

T H E  
**DATABASE PUBLISHING**  
H A N D B O O K

*.....Pat Bensky*



How to turn your database or  
spreadsheet into a beautifully formatted publication

**The Most Comprehensive Reference to Database Publishing:  
*How to turn your database or spreadsheet into a beautifully formatted catalogue,  
directory, price list ... or any sort of publication***

by

Pat Bensky

First Edition

Published March 2009

© 2009 Tracker Press

[www.trackerpress.com](http://www.trackerpress.com)

[www.database-publishing-book.com](http://www.database-publishing-book.com)



You may give copies of this ebook away but you may NOT:

- Alter it in any way
- Dissassemble it
- Extract text from it
- Charge for it

Please [contact me](#) with any questions, suggestions, or feedback about this book!

**If you did not receive this ebook by downloading it from our web site, it might not be the most current edition!**

**[Get the latest edition for free from our website now!](#)**

## Contents

Chapter One: What is Database Publishing?	1-1	FrameMaker	4-5
What can you do with database publishing?	1-1	InDesign	4-5
Step 1: Getting the data out of the database	1-1	QuarkXPress	4-6
Step 2: Getting Data Into a Nicely Formatted Publication	1-1	RTF	4-6
Terminology	1-1	Chapter Five: ODBC	5-1
Chapter Two: Text Formats	2-1	What's ODBC?	5-1
Delimited Text Files	2-1	Chapter Six: Plugins for Page Layout Programs	6-1
Summary	2-2	CatBase	6-1
Chapter Three: Delimited Text Files	3-1	DataLinker	6-1
CSV (Comma-separated) Text Files	3-1	DataLinker ODBC Add-on	6-1
Tab-Delimited Text Files	3-1	EasyCatalog	6-1
Custom-Delimited Text Files	3-1	Incatalog	6-1
Creating a delimited text file	3-1	LinkUp	6-1
Exporting a Delimited Text File from a Spreadsheet	3-2	Xcatalog	6-1
Publishing a Delimited Text File	3-2	Xtags	6-1
Data Merge Methods and Plugins	3-3	Chapter Seven: Other Useful Plugins	7-1
Chapter Four: Tagged Text Formats	4-1	Other Useful Plugins	7-1
What are Text Tags?	4-1	Chapter Eight: Resources	8-1
InDesign tags	4-1	Chapter Nine: Glossary	9-1
HTML	4-1	Chapter Ten: Index	10-1
MIF	4-1		
QuarkXPress Tags	4-2		
RTF	4-2		
XML	4-2		
Xtags	4-3		
Creating Tagged Text Files	4-3		
Anatomy of a Tagged Text File	4-3		
Making the Text File	4-4		
Publishing Pictures	4-5		
Publishing a Tagged Text File	4-5		

## Chapter One: What is Database Publishing?

The short answer is:

*Database Publishing is the art of taking data from a database and publishing it in a human-digestible form such as a catalogue, directory, price list etc.*

As well as databases, many database publishing projects originate from spreadsheets. For the purposes of this book, you can generally think of “database” as meaning “database or spreadsheet”.

### What can you do with database publishing?

You can publish your data in print, using a page layout program such as QuarkXpress or InDesign, or electronically such as on your web site, as a PDF document, or on a CD.

However you are planning to publish your data, there are two basic steps to take:

1. Get the data out of the database
2. Get it into a “publishing destination” - the end result that you want to distribute to customers, friends, members ... or any other people.

### Step 1: Getting the data out of the database

If you're going to publish your data, you need to get it out of the database and into a format that can be used by the publishing solution you want to use (page layout program, web site, etc.)

The basic, simple way to do this is to save (or export) the data into a text file. This is placed in your publishing solution and then styled: you select the font, paragraph format and so on, and insert any required graphics. This approach might be fine for simple projects such as, for example, a plain telephone directory. But it can become a very tedious and error-prone project if your publication has anything more than very simple formatting.

Suppose, for example, that you're producing a directory of members of a club or association. Every member is entitled to a free, plain entry but each member can pay to have an enhanced entry; it may be set in bold, or include additional information about the member's business, or perhaps a logo or picture.

Another example is a product catalogue. Perhaps there's a picture that goes with each product; maybe some products have two or more pictures. Maybe you want to have some static text such as “Price:” inserted in front of each price or “Tel:” in front of each telephone number. And to complicate things further, that text needs to be set in bold.

And what about tables? Many catalogues and price lists contain tables of information about products, and you may want to have them nicely formatted, possibly with different styles in different columns or with coloured backgrounds to some columns. How tedious it is styling all those tables manually!

You don't want to have to go through the entire publication, manually inserting pictures and static text, choosing stylesheets, perhaps applying different styles to different types of entry. That's very tedious, time-consuming, and error-prone.

What if all of that could be handled automatically when the data is published? That's what database publishing is all about.

### Step 2: Getting Data Into a Nicely Formatted Publication

There are two basic ways to automate the formatting of the publication:

1. Import a delimited text file into the publishing destination, and use a plugin or script to style it.
2. Instead of exporting the data from the database into a plain text file, export it to a **tagged text file**.

We'll look at both methods in detail over the following chapters.

### Terminology

Before we proceed, we need to be sure that everybody understands what certain things mean.

**Database:** An ordered collection of information

**Field:** A specific piece of data within a specific **table** in a database. For example, in the Product table in a database there would be fields such as “productName”, “Price”, “Weight”, “Description” and so on. In a spreadsheet, you can think of each column as a field.

**Publishing destination:** The program you will use to create the publication – page layout program, word processor, html files, etc.

**Record:** In a database, each individual item's set of details (a collection of **fields**) are stored in a record. In a spreadsheet, you can think of each row as a record.

**Relational database:** A **database** in which two or more tables have been set up, with the tables containing related, or linked, data. For example: products and parts; parents and children; manufacturers and suppliers; companies and employees. The most common type of relation is a many-to-one relation, which is like a parent-child relationship: each “parent” record can have any number of “child” records and each child record is related to **one** parent record. A typical example would be Manufacturers and Products. In a many-to-many relationship each parent record can be related to any number of child records, and each child record can be related to any number of parent records – for example, Products and Components or Students and Classes.

**Tagged text file:** A text file in which special “tags” have been inserted to tell the publishing destination what to do with the data. An HTML file is an example of a tagged text file. A detailed explanation can be found in Chapter Four.

**Table:** i. A subset of data elements within a *relational database*. For example, suppose your database contains information on products and parts. The basic product data will be stored in one table, and the details of individual parts will be stored in another table.

ii. A group of data presented in tabular format in a publication.

You can find a complete glossary in [Chapter Nine](#).

