Xcatalog
Xcatalog Pro

QuickStart
Tutorial

Welcome to the Xcatalog QuickStart Guide

Xcatalog is a powerful bi-directional linking tool for creating transparent links between your QuarkXPress documents and external data. With these links in place, you never have to enter or update your data in two places (documents and database), but can simply update one or the other from the most current version.

Xcatalog is a great solution for building modular layouts (where the design or the product information doesn't work well as a continuous text stream), and for updating information once your document has been laid out (either in modules or in a text flow). It also works well combined with our other page building products, Xdata and Xtags.

This tutorial is intended as a basic overview of Xcatalog and its functions—all of the topics covered here are thoroughly detailed in the Xcatalog user guide. The goal of this tutorial is to launch you in your use of Xcatalog to the point where you can make progress on your own.

Please visit us on the web at www.emsoftware.com. For technical support, please email us at support@emsoftware.com.

About this Tutorial

There are four major sections in this tutorial:

1. **Xcatalog Core Concepts.** This is foundational information—please read this before proceeding to the tutorials.

2. **Tutorial 1—Working with a Linked Document.** This tutorial is a walk-through of how Xcatalog works. A document has already been set up with links, and you get to build it.

3. **Tutorial 2—Creating Xcatalog Links in a Document.** This tutorial gets you behind the scenes, showing you how to use Xcatalog to recreate the first tutorial.

4. **Tutorial 3—Linking Directly to a FileMaker Pro Database.** The first two examples used a text file as a data source. This tutorial shows you how to connect directly to a database.

Xcatalog Core Concepts

### An introduction to the Xcatalog Data Descriptor File

Most traditional data sources can be thought of as tables consisting of columns and rows (in other words, a spreadsheet). Each row is a single data *record* containing all of the pieces of information for a particular item, such as product number, product name, price, description, etc. All of the product numbers would fall into a column, all of the product names would fall into another column, the prices into another, and so on. Columns are referred to as *fields*. Here is a table (opened in Excel) containing some of the data used in this tutorial:

In order for Xcatalog to interact with your data, it first needs to understand the schema (the data structure) it will interact with. The schema is a map describing the different parts of your data. We call this map a *data descriptor file*. Data descriptors provide a great deal of flexibility with regards to the types of files Xcatalog can work with. It doesn't require that there be a descriptive "header row" in the data source file. It also lets Xcatalog read data directly from databases like FileMaker on the Mac and any other ODBC compatible database on any platform.

The data descriptor is an ASCII file. It contains a tab-separated list of the field names (for spreadsheet folks, think column names) in the data source. The data descriptor for one of the following tutorials looks like this:
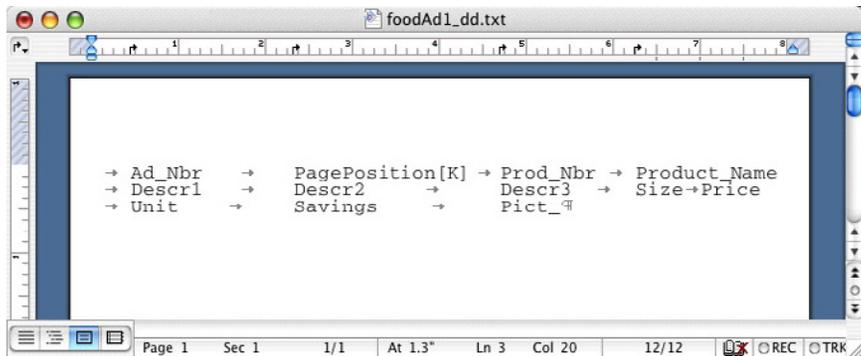
Notice that the data descriptor starts with a tab, and each field name is separated by a tab. Field names in your data descriptor don't have to match the actual names of your fields, and the only characters not allowed in a field name are brackets [].

See the [K] after PagePosition? That's a marker used to define the **key field**. An important concept to Xcatalog is the **key value** or **key**. In the analogy of a map, you can think of the key field as the map's key that provides information you need to interpret the map.

The key field tells Xcatalog, "In this data structure, I'm one-of-a-kind; once you find me, you can use me as a reference to anything else within this record." In other words, the key field contains information that is unique—no other record has the same content in this field—and is used by Xcatalog as a hook into the rest of the data inside that record. In many cases the key field is a SKU number. Let's look at the data sample again—which fields would make a good key field?

In this data sample, PagePosition, Prod_Nbr, and Product_Name could all qualify as key values (all these fields contain unique information); we will be using PagePosition as our key for this tutorial.

Note: *It is good practice to put your data descriptor file in the same location as your data extract and your Quark document.*

For more information on data and data descriptor files, please see the Xcatalog user guide, *Chapter 6: Data Descriptor Sets.*

Xcatalog Link Markers

Xcatalog places links on data in your document. You can quickly tell what page content has Xcatalog links — linked text is surrounded by non-printing link markers [brackets]. (For graphic boxes, there is currently no visual clue that a link is on place. You need to use the Xcatalog linker palette to determine if a link is there or not.) Below is an example of link marker turned on and turned off.


link markers showing   *link markers hidden*

You can toggle change the visibility by going to the Xcatalog menu and selecting Hide Link Markers or Show Link Markers.


Note: *The color of the link markers can be changed in* Xcatalog/Edit Preferences.

Xcatalog Preferences

Before you begin to work with the tutorials, this section will help you set up Xcatalog preferences and cover some basic concepts.

After you install Xcatalog (by dragging the XTension into the XTension folder, located inside your QuarkXPress folder), you should change the preferences so you can run this tutorial. Here's how to do it.

