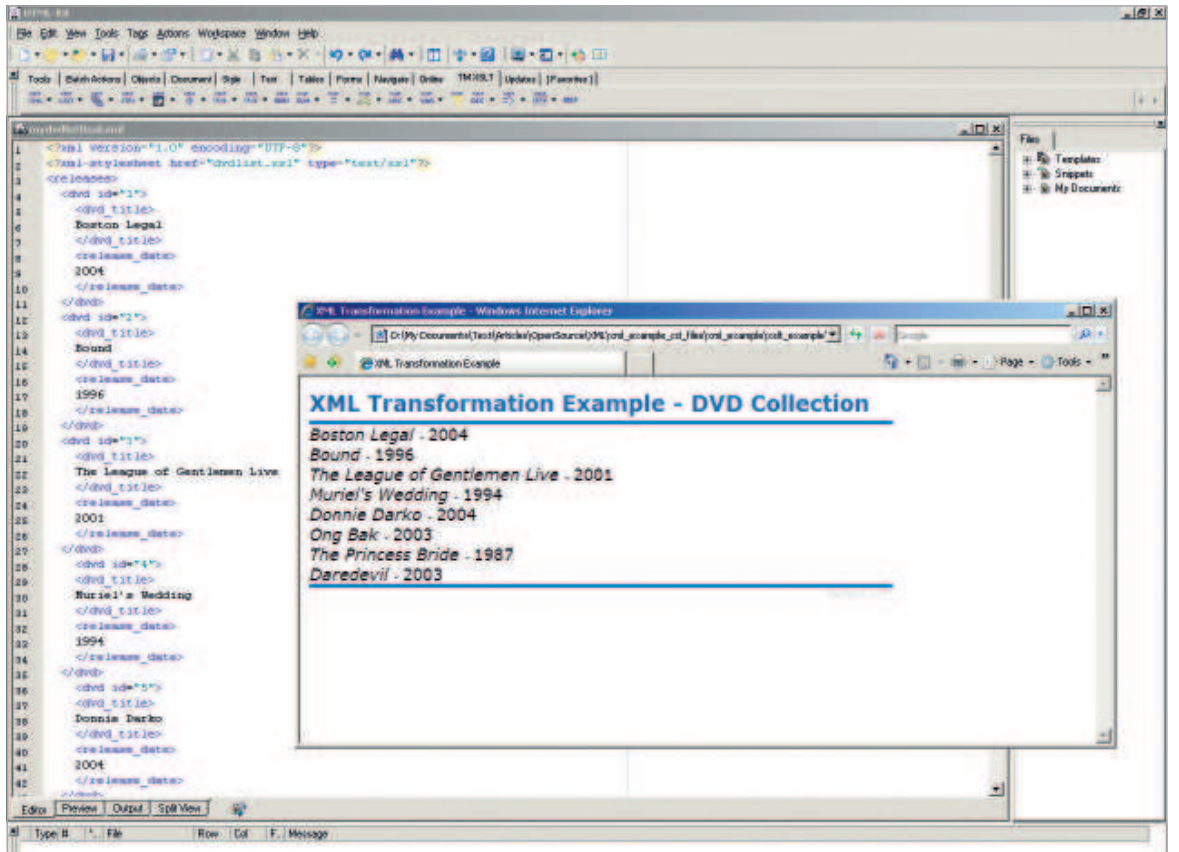
16 Add more content Insert the cursor between the opening and closing tags. Insert the <xsl:value-of select="/> tag, typing release_date between the empty quote marks. Classes referred to in the tags have been defined in an external CSS file named 'examplestyles.css'.



17 XML values The XSL values you've just inserted into your XSL style sheet refer directly to data embedded in the XML list you were working on at the beginning of the tutorial. Flick back to the source code of that document and you'll find elements marked up with dvd_title and release_date.



18 Check the result Save the XSL file as 'dvdlist.xsl' in the same folder as the example files. Try opening 'mydvdlist.xml' again in your browser. If all has gone well, this time the list will be displayed using the formatting embedded in your XSL file. That's the complete list, not just the first element.



FINISH Client-side transformation The XSL style sheet takes the data from the XML document and applies a transformation, creating a third virtual document, client-side, on-the-fly. It's a quick and easy alternative to server-side scripting.