## Chapter Two: Text Formats

You're going to need to export the data from your database or spreadsheet in some sort of orderly format for use by your chosen publishing destination; different publishing destinations can use various formats, and various plugins are available for most of them to help with the task of formatting imported data.

First I'll describe the different options, and then each will be discussed in detail in its own chapter.

If you're not sure of any of the terminology used here, please refer to the [Terminology](#) section in Chapter One or the [Glossary](#).

### Delimited Text Files

In a delimited text file, each piece of data (or field) is separated by a specific character - eg a comma, tab, or some other character. The two most common delimited text formats are CSV and tab-delimited.

Delimited text files are a very versatile way to publish your data; there are a number of plugins for page layout programs that can format a delimited text file according to criteria that you establish. These plugins range from the quite basic (but perfectly fine for many projects) InDesign Data Merge, which is included with the InDesign program, to more sophisticated solutions which offer more options.

#### Comma-delimited text (csv)

"csv" stands for "comma-separated values" and, as you might guess, this means that each field is separated by a comma. If the text in a field contains a comma, that field will be surrounded with double quote marks. Some applications that create csv files put double quote marks around every field.

#### Custom-delimited text

Some databases let you choose the delimiters you want to use. This can be useful when your

data might contain the usual delimiters: the most common example is when you have text such as descriptions in your database. This text might contain return characters (new lines) which is the delimiter that is usually used to denote the end of a record. If you don't do anything about this, your data import might be a mess because the program gets confused about where one record ends and another begins. A typical solution is to use the "pipe" character (|) as the record delimiter or, alternatively, to replace the return characters in the text with another character.

#### Tab-delimited text

Each field is separated by a tab character and each record usually ends with a return character. So, each record is one line, or row, in the text file and each one must contain the same number of tabs, even if a field is empty. The first line might contain field headings (field names). If you were to import a tab-delimited text file into a spreadsheet, you would see all the fields dropping nice and tidily into columns. One of the most common ways in which tab-delimited files are created is by "saving as" a tab-delimited file from a spreadsheet.

#### Fixed-length text

Every field contains a specific number of characters. If any field doesn't contain the required number of characters, it is padded out with spaces or zeroes, depending on the type of data (spaces for text; zeroes for numeric data); conversely, if a field contains too many characters, it is truncated. So every row contains exactly the same number of characters and there are no delimiters. This file format is not very useful for database publishing; I've just included it for completeness.

### ODBC

ODBC stands for Open DataBase Connectivity. It's a protocol that enables different programs such as databases and plugins for page layout programs to "communicate" with each other. This topic is discussed in detail in [Chapter Five](#).

### Tagged Text Files

In a tagged text file, each piece of data is "tagged" with special markers that tell the publishing destination how to style it. This topic is discussed in detail, with lots of examples, in Chapter Four.

### XML

XML stands for eXtensible Markup Language. XML is a tagging system in which you can create your own tags and define what they mean. They look similar to HTML tags, but there is a big difference: HTML tags define the *formatting* of data, whereas XML tags define the *content* of data. For example, this HTML tag says that the text "Set me in bold" should appear in a bold font:

```
<strong>Set me in bold</strong>
```

This XML tag tells us that the text "Super Scooper" is a product name:

```
<productname>Super Scooper</productname>
```

It is up to the publishing destination to determine what to do with the data: how to format it etc. For example, when you import an XML file into an InDesign layout you can tell InDesign to automatically apply a certain style sheet to a certain XML tag. There's a lot more to it than that, of course! If you want to find out more about XML, the best place is probably [the W3 web site](#). XML is an excellent way to transfer data between various kinds of applications.

## Summary

So, to summarise, there are basically three ways to approach database publishing:

Method	How it works	Advantages	Disadvantages
Delimited text file	Data is exported from a database or spreadsheet into a plain text file with each field and record delimited by the specified delimiters. As the data is imported into the publishing destination, a plugin can handle the formatting of each field according to rules that have been set up.	<ul style="list-style-type: none"> <li>• Quite easy to set up</li> </ul>	<ul style="list-style-type: none"> <li>• Not all support picture placement</li> <li>• Most cannot handle conditional formatting or conditional data inclusion</li> <li>• formatting options are generally somewhat limited compared to a tagged text file.</li> </ul>
ODBC connection	<b>ODBC Connection:</b> Via a plugin, data is extracted directly from another application, such as a MySQL or Sql Server database, and placed in the document.	<ul style="list-style-type: none"> <li>• Saves the step of exporting the data from the database; it is just extracted directly</li> <li>• Enables bi-directional updating</li> </ul>	<ul style="list-style-type: none"> <li>• Can be a bit tricky to set up for the non-technical</li> </ul>
Tagged text file	A text file is created which contains the data plus “tags”, which specify how the data should be formatted.	<ul style="list-style-type: none"> <li>• Extremely flexible</li> <li>• Makes it possible to do conditional formatting (eg, if it’s a new product, set the product name in red instead of black)</li> <li>• Makes it possible to do conditional data inclusion (eg, if a society member has paid for an upgraded listing in a directory, include additional details such as a logo and a description of their business; otherwise, only include their basic name and contact details)</li> </ul>	<ul style="list-style-type: none"> <li>• Requires a knowledge of programming or scripting for the database application being used</li> </ul>

XML is a sort of hybrid: it’s a tagged text file that can be mapped to stylesheets when it is imported into the page layout program. More on that in the Tagged Text Files chapter.

In the following chapters we’ll look at working with [delimited text files](#), [tagged text files](#), and [ODBC connections](#).

Delimited text files are plain text files containing fields and records of data organised in an orderly format. Each field is separated by a certain delimiter, and each record ends with another delimiter. The most common delimited formats are CSV (comma-separated) and tab-separated.

Delimited files don't contain any formatting instructions or tags. If you are going to publish the data using a page layout program, you'll need to use a plugin to format the data as it is imported into your document. I've included information about some plugins later in this chapter.

### CSV (Comma-separated) Text Files

"CSV" stands for Comma-Separated Values. Each field is separated by a comma. Some applications create csv files in which only fields containing commas are surrounded by double quote marks; some applications create csv files in which every field is surrounded by double quote marks. Usually (but not always) the first line contains a list of the fields.

