

# User Manual





First Edition March 1998  
User Manual Release 2.94

© 1993-1998 MC Research Ltd

MC RESEARCH LTD.  
SUITE A, SQUIRES HOUSE, 81-87 HIGH STREET  
BILLERICAY  
ESSEX CM12 9AS, UK  
TEL: +44 (0) 1277 632 887  
FAX: +44 (0) 1277 632 023

EMAIL: [support@mcresearch.co.uk](mailto:support@mcresearch.co.uk)

# MC Research Limited END USER LICENCE AGREEMENT

**IMPORTANT: THIS IS A LICENCE AGREEMENT BETWEEN YOU AND MC RESEARCH LIMITED. BY OPENING THE SEALED ENVELOPE CONTAINING THE SOFTWARE DISK(S) YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT ACCEPT THIS AGREEMENT, DO NOT OPEN THE ENVELOPE CONTAINING THE SOFTWARE DISK(S) BUT PROMPTLY RETURN IT TOGETHER WITH ALL ACCOMPANYING ITEMS TO THE PLACE OF PURCHASE WITH PROOF OF PURCHASE AND YOU WILL RECEIVE A FULL REFUND, PROVIDED THE ENVELOPE CONTAINING THE SOFTWARE DISK(S) REMAINS UNOPENED AND ALL ACCOMPANYING ITEMS HAVE BEEN RETURNED WITHIN 14 DAYS OF THE DATE OF PURCHASE.**

## 1 Licence

### 1.1 You are permitted to:-

- (a) load the Software into and use it on a single computer which is under your control;
- (b) use the Software on a number of computers provided you have purchased such number of copies of the Software
- (c) transfer the Software from one computer to another provided it is used on only one computer or the specific number of computers for which copies of the Software have been purchased at any one time;
- (d) make up to 3 copies of the Software for back-up purposes only in support of the permitted use. This copy must reproduce and include the Owner's copyright notice on each copy.
- (e) transfer the Software (complete with all its associated documentation) and the benefit of this Agreement to another person provided he has agreed to accept the terms of this Agreement and you contemporaneously transfer all copies of the Software you have made to that person or destroy all copies not transferred. If any transferee does not accept such terms then this Agreement shall automatically terminate. The transferor does not retain any rights under this Agreement in respect of the transferred Software.

### 1.2 You are not permitted:-

- (a) to load the Software on to a network server for the purpose of distribution to one or more other computer(s) on that network or to effect such distribution (such use requiring a separate or appropriate licence), except where more than one copy has been purchased;
  - (b) except as expressly permitted by this Agreement and save to the extent and in the circumstances expressly required to be permitted by law, to rent, lease, sub-license, loan, copy, modify, adapt, merge, translate, reverse engineer, decompile, disassemble or create derivative works based on the whole or any part of the Software or its associated documentation or use, reproduce or deal in the Software or any part thereof in any way.
- 1.3 To the extent that local law gives you the right to decompile the Software in order to obtain information necessary to render the Software interoperable with other computer programs, the Owner hereby undertakes to make that information readily available to you. The Owner shall have the right to impose reasonable conditions such as a reasonable fee for doing so. In order to ensure that you receive the appropriate information, you must first give the Owner sufficient details of your objectives and the other software concerned.

## 2 Term

This Agreement is effective until you terminate it by destroying the Software and its documentation together with all copies. It will also terminate if you fail to abide by its terms. Upon termination you agree to destroy all copies of the Software and its documentation including any Software stored on the hard disk of any computer under your control. Termination of the licence will not entitle the licensee to repayment, wholly or partially, of the licence fee.

## 3 Ownership

- 3.1 You own only the diskette (or authorised replacement) on which the Software is recorded. You may retain the diskette on termination of this Agreement provided the Software has been erased. The Owner shall at all times retain ownership of the Software as recorded on the original diskette and all subsequent copies thereof regardless of form. This Agreement applies to the grant of the licence contained herein only and not to the contract of sale of the diskette. The Owner's warranties and telephone support service under this Agreement are available only to the original registered user (being the person who has signed and returned duly completed the enclosed Licence Registration Form to the Owner within 30 days after the date of original purchase).
- 3.2 You acknowledge that the Software is an XTension™ for use with a valid registered copy of QuarkXPress™ software. QuarkCopydesk, QuarkDispatch, Quark Publishing System, QuarkImmedia and QuarkXTensions are trademarks of Quark, Inc. which have been registered in the U.S. Patent and Trademark Offices, and in many other countries.

#### 4 Warranties

- 4.1 The Owner warrants that the diskette on which the Software is supplied, if any, will be free from defects in materials and workmanship under normal use for a period of 60 days after the date of original purchase ('the Warranty Period'). If a defect in the diskette shall occur during the Warranty Period it may be returned with proof of purchase to the Owner who will replace it free of charge
- 4.2 The Owner warrants that the Software will perform substantially in accordance with its accompanying documentation (provided that the Software is properly used on the computer and with the operating system for which it was designed) and that the documentation correctly describes the operation of the Software in all material respects. If the Owner is notified of significant errors during the Warranty Period it will correct any such demonstrable errors in the Software or its documentation within a reasonable time or (at its option) provide or authorise a refund (against return of the Software and its documentation).
- 4.3 The above represent your sole remedies for any breach of the Owner's warranties, which are given only to the original registered user.
- 4.4 The express terms of this Agreement are in lieu of all warranties, conditions, undertakings, terms and obligations implied by statute, common law, trade usage, course of dealing or otherwise all of which are hereby excluded to the fullest extent permitted by law.
- 4.5 The Owner does not warrant that the Software will meet your requirements or that the operation of the Software will be uninterrupted or error-free or that defects in the Software will be corrected. You shall load and use the Software at your own risk and in no event will the Owner be liable to you for any direct loss or damage of any kind (except personal injury or death resulting from the Owner's negligence)
- 4.6 The Owner will also not be liable for lost profits or other indirect or consequential loss arising from your use of or inability to use the Software or from errors or deficiencies in it whether caused by negligence or otherwise except as expressly provided herein.
- 4.7 In no event shall the Owner's liability exceed the amount paid by you for the Software.

#### 5 Law

This Agreement shall be governed by English law and be subject to the exclusive of jurisdiction of the English Courts.

**If you have any questions concerning this Agreement please write to the Commercial Director of MC Research Ltd**

# Table of Contents

<b>Start Here</b> .....	<b>9</b>
How to use this manual .....	9
Where to start.....	10
Support .....	11
Registration .....	11
When contacting MC Research .....	11
Services Available.....	12
Maintenance Contracts.....	13
Contacting MC Research for consultancy service.....	14
Version 6 Users .....	15
New Features .....	15
New Concepts .....	16
Terminology and Interface Changes .....	16
<b>Installation Guide</b> .....	<b>18</b>
What's on the CD-ROM?.....	18
Minimum Requirements.....	21
What do you Install? .....	21
What do you want to do?.....	22
Installation - if you would like to work with an ASCII text file .....	23
Installing the XTension.....	23
Installing the Dongle .....	24
Confirming the Installation.....	25
Installation - if you would like to work with an ODBC Compliant Data Source ....	26
Installing the XTension.....	26
Installing the Dongle .....	27
Confirming the Installation.....	28
Using LinkUP! and ODBC .....	29
<b>Concepts</b> .....	<b>34</b>
What is LinkUP!.....	34
Why is LinkUP! useful?.....	34
What is a Database? .....	35
Databases types .....	37
How does LinkUP! work with XPress documents?.....	38
Administrator / Operator .....	38
What is ODBC?.....	39
LinkDB .....	39

Working with Simple Databases and LinkUP!.....	40
Row, or RECORD.....	40
Column or FIELD.....	40
Field Names.....	41
Key Field.....	41
Assigning a Key Field.....	41
Hiding Key field data on the page.....	42
The LinkUP! Record Number (LRN).....	42
<b>Navigation.....</b>	<b>45</b>
Guide to Floating Palettes.....	46
Markup Tools Palette.....	46
Markup Configuration Palette.....	47
Markup Type Icon.....	48
Index Tools Palette.....	49
Data Viewer Palette.....	50
Command-by-command reference.....	52
About LinkUP!.....	52
Add Index Entry.....	53
Add Special Markup.....	53
Add TOC Entry.....	54
Batch.....	54
Clear Markups.....	55
Configure Import Links.....	55
Copy Markups.....	56
Create Index.....	56
Create TOC.....	57
Create Header/Footer.....	57
Define New Field.....	58
Export.....	58
Load Project.....	59
Mark Key Field.....	59
Mark Field.....	60
Next Oversight Box.....	60
Paste Markups.....	61
Picture Tags.....	61
Process Markup Tags.....	62
Proof by Field.....	62
Proof by Key.....	63

Resequence LRNs.....	63
Save Project.....	64
Show Data Viewer Palette.....	64
Show Index Tools Palette.....	65
Show Markup Configuration Palette.....	65
Show Markup Tools Palette.....	66
Update QuarkXPress.....	66
<b>Tutorials.....</b>	<b>67</b>
Tutorial 1 - Updating a QuarkXPress Document.....	68
Overview.....	68
Making it Happen.....	68
Tutorial 2 - Template Building.....	75
Overview.....	75
Making it Happen.....	75
Tutorial 3 - Preparing a Document for LinkUP!.....	81
Overview.....	81
Making it happen.....	81
<b>Reference.....</b>	<b>91</b>
ODBC.....	91
How does ODBC work?.....	91
So why should you use ODBC?.....	92
LinkUP! and ODBC.....	92
The Standard.....	93
ODBC Requirements.....	93
Configuring a Data Source (LinkDB Driver).....	95
Configuring a Data Source (Other Drivers).....	95
Installing the correct ODBC version of LinkUP!.....	95
Establishing a Connection.....	96
ODBC Connectivity Issues:.....	97
Field Names & Data Setup.....	98
Defining Fields.....	98
Editing Fields.....	103
Deleting Field Names.....	104
Views.....	105
How do I create a New View.....	107
Configuring Import Links.....	110
Marking Up.....	117
Overview.....	117