❶ Select Xcatalog/ Edit Preferences/Data Snapshot.
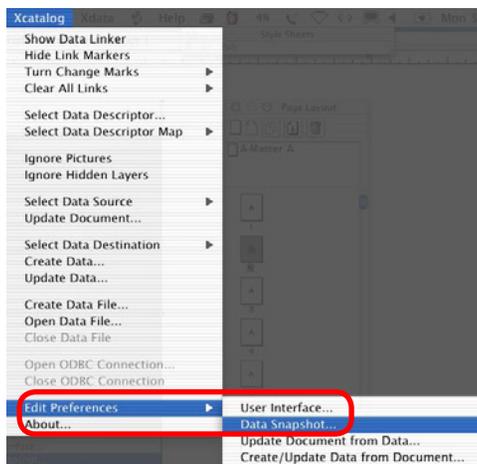
❷ Select Tab-delimited format, and be sure the checkbox for Unquoted fields box is **NOT** checked. Then click on OK.
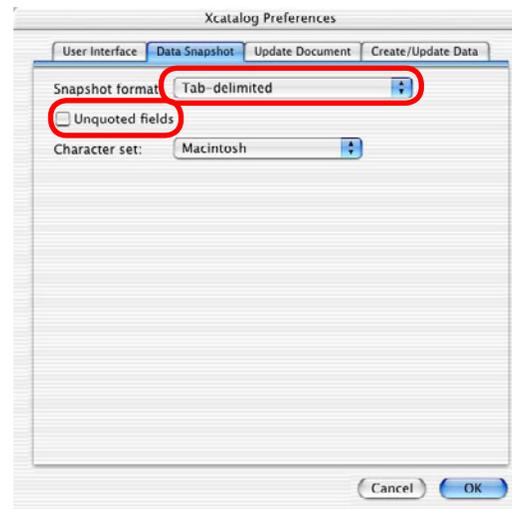
❸ Be sure that Xcatalog/Data Source and Xcatalog/Data Destination are both set to Snapshot File.
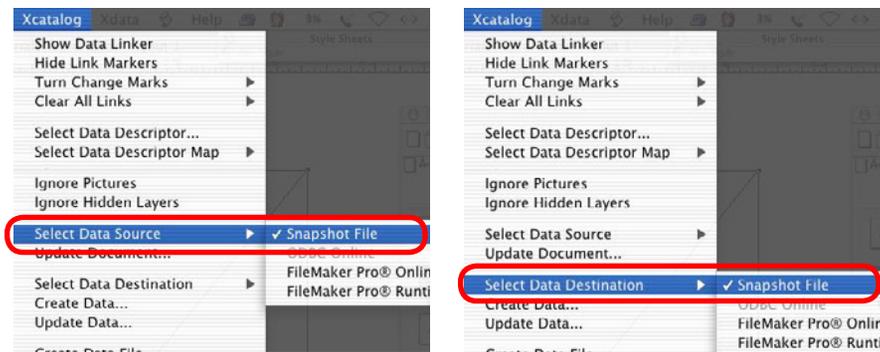
Your preferences are now set, and you are ready to start the tutorial

Tutorial 1: Working with a Linked Document

To gain familiarity with the most basic capabilities of Xcatalog, we will begin by first updating a document that has links already in place. Then we'll update the data on the page and send the updates back to the data source. Finally, we'll creating a brand new data file from the links on the page.

This tutorial outlines the most basic aspects of updating documents and data; for detailed information on working with links, please see the user guide, *Chapter 7: Linking*.

Updating a Linked Document

❶ Open the file CatTutorial_1.qxd. This is a completed grocery ad template containing Xcatalog links. We will be re-creating this template throughout this tutorial.

*Note: the visibility of the link markers is turned off. To see the link markers, select Xcatalog/Show Link Markers.*

❷ Select the data descriptor by choosing Xcatalog/Select Data Descriptor and navigate to and select the file foodAd1_dd.txt. Click on Open to select the file.
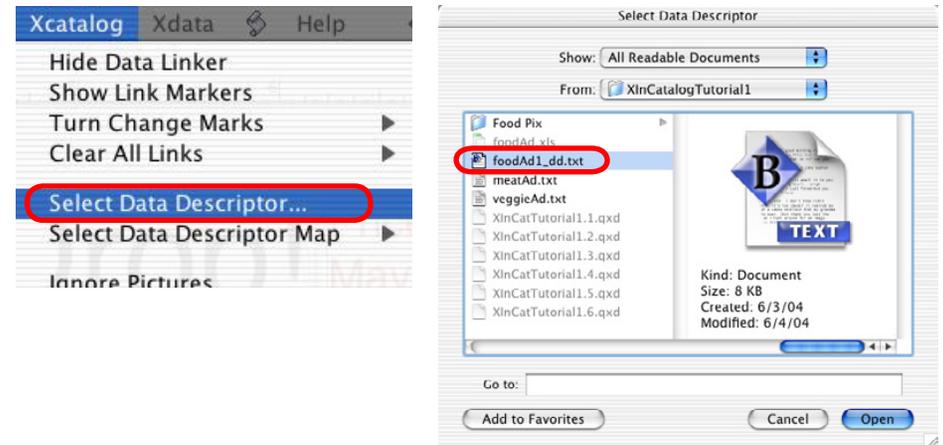
❸ From the Xcatalog menu, select Update Document.
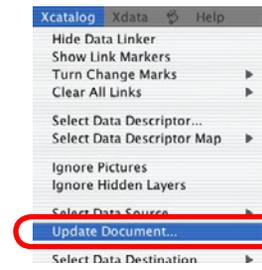
❹ Xcatalog wants to know what data file to use in order to update the document. (The data structure of this file is described in the data descriptor file you just opened.) Navigate to and select the file meatAd.txt and click the Open button . . .
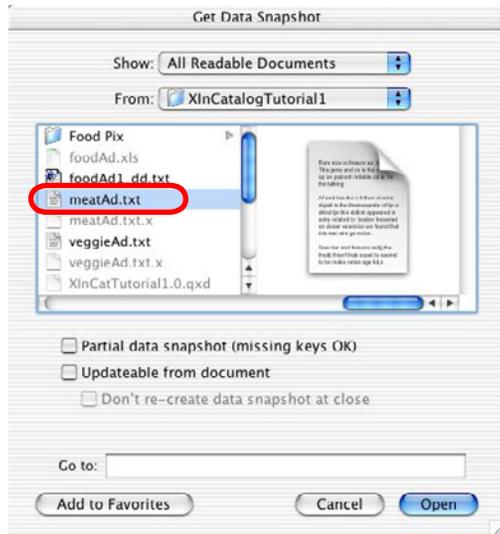

❺ . . . And presto, the linked elements in the document are updated to reflect the contents of the data file.