Here's an example of a csv file:

```
"First Name","Last Name","Email","Phone"  
"Fred","Bloggs","fred@bloggs.com","1234 5678"  
"Kate","Smith","kate@smith.com","1234 5678"  
"Nick","Briggs","","1234 5678"
```

Note that:

- The first line contains the field names (in this example)
- Each line contains the details of one record
- Each line contains the same number of fields, even if a field is empty – as in the record for Nick, who doesn't have an email address
- The fields must be in exactly the same order in each line (record)

### Tab-Delimited Text Files

In a tab-delimited text file, each field is separated by a tab character. As with csv files, the first line usually - but not always - contains the field names. Here's an example tab-delimited file:

First Name	Last Name	Email	Phone
Fred	Bloggs	fred@bloggs.com	1234 5678
Kate	Smith	kate@smith.com	1234 5678
Nick	Briggs		1234 5678

Notice how the fields line up smartly into columns. As with csv files (or any delimited format, for that matter), each record **must** contain the same number of fields, in the same order.

### Custom-Delimited Text Files

Some database applications, such as [CatBase](#), allow you to specify your own field and record delimiters. Why would you want to do that? Your data might contain the characters that are normally used as field or record delimiters, and that would cause confusion when the data was imported into the publishing destination. A typical example is a product description field, which might contain several paragraphs of information: each paragraph is separated by a Return character, which, as it happens, is the default character used as the end-of-record delimiter. If you export the data in a standard format, without doing anything about those embedded Return characters, your publishing destination will think that each time it encounters a Return, that's the end of a record. It will be a mess! You have two options:

1. Change the embedded Return characters to some other character – a new line (\n), or a pipe ("|") perhaps – before you export it from your database or spreadsheet. Then, after it has been published, replace each of those special characters with a Return character.

Many applications will automatically interpret a new line tag as a paragraph break.

2. Choose a different record delimiter. Not all programs allow you to do this, and also you must first make sure that you will be able to specify a custom record delimiter when you import the data into your publishing destination.

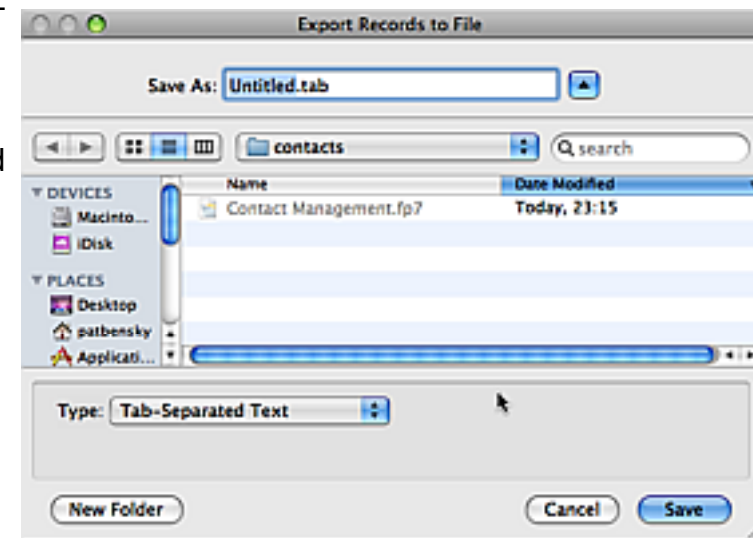
### Creating a delimited text file

Most database and spreadsheet applications offer an easy way to create delimited files.

### Databases

There will usually be an "Export" or "Export Records" or similar option under the **File** menu. Choose that and you'll see a "wizard" to help you export the data. For example, in FileMaker:

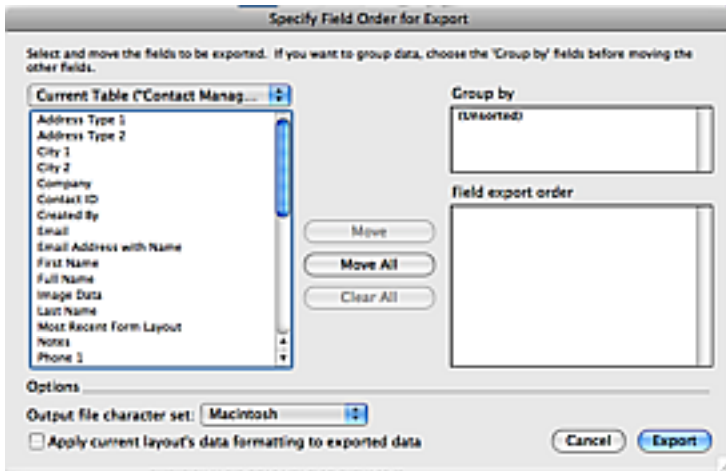
1. First, find the records you want to export.
2. Choose **Export Records ...** from the **File** menu
3. You'll then see a **Save As** dialogue:



4. Change the name in the **Save As:** area to something appropriate



- The default format is Tab-Separated (Tab-delimited). If you want to use a different format, click on the Type pop-up menu and choose (for example) Comma-separated Text. Note that the file extension in the **Save As:** area changes to csv.
- Click the **Save** button
- Next comes the field selection dialogue:



- Select the fields you want to export by clicking on them in the list on the left and then clicking on the **Move** button. Each field will be added to the **Field export order** area as you click on it.
- Our example export would look like this:



- Finally, click the **Export** button to create your file.

Most database programs offer something similar; if you can't find it, consult the program's documentation and look for something like "exporting data" in the index. If you still can't find it, then look under "Reporting".

## Exporting a Delimited Text File from a Spreadsheet

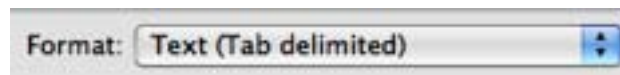
The **Save As ...** option under the **File** menu is your best bet for exporting data from a spreadsheet. Let's look at an example using Microsoft Excel.

Here's part of a spreadsheet:

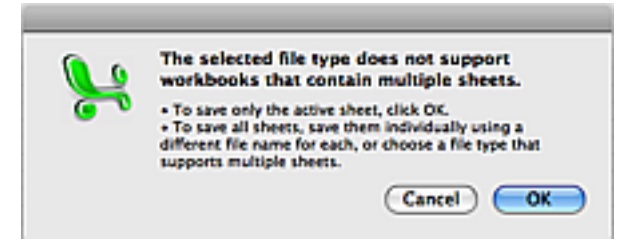
	A	B	C	D
1	First Name	Last Name	Email	Phone
2	Fred	Bloggs	fred@bloggs.	1234 5678
3	Kate	Smith	kate@smith.	1234 5678
4	Nick	Briggs		1234 5678

To export the data to a delimited file:

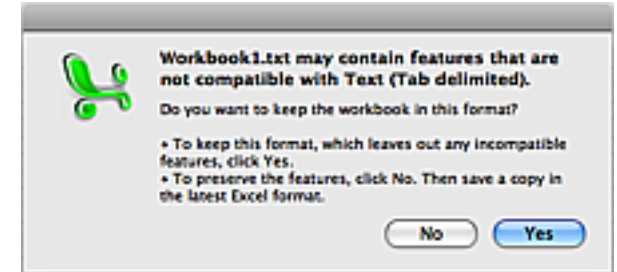
- Create a spreadsheet that contains only the data you want to export
- If your spreadsheet consists of multiple worksheets, you can only export from one at a time - each one will become an individual text file
- Choose **Save As ...** from the **File** menu
- In the Save dialogue, choose the appropriate format from the **Format** popup menu. For example, to create a tab-delimited file:



- Click **Save**.
- You'll then see Annoying Message no. 1:



- Click **OK**.
- Next comes Annoying Message No. 2:



- Click **Yes**.
- Your file will then be created.
- It's possible that your version of Excel, or whichever spreadsheet application you're using, may behave a little differently or have different options; consult the program's documentation for more info.