Marking Up Database Links.....	120
Picture markups.....	150
Editing Markups.....	151
Marking up an item out of sequence.....	152
Deleting a Markup.....	153
Relational Markups.....	154
Projects.....	159
Working with Project Files.....	159
Updating.....	162
Restricting the update range.....	163
Working with Pictures.....	165
Configuring Picture Tags.....	166
Creating Indexes From your QuarkXPress Documents.....	167
Indexing Fundamentals.....	167
Marking and Altering Index Entries.....	168
Generating an Index.....	169
Creating Table of Contents (TOC) from your QuarkXPress documents.....	175
TOC Fundamentals.....	175
Marking and Altering TOC Entries.....	176
Generating a Table of Content.....	177
Other uses of TOC.....	181
Header/Footer.....	182
Creating a Header/Footer.....	182
Relational Reference.....	184
What is a Relational Database?.....	184
Why is being relational important to LinkUP!.....	187
Configuring Relations with LinkUP!.....	188
Proofing.....	191
Proof by Field.....	191
Proof by Key.....	192
Proof Reports.....	194
<b>Advanced Topics.....</b>	<b>196</b>
Next Overset Box.....	196
Exporting.....	197
Why you do it.....	197
How you do it.....	198
Configuring the Export Parameters.....	199
The Export.....	200

Tagging.....	203
What are LinkUP! Tags .....	204
Using LinkUP! Style Tags.....	204
Using LinkUP! Markup Tags.....	205
Picture Tags .....	206
XPress Tags.....	207
Processing Tags on Update .....	208
Processing Tags in Entire Document.....	208
Character Maps .....	209
Apple Scripting .....	211
What is Apple Scripting? .....	211
Where do I find it?.....	211
LinkUP! and AppleScript .....	212
<b>Appendices .....</b>	<b>213</b>
XPress Tags.....	214
Character Attributes .....	214
Paragraph Formats .....	215
Glossary .....	216
Index.....	217

# START HERE

## How to use this manual

The most important section of this manual for finding your way around both the manual and LinkUP! will be the navigation chapter. There are two key elements to the navigation chapter, the palettes Exposed and the Command Reference. The Guide to Floating Palettes explains what all of those quirky little icons and buttons mean and do while the command reference will give you instructions as to go in this manual for a detailed explanation of that particular command and the process it fits into.

# Where to start

There is only one starting place in this manual and that is with the Concepts Chapter. Having a good grasp of the concepts will enable you to go on and get the most from this manual and more importantly from LinkUP! We cannot stress too strongly how important it is for you to read this before anything else.

Next we would suggest you find a tutorial in the tutorial section that is relevant for the type of publication you are going to produce. Obviously every publication is different but in terms of publishing with LinkUP! your documents fall into one of 3 categories. These categories are explained in the Concepts Chapter (the one you are going to read next).

After working through one or more of the tutorials, have a look through the planning and implementation Chapter, this will point you towards all of the aspects you should be considering when implementing LinkUP! on your document and it is obviously better to have this information at the beginning than discovering a flaw in your plan further down the road.

As we stated, every publication is different so there probably won't be a 10 step point-and-click guide to building your specific publication. However, once you have a grasp of the concepts and functionality of LinkUP! then by using this manual as a reference point you will be able to get the most from LinkUP! and go on to reap the benefits of using this ground breaking solution.

If you are upgrading from Version 6 of LinkUP! you will benefit from reading the section *Version 6 Users* on page 15.

# Support

## Registration

For some of the support options available, such as telephone support, you will have to have registered your product first. Product registration can be achieved either by mailing the Product Registration Card to MC Research or On Line on the MC Research Internet Site ([www.mcresearch.co.uk](http://www.mcresearch.co.uk)). Make sure you register as soon as you receive your product for uninterrupted support.

MC Research is committed to providing its customers with the best support available wherever they are. If you have problems using LinkUP! that are not covered in this manual then the following are resources that are available to you as a registered customer of LinkUP!

## When contacting MC Research

When contacting any of the MC Research support services, as well as your serial number (given below) you should always make sure you have the following information to hand:

SERIAL NUMBER GOES HERE

- LinkUP! Serial Number
- QuarkXPress Version
- LinkUP! Version
- Version of ODBC Drivers being used (If Appropriate)
- Macintosh Model
- MacOS Version
- A Description of the Problem

# Services Available

## MC RESEARCH BY MAIL:

You can always contact us using the conventional method of mail should you wish. Please send you queries to:

LinkUP! Support Services  
MC Research  
Suite A  
Squires House  
81-87 High Street  
Billericay  
Essex CM12 9AS  
United Kingdom

This is also the address to send any data files or QuarkXPress documents you would like us to examine. We will accept most standard media formats.

## TELEPHONE SUPPORT

For your first year after purchasing LinkUP! you are entitled to free telephone support. All you have to do to be eligible for this is to register your copy of LinkUP! This will put you in touch with the experts for your software, trained to help you overcome any initial installation and configuration problems. After this period of time you will need to take out one of the maintenance contracts to continue to benefit from this. **Please make sure you have thoroughly looked in this manual for the solution to your problem before contacting the telephone support desk.**

MC Research's telephone support service is available from 9:00 AM to 6:00 PM GMT, Monday through Friday although you are welcome to try us outside of these hours should you wish. All the electronic services are available 24 hours a day, 7 days a week.

Telephone +44 (0)1277 632887

## **MC RESEARCH FTP SERVICE**

Download the latest interim updates to your MC Research Products by simply keying in your serial number. You will have to have registered your product first for your serial number to have been validated.

## **MC RESEARCH INTERNET SITE**

<http://www.mcresearch.co.uk/support>

The MC Research Internet Site gives you access to Frequently asked questions, technical documents and e-mail access to Product Technicians. If you are a registered user you can simply complete an online support request form for prompt support world-wide.

## **MC RESEARCH BY FAX**

Fax your support requests to MC Research remembering to include all of the information listed on page 11 along with a detailed account of the problem.

Fax +44 (0)1277 632023

# **Maintenance Contracts**

Once your first years support has expired, if you wish to continue receiving the benefits of all of the support services available, you can take out a maintenance support contract with us. Please call for up-to-date pricing on this option and for more details on the benefits you will receive.

# Contacting MC Research for consultancy service

Selling shrink-wrapped software is not all we do at MC Research, in fact there are a range of services available targeted towards the publishing industry. MC Research specialise in database specification and design. Our unique marrying of publishing expertise with database knowledge ensures that these two vital parts of your business will operate hand in hand. MC Research undertake bespoke projects when appropriate and have many impressive publishing systems installed in the UK. Please contact us to discuss your requirements or to arrange a meeting with one of our consultants.

Outside of the UK MC Research have several partners that we work closely with that can bring you a complete consultancy service in your area. Several of these partners also undertake specification and design of bespoke publishing systems.

Other shrink wrapped MC Research products include Pricelt - in place price updating for XPress, LinkIndex for your complete Indexing and Table of Contents Requirements, The Genesis Directory Publishing System and Mergelt - mail merging for XPress. These products are available direct from MC Research or from our recognised resellers and integrators.

# Version 6 Users

Version 7 of LinkUP! has a wealth of new and improved features that should make users even more productive than previously. With these new and improved features have also come some terminology changes and changes to LinkUP! working methods.

## New Features

**Support for relational database structures.** This is one of the most significant changes to the new version. Even if you don't understand the intricacies of relational data, the application of this technology will be apparent to all. It means you can now work with grouped database items directly on the QuarkXPress page. Simply place a template for a group of products on the page and then LinkUP! will automatically bring all the related products for that group into the document. As LinkUP! is bi-directional, when using LinkUP! 7 to create your database, it can create a fully relational database structure for you.

**The Data Viewer.** This gives you the ability to directly search and query your database from within QuarkXPress. Once you've located the item you need, simply assign it to an element on the XPress page. XPress operators will be able to put together pages in minutes instead of hour. All of the cells within the Data Viewer are resizable making viewing absolutely anything within XPress a breeze.

**Operator/Administrator Mode.** This makes using LinkUP! even easier for your users. Running LinkUP! in Operator mode greys out certain configuration options on the LinkUP! menu, removing unnecessary complication.

**Automatic Character Translation.** When transferring data between platforms one of the most tedious tasks can be handling the characters that change during this process. By setting up a character translation table, LinkUP! seamlessly converts the characters as they come from the database into XPress and vice versa so the operators need no longer concern themselves with it.

**Conditional Updating.** Specify whether or not certain items should be updated or even placed on the QuarkXPress page dependent on their value. For example a car guide could have the 'recommended' logo placed or not placed on the page dependent on whether or not the car model is specified as being recommended in the database.

**Automated Building functions.** As well as updating documents, of equal importance is LinkUP!'s ability to automatically build documents in several ways. MC Research have extended and automated these functions making automated building of car guides, TV listings, directory listings, holiday brochures and more, a simple process.

**AppleScript Support.** LinkUP! now supports Apple Computer's AppleScript Language. This means if you work with AppleScript you can now automate the use of LinkUP! within your production environment should you wish.

**Other Features.** Apart from the ones listed above, LinkUP! 7 also includes an enhanced user interface with new toolbars and palettes for ease of use, data source automation and batch processing.

## New Concepts

The largest new concept for previous users to grapple with in version 7 is the concept of relational database structures. Obviously if you only wish to carry on using LinkUP! as you have previously then you will not need a detailed understanding of this. However if you want to take full advantage of this added functionality and start automatically building items, groups of items or even your whole publication from a relational database structure you should get to grips with this early on. Fuller explanations are given in *Concepts* on page 34 and *Relational Reference* on page 184.

## Terminology and Interface Changes

The good old ISN (Internal Sequence Number) has undergone a name change and is now forever known as the **LinkUP! Record Number (LRN)**. Its function is still the same, the only change is the name.

The Edit Palette has been replaced by the **Markup Configuration Palette**, which reflects the major changes it has undergone and the extra functionality provided. The Markup Palette has simply been renamed the **Markup Tools Palette** as it now contains all of the tools necessary for marking up a document, including the load project button. A full description of the buttons on this and the other palettes is given in the navigation chapter on page 45. The Index Palette has been renamed the Index Tools Palette and while still

functionally the same, now has an improved interface and the ability to preview index entries.

You will notice that the LinkUP! Menu itself has undergone a complete rebuild into what we feel is a far more logical structure with items grouped in a way that fit in better with the new User Mode Feature. A number of the menu items have been moved onto the palettes. Most of these changes are simple menu to palette changes with the only real exception to this being the **Import Field Names** option. Import field names is now part of a larger process, that of creating a View. If you select **<New View>** from the Data Viewer Palette, after selecting the Data Source you are presented with the Import Field Names Dialogue screen. Here you can select which fields to import and if they require a prefix to be automatically applied. You will notice that there is far more control here than in previous versions, i.e. you can actually select which field names to import rather than importing the whole lot.