before . . .   *. . . and after*

❻ Now, to show you how easy it is to update this document with different information, select Xcatalog/Update Document once more. Navigate to and select the file veggieAd.txt, then click on Open.
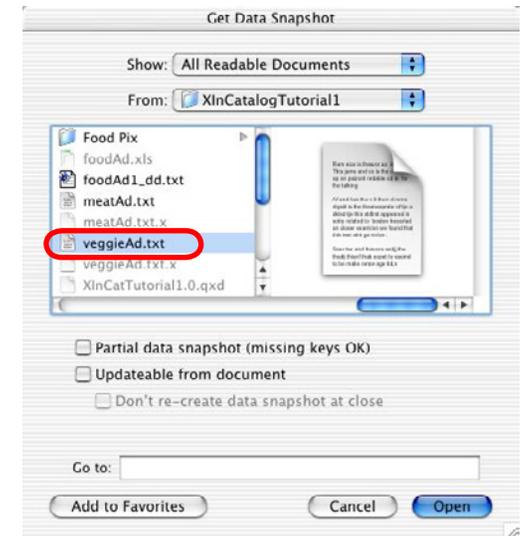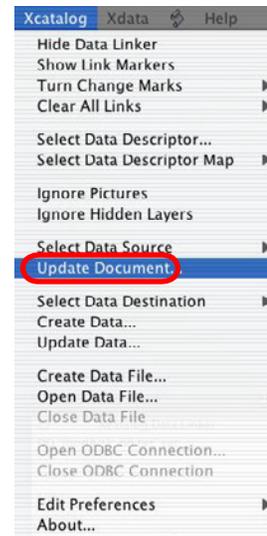
❼ And presto, here's a new ad using the same layout as before.

before. . .   *. . . and after*

How does it do this? Remember, the key field being used here is PagePosition, not Prod_Number or Product_Name. Each record has a unique PagePosition, and that's what Xcatalog is using to bring in the new found set.

Sending Updates from the Page Back to the Data Source

Changes made to content on the page can be sent back to the original data source, or, an entirely new data file can be created from a document containing Xcatalog links. In the following two exercise, we'll do both.

Updating the Original Data Source

❶ Let's have a fire sale and lower all the prices by 10 cents. You'll need to manually change all the prices circled below. Make sure links are visible (if they're not, go to Xcatalog/Show Link Markers, and adjust the prices

Note: *Currently editing the first and last character inside the brackets is a bit tricky; you need to position your cursor to the right of the first character, or to the left of the last character. If you don't, you'll be editing outside the link. We can hear your minds already forming the question, "Why?" Without getting into the technical details, this was the only way to do it in previous versions of XPress. We hope to change this behavior in the near future, but for now, just remember to edit your text in the way we've already described.*

❷ After you've changed the prices, select Xcatalog/Update Data. Navigate to and select the file veggieAd.txt, and click on Open.
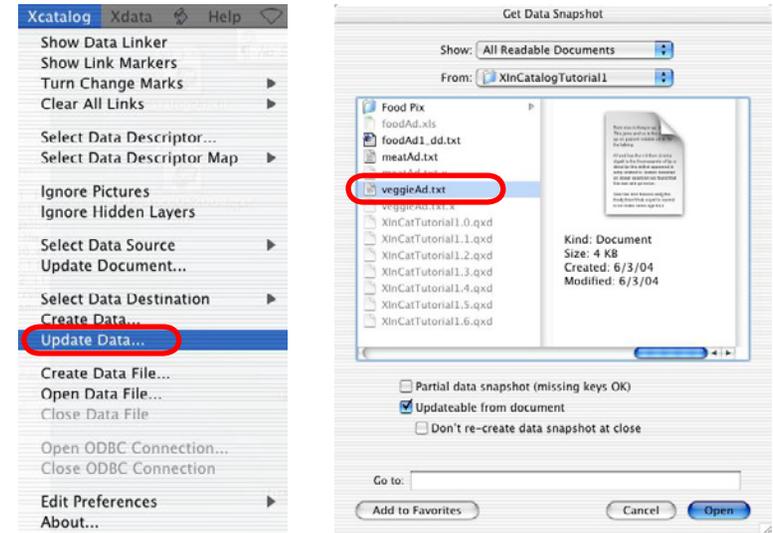
❸ Now open the data file; you'll see that the data has been updated.

data file before update   *data file after update*

❶ With the Ad document still open, select Xcatalog/Create Data. Name the new file (the default name is data.tsv) and click on Save.
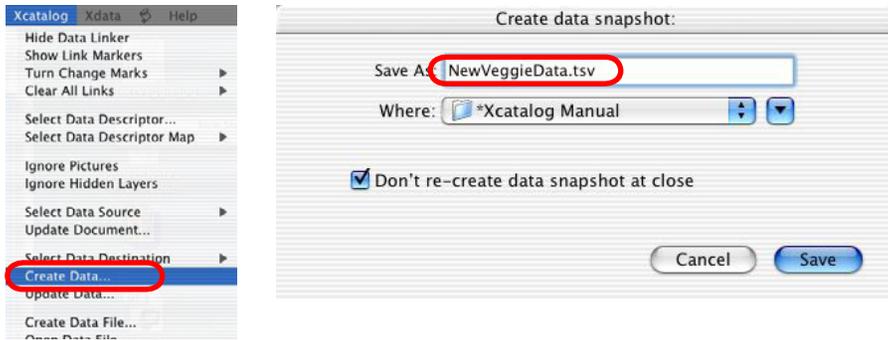
❷ The new data file is created. You can open it up in your favorite spreadsheet application; it should look similar to the photo below.

Where are all these .x files coming from . . . ? *During the data extraction or updating process to a snapshot file, Xcatalog actually creates an associated index (".x") file. These files may be deleted at any time, as Xcatalog will re-create them as necessary (though this will make the extraction or update take a bit longer).*

Tutorial 2: Creating Xcatalog Links in a Document

You can place Xcatalog links on data inside existing QuarkXPress documents or insert them while building new documents. These links can be made directly to selected text, or to entire text or picture boxes. **HOT TIP:** You can also insert "placeholder" links into library items, through which data can be injected semi-automatically when the library element is placed in a document.