## Publishing a Delimited Text File

OK: you've created a delimited text file that contains the data you want to publish. But it's just plain, unformatted text. How do you get that set in the fonts and styles you want in your page layout program? You're probably going to need a plugin. There are a number of plugins available, ranging from relatively basic to quite sophisticated, and with prices ranging from free to quite expensive. On the next page you'll find a table containing information about some of those plugins. I'm just going to discuss the options for the two most popular page layout programs - InDesign and QuarkXPress - as these have the best plugins for this purpose, and Microsoft Word, which has a useful Data Merge feature.

# The Database Publishing Handbook

## Data Merge Methods and Plugins

<b>Application</b>	<b>Plugin Name</b>	<b>Description</b>	<b>Price</b>	<b>More Info</b>
InDesign	InDesign Data Merge	Included with InDesign. Provides fairly basic formatting options.	Free	<a href="#">Click here</a>
	InData	Data merge on steroids! Includes a scripting language that enables you to build rules for including or excluding fields, pictures and static text.	\$400	<a href="#">InData Info</a>
Microsoft Word	Data Merge	Enables you to merge data to printer, documents, or email.	Free	
QuarkXpress	Xdata	Data merge on steroids! Includes a scripting language that enables you to build rules for including or excluding fields, pictures and static text.	\$400	<a href="#">Xdata info</a>

## Chapter Four: Tagged Text Formats

### What are Text Tags?

Tags are markers that are placed around text to tell an application what to do with that text - usually, how to format it. For example, a tag might specify a stylesheet to be applied to that text or that certain text should be set in a bold or italic font. The tags usually come in pairs: an opening tag and a closing tag.

There are many advantages to using tagged text files. You have a great deal of control over the formatting of each piece of data. For example, you can have mixed formatting within a paragraph, like this:

**Price:**     \$123.45

Some programs will even allow you to tag individual words or phrases within a block of text with different character formatting. For example, perhaps you need to set all references to publications in italics, something like this:

A recent article in *The Lancet* discussed various ways of medicating patients with this illness.

Some programs, such as [CatBase](#), provide a way for you to “tag” words or phrases in the database. When the data is exported to a tagged text file for publishing in a page layout program or web page, the appropriate tags are inserted and the text will automatically be styled as required when it is published.

If you’ve ever looked at any HTML code, you’ve seen a tagged text file. Here’s a simple example:

```
<strong>Show me the bold</strong>
```

The opening tag (<strong>) tells your web browser to display the text “Show me the bold”

in bold; the closing tag (</strong>) indicates that that’s where the bolding of the text ends.

XML is a special case in which the tags specify the *content* of the data rather than the *formatting* of it: instead of saying “this text needs to be set in the Product Name stylesheet” it says “This text is the name of a product”. It is up to the program to know what to do with each particular tag (or element, to use the correct XML terminology).

There are a number of text tagging formats out there - I’ll discuss the ones that are most widely used. First: a brief overview of the various formats, followed by detailed discussions of each.

[InDesign Tags](#): A special set of tags that are used by InDesign.

[HTML](#): Understood by web browsers.

[MIF](#): Maker Interchange Format. Used by the page layout program, FrameMaker.

[QuarkXPress Tags](#): The tagging system used by page layout program QuarkXPress.

[RTF](#): Rich Text Format. A widely-recognised format that is understood by many word processing and page layout programs.

[XML](#): A *specification* that lets you create your own markup language

[Xtags](#): A proprietary tagging system created by Em Software which extends the QuarkXpress and InDesign tagging systems.

### InDesign tags

This tagging system is used by the page layout program, InDesign, to specify the styling of text (and other things) within the document. You don’t need any special plugins to import an InDesign tagged file into an InDesign document; it’s included with the program.

Here’s an example of a tag that tells InDesign to apply the paragraph style “product name” to the text “Chocolate Soldiers”:

```
<ParaStyle:product name>Chocolate Soldiers
```

Note that some InDesign tags don’t have (or don’t *have* to have) a closing tag: the “product name” style will be applied to all text until a new style tag is encountered. On the other hand, the following snippet shows some text that has been bolded:

```
<cTypeface:Bold>bold<cTypeface:>
```

The ending tag resets the style to whatever it was before the bold was applied.

InDesign tags let you specify a great many attributes, but they do have one or two significant limitations, the main ones being that there are no tags to create text or picture boxes or import images. If you need to do this you’ll need a plugin called [Xtags](#).

You can find a detailed description of InDesign tags and a specification listing all the tags on [our web site](#).

### HTML

You’ve probably seen HTML tags, and quite possibly even used some of them; if you’ve created any web pages you’ve almost certainly encountered these tags.

You can find a complete reference, and some learning materials, on the W3 website at [www.w3.org](#). There are also some excellent free learning resources at the [w3schools website](#).

### MIF

The Maker Interchange Format is a proprietary specification for a tagging system that works with Adobe FrameMaker. You may not have heard of

FrameMaker, as it's not as well-known as other page layout programs such as QuarkXPress and InDesign, but it does offer certain advantages for the more advanced (or more determined) user. Unlike most tagging systems, the MIF format can describe an entire document - not just the content of a document. It also offers tags to create complex equations, graphs and charts, so it's popular with the scientific community.

To put this into perspective, with Quark and InDesign you must first create a document, and then import or place the text into it. You can do that with MIF, but you can also create a MIF file that contains all the information required to create a document. This gives tremendous scope for creating custom documents - although obviously it will require quite a lot of work to set up.

Like most tagging systems, MIF tags consist of statements surrounded by angle brackets. However the structure is somewhat more complex. Here's an example:

```
<PgfTag `Body ' >
<ParaLine
<String `Here is some sample text '>
>
```

As you can see, MIF statements look more like programming code. The first tag

```
<PgfTag `Body ' >
```

tells FrameMaker which stylesheet to use; the

```
<ParaLine
```

tag indicates the start of a paragraph, and the

```
<String
```

tag indicates that what follows is text to go on that line. The whole document is very structured, beginning with statements that describe the stylesheets, pictures, document settings, and other details.