The following is a brief summary of the menu to palette moves;

LinkUP! 6 Menu Item	New Name	New Location
Define New Field	Define New Field	Markup Tools Palette
Mark Key field	Mark Key field	Markup Tools Palette
Mark Field	Mark Field	Markup Tools Palette
Copy Markups	Copy Markup	Markup Tools Palette
Paste Markups	Paste Markups	Markup Tools Palette
Clear Markups	Clear Markups	Markup Tools Palette
Add Index Entry	Add Index Entry	Index Tools Palette
Add TOC Entry	Add TOC Entry	Index Tools Palette
Add Header/Footer Markup	Add Header/Footer Markup	Markup Tools Palette
Next Overset Box	Next Overset Box	Data Viewer Palette
Import Field Names	Import Field Names	Data Viewer Palette
Update Quark from Database	Update Quark	Data Viewer Palette
Update Database from Quark	Export	Data Viewer Palette

# INSTALLATION GUIDE

Whether you're installing LinkUP! for the first time or upgrading to the latest version, it only takes a few minutes to get your computer up and running with LinkUP!

Inside your LinkUP! package you should have received the following:

- 1 x LinkUP! CD-ROM
- 1 x Manual
- 1 x Dongle (Security Key)
- 1 x Registration Card (Please complete and return)

If any of these components are missing, please contact your supplier immediately.

## What's on the CD-ROM?

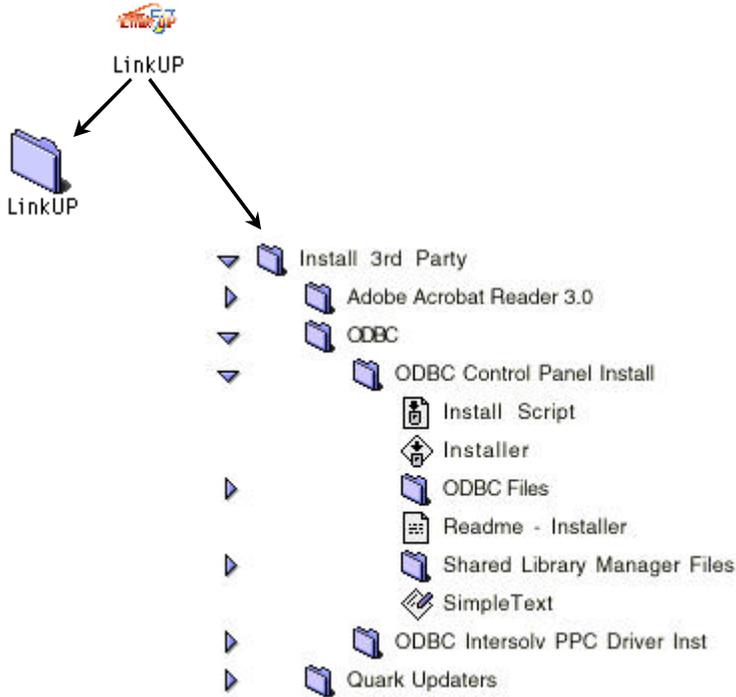
The CD-ROM contains the following components;

- The LinkUP! 7 LinkDB XTension for working with ASCII data files
- The **LinkUP! 7 68k and PPC XTensions** for working with ODBC Data Sources
- LinkDB 68k ODBC Driver
- LinkDB Utility
- The LinkIndex Utility
- Documentation
- ODBC Support Files
- Adobe Acrobat Reader
- INTERSOLV DataDirect v3 Evaluation ODBC Driver Set
- QuarkXPress™ 3.32v5 Updater (UK and International English)
- QuarkXPress™ 4.02 Updater (UK and International English)

If any of these components are missing, please contact MC Research or your local distributor.

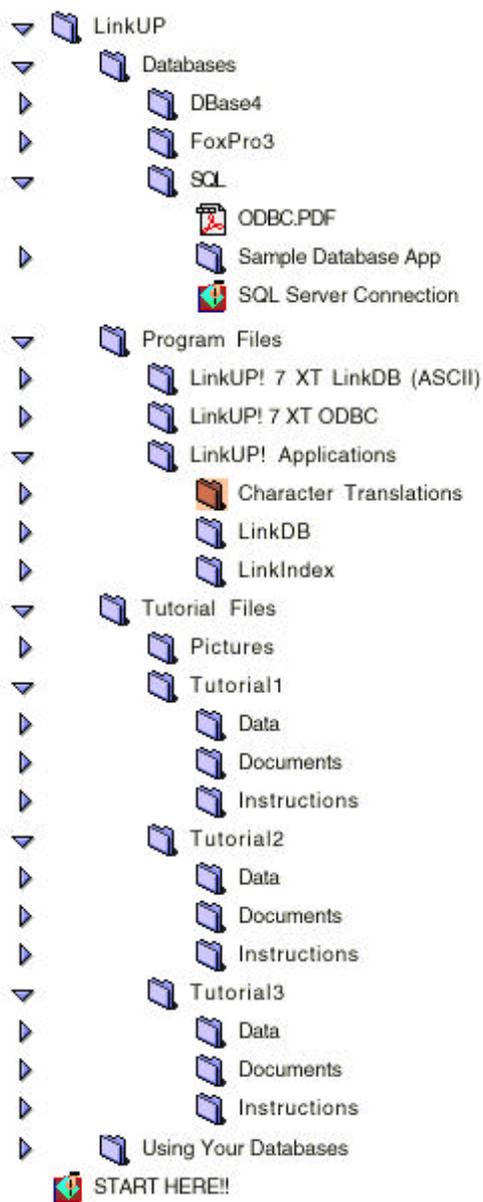
The CD-ROM has been split into two main categories, 3<sup>rd</sup> Party and LinkUP!, this is therefore a logical dividing line for the installation guide for these components. The following is an overview of the contents of each of these folders to enable you to get a feel for the contents and where everything is located.

The following the outline structure of the Install 3<sup>rd</sup> Party Folder on the CD-ROM



**INSTALLATION**

The following the outline structure of the LinkUP! Folder on the CD-ROM



# Minimum Requirements

## LINKUP! 68K XTENSION FOR MACINTOSH

- QuarkXPress 3.32r5 or 4.03
- System 7.5
- 32MB RAM with at least 8MB assigned to QuarkXPress

## LINKUP! PPC XTENSION FOR MACINTOSH

- QuarkXPress 3.32r5 or 4.03
- System 7.6 or 8.0
- 32MB RAM with at least 10MB assigned to QuarkXPress

Updaters for both QuarkXPress 3.32v5 and version 4.02 have been included on the CD in the 3rd Party Folder. Although LinkUP! will run without both of these updates, there are several known bugs in the non-updated versions of XPress that will adversely effect LinkUP! 7s performance and could compromise your document's integrity. We would therefore strongly recommend installing the appropriate Updater.

MC Research maintain an up-to-date document that lists the latest recommended hardware and software that LinkUP! has been tested with. This document is available from MC Research and on the MC Research Web site.

## What do you Install?

When you install LinkUP! on a computer alongside a copy of XPress the main LinkUP! program effectively becomes part of your copy of XPress.

A new menu, 'LinkUP' appears at the right-hand end of your QuarkXPress menubar as below.



The installation of LinkUP! varies significantly dependant on exactly what you want to do. By answering this simple questions you will be able to move straight to the installation path designed for your situation.

## **What do you want to do?**

1. I want to connect QuarkXPress to an ASCII text file - Then goto page 23
2. I want to connect QuarkXPress to an ODBC compliant Database - Then goto page 26

# Installation - if you would like to work with an ASCII text file

If you would like to use an ASCII data source with QuarkXPress you will require the following components from the CD-ROM;

- LinkUP! 7 LinkDB XTension
- LinkDB Utility
- LinkIndex Utility
- Tutorial Files

For this situation the installation of LinkUP! is very straight forward. The first thing you need to do is copy these components from the LinkUP! CD-ROM to your hard disk.

1. Insert the CD-ROM into the disk drive
2. Copy the **LinkUP Folder** into a location of your choice on your computer

## Installing the XTension

With all the files now installed onto your hard disk it is now necessary to copy the LinkUP! XTension into the **QuarkXPress/XTensions** folder. This will cause LinkUP! to be automatically loaded when QuarkXPress is loaded.

3. Locate the **QuarkXPress/XTensions** folder on your hard disk.
4. Locate the **LinkUP! 7 LinkDB** file from within the **LinkUP/Program Files/LinkUP! 7 XT LinkDB (ASCII)** folder you have just copied to your hard disk.
5. Copy the **LinkUP! 7 LinkDB** file into the **QuarkXPress/XTensions** folder

You have now installed the LinkUP! XTension on your computer provided you followed the instructions above correctly. To verify that the LinkUP! XTension is correctly installed where QuarkXPress will recognise it, you should now try and load XPress in the normal manner. If

everything has installed correctly, then you will see an alert similar to the one below, as soon as XPress loads.



Don't worry this message is perfectly correct. We will install the dongle in the next section.

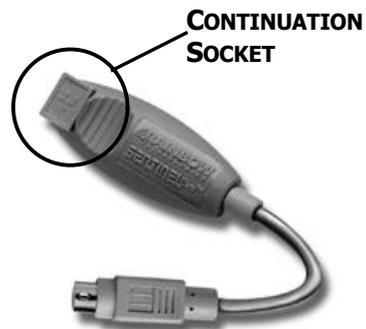
If this message does not appear

1. Check that you have copied the LinkUP! XTension into the correct folder and not a folder that contains an alias to Quark
2. Also check that the XTension actually copied across correctly. You can do this by using the 'Apple I' command whilst clicked onto the XTension file . Note down the size of the file on the CD-ROM and compare it to the size of the XTension copied into the QuarkXPress/XTensions folder. If these are different, try copying the XTension again
3. Call MC Research or your reseller for assistance

## Installing the Dongle

The dongle supplied with LinkUP!, has what is known as a flying lead that connects into an ADB connector on your Mac. A continuation socket is provided, so that you can still use whatever is currently plugged into your ADB connector by simply reconnecting it into the dongle continuation socket.

NB. Connecting devices to the ADB port when your Mac is not shutdown risks damaging both the port and the devices.



1. First shut down your computer.
2. Locate the ADB socket (one on your computer keyboard is recommended, such as the one that the mouse is connected to).
3. Unplug the device currently attached to the ADB port.
4. Plug the Dongle flying lead into the now vacant ADB port.
5. Reattach any devices unplugged during stage 3 into the Continuation Socket on the LinkUP! dongle. The device will now work in the same way as it did previously.
6. Restart your computer.

If the system fails to start, unplug the dongle, verify operation without the dongle and then try the connection process again.

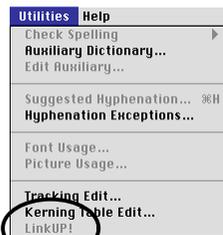
## Confirming the Installation

At this stage, the installation of the LinkUP! XTension should be complete. Start up your QuarkXPress application again, and you might observe a LinkUP! icon scrolling across the initial XPress sign-on screen. This is fast so you will have to watch carefully.