This tutorial will outline the basics for creating Xcatalog links. For a complete discussion of Xcatalog linking, please see *Chapter 7: Linking* in the user guide.

Preparing to Link

❶ Open the XPress document CatTutorial_2.qxd. This document contains a single un-linked presentation for our grocery ad.

❷ The primary interface for linking is the Data Linker Palette, pictured below. If it's not open already, you can make it visible by selecting Xcatalog/Show Data Linker. For a complete description of the Data Linker palette and its functions, please see the Xcatalog user guide, *Chapter 7: Linking.*
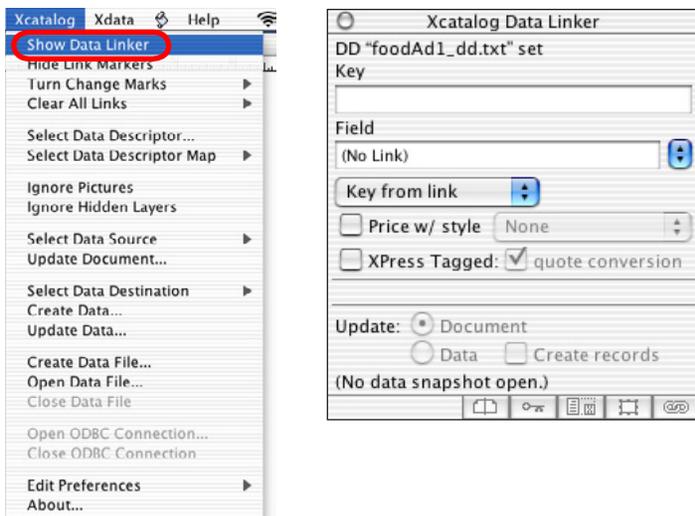

❸ After you've opened the Data Linker palette, select the data descriptor by clicking on Xcatalog/Select Data Descriptor... Now navigate to and select foodAd1_dd.txt and click on Open. Now you're ready to begin linking.
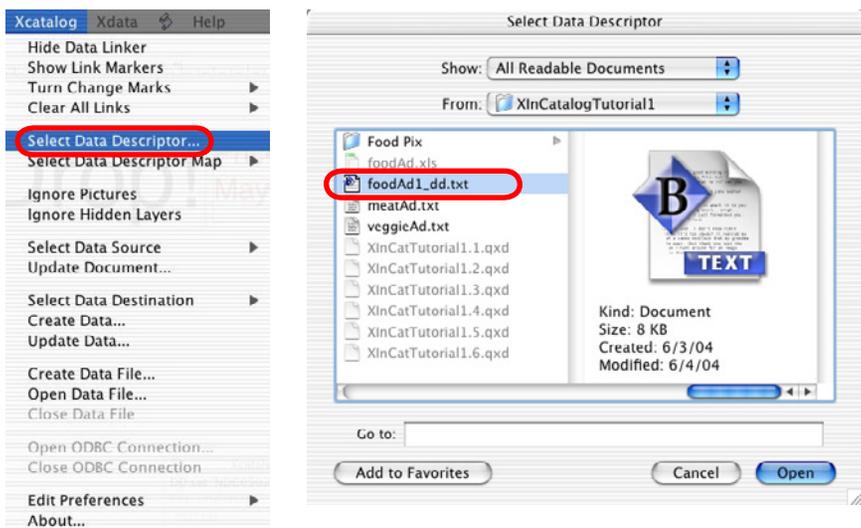
Using Direct Keys

The most basic type of Xcatalog link is of the type ***key from link***. For this type of link, you must supply a key value for each and every link that is created.

❶ With the Content tool, Select the text Product Name. Then select Product_Name from the drop-down Field menu in the linker palette.
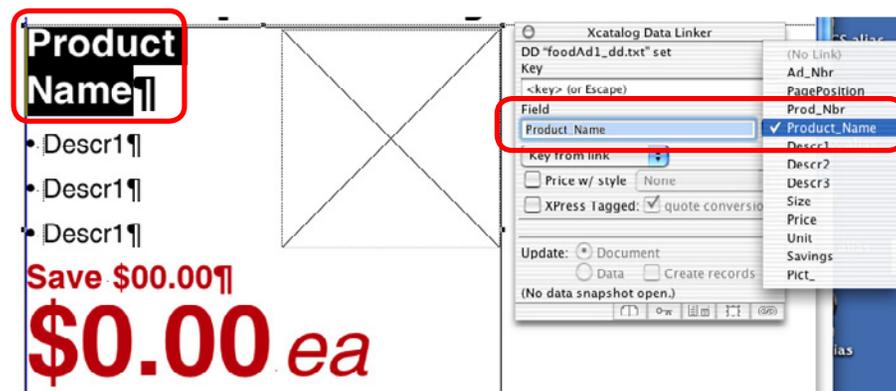

Tip: Activating View/Show Invisibles, along with showing Xcatalog link markers, helps you avoid getting unwanted paragraph returns inside of the links, and in general helps you see exactly what's going on at linked text boundaries. Zooming in can also help you find the exact boundaries of a text selection link.