There is no Macintosh version of FrameMaker.

You can find a complete MIF specification in the [Resources area](#) of this book's website, and

more information about FrameMaker on Adobe's website. Developer tools can be found [here](#).

### QuarkXPress Tags

Quark tags are very similar to InDesign tags (actually, it would be more accurate to say that InDesign tags are similar to Quark tags, since Quark has been around way longer than InDesign!). But, like InDesign, they are lacking in a few key areas - particularly in the creation and filling of text and picture boxes. If you want to import pictures or create and place text into text boxes automatically when you import a tagged file, you'll need to get a copy of the [Xtags](#) plugin.

### RTF

Rich Text Format is one of the oldest tagging formats, and it's still popular and is supported by a variety of applications from word processors to page layout programs. It was created by Microsoft and intended for use as an interchange format between Word and other applications.

RTF uses curly braces, angle brackets, and backslashes to delineate its tags (which are referred to as control words). Like MIF, you can create an entire document from scratch with RTF; you can specify the size and characteristics of the document, headers and footers, and pretty much all other characteristics of a word processing document. Plus, of course, the content. You can use RTF to create all kinds of custom documents automatically. But you'll need to do some programming.

You can find a complete RTF specification in the [Tools](#) section on our website.

### XML

XML is an excellent way to transfer data between applications, and it's supported by most of the publishing solutions covered in this book.

An XML file contains tags that specify what the content of the data is, rather than how it should be

formatted. An XML file can be used in combination with [XSLT transformations](#) and [DTD](#) files so that the same data can be presented in different ways.

Here's a simple XML example:

```
<catalog>
  <product>
    <name>Chocolate Solders</name>
    <description>Keep these guys cool</description>
    <price>14.95</price>
  </product>
  <product>
    <name>Pink Mice</name>
    <description>No cheese please</description>
    <price>9.95</price>
  </product>
</catalog>
```

<catalog> is the **root element**: this is the one thing that every XML document **must** have (along with its matching end tag - </catalog> in this example). The root element is also known as the document element. This denotes the beginning and ending of the XML data.

Next we have a <product> element. Between this and the ending </product> element are more elements for the data that belongs to that product.

After the closing </product> element, there's another set of <product> elements for another product. It's vital that the opening and closing elements are correctly matched up in a structured way as shown here; most XML viewers will automatically indent the elements as we've shown here for clarity and so that you can easily see that the "tree" is correctly structured. Correctly structured XML is known as well-formed XML.

The tags identify the various pieces of data - fields, if you like. They don't say anything about how that data should be presented: that's up to the publishing destination to determine. In InDesign, for example, you can map stylesheets to XML elements so that when you import an XML

file, each item will automatically be formatted with the corresponding stylesheet. You can see an example of a catalog being published with InDesign and an XML file on [the CatBase website](#).

An excellent book which covers the subject of publishing with InDesign and XML in detail is *A Designer's Guide to Adobe InDesign and XML* by James J, Maivald and Cathy Palmer.

## Xtags

If you use, or are going to use, QuarkXPress or InDesign for your database publishing project and you need to import pictures and/or create and fill text boxes automatically, you're going to need the Xtags plugin. (Actually you can import images with an XML file, but the formatting options are very limited).

Xtags extends the Quark and InDesign text tagging systems by adding some new tags. With Xtags you can:

- Create anchored and unanchored picture boxes
- Import a picture into a picture box, size it, give it a background, a frame and pretty much any attribute the page layout program offers
- Create anchored and unanchored text boxes and fill them with text; give them background colours, borders, and most other attributes offered by the page layout program
- Apply Master Pages to the pages that certain text falls on
- Apply text translations

Of course, Xtags also supports all the basic text tags for the page layout program.

Let's take a look at the possibilities these features offer ...

## Picture Boxes

You can create picture boxes with your choice of border, corner shape, and so on. Create the picture box at a certain size and Xtags will resize the picture to fit.

## Text Boxes

Create display ads, product boxes, and other boxed content automatically. You can tell Xtags to automatically size the box to fit the content.

You can find more information about Xtags [on this page](#).

## Creating Tagged Text Files

So, now you have an understanding of what a tagged file is, and you've probably got a pretty good idea about which one you want to use for your database publishing project. The next step is working out how to create a file in that format.

## Anatomy of a Tagged Text File

We've seen, briefly, what the tags look like. But what else needs to be in the file? The actual data you want to publish, of course. But there's also other information that the publishing destination will need to know when it is importing your file, such as:

- Which protocol are you using, and what version?
- How are the colours specified?
- How are the paragraph styles set up?
- and so on

Most tagging systems consist of at least two sections: a header section and a body section. The header section is where the details of the file are provided, and the body section contains the actual data.

I'm not going to throw you into confusion by trying to describe the formats of each tagging system; I'll just give one example to give you a

general idea. You can find the specific details for your chosen system in its documentation, which you can find [in the Tools area on our web site](#).

Here's a simple example of a complete InDesign Tags file:

```
<ASCII-MAC>
<Version:5>
<ParaStyle:body text>This is a very short document.
```

The last line contains the actual contents of the document; all the previous lines comprise the header section.

## Header Section

The first line contains the start tag:

```
<ASCII-MAC>
```

This tells InDesign which character set to use and which platform the file was created for.

The next tag tells InDesign which version of the tagging specification to use:

```
<Version:5>
```

## Body Section

Finally we come to the actual meat of the document – the body, or content, section:

```
<ParaStyle:body text>This is a very short document.
```

The tag `<ParaStyle:body text>` tells InDesign to set the text "This is a very short document" in the paragraph style "body text".

This example is obviously a very basic one. In practical use there would be much more information: it might contain colour and stylesheet definitions; perhaps some character style definitions; various other definitions; and some useful content of course. You can find a complete list of InDesign tags in the specification document, which you can

download from [the InDesign Tools page on our web site](#)

## Making the Text File

Now all we need to do is combine the data from the database with the correct tags for the chosen tagging system and produce a text file containing all this information.

Most database systems include either a programming or scripting language, and this is what you can use to create your files.

Let's look at an example in which we want to publish a price list from our database. We're going to publish the price list using InDesign, so we will create an InDesign tag file. It's going to look like this:

Bolts.....	1.22
Nuts .....	2.35
Gadgets.....	3.45
Widgets.....	4.22

This example is a basic one in which we just have the product name and prices, separated by a tab character. This can easily be accomplished by exporting the data to a delimited text file (as described in Chapter Three), without bothering with tags. We can create a file like this quite easily:

1. Find the products we want to publish
2. Sort them
3. Export them to a tab-delimited text file
4. Import that text file into an InDesign (or whatever) template
5. Select all the text and apply the required stylesheet.