Provided the dongle is installed correctly, you will not see the error message that was displayed in your first test.

If you click on the Utilities menu in QuarkXPress, it should now contain an entry for LinkUP! as below;

If you do not have LinkUP! listed on the QuarkXPress Utilities menu either repeat the above process again or see the troubleshooting section at the end of this User Guide.



# Installation - if you would like to work with an ODBC Compliant Data Source

If you would like to connect QuarkXPress to an ODBC compliant Data Source you will require the following components from the CD-ROM;

- LinkUP! 7 68k or PPC XTension
- LinkIndex Utility
- Tutorial Files

The first thing you need to do is copy these components from the LinkUP! CD-ROM to your hard disk.

1. Insert the CD-ROM into the disk drive
2. Copy the **LinkUP Folder** into a location of your choice on your computer

## Installing the XTension

With all the files now installed onto your hard disk it is now necessary to copy the LinkUP! XTension into the QuarkXPress folder. This will cause LinkUP! to be automatically loaded when QuarkXPress is loaded. You have two options of which LinkUP! 7 XTension to install, either the 68k version or the PPC version. The version you decide to use does not actually depend on the Macintosh you are using but on the ODBC drivers you need to use. The functionality is identical. If the ODBC driver for your database is a 68k driver then you will need the **LinkUP! 7 68k** XTension. If on the other hand your ODBC driver is a PPC ODBC driver for your Mac, then you will need the **LinkUP! 7 PPC** XTension.

3. Locate the **QuarkXPress/XTensions** folder on your hard disk.
4. Locate the **LinkUP! 7 68k or PPC** file from within the **LinkUP/Program Files/LinkUP! 7 XT ODBC** folder you have just copied to your hard disk.
5. Copy the **LinkUP! 7 68k or PPC** file into the **QuarkXPress/XTensions** folder

You have now installed the LinkUP! XTension on your computer provided you followed the instructions above correctly. To verify that the LinkUP! XTension is correctly installed where QuarkXPress will recognise it, you should now try and load XPress in the normal manner. If everything has installed correctly, then you will see an alert similar to the one below, as soon as XPress loads.



Don't worry this message is perfectly correct. We will install the dongle in the next section.

If this message does not appear

1. Check that you have copied the LinkUP! XTension into the correct folder and not a folder that contains an alias to Quark
2. Also check that the XTension actually copied across correctly. You can do this by using the 'Apple I' command whilst clicked onto the XTension file . Note down the size of the file on the CD-ROM and compare it to the size of the XTension copied into the QuarkXPress/XTensions folder. If these are different, try copying the XTension again
3. Call MC Research or your reseller for assistance

## Installing the Dongle

The dongle supplied with LinkUP!, has what is known as a flying lead that connects into an ADB connector on your Mac. A continuation socket is provided, so that you can still use whatever is currently plugged into your ADB connector by simply reconnecting it into the dongle continuation socket.



NB. Connecting devices to the ADB port when your Mac is not shutdown risks damaging both the port and the devices.

1. First shut down your computer.
2. Locate the ADB socket (one on your computer keyboard is recommended, such as the one that the mouse is connected to).
3. Unplug the device currently attached to the ADB port.
4. Plug the Dongle flying lead into the now vacant ADB port.
5. Reattach any devices unplugged during stage 3 into the Continuation Socket on the LinkUP! dongle. The device will now work in the same way as it did previously.
6. Restart your computer.

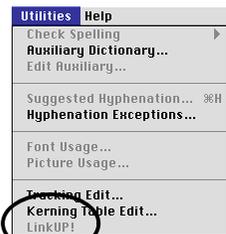
If the system fails to start, shut down your computer, unplug the dongle, verify operation without the dongle and then try the connection process again. If the computer comes up with the same error dialogue you achieved above, try the connection process again.

## Confirming the Installation

At this stage, the installation of the LinkUP! XTension should be complete. Start up your QuarkXPress application again, and you might observe a LinkUP! icon scrolling across the initial XPress sign-on screen. This is fast so you will have to watch carefully.

Provided the dongle is installed correctly, you will not see the error message that was displayed in your first test.

If you click on the Utilities menu in QuarkXPress, it should now contain an entry for LinkUP! as below;



If you do not have LinkUP! listed on the QuarkXPress Utilities menu either repeat the above process again or see the troubleshooting section at the end of this User Guide.

As you would like to use LinkUP! connected to an ODBC compliant database, you also need to install and configure various ODBC components on your computer.

## Using LinkUP! and ODBC

### OVERVIEW

In order to use LinkUP! with ODBC Compliant databases you will need to configure your system to work with ODBC. ODBC (Open Database Connectivity) is a universal standard that allows a client application to exchange data with any ODBC compliant database. In this case, LinkUP! turns QuarkXPress into the ODBC client application, allowing you to interact with a wide range of databases.

There are several components that are required to successfully establish a connection between the client application and the server:

#### On your Mac:

1	The Client Application	(QuarkXPress running with LinkUP!)
2	An ODBC Manager Environment	
3	An ODBC Driver	(Specific to the database that you are connecting to)
4	Database specific support applications	(Optional, and rarely required)

#### Either on your Mac, a network accessible Mac or a database server:

5	The Database Server/File and support applications	(Your Database)
---	---	-----------------

Under Mac OS 7 & 8 there are two different versions of the ODBC environment; one that uses PowerPC native code and one that uses 68K native code. It is possible to run both of these environments on a PowerPC Mac, but you can only run the 68K environment on any pre-PowerPC system. This corresponds to the PowerPC and 68K versions of QuarkXPress. If you are running on a PowerPC system, you should be using the PowerPC version of

QuarkXPress, although it is possible to run the 68K version. On earlier systems you will only be able to run the 68K version.

ODBC Drivers are also required to provide access to specific databases. Whilst only one manager controls the whole ODBC environment, it may be used to supervise connections to many different database types. Each of these database types will require its own ODBC driver. For example, you might have a driver to link to Microsoft SQL Server across a network, and another driver to link to dBase files that are stored on a network hard drive. These two options show the flexibility of the ODBC interface – in one case, a connection is established to a database server, whereas in the other the connection is simply to a static file which does not require a supporting application to be running.

In a very few cases, an ODBC driver can require additional supporting applications to be running on your Mac in order to gain access to a database server. These will almost always be setup by the driver at the point of installation.

Note that an ODBC driver will be targeted to work with either a 68K or a PowerPC ODBC environment, but not both. It is important to identify which environment a driver supports, as this dictates the version of LinkUP! that you need to install (both 68K and PowerPC versions are included as standard with the demo and the release versions).

The final element that you need for your connection is, of course, the database itself. Remember that you may also need to provide network access from your Mac to the database in order to be able to exchange information with it.

## **OBTAINING THE COMPONENTS**

The LinkUP! CD-ROM contains all the components necessary for parts 1 and 2 of the connection. The LinkUP! XTension provides the functionality to enable QuarkXPress to work with ODBC connections, and you will find Apple's standard installer which will load both 68K and PPC ODBC manager environments.

You will also find an ODBC driver that you can use to access files that have been processed by LinkDB – MC Research's ASCII file pre-processing utility. This allows you to view and interact dynamically with ASCII data from within QuarkXPress. This ODBC driver is contained both within the LinkDB Folder on your LinkUP! CD-ROM.

Intersolv produce a 30 day evaluation version of their standard PowerPC ODBC driver set. This will provide access to the following databases:

- Sybase SQL Server
- Dbase III and IV
- FoxPro Version 2 and 3
- Oracle 7 or later
- Text

Note that the Oracle driver also requires you to be running SQL\*Net on your Mac.

This 30 day evaluation pack is also included on the LinkUP! CD-ROM, therefore providing you with the 3rd component in the above list as long as the ODBC compliant database you wish to connect to is one of the ones listed above. Intersolv supports a wider range of database platforms than the driver set which is included in their demo so check out their web site for details. ([www.intersolv.com](http://www.intersolv.com))

If your database environment does not fall into any of these categories, then you can either export your data to an ASCII file for use with the LinkDB driver, or you can obtain drivers from a third party supplier.

If you are having difficulty obtaining a driver for your own usage, then contact our customer support services, who will be happy to assist where they can. We maintain a database of supported platforms and drivers. Please also check our Website, as we will be publishing a list of tested connections. We would also be grateful if you could let us know of any drivers that you use, so that we can make this information available to other LinkUP! users.

You may wish to note that if you have installed a copy of Microsoft Office on your system, then you will already have drivers installed for accessing certain Microsoft products.

## **CHECKING WHAT'S THERE ALREADY**

The simplest way to check if you already have any form of ODBC support on your Mac is to look in the Control Panels, which are accessed off the Finder's Apple Menu. If you see either an ODBC Setup or ODBC Setup PPC listed, then you have either the 68K or the PowerPC environment already present. Note that it is possible to have both installed at the same time on a PowerPC system.

If you don't have either of these options listed, then the first step is to install the ODBC environment – skip to the *Setting up an ODBC* environment on page 33 section of this document.

If you do have either setup option present, then you can easily check what drivers are already available. Open up the ODBC Setup (or ODBC Setup PPC) control panel and you will be given a list of configurations that are already present on your system. If you now press the Add button, you will be given a list of the drivers that are available. If you see the driver that you need, then you are almost ready to get going.

Note that the ODBC environment has to match the driver that you are going to use. In other words if you have a 68K driver then you need the 68K environment, and similarly for a PPC driver you will need a PPC environment.

## SETTING UP AN ODBC ENVIRONMENT

If you either have no ODBC environment set, or you do not have the correct one, then you will have to run the ODBC installer available on the LinkUP! CD-ROM. This installer will install the standard Apple Control Panels and support environments for both 68k and PPC ODBC, if required. It is located in the **Install 3<sup>rd</sup> Party/ODBC** folder.

**Warning:** If you have Version 2 drivers from Intersolv already installed on your system, then you will already have an ODBC Setup control panel available. In this case **DO NOT** install the standard Apple ODBC 68K environment as this is **INCOMPATIBLE** with the Intersolv Version 2 environment. Please contact our customer support desk for further information on how to use the supplied drivers. Note that you **CAN** use the PowerPC ODBC environment from either Apple or Intersolv in conjunction with the Intersolv Version 2 drivers, as these are 68K only.

Optionally, you may also install the Intersolv 30 day evaluation drivers, which also install a complete ODBC environment for PowerPC only. Please read the terms that are enclosed with the installer before loading them onto your system.