❷ Type the key value "1" (one) in the Key window of the Data Linker and then hit return or enter; be sure the link type is the default Key from link.
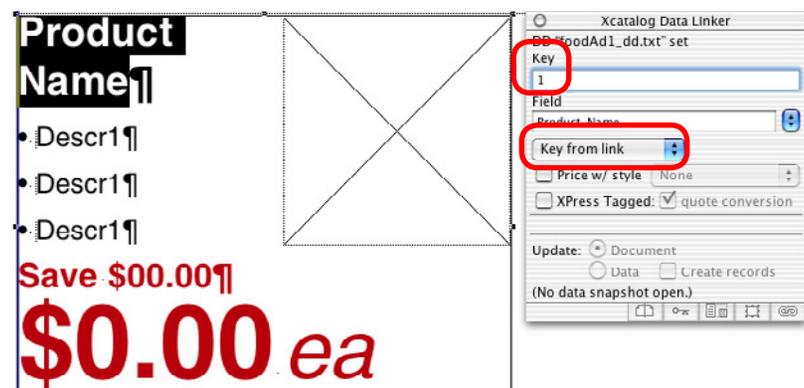
❸ Create the other desired text links with fields, link type, and key values as follows:
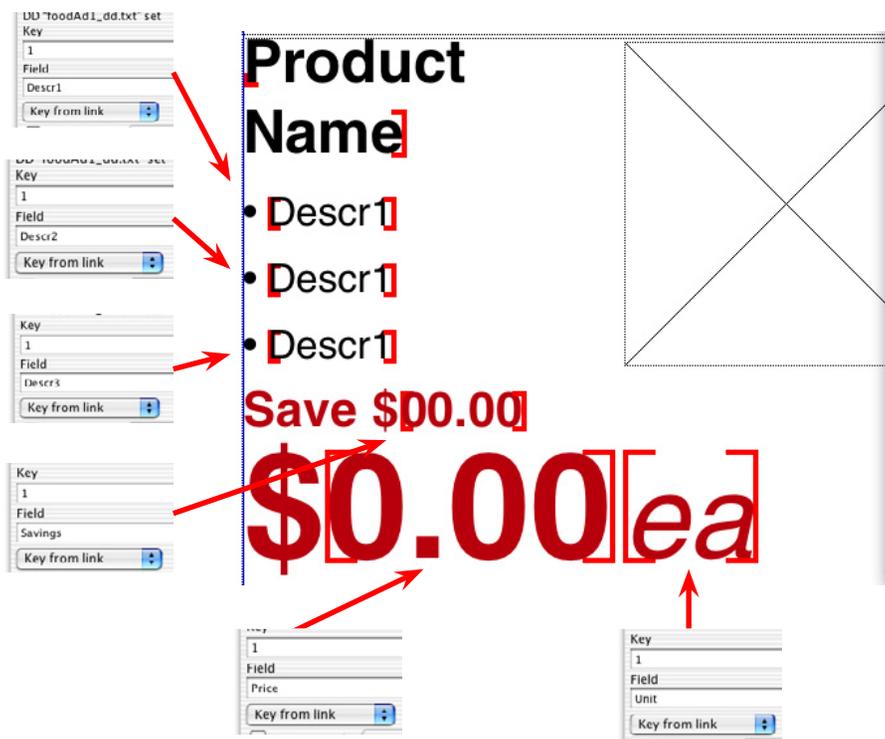
❹ Click onto the picture box to select it, use these settings for the picture box link. When you're done, save your document.
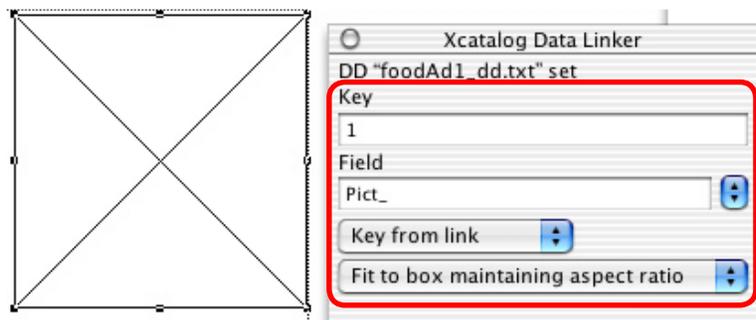
Understanding Indirect Keys

Our template has been set up using direct keys. In order to update the template with a different set of data, it will be necessary to place the cursor into each Xcatalog link, and supply a new key value in the Key window in the Data Linker. Changing every link is often a painful process. Fortunately there's another way to do set up links that make it a much faster process —indirect keys.

Indirect keys let you use a single field to supply the key for all of the links in an entire group. In this tutorial, we will look at *key from text* and *key from group* links.

Key from group

Key from group links obtain their key values from another link in the group—from one link that is set to a direct key link. Here's how to do it:

❶ First, group the text and picture boxes — hold down the shift key and click on each box with the Item tool, then press Command+G (Mac) or Control+G (Windows), or select Item/Group from the Quark menu.

Place your cursor any link in the ad. This is what your grouping should look like at this point:
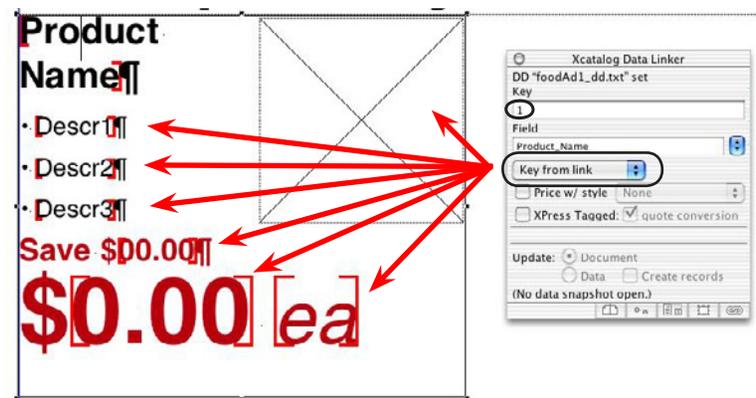

Notice that the key values for all links are "1" — in this type of layout, direct keys are not as useful as indirect keys, because these links will all need to be changed individually. That's a lot of work, and it's the reason we created the **key from group** concept.