Easy!

If your data is in a spreadsheet the steps are very similar:

1. Select the products you want to publish
2. Sort them
3. Choose **Save As ...** from the **File** menu and save the data as a Tab-Delimited text file
4. Import that text file into your template
5. Select all the text and apply the required stylesheet.

But what if you need to produce a more complex price list? Consider this example:

Bolts.....	1.22
Nuts .....	2.35
<b>Gadgets</b> .....	3.45
Widgets.....	4.22

- One of the products is set in red to indicate that it's on special offer
- The prices are set in italics

So we are faced with two issues:

1. How do we tell the page layout program that Gadgets is to be set in a different style to the rest of the products?
2. How do we switch text formatting mid-stream to set the prices in italics?

These things can't be accomplished with a simple text export. We're going to have to do a bit of programming.

The steps we will need to take to create the tagged text file are as follows:

1. Find the products you want to publish
2. Sort them
3. Open a new text file
4. Save the header details into the text file
5. Loop through the product records, adding style information and data to the text file line-by-line. The programming logic goes something like this:

## Do you have these problems when publishing your catalogue, directory, or price list?

- Inconsistent formatting
- Last-minute updates (eg new prices)
- It takes way too long!
- Wrong or missing pictures
- Different types of entries need to be formatted in different ways

CatBase rapidly turns your database or spreadsheet data into a perfectly formatted catalogue, directory, or price list

- All text is automatically formatted
- Pictures can be automatically placed
- Conditional styling and content is automated
- Works with the popular page layout programs
- PC and Mac versions available



See a video that shows how easy it is to create your publication with CatBase!

Go to [www.catbase.com](http://www.catbase.com) to find out how to get a free trial version of CatBase  
Or call:

01462 454522  
(+44 1462 454522)

CatBase Software Ltd.

1. If it's a new product, use the stylesheet "newproduct"; otherwise, use the stylesheet "prices"
2. Add a Tab character
3. Add the "italic" character style
4. Add the price
5. Add a Return character
6. Go to the next record and repeat steps 1 through 5
7. After all the products have been added to the text file, close the text file

We will end up with a text file that looks something like this (using InDesign tags):

```
<ASCII-MAC>
<Version:5>
<ParaStyle:prices>Bolts      <CharStyle:italic>1.22<
CharStyle:>
<ParaStyle:prices>Nuts      <CharStyle:italic>2.35<
CharStyle:>
<ParaStyle:newproduct>Gadgets  <CharStyle:italic
>3.45<CharStyle:>
<ParaStyle:prices>Widgets  <CharStyle:italic>4.22<
CharStyle:>
```

If you are able to write programming code or scripts to accomplish this, you're on your way to becoming a database publisher. If not, don't worry: there are a number of software tools available to help you, in the form of applications and plugins. An example application is [CatBase](#), a stand-alone program that can create tagged files in all the popular formats. You don't need to do any programming: just make selections in dialogue boxes.

## Publishing Pictures

Most text tagging formats support pictures; two do not. If you're working with InDesign or QuarkXpress you will need to use [Xtags](#) instead of the native InDesign or Quark tags. Xtags is a plugin that extends the native tagging systems by adding

the ability to create and fill text and picture boxes, and more.

## Publishing a Tagged Text File

So, you've created your tagged text file. What do you do next – how do you publish it? That varies a little depending on the publishing destination you're using, but generally you simply need to import your tagged file into a template. I'll briefly explain the options for the various publishing destinations below.

## FrameMaker

### MIF

You can either import a MIF file into an existing FrameMaker document, or you can open a MIF file as a brand new document.

#### Importing a MIF File into an existing document

1. Open your FrameMaker document
2. Choose **Import -> file** from the **File** menu.
3. Select the MIF file you want to import
4. Select the **Copy into document** radio button
5. Click on the **Import** button

#### Opening a MIF file as a new document

1. That's easy ... just choose **Open** from the **File** menu and open your MIF document!

## InDesign

InDesign supports InDesign tags, Xtags, RTF, and XML. (It also supports delimited text files of course – see Chapter Three for more info on that format).

First you will need to set up an InDesign template with at least the following elements in place:

- One or more Master Pages with a text frame set up as required (no. of columns etc.)
- One or more Paragraph Styles

You'll probably also need to have things like automatic page numbering, page headers, and that sort of thing, plus perhaps Character Styles, Swatches, Table Styles, etc. set up.

## InDesign Tags

1. Open your InDesign template
2. Go to the place where you want to insert the data
3. Click in the text box
4. Choose **Place ...** from the **File** menu
5. Locate the tagged text file you want to import and double-click on it
6. The data will be imported into the document and formatted according to the styles that are specified in the tagged text file
7. If the tags include styles that do not already exist in the document, they will be created with default settings. You'll just need to modify them, choosing the font, size, etc. that you want to use for each one.

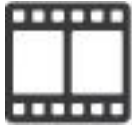
You can also import a tagged text file if you haven't selected a text box: in this event, you'll see a "loaded cursor" like this:

```
.....
|  Cornwall River
|  Fishing Camel
|  Bodmin Anglers
|  Association ( Camel)
|  Contact: Ivan Lyne,
|  Ivy Cottage, Dunmere,
|  Bodmin. Tel: 01208
```

Just click in the document where you want to place that text and a new text box will be created with the formatted text inside.

## XML

XML files don't contain formatting tags; what you need to do is map the various elements to stylesheets. There's a lot you can do with XML data.

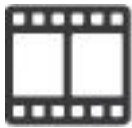


View a video and tutorial that show how to import XML data into an InDesign document.

An excellent source to find out more is the book, *A Designer's Guide to InDesign and XML*.

## Xtags

Importing an Xtags file works very much like importing an InDesign Tags file, the only difference being that in Step 4 you choose **Get Text with Xtags ...** or **Import Text with Xtags ...**, depending on which version of InDesign you're using, from the **File** menu. If you don't see an option like that on your **File** menu, then you haven't installed the Xtags plugin.



View a video that shows data being published with Xtags.

You can find more information about Xtags, and download a free trial version, [here](#).