## INSTALLING DRIVERS

If you wish to use the LinkDB ODBC driver provided on LinkUP! CD-ROM, then you need to copy the component driver files found inside **LinkDB/ODBC Drivers/Drop Content onto System Folder** that you will have copied from the CD-ROM, to your current **System/Extensions Folder**. They will immediately be available for configuration with a 68k ODBC Setup Control panel.

For more information on ODBC and setting up and configuring ODBC Data Sources see *ODBC* on page 91.

# CONCEPTS

In entering into the realm of data publishing there are many new concepts that need to be grasped if you intend to utilise the full potential of LinkUP! We would suggest that it is essential that you both read and understand this section of the manual before you start to use LinkUP! It may take some time for you to familiarise yourself with this whole new language and way of working but it will make the whole process that much easier.

## What is LinkUP!

LinkUP! is a program that works with QuarkXPress to extend the functionality of XPress. It is a QuarkXPress XTension that has been developed over five years by database publishing experts MC Research. It is used to connect your copy of QuarkXPress with the information held in almost any database file that can be accessed from your computer.

LinkUP! provides a suite of tools that allow you to work with information held in database files without leaving XPress - in effect, LinkUP! builds a bridge between your database files and QuarkXPress. It turns QuarkXPress into a powerful system for creating catalogues, directories, price lists and other publications - text and picture information appearing on the pages, is imported automatically into XPress from data that is held in one or more database files.

## Why is LinkUP! useful?

Using LinkUP! allows you to produce your final XPress pages much later in the production cycle. Because it only takes a few minutes to import information from your database into XPress pages created with LinkUP!, the final version of your document can be produced just before it needs to go to film or to plate.

As data isn't re-keyed, you can be certain that the information that appears on your pages is identical to the information held in your database files. So as long as the information that is

held in the database files you use to generate a document is accurate, correct and up-to-date, then there is no need to proof the pages of your document for accuracy of content more than once in its life.

Using LinkUP! means that producing updates to even the most complex documents can become an almost trivial task. For instance, if the layout and overall content of a publication doesn't need to change but all the prices need revising, it will take literally minutes to import the latest prices from your database file.

## What is a Database?

*(If you already know what a database is, skip this section)*

Most of us have, or are familiar with, a Filofax or card filing system such as a Rolodex. These are paper-based databases – they provide a structure in which we store information in a logical format for easy recall. A Filofax address page provides a good model for any database. Each address has an alphabetic location and a clearly defined place in which to write name, address and telephone number. In database terminology each of these is a 'field'. When we want to find a person's details we recall the first letter of their name, locate the appropriate alphabetic tab and then 'search' that 'subset' of the database for their name.

However, the Filofax 'database' has a major limitation. Even if you took the trouble to alphabetise all your contacts' names when you first entered them, the system breaks when you want to record a new contact name. Although they can be entered into the right alphabetic section A, B, C etc., they will inevitably be entered at the end of the section and will be out of order.

If we were entering the same information in an electronic form – a database or spreadsheet - we would be able to create a more logical structure of fields and then to sort the data and search on any field sequence. For example, if we were creating a database in FileMaker 3 or ClarisWorks we might create the following fields in the example below. Such a structure would enable the data to be stored in any order. For example, we could sort alphabetically on surname and then first name to give a logical Filofax type result or search on the contents of any field – perhaps to locate all contacts whose office telephone number started with 0181.

First Name	<input type="text"/>
Middle Name	<input type="text"/>
Surname	<input type="text"/>
Address	<input type="text"/>
Town/City	<input type="text"/>
Postcode	<input type="text"/>
Country	<input type="text"/>
Direct Line	<input type="text"/>
Home Telephone	<input type="text"/>
Office Telephone	<input type="text"/>
Fax	<input type="text"/>
Mobile	<input type="text"/>
Email	<input type="text"/>
Notes	<input type="text"/>

This functionality is provided by a DBMS – database management system and it is this that turns a collection of data into a useful and productive resource.

A database system consists of four major components:

**data** – within a company or organisation, data (information) is often stored centrally in a single database and that data can be shared by multiple users. Shared databases often allow different users to access and utilise that data in different ways depending on their job function. Although users are sharing the same data how they view and use the data could be very different.

**hardware** – the database programme physically resides on one or more hard discs of the computer system and users interact with it via software tools.

**software** – provides the interface between the data and the user. This interface is known as the DBMS (database management system) or database manager (DB manager) and provides tools for database administration, data entry, data modification and retrieval.

**users** – the database administrator is a ‘super-user’ with access to and responsibility for all aspects of the database system and its administration; end-user(s) interact with the database either as a ‘stand-alone’ user or as a ‘client’ on a network. The end-user may have complete or limited access to the database dependant on the ‘access privileges’ assigned by the database administrator. In a telephone sales office for example, the telesales staff

may have access to contact information, stock inventories and purchase records but not the accounting records for the company.

## Databases types

Two types of database are likely to be encountered by the potential database publisher – flat-file and relational.

### FLAT-FILE

ClarisWorks and FileMaker, up to and including version 3, are typical examples. A flat-file database can usefully be compared to a wordprocessor file in which each sentence is a 'field' separated by a full stop and each 'record', a complete section of data, as a paragraph separated by a carriage return. The inherent limitation of a flat-file database is that each record is a separate entity and has no relationship with any other record. Some databases, such as FileMaker, over-come this limitation to some extent by offering a 'lookup' function which allows data to be transferred between database files.

### RELATIONAL DATABASES

A relational database can be thought of as a series of 'mini-databases' which may be dynamically linked.

In our example of a telesales environment the client contact database would contain name, address and telephone numbers; the stock database would contain stock numbers, description and pricing information; the order database would need to show date of order, date of despatch, name and address, delivery details, products purchased, individual item pricing and generate an invoice. In a relational model if the client advised a change of address at the time of placing an order the telesales staff could input the necessary information in the order screen and this information would automatically be updated in the contact database and invoice database.

# How does LinkUP! work with XPress documents?

In three ways:

1) LinkUP! allows you to build XPress documents from scratch. These documents can be filled with information by automatically importing text and picture information from one or more database files.

2) LinkUP! also allows you to take an existing XPress document and 'mark it up', establishing links to one or more database files so that future editions of that document can be produced by automatically importing text and picture information from one or more database files.

3) And if you don't have a database file that contains the information that you published in your catalogue or directory, then LinkUP! can be used to generate a database file for you from the QuarkXPress pages you used to produce a printed document.

There are also variations on each of these 3 categories especially the building of XPress documents and some publications may require combinations of all three types of use of LinkUP!

## Administrator / Operator

There are two distinct elements to using LinkUP!, the everyday users operations such as building documents, updating QuarkXPress elements etc. and the one-off administrator functions that include the more complex tasks of setting up connections to the database files and specifying how these will behave. LinkUP! now incorporates a **User Mode** facility so that operators only have available the tools necessary for actually using LinkUP! while administrators have the tools for configuring and setting up LinkUP!

In many cases the operator and the administrator will be exactly the same person however in larger site installations they are more commonly two separate groups of people.

# What is ODBC?

ODBC - or Open DataBase Connectivity - is a software standard that allows two-way communication between files generated by almost all database software applications as well as other 'ODBC Compliant' programs. It's a common interface protocol if you like.

ODBC provides the software interface and the translation tools that allow a LinkUP! enabled copy of QuarkXPress to open a live link with a database file. LinkUP! uses SQL (Standard Query Language) to communicate with the contents of an open file. Effectively, your LinkUP! enabled copy of XPress becomes the application that opens and uses the database file you connect to.

LinkUP! uses the facilities provided by ODBC and SQL to allow you to examine, sort, import, manipulate and use the data held in the database file from within XPress. It also allows you to write information from an XPress document back to a database file.

Database software packages - for instance FileMaker Pro, Dbase, or Access - were not generally designed or written so that they would work directly with other specific applications packages such as spreadsheets or QuarkXPress. What they did do however is put in place the 'hooks' that allow other applications to access their data using ODBC.

## LinkDB

LinkDB, a separate application, allows you to take almost any ASCII text file generated as a report by a database or spreadsheet program and process that file so that it can be used by LinkUP! as if it were an ODBC-compliant file. A copy of the LinkDB Utility is included on the LinkUP! CD-ROM and should have been copied to your hard disk during installation (See separate LinkDB User Manual for Details on using the LinkDB Utility).

# Working with Simple Databases and LinkUP!

We suggest you spend a while with this section of the manual whether or not you are already very familiar with databases and how they work. Terms and concepts that are referred to throughout this manual will be defined and explained here.

A full understanding of the concepts of KEY FIELD and LinkUP! Reference Numbers (LRNs) is essential if you are to understand LinkUP!

In its simplest form, a database file can be represented as a table containing information. Each cell within the table contains a single item of information or data.

<i>Part Number</i>	<i>Name</i>	<i>Description</i>	<i>Price</i>
9087654	Radio	AM/FM	25.77
980343	Cassette	dual-deck	199.00
89234	Television	widescreen	599.00
892342	Alarm clock	travel/radio	12.25

## Row, or RECORD

A horizontal row in the database table contains all the data that is associated with a single entity - a product, perhaps, in the case of a database file that will be used with LinkUP! to create a price list.

## Column or FIELD

A vertical column in the database table contains data of a particular type or category - prices, perhaps, in the case of a database file that will be used with LinkUP! to create a price list.

## Field Names

The first row in the database table does not contain data associated with a single entity; instead each cell in the row is used to store a label that is used to refer to the type or category of the data or information that is stored in the cells below. In effect, the cell at the top of each vertical column is used to store a 'header' or label that can be used to identify that column. This label is called the Field Name.

In our illustration, PART NO, NAME, DESCRIPTION and PRICE are the four FIELD NAMES. LinkUP! uses FIELD NAMES to refer to the contents of fields and the Key Field to refer to individual records within the database table.

## Key Field

LinkUP! needs to be able to use one of the columns - or FIELDS - in a database table as the KEY FIELD. The content of each field in the column that LinkUP! uses as the KEY FIELD must be unique in the database table, because LinkUP! uses the KEY FIELD to identify individual records in the database table.

LinkUP! must be able to identify each record in a database table by looking at a single field in that database table where it will find data that is unique to each record. The diagram below shows the Key Field as part of our example database.

<i>Part Number</i>	<i>Name</i>	<i>Description</i>	<i>Price</i>
9087654	Radio	AM/FM	25.77
980343	Casette	dual-deck	199.00
89234	Television	widescreen	599.00
892342	Alarm clock	travel/radio	12.25

## Assigning a Key Field

You can assign one of the existing columns in a database table to act as the Key Field for that data source, in which case you should chose a column that will contain data that is unique to each record. If the database contains a field that holds part number information,

then it would probably be a good choice as the Key Field. The column you choose doesn't have to be the first column in the database table.

