

How to publish Databases on the Web



This month Simon Bisson looks at using database packages to create entire Web sites

The Web stopped being a static place a long time ago. Visit any large Web site, and you can be sure that the content you're seeing wasn't hand-coded in Notepad by some caffeine-crazed hack.

Dynamic Web content means that pages can be tailored to browsers, and personalised if you've left your details on the Web site. It also means that the site's developers don't have to worry about the page layout – as templates and content delivery engines enable text and images stored in databases to be turned into Web pages.

Unfortunately, the dynamic Web hasn't reached the free Web space most ISPs give their users. To be honest, this isn't surprising because large-scale Web application servers are expensive beasts to buy, and even more expensive to run. Disk space and a copy of Apache are much cheaper, so your ISP-hosted



Slashdot began as a simple Web log and now is one of the busiest open source community Web sites around.

site is going to be limited to static HTML pages. This doesn't mean you can't take a leaf out of the larger sites' books, though, and use

a database to create and manage the content on your site.

One of the most popular Web sites around is Slashdot.org – an Open Source news and gossip site. It's what's known as a Web log or 'blog', a site made up of random musings and their responses. Web logs don't have to be interactive like Slashdot, and you could create your own to keep the world informed of developments in any projects you're running – or just to have an on-line journal.

In this *Web Workshop* we're going to create a simple Web log application in one of the most widely available home database packages: Microsoft Access. You don't have to use Access – the basic techniques are valid for most database packages. Another tool that will give good results is FileMaker Pro, the latest version of which includes a selection of HTML publishing templates that can create attractive Web sites quickly.

As Web-ready office tools haven't been around that long (the first HTML add-ons for Word are only five years old) you'll need to have a reasonably up-to-date database package. Access 97 or Access 2000 will have all the features you need and, if you're using an older database, you can use a text processing language like Perl to turn a CSV text export into HTML.

The first thing you need to do is to design the database you're going to use for your Web log.

ADD YOUR OWN FORMATTING

Using HTML templates with Access

You can use HTML templates to add your own formatting to pages generated by Access

Access' generated HTML isn't the best around. While it doesn't contain the extraneous XML added to Word's HTML, you're probably not going to be completely happy with the tables it generates.

If you want to add content around the HTML produced from an exported report or query, then you'll need to use the following tags. These enable Access to fill in your pages – so you can use your own look and feel instead of Access'. There are only seven tags required to create a Access HTML template, including four for handling navigation between Access generated pages.

<code><!--AccessTemplate_Title--></code>	The name of the table, query, form or report placed in the title bar of the Web browser.
<code><!--AccessTemplate_Body--></code>	The output of the table, query, form or report.
<code><!--AccessTemplate_FirstPage--></code>	An HTML anchor tag that links to the first page of a report.
<code><!--AccessTemplate_PreviousPage--></code>	An HTML anchor tag that links to the page previous to the current report page.
<code><!--AccessTemplate_NextPage--></code>	An HTML anchor tag that links to the next page after the current report page.
<code><!--AccessTemplate_LastPage--></code>	An HTML anchor tag that links to the last page of a report.
<code><!--AccessTemplate_PageNumber--></code>	The current page number.

Another tool that enables you to create HTML from databases quickly and easily is FileMaker Pro 5.0.



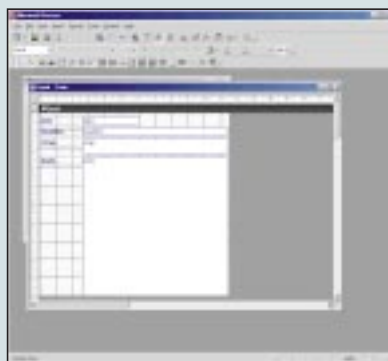
WALKTHROUGH

Building a Web Log application in Microsoft Access

Creating a Web log application in Microsoft Access is a simple process. Just follow our simple guidelines to create your own blog...



1 First design your database. A Web log is a simple single table database – all you need is an automatically generated date field, and then the text fields you want to use. We've just chosen a headline, a sub-heading and then a memo field for the body text.



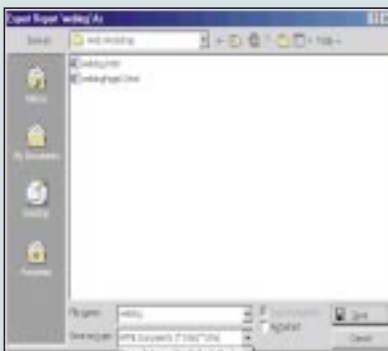
2 Access can be used to create applications that don't require you to input data directly into a table – you can design a form that will enable you to quickly add new content to your log. It's not important to spend time on this, unless you're sharing it with colleagues.



3 You should design your Access reports with the Web in mind. Think about how the page will be seen in a browser – and remember that you won't be able to control the way a page is laid out in the browser.



4 Once you've added some content to a database, take a look at the report Access creates. This will enable you to check what you've written – and you can go back to your input form to make any changes you want.



5 Creating a Web page from an Access report is very easy. Just select File/Export while you're previewing a report. A standard Windows file save dialog will appear. From the 'Save as type' choose HTML, and then save the exported report.



6 Once you've created the HTML, check it in your Web browser before uploading it to your Web server. Access will automatically create links between pages, so you don't need to worry about multipage reports. You can then use your favourite FTP package to upload the new pages to your server.

Generally these are simple databases – and a single table will serve all your needs initially. You really don't need a relational database to create a Web log, however applications like Access have the additional features you need to turn a set of tables into a simple application.

You'll need to decide how many fields you're going to be using. As this will be a personal Web log you can keep formatting information for the report you'll use to create the HTML. It's a good idea to keep the number of fields to a minimum – a headline, a sub-heading and a section of body text are all you're likely to need. A Web log is for recording short pieces of information – not for writing *War and Peace*!

It's important to think about the design of your Web log at an early stage, so that you're ready to start building the reports that will be used to export the HTML. It's not difficult to

design reports in Access because it contains a drag and drop design tool. This isn't WYSIWYG for the Web though, as when you export an HTML document from Access, the table structure it generates isn't a duplicate of the report you've designed.

You can improve formatting by applying an HTML template to the output. You'll need to use Access's own tags to control how the template interacts with the HTML generated by Access. See *Using HTML templates with Access*.

Once you've created your Web log application and begun publishing regular updates, you'll need to start thinking about how you might manage a more complex set of pages. The application we've built here is only designed to create a single Web page, but you can use a combination of reports and queries to generate an index page and then either daily or weekly pages.

Developing a simple Web log like this is a step on the road to the dynamic Web, as these are the techniques used to run large data-driven Web sites. Instead of using canned reports to generate content, an application server will extract content from a database on the fly and then format it. On a smaller scale, you can use CGI scripts and ASP or PHP to do the same on a personal Web server.

The largest sites around use tools like Apple's WebObjects, ATG's Dynamo and BEA's WebLogic. All are able to work with multiple databases and services, creating some of the busiest and most complex e-commerce sites around. **PCP**

NEXT MONTH

Simon Bisson looks at designing pages to display on mobiles and other WAP-enabled devices.