## InDesign Tables

Worthy of a special mention here are InDesign's excellent table-creation facilities. You can create Table Styles to help automate the formatting of tabular data. Better yet, with InDesign tags and Xtags you can build all kinds of tables in various styles! Suppose you want to make tables in your price list with columns that have varied formatting and background colours ... something like this, perhaps:

COLOUR/TYPE	PACK	30	20	10	1	CODE
Blue	6	0.00	0.00	0.00	0.00	331302
Red	6	0.00	0.00	0.00	0.00	331303
Green	6	0.00	0.00	0.00	0.00	331304
Yellow	6	0.00	0.00	0.00	0.00	331305
Grey	6	0.00	0.00	0.00	0.00	331309
Black	6	0.00	0.00	0.00	0.00	331301
Charcoal	6	0.00	0.00	0.00	0.00	331308
Chrome Plated Risers 114mm Pk 4	set	0.00	0.00	0.00	0.00	331350
Chrome Plated Risers 152mm Pk 4	set	0.00	0.00	0.00	0.00	331355

[See a tutorial](#) that shows how this sort of table can be created automatically.

## QuarkXPress

Like InDesign, Quark supports its own tagged text file, Xtags, RTF, and XML.

First you'll need to create or open a Quark template or document with a master page containing a text box.

### QuarkXPress Tags

1. Open your Quark document or template
2. Click in the text box
3. Choose **Get Text ...** from the **File** menu
4. Locate your tagged text file and double-click on it
5. The file is imported and formatted according to the tags in the file.

### XML

XML importing is available in Quark version 6 and 7 via the XML Import Xtension. At the time of writing (March 2009), Quark is still working on a version of this for Version 8.

There are also third-party Xtensions available, such as Atomik XML Publisher by Easypress.

### Versions 6 and 7

You'll first need to install an Xtension (which is Quark's name for plugins). You can get it [here](#). Just put it into your Xtension folder, which is

inside your QuarkXPress program folder (you'll need to restart Quark if it was already running).

To import XML files into a Quark layout, there are two criteria to meet:

1. Your XML file must be "well-formed" (ie, it must be correctly structured)
2. You will also need to have a DTD (a Document Type Definition)

There's a step-by-step guide to importing an XML file into a Quark layout [on Quark's website](#). Note, though, that it says you need to have a well-formed XML document OR a DTD but in fact you can't import an XML file without a DTD.

## Xtags

To import an Xtags file into a Quark layout:

1. Open your layout, or create a new document with a text box
2. Click in the text box that you want to import the tagged text into
3. Choose **Get Text with Xtags** from the **File** menu
4. Locate your Xtags file and double-click on it
5. Your file will be imported and formatted according to the tags.

## RTF

An RTF file can either be imported into a document or it can be opened in Microsoft Word as a new document. There are many applications that will allow you to import an RTF file: most page layout programs, for example. They might require a filter to be present - usually you'll find a folder called Filters in the application folder, and that's where it should be. In InDesign, the Filters folder is usually found within the Plugins folder; with Quark, they are in the XTension folder. So, if your application refuses to import RTF files, check the documentation for information about filters.

To open an RTF file in Microsoft Word, simply choose **Open ...** from Word's **File** menu, select your document, and open it.



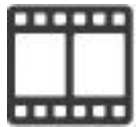
## Chapter Five: ODBC

### What's ODBC?

ODBC stands for Open DataBase Connectivity. It's a protocol for enabling applications to "talk" directly to an ODBC-enabled database or spreadsheet.

With ODBC, you can eliminate the step of exporting data from your database. Instead, you set up an ODBC connection to reach into the database and extract the data you want to publish. In some cases you can also "push" data the other way - in other words, you can modify the source database from elsewhere. Why would you want to do that? Let me give you an example.

Suppose you are using the [Data Chameleon edition](#) of [CatBase](#) (CatBase is a complete database publishing solution). Perhaps your data is stored in your MySQL or SqlServer database and you're just using CatBase to publish your catalogue. Once your database connection has been set up, whenever you are ready to publish a catalogue you simply tell CatBase to import the up-to-date data and then to publish it in the required format.



View a video that demonstrates extracting data from a MySQL database and publishing it with InDesign.

You might be using CatBase as your main database management system, and you need to make sure that your online database is always up to date. Using the ODBC option in CatBase Data Chameleon you can very quickly update your web site at any time - it can even be automated so that it's updated, say, every night.

A common problem when publishing catalogues is the last-minute pricing problem: you spend a week producing the catalogue, making each page look just right, and then you find out that all the prices have been changed. There are

several plugins that enable the prices to automatically be updated via ODBC or from a delimited text file – some of them are:

### ***DataLinker by Teacup Software***

This plugin for InDesign enables you to link data in your document to a database, so that data can be automatically updated.

### ***LinkUp by MC Research***

Database publishing XTensions for QuarkXpress.

### ***Xcatalog and Incatalog by Em Software***

Data linking plugins for QuarkXpress and InDesign that enable bi-directional linking of data between the data source and the document (note that the Pro versions support ODBC).

You can find quite a good introduction to ODBC on [Wikipedia](#).

## Chapter Six: Plugins for Page Layout Programs

### CatBase

Publisher: CatBase Software Ltd.

A complete database publishing solution which combines a database with an easy-to-use point-and-click interface to publish your data in any number of ways to lots of different publishing destinations. [See a short demonstration movie.](#)

### DataLinker

Publisher: Teacup Software

Connect data quickly and efficiently to Adobe InDesign documents. Allows you to link data anywhere on an InDesign page, or merge data into a document and into tables too. Once your data is linked, you can update from the database with the simple click of a button.

### DataLinker ODBC Add-on

Adds ODBC connectivity to the DataLinker plugin.

### EasyCatalog

Publisher: 65bit Software

A database publishing plugin for Adobe InDesign.

### Incatalog

Publisher: Em Software

A plugin for InDesign which enables bi-directional data updating between the source data and the InDesign document.

### LinkUp

Publisher: MC Research

Database publishing XTensions for QuarkXpress that cover publishing tasks from updating simple price lists to full automatic pagination using a web front end.

### Xcatalog

Publisher: Em Software



A plugin for QuarkXpress which enables bi-directional data updating between the source data and the Quark document.

### Xtags

Publisher: Em Software

Plugins for QuarkXPress and InDesign that extend the basic tagging system to enable automatic picture and text box creation and filling, and more.