However, you may find you are working with a database table that does not contain a field that can be used in this way. If you are working with an ASCII data source LinkUP! can create a new field at the start of the database table that you are working with and populate that field with sequential numbers. This new field will be used as the KEY FIELD by LinkUP! when it needs to identify individual records from the table.

## **Hiding Key field data on the page**

The content of the field that LinkUP! uses as a Key Field - whether it was an existing field in the data source you are using or was added by LinkUP! as a Key Field - does not have to appear on the page. When you markup an XPress document to tell LinkUP! where on the page to place the data from each record in the data source you are using, you can choose to 'hide' the content of the Key Field. Further details can be found in the section '*hidden*' Key Field - data in Key Field does not appear on the page on page 126

## **The LinkUP! Record Number (LRN)**

The Key field allows LinkUP! to identify which record from the database file it is working with. The easiest way to think about the LinkUP! Record Number is to relate it to shipping parcels. When you send a parcel through a shipping company you are usually given a tracking number that is unique to that parcel or group of parcels. This enables the shipping company, and indirectly you, to keep track of where your parcel is at any particular time between leaving you and arriving at its destination. The LinkUP! Record Number (LRN for short) is in essence the unique tracking number that LinkUP! assigns to groups of items on your QuarkXPress page. In the same way that the shipping company keeps track of your packages with a tracking number, LinkUP! can keep track of all of the marked up elements and where they are at anyone time. Just as importantly it allows LinkUP! to know which database records the elements on the page belong to.

LinkUP! uses a sequential numbering system to track the way that the content of a database record is to be placed on the pages of your document. The system works with numeric data

that is dynamically assigned by LinkUP! as a document is prepared - these numbers are the LinkUP! Record Numbers.

This tracking process is automated: you do not have to assign LinkUP! Record Numbers, although you can assign, change and edit LinkUP! Record Numbers when you need to perform certain advanced functions. It is important to understand that LinkUP! Record Numbers do not appear in the database table that you use as a data source: LinkUP! Record Numbers are not read from or written to the file that holds the database information, but travel with your XPress document as part of LinkUP!'s system for managing the content of that document. A set of LinkUP! Record Numbers is specific to a QuarkXPress document.

All the elements on the QuarkXPress page that relate to a single record in the database will have the same LinkUP! Record Number. They are in essence a group of elements. Imagine a clothing catalogue, and a particular leather jacket within that catalogue. The leather jacket's price, picture, heading, description etc., in fact anything to do with that leather jacket, all belong to a single record in the database. Hence all of these items will be grouped together with the same LinkUP! Record Number.

LinkUP! Record Numbers allow LinkUP! (and you!) to keep track of what is happening on the pages of your XPress document. All the fields that belong to one logical record from the database table should have the same LRN assigned to them when they appear in XPress elements on your pages.

When you create a set of XPress page elements that relate to the particular fields of a record in a database file, LinkUP! uses the Key Field to identify which record in the database file it is to use for that group of elements. It assigns the next available LinkUP! Record Number (explained above) to that particular implementation of the content of the database record on your pages. All the fields that belong to that particular appearance on the page of the record in the database file have the same LinkUP! Record Number, effectively as a badge that allows LinkUP! to identify them as belonging to that particular set of page elements in which the contents of the database field will appear.

Within a document you may wish to use the content of the same database record more than once - if you do this, LinkUP! identifies the database record you are using by its unique Key Field, and will identify each appearance of the database record on your pages by a separate LinkUP! Record Number.

LinkUP! Record Numbers may not appear to run in strict numeric sequence, especially if you have edited the layout of your document. Don't worry about this. If you create an element on your page to hold the contents of a record in a database file, then LinkUP! assigns the next available LinkUP! Record Number to that incarnation of the record on your page. If you subsequently change your mind and delete all the elements on the page used to hold the fields of that database record, then LinkUP! does not free up the LinkUP! Record Number for use in the future.

There will be occasions when you want to use the same record from the database table more than once on the pages of a single document. Each time you use a record, the collection of XPress page elements that will be used to contain data from a single record will share their own unique LRN. So a unique LRN is used to keep track of each separate usage on the page of the content taken from a single record in the database table

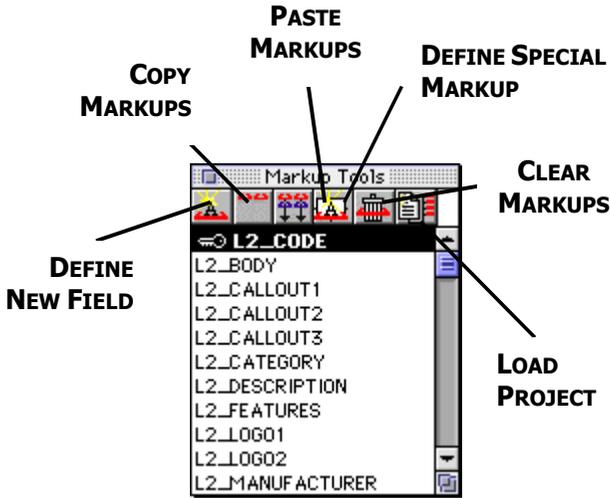
# NAVIGATION

The navigation section provides two sections that will assist you in finding your way around both LinkUP! and this manual. The guide to the four floating palettes explains the functions of all the palette buttons and controls. The command reference section is your one-stop route to navigate your way to specific topics within this user guide.

# Guide to Floating Palettes

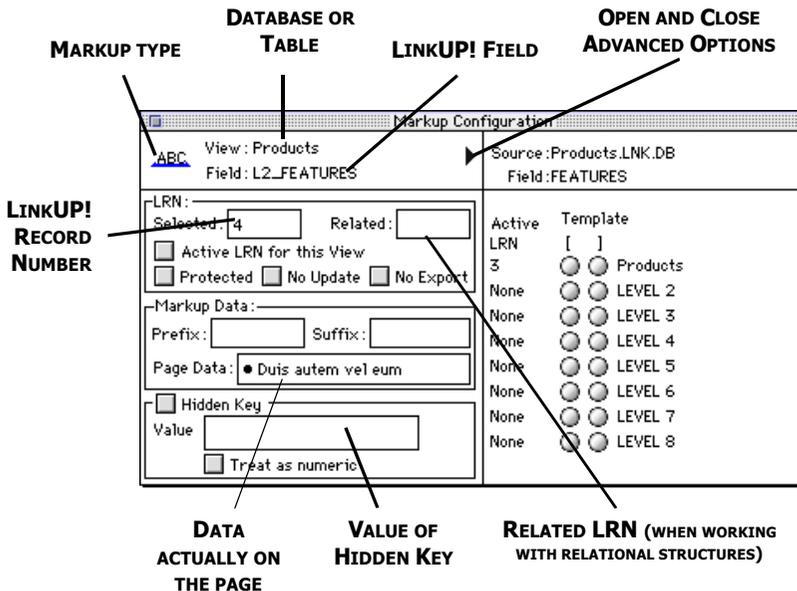
## Markup Tools Palette

This is the main palette you will use when preparing a document for use with LinkUP! It contains a list of all of the available database fields for the open document as well as various buttons that you will need for the process. The following diagram explains the function of each of these.



# Markup Configuration Palette

This is your peek into the LinkUP! Markups. It allows you to see exactly what is hiding behind each of the marked items on the page. You can also use the Configuration Palette for editing your data, ensuring that none of the markups are damaged.



NAVIGATION

The Markup palette is divided into two main sections, the left-hand-side is what's displayed as default. This holds all of the information about the markup currently selected in the document. The right-hand-side, only visible when you click the Open/Close icon at the top right, is where you specify the start and end points for relational levels in a document. This is essential for template building with LinkUP!

## Markup Type Icon

The Markup Type Icon changes dependent on the type of markup currently selected in the document. The four icon types are as follows;

 Text Field: - The piece of text is marked up as a LinkUP! field

 Picture Field: - The Picture is marked up as a LinkUP! picture field

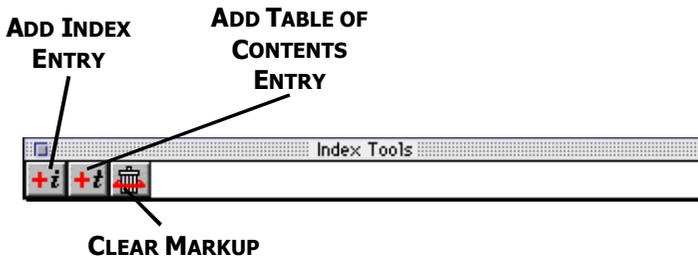
 Key Field: - The piece of text is marked up as a LinkUP! Key Field

 Hidden Key Field: - The piece of text is marked up as a Hidden Key Field

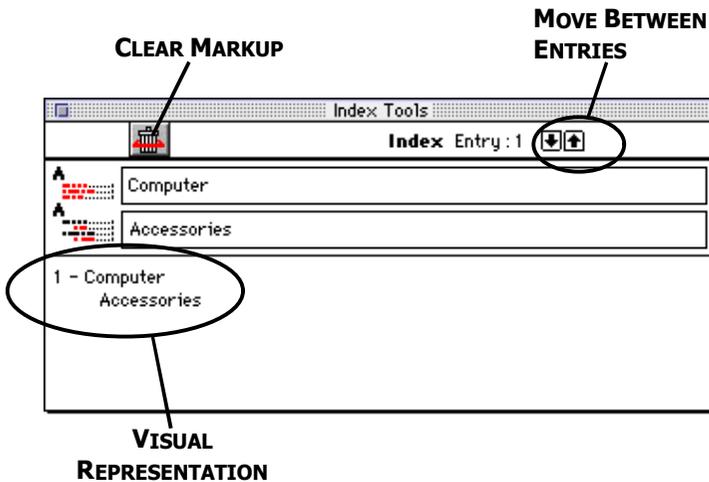
# Index Tools Palette

Use this palette to mark index and table of contents entries in your documents. The palette also gives a visual indication of how the entries will appear in the final index and table of contents.