❷ Place the cursor in each of the links (except the product name link—it's going to be our direct key, so we're not going to change that one) and change the link type to key from group. Don't forget to select the picture box by clicking on it with the Content tool, and change its link to key from group as well. This is what it should look like now:
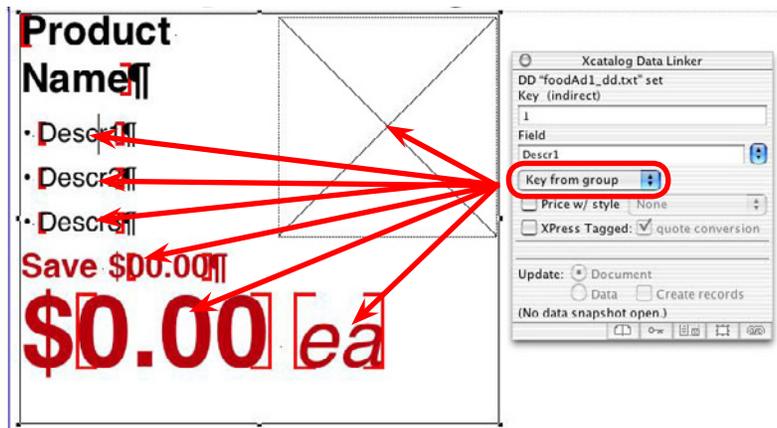
❸ Now we're ready to do some magic. With the cursor in the Product Name link, enter "2" as new key value in the Key window of the Data Linker (don't forget to press return or enter. Once you've done this, go ahead and place your cursor in the other links in the group — you'll now see that their key values is "2", the same as the product name.
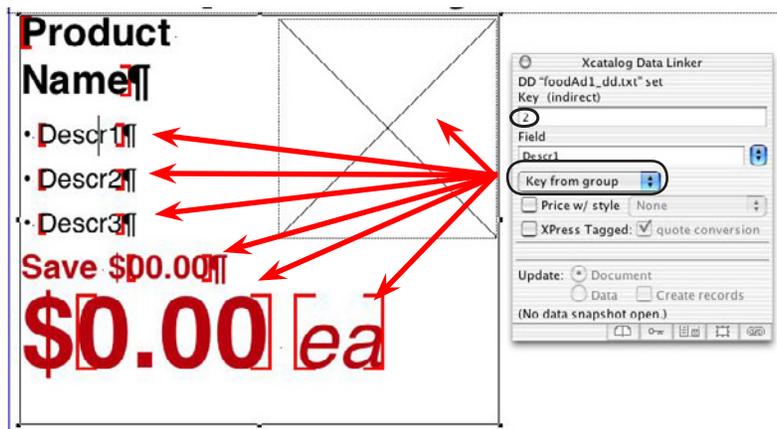
In short, setting up this type of key arrangement lets all other links pick up their key from whatever link is set to Key from Link. The key value for the In

this case, the Product Name link is the master key, and all other links act as clone keys, changing to whatever the master key is.

Key from text

Key from group is a great way to handle modular grouped items, but you can do the same thing working in a text flow by using the **key from text.** This key works by looking to the left of the linked text (key from text <<) or to the right of the linked text (key from text >>) and will use the first direct key it finds. Here's how to use it:

❶ At the bottom of the CatTutorial_2.qxd layout there's a two paragraph ad listing.
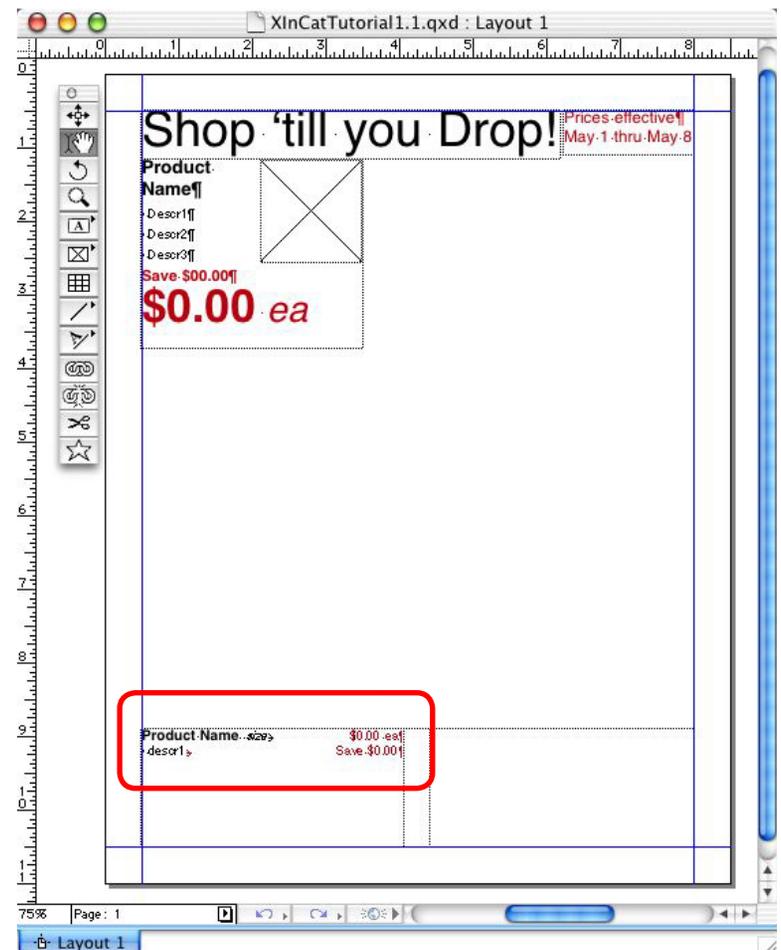
❷ We're going to ad links to this ad, just like the following illustration shows. Note that the Product_Name link contains the direct key which the other links will point to.
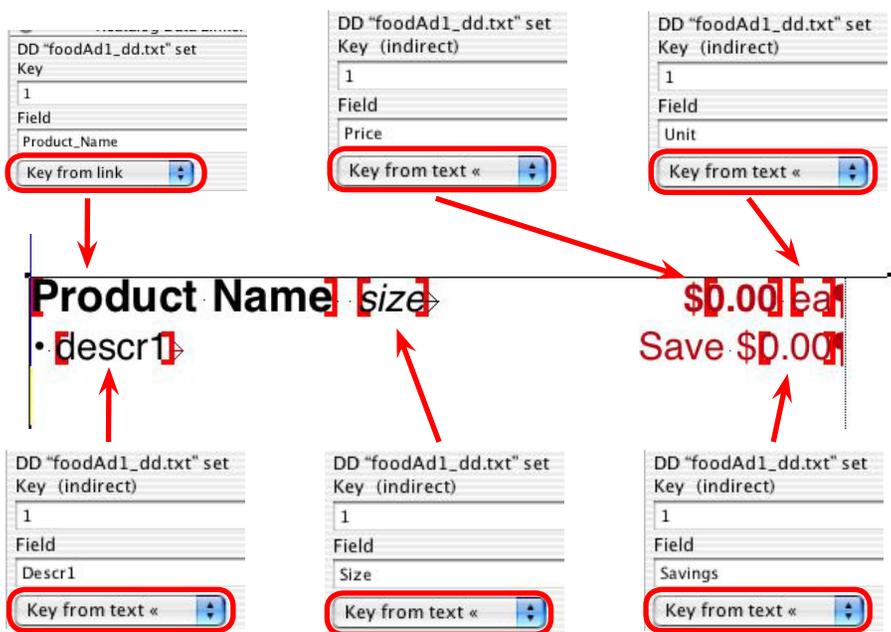
❸ Once you've established your links, you can change the key value in the Product_Name link, and then put your cursor in any of the other links. You'll see that the link you're in has picked up the key value from the Product_Name link.