### Other Useful Plugins

Application Name	Description	More info
Auto Flow	InDesign plug-in which dynamically flows text and fits text frames	<a href="http://www.in-tools.com">www.in-tools.com</a>
Column Flow	InDesign plugin which flows text in variable number of columns based on paragraph style	<a href="http://www.in-tools.com">www.in-tools.com</a>
ID2Q	Converts Quark files to InDesign	
InFlow	Plugin for InDesign which automatically creates new pages as needed when text is imported into an InDesign layout. (Note - included for free with Xtags)	<a href="http://www.emsoftware.com">www.emsoftware.com</a>
Q2ID	Converts Quark files to InDesign	
Running Headers	Plugin for InDesign that enables the automatic creation of running headers.	<a href="http://www.in-tools.com">www.in-tools.com</a>
Side Heads	InDesign plugin which creates anchored side heads based on either character or paragraph style	<a href="http://www.in-tools.com">www.in-tools.com</a>

## Chapter Eight: Resources

You can find lots of useful, constantly-updated resources on this book's Resources page:  
[www.database-publishing-book.com/resources/index.html](http://www.database-publishing-book.com/resources/index.html)

### HTML

W3 pages: [www.w3.org/html](http://www.w3.org/html)  
w3Schools tutorials [www.w3schools.com/html](http://www.w3schools.com/html)

### ODBC

[Wikipedia](#) is a good place to start!

### XML

To find out more about XML, visit the W3 web site at <http://www.w3.org/XML>.

## Chapter Nine: Glossary

**Comma-delimited text (csv):** “csv” stands for “comma-separated values” and, as you might guess, this means that each field is separated by a comma. If the text in a field contains a comma, that field will be surrounded with double quote marks. Some applications that create csv files put double quote marks around every field.

**Conditional formatting:** The option to have text formatted in a certain way depending upon some criteria that you set. For example, in an association membership directory you might want to set a member’s name in black if they are a Standard member, but in blue if they are a Patron. Or in a product catalogue, perhaps you want to set the price in red only if the product is on special offer.

**Custom-delimited text:** Some databases let you choose the delimiters you want to use. This can be useful when your data might contain the usual delimiters: the most common example is when you have text such as descriptions in your database. This text might contain return characters (new lines) which is the delimiter that is usually used to denote the end of a record. If you don’t do anything about this, your data import might be a mess because the program gets confused about where one record ends and another begins. A typical solution is to use the “pipe” character (|) as the record delimiter.

**DTD:** Document Type Definition. A DTD describes the structure of an XML document. It specifies which elements can contain which other elements, which elements are optional, and which are required, and which elements contain data. For more information about DTDs, see [the W3 website](#).

**Fixed-length text:** A delimited text format in which every field contains a certain number of characters. If the contents of a particular field don’t contain the required number of characters, it is padded out with spaces or zeroes, depending on the type of data (spaces for text; zeroes for numeric data). So every row contains exactly the same number of characters and there are no delimiters.

**InDesign Tags:** A text tagging system native to InDesign.

**Tab-delimited text:** A delimited text file format in which each field is separated by a tab character and each record usually ends with a return character. So, each record is one line, or row, in the text file and each one must contain the same number of tabs, even if a field is empty. The first line might contain field headings (field names). If you were to import a tab-delimited text file into a spreadsheet, you would see all the fields dropping nice and tidily into columns. One of the most common ways in which tab-delimited files are created is by “saving as” a tab-delimited file from a spreadsheet.

**Tagged text file:** A text file in which special “tags” have been inserted to tell the publishing destination what to do with the data. An HTML file is an example of a tagged text file. A detailed explanation can be found in Chapter xxx.

**W3C:** The World Wide Web Consortium develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. On [the W3 website](#) you will find information and learning tools for technologies such as HTML, XML, and much more.

**Well-formed XML:** An XML file that is correctly structured. Most XML viewers include a “well-

formedness” checker, and there is also one that you can use for free [on the xml.com website](#).

**XML:** eXtensible Markup Language. XML is a tagging system in which you can create your own tags and define what they mean. They look similar to HTML tags, but there is a big difference: HTML tags define the *formatting* of data, whereas XML tags define the *content* of data. If you want to find out more about XML, the best place is probably [the W3 web site](#). XML is an excellent way to transfer data between various kinds of applications. See also [DTD](#).

**XPress Tags:** A text tagging system native to QuarkXPress. Not to be confused with Xtags.

**XSLT:** Extensible Stylesheet Language Transformations. An XML-based language used for the transformation of XML documents into other XML or “human-readable” documents. A good explanation can be found on [Wikipedia](#).

**Xtags:** A plugin, published by Em Software, for QuarkXPress and InDesign. It extends the native tagged text support by adding a number of useful features, such as creating and filling text and picture boxes. See more info [here](#).

### A

Adobe FrameMaker 4-1  
Auto Flow 7-1

### C

CatBase 5-1  
Column Flow 7-1  
Comma-delimited text 2-1  
Conditional formatting 9-1  
csv 2-1, 9-1  
CSV (Comma-separated) Text Files 3-1  
csv file, from databases 3-1  
Custom-delimited text file 2-1, 9-1  
Custom-Delimited Text Files 3-1  
Custom delimiters 2-1, 9-1

### D

Database 1-1  
Database Publishing 1-1  
Data Chameleon 5-1  
Delimited Text File, publishing 3-2  
Delimited text files 3-1  
document element 4-2  
Document Type Definition 9-1  
DTD 9-1

### E

Excel 3-2  
eXtensible Markup Language 2-1, 9-1

### F

Field 1-1  
FileMaker 3-1  
filters 4-6  
Fixed-length 2-1, 9-1  
FrameMaker 4-2  
FrameMaker, publishing a tagged file 4-5

### H

Header Section 4-3  
HTML 8-1

### I

InDesign 4-5  
InDesign, importing files 4-5  
InDesign Tags 4-5

InDesign Tags, publishing 4-5  
InFlow 7-1

### M

MIF, importing into FrameMaker 4-5

### O

ODBC 5-1  
Open DataBase Connectivity 5-1

### P

Picture Boxes 4-3  
Publishing destination 1-1

### Q

QuarkXPress, importing files 4-6  
QuarkXPress Tags 4-2, 4-6

### R

Record 1-1  
Relational database 1-1  
Rich Text Format 4-2  
root element 4-2  
RTF 4-2, 4-6  
RTF, publishing 4-6  
Running Headers 7-1

### S

Side Heads 7-1  
Spreadsheet, exporting data from 3-2

### T

Tab-delimited 2-1  
Tab-Delimited Text Files 3-1  
Table 1-2  
Tables, InDesign 4-6  
Tagged text file 1-1  
Tagged Text File, publishing 4-5  
Text Boxes 4-3  
Text Tags 4-1

### W

W3 9-1  
W3C 9-1  
Well-formed XML 9-1  
Word 3-2

### X

XML 2-1, 4-2, 9-1  
XML, importing into InDesign 4-5  
XML, importing into QuarkXPress 4-6  
XML, publishing with InDesign 4-5  
XSLT 9-1  
Xtags 4-3  
Xtags, importing into InDesign 4-6  
Xtags, importing into Quark 4-6