**State 1.** While not highlighting an Index or TOC entry

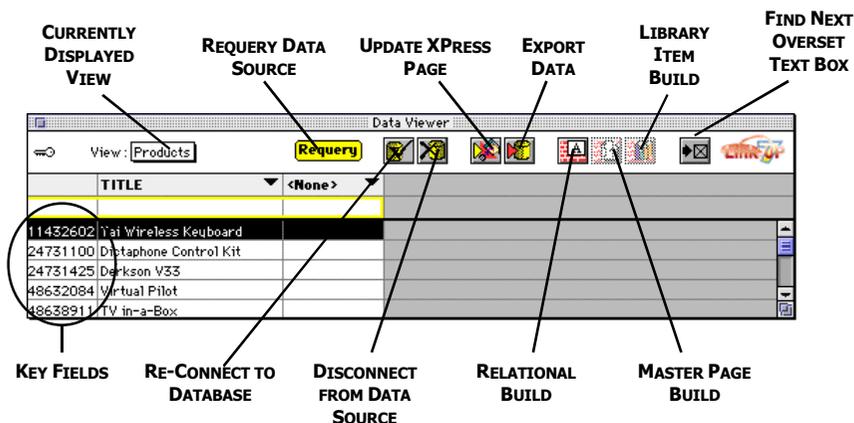


**State 2.** While highlighting an Index or TOC entry



## Data Viewer Palette

This is arguably the most powerful palette to be found within the QuarkXPress environment. It contains all of the action buttons you will need to connect to your database and update and build documents. More importantly than this it lets you view, search and query your database from within QuarkXPress and then assign items onto the page by simply double-clicking your mouse.



The first column in the Data Viewer lists the Key Field for each database record. This is the most important column in the data as it is the field that LinkUP! uses to track which elements belong together. This way LinkUP! can know that a price on one page belongs to a picture on another and that in turn they all relate to a particular database record.

To view your data within the Data Viewer you will have to be connected to a data source. This could be an ODBC data source, in which case you will have had to of set one up within the ODBC Control Panel on your Mac. Alternatively it could be a LinkDB file on your hard disk, depending on which version of the XTension you are running.

You can view different ODBC sources by simply selecting a different source on the View: pull down menu. As default the viewer only displays the Key field in the first column, with the next column set to none. If you click on the down arrow next to none, you can select from a

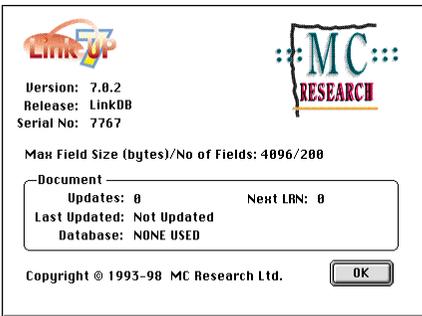
list of available fields in that data source. This is a list of ALL available fields, not just the ones defined or imported into XPress. Select the field you want and automatically the next column is enabled with the word none above it. You can repeat this process for as many fields as you have screen space.

Once you have all the fields you want to see in the Data Viewer, click the requery button and LinkUP! will fill the columns with the data you asked for. This process can be repeated as many times as you like. You must click the requery button each time though. The rows and columns can be resized by dragging the grid lines. If you drag the mouse across several rows or columns and then resize just one of those columns, the Data Viewer will resize all of the selected rows and columns to the new dimensions.

NB. The buttons on the Data Viewer will always be greyed out if you do not have a data source or view defined.

# Command-by-command reference

## About LinkUP!



### PURPOSE:

Displays the about LinkUP! Dialogue Box

The dialogue informs you of the version and release of LinkUP! that you are running. It also tells you what the next LRN that will be used in the document is, and gives details relating to when the document was last updated and the total number of times that it has been updated. This will help whoever is providing support solve your problems more easily.

## Add Index Entry



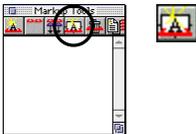
**PURPOSE:**

Turns the currently highlighted text selection into an index entry.

**FURTHER READING:**

Creating Indexes From your QuarkXPress Documents      167

## Add Special Markup



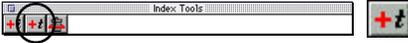
**PURPOSE:**

Turns the currently highlighted text selection into an index entry.

**FURTHER READING:**

Marking Up Header/Footer (Page Reference) Links      119

# Add TOC Entry



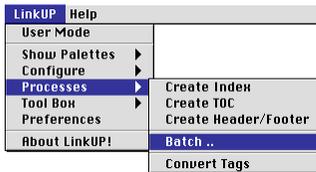
## PURPOSE:

Turns the currently highlighted text selection into a Table of Contents entry.

## FURTHER READING:

Creating Table of Contents (TOC) from your QuarkXPress documents 175

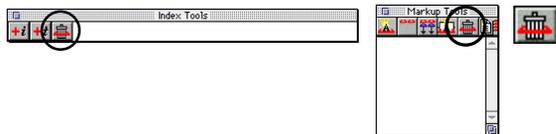
# Batch



## PURPOSE::

To automatically update the contents of multiple QuarkXPress documents from their specified data sources.

## Clear Markups



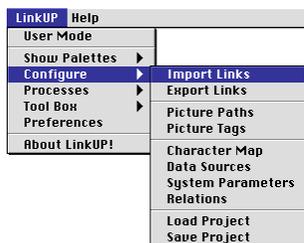
**PURPOSE:**

To remove all LinkUP! markups from a specified section of text.

**FURTHER READING:**

Marking Up 117

## Configure Import Links



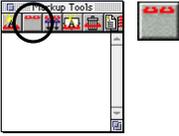
**PURPOSE:**

This option indicates to LinkUP! the relationships between the fields in LinkUP! and the fields in your database.

**FURTHER READING:**

Configuring Import Links 110

## Copy Markups



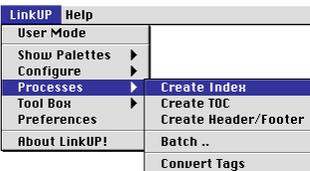
### PURPOSE:

To alleviate the manual marking of tables, LinkUP! allows you to copy the structure in a single line of a table so that it can be applied to the rest of a table.

### FURTHER READING:

Dealing with Tables 129

## Create Index



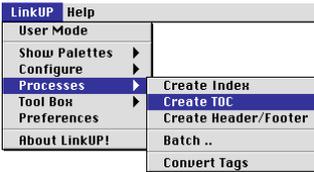
### PURPOSE:

To cause LinkUP! to create an Index Data File

### FURTHER READING:

Creating Indexes From your QuarkXPress Documents 167

## Create TOC



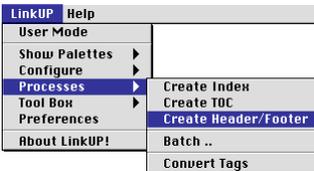
### PURPOSE:

Causes LinkUP! to create a Table of Contents Data File

### FURTHER READING:

Creating Table of Contents (TOC) from your QuarkXPress documents 175

## Create Header/Footer



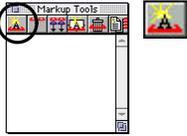
### PURPOSE:

This option inserts information into designated header/footer text boxes.

### FURTHER READING:

Header/Footer 182

## Define New Field



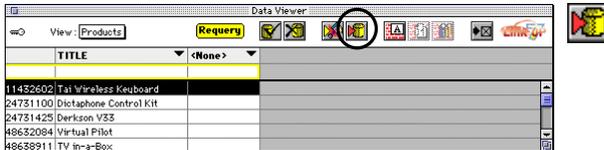
### PURPOSE:

To create a new field that can be used to Markup the current document.

### FURTHER READING:

Defining Fields Manually                      98

## Export



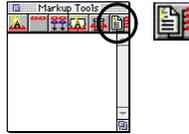
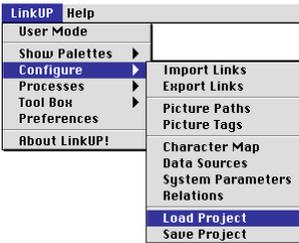
### PURPOSE:

To export all marked items in your document selected for export.

### FURTHER READING:

Exporting                      197

## Load Project



### PURPOSE:

To load a predefined set of field definitions and configurations from an existing project file.

### FURTHER READING:

Projects 159

## Mark Key Field

### PURPOSE:

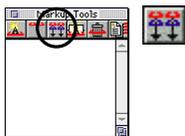
To identify the currently highlighted text as a Key Field.

### FURTHER READING:

Marking Up 117



## Paste Markups



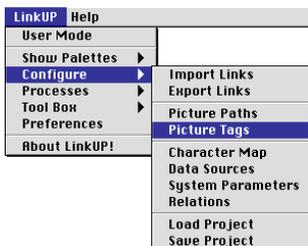
### PURPOSE:

To identify an area of text within your document that requires the same Markup structure that was read using the copy markups option.

### FURTHER READING:

Dealing with Tables 129

## Picture Tags



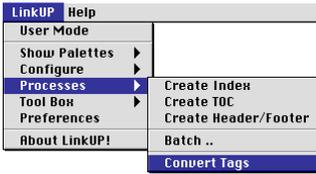
### PURPOSE:

Allows embedded boxes to be automatically inserted at the relevant places in the text stream. Up to nine different styles and sizes of text boxes are supported.

### FURTHER READING

Working with Pictures 165  
 Tagging 203

# Process Markup Tags



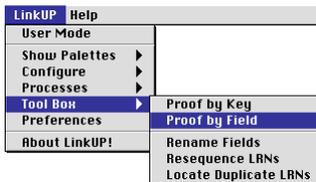
## PURPOSE:

To locate and act upon character sequences that allow an imported text file to be automatically marked up.

## FURTHER READING

Configuring Picture Tags	166
Tagging	203

# Proof by Field



## PURPOSE:

To replace the contents of all marked fields with the name of the field that they are marked with.

## FURTHER READING:

Proofing	191
----------	-----

## Proof by Key

LinkUP	Help
User Mode	
Show Palettes	▶
Configure Processes	▶
Tool Box	▶
Preferences	▶
About LinkUP!	
	Proof by Key
	Proof by Field
	Rename Fields
	Resequence LRNs
	Locate Duplicate LRNs

### PURPOSE:

To replace the contents of all marked fields with the value of the Key Field they relate to.

### FURTHER READING:

Proofing 191

## Resequence LRNs

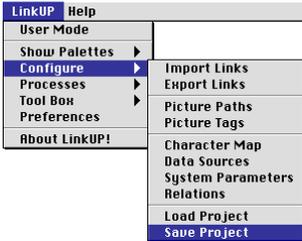
LinkUP	Help
User Mode	
Show Palettes	▶
Configure Processes	▶
Tool Box	▶
Preferences	▶
About LinkUP!	
	Proof by Key
	Proof by Field
	Rename Fields
	Resequence LRNs
	Locate Duplicate LRNs

### PURPOSE:

To alter the order of the LinkUP! Record Numbers within your document.

### FURTHER READING:

# Save Project



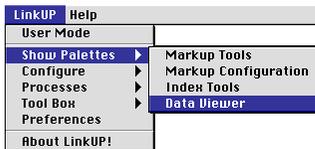
## PURPOSE:

To save the current Configuration parameters to a project file for later retrieval.