In this example, all our keys are pointing left (Key from text <<) to catch the direct link on the Product_Name field. You could just as easily put the other linked fields in front of the product name field. These links would use key from text >> and catch the direct link on the Product_Name field.

Key from group and ***key from text*** give you the same results, but which one you should use depends upon the kind of layout you're working with. You can use both of them together as well.

Building a Document

Now that both presentations are complete, simply copy and paste to build the page.

❶ Select the grouped ad containing links with the Item tool, and copy it to the clipboard. Then paste and arrange the presentations (you can lay these out anywhere and any way you want).

❷ Select the ad listing containing links with the Content tool and copy it to the clipboard. Now paste a few copies into the text box.
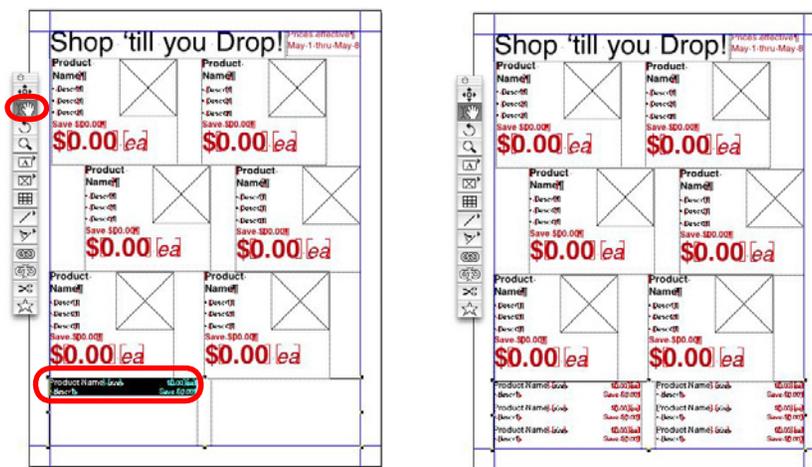
❸ Change the key value of the Product_Name link. Assign the values 1–12 as follows:
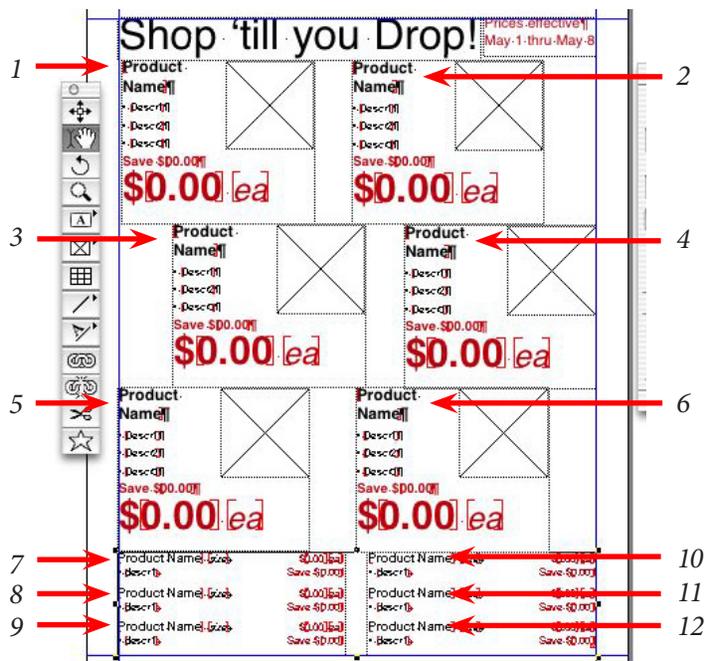
❹ Select Xcatalog/Update Document and watch the page build almost instantly.

Tutorial 3: Linking Directly to a FileMaker Pro Database

Working with a delimited text file as your data source is a common way to work, but Xcatalog lets you connect directly to FileMaker (and to any database via ODBC). This section shows how to use Xcatalog to link directly to a FileMaker database on the Mac platform. We assumed that you already have some experience with working with FileMaker and are familiar with such concepts as performing finds, found sets, etc.

Working with Found Sets

❶ Open the FileMaker Pro database XIncatTutorial.fp5. It looks like this:

Xcatalog operates on the current found set of records. Notice that this database contains two "sets" of data, as identified by the AdNumber field. Begin by performing a "Find" for all of the AdNumber 1 records:

❷ Select Find Mode from FileMaker's View menu (or type command+F). Then Enter "1" into the AdNumber field in the Find Mode layout, and press **Enter.**
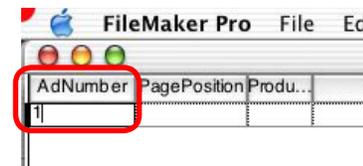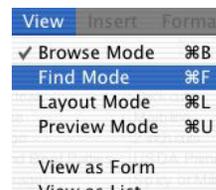
The Ad 1 records become the found set.