## FURTHER READING:

Projects 159

# Show Data Viewer Palette



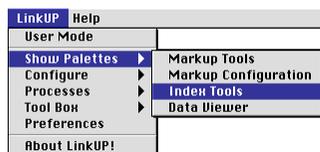
## PURPOSE:

Displays or hides the Data Viewer Palette.

## FURTHER READING

Data Viewer Palette 50

## Show Index Tools Palette



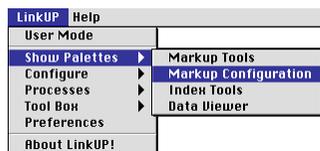
**PURPOSE:**

Displays or hides the Index Tools Palette.

**FURTHER READING:**

Index Tools Palette	49
Creating Indexes From your QuarkXPress Documents	167
Creating Table of Contents (TOC) from your QuarkXPress documents	175

## Show Markup Configuration Palette



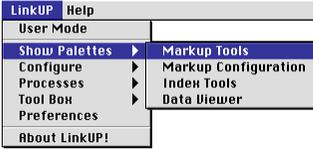
**PURPOSE:**

Displays or hides the Markup Configuration Palette.

**FURTHER READING:**

Markup Configuration Palette	47
Marking Up	117

# Show Markup Tools Palette



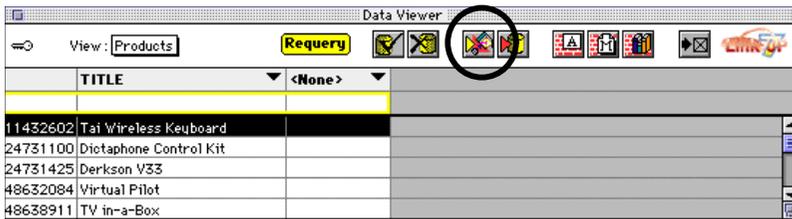
## PURPOSE:

Displays or hides the Markup Tools Palette.

## FURTHER READING:

Markup Tools Palette      46  
Marking Up                    117

# Update QuarkXPress



## PURPOSE:

To instruct LinkUP! to place the latest information from your database into your QuarkXPress publication.

## FURTHER READING:

Updating      162

# TUTORIALS

**You will need the LinkUP! 7 LinkDB XTension installed** to be able to work through these tutorials successfully. Instructions for this are given on page 23. You must make sure that all of the tutorial files are installed on your local hard disk and NOT running off of the CD-ROM, otherwise they will not function correctly.

# Tutorial 1 - Updating a QuarkXPress Document

## Overview

In this tutorial you will see LinkUP! automatically update information directly onto QuarkXPress pages. You will also be able to see that LinkUP! places no restrictions on your page layout. This is extremely powerful as it means that LinkUP! can find and update your prices, part numbers, specifications, pictures etc. and any other XPress element regardless of their location and association in your publication.

The fastest way to see the speed, power and accuracy of LinkUP! is to watch it in action on an actual QuarkXPress page, in this case a catalogue page but I'm sure you can translate this to whatever type of publication you produce. For the purpose of this Tutorial we have prepared an XPress document and the associated database for you in advance.

## Making it Happen

All the files necessary for the successful completion of Tutorial1 are located in the **Tutorial1** Folder that you copied across to your hard disk during the installation. The only exception to this are the pictures, which are located in the **Pictures** Folder.

1. Launch QuarkXPress with the **LinkUP! 7 LinkDB** XTension properly loaded as described on page 23.
2. Locate and open the **Tutorial1** document that you copied across during the installation (it's located in the **Tutorial Files/Tutorial1/Documents** Folder)

If you take a look at the document you will see we have a typical catalogue page containing all of the types of elements you'd expect to see. Pictures, prices, feature lists, call outs, logos, etc.

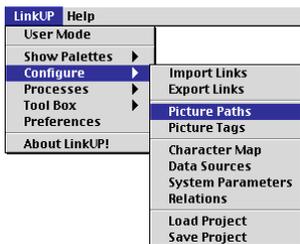
3. Switch LinkUP! on by selecting **LinkUP!** from the **Utilities Menu**.

You will notice that most of the elements on the XPress page now have underlines. These underlines signify that the underlined elements are linked in some way to an item in a database. In other words LinkUP! knows and understands the relevance of each element. These underlined items have been 'marked up'. You can learn how this is achieved in **Tutorial 3 - Preparing a Document for LinkUP!**

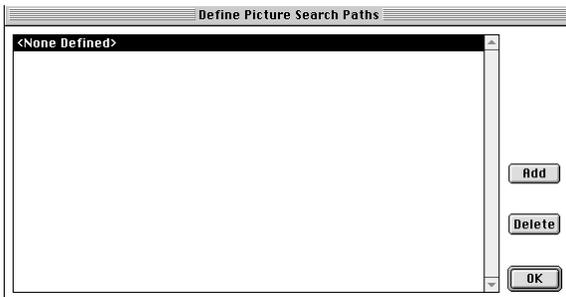
Before we update the page it is necessary to tell LinkUP! where both the pictures and the data you wish to work with are located. LinkUP! works with absolute paths so when you copied the files from the CD-ROM to your hard disk, the paths changed. We'll start with the picture paths.

An unlimited number of picture paths can be set for any one document so you can point LinkUP! to use Low Res. pictures during the design stage and then point it to the High Res. images on your OPI server when the time comes to output the film.

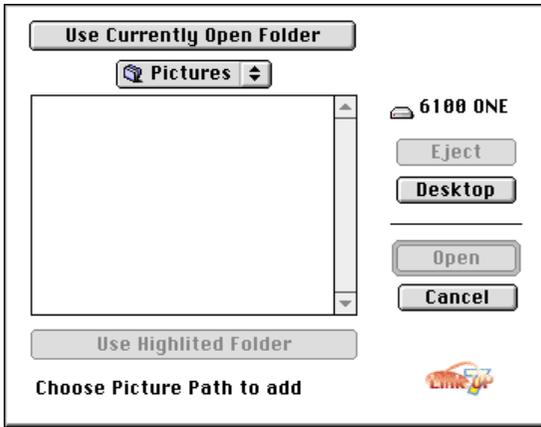
To configure your picture paths;



4. From the LinkUP! Menu, select **Configure Picture Paths**



5. Click the Add button
6. Browse your computer to find the location of the pictures. (They will be located in the **Tutorial Files/Pictures** Folder you copied to your hard disk during the installation)

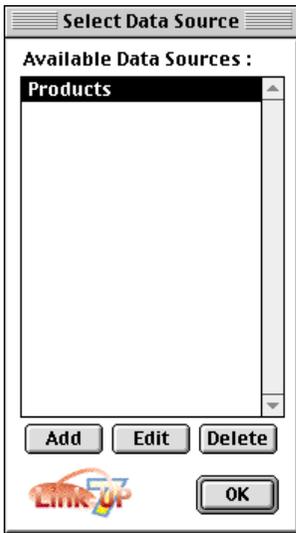


7. Click either the **Use Currently Open Folder** Button if you are inside the **Pictures** folder, otherwise highlight the pictures folder and click the **Use Highlighted Folder** Button. You will see that the new picture path has been added.
8. Click the **OK** button

The next process is to tell LinkUP! where to find the data needed for updating the document. As you are working with LinkDB files it is simply a case of letting LinkUP! know which Folder contains the LinkDB file. LinkUP! comes with a 'learn' feature that gives it the ability to learn all of the necessary information about a data source without you having to type it all in. We'll use this function now.

To select your data source;

9. From the LinkUP! menu select **Configure Data Sources**
10. Select the **Products** Data Source
11. Click the **Edit** Button



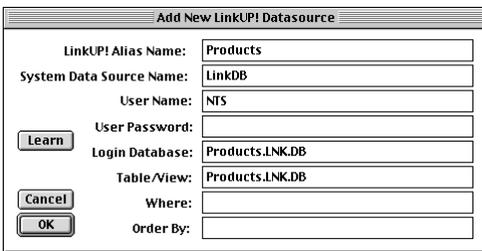
The data for this tutorial is help in the **Tutorial1/Data** Folder

12. Click the **Learn** Button
13. You will now need to locate your **Data** Folder (located in the **Tutorial1/Data** Folder)
14. Click either the **Use Currently Open Folder** Button if you are inside the **Data** folder, otherwise highlight the data folder and click the **Use Highlighted Folder** Button

Providing you have chosen the correct location, you are now presented with a list of LinkDB files located in this folder. In this case there is just the one, Products.LNK.DB

15. Select this and click the **Open** button

LinkUP! has now automatically re-configured itself to the new location of the data source. Your dialogue will look similar to this apart from the Login Database will contain your local path to the data.



16. Click **OK** on the Add New LinkUP! Data Source dialogue

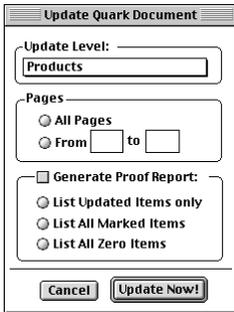
17. Click **OK** on the available Data Sources dialogue to finish

All of the initial configuration is complete. LinkUP! knows where the pictures are stored and also where the Data Source is located. These two actions are one-off actions. You need only perform these two configurations once. You are now ready to update the QuarkXPress document.

18. From the LinkUP! menu select **Show Palettes Data Viewer**

19. Click the Update button on the Data Viewer Palette 

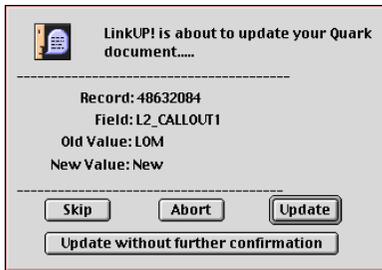
A dialogue will appear as below;



We are about to see LinkUP! use this database to update the QuarkXPress page to reflect the information held in our database for each product on the page. The process is very fast...

20. Click the **Update Now!** Button

LinkUP! will present its **LinkUP! is about to Update** windows, allowing you to observe and confirm each individual update. For now we will just proceed with the process automatically.



21. Click the **Update without further confirmation** button and watch LinkUP! do the hard work!!

Notice how the real prices, pictures and descriptions have been brought onto the page. Notice also some of the more intricate details that LinkUP! has taken care of. The currency symbol in front of the prices, the currency units have been place in superscript

automatically. Even the feature list has had bullets placed in front of each bullet point. All of the styling on the XPress page is maintained unless you specifically ask LinkUP! to do otherwise.

These simple steps you have just performed demonstrate the speed, power and accuracy of LinkUP! in updating QuarkXPress documents