❸ Open the tutorial document CatTutorial_3.qxd. Set the preferences for Data Source and Data Destination to FileMaker Pro Online.
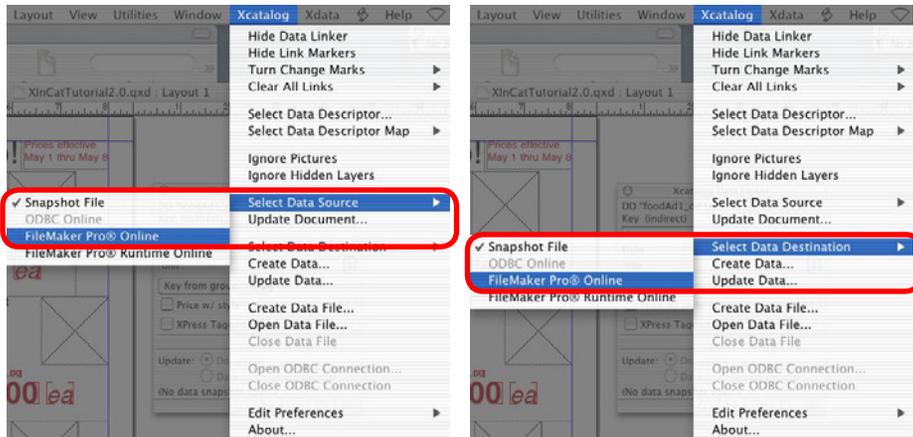
❹ Use the procedures outlined in the previous sections to select the data descriptor foodAd1_dd.txt from the FileMaker Document Descriptor folder.

FileMaker and the Data Descriptor

In order to link to a FileMaker database, the "D" and "F" field qualifiers must be used in the data descriptor file. The "D" qualifier contains the database name; the "F" qualifier contains the field name, as follows:

Product_Name[D"XIncatTutorial.fp5" F"Product Name"]

Note: the field name, as specified in the "F" qualifier, must exactly match the field name as it appears in the database—in fact, if the field name used in the data descriptor exactly matches that in the database, the "F" qualifier is not necessary.

Compare the data descriptor used in this section of the tutorial with the original data descriptor, which linked only to the delimited text files ("�" indicates a tab).

The original data descriptor:

➙Ad_Nbr➙PagePosition[K]➙Prod_Nbr➙Product_Name➙Descr1➙Descr2 ➙Descr3➙Size➙Price➙Unit➙Savings➙Pict_

And the revised Data Descriptor:

➙Ad_Nbr[D"XIncatTutorial.fp5" F"AdNumber"]➙PagePosition[K D"XIncatTutorial. fp5" F"PagePosition"]➙ Prod_Nbr[D"XIncatTutorial.fp5" F"ProductNumber"]➙Pr oduct_Name[D"XIncatTutorial.fp5" F"Product Name"]➙ Descr1[D"XIncatTutorial. fp5" F"Description1"]➙Descr2[D"XIncatTutorial.fp5" F"Description2"]➙ Descr3[D"XIncatTutorial.fp5" F"Description3"]➙Size[D"XIncatTutorial.fp5" F"Size"] ➙Price[D"XIncatTutorial.fp5" F"Price"]➙Unit[D"XIncatTutorial.fp5" F"Unit"]➙Savi ngs[D"XIncatTutorial.fp5" F"Savings"]➙Pict_[D"XIncatTutorial.fp5" F"Picture1"]

Updating Your Document With A New Found Set

In Tutorial 1 you learned how to upda te a document with a particular data source, and then pick another data source and update the document again. You can do the same kind of thing when directly connected to a database. The found set can be quickly redefined, and based on the new found set your document can be re-updated. In other words, you don't have to change any key numbers on the direct link fields. The process is just like the previous one to determine your found set. Let's do it right now.

❶ In the previous section we set up the data source and destination to point to FileMaker. We're ready now to update the document from the existing found set. Select Xcatalog/Update Document, and the document should update based on the contests of the current found set.
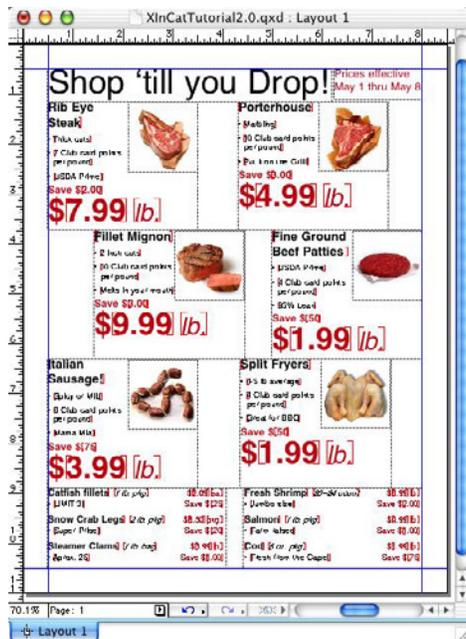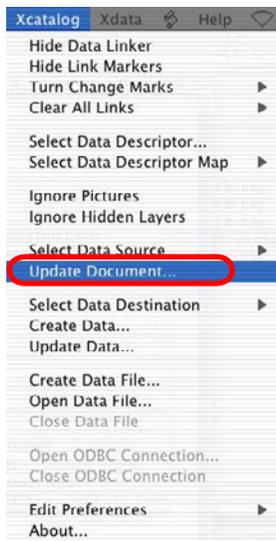
❷ Now we're going to change the found set. Switch to FileMaker, and select Find Mode from FileMaker's View menu (or type command+F). Enter "2" into the AdNumber field in the Find Mode layout, and press return or enter.
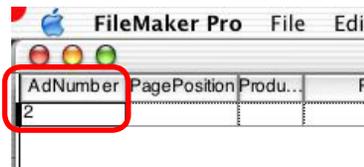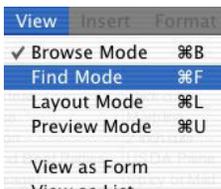
The Ad 2 records become the found set.

❸ Switch back to your Xpress document. Select Update Document from the Xcatalog menu, and the document is updated.
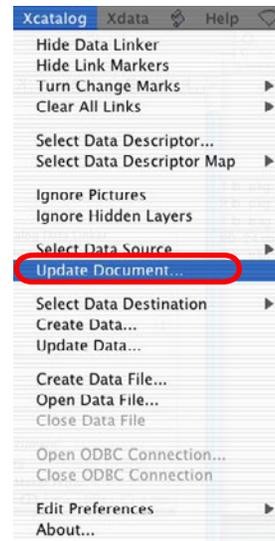
How does it do this? Remember, the key field being used here is the PagePosition field, not the Product_Name field (and certainly not the found set). Each entry has a unique PagePosition, and that's what Xcatalog is using to bring in the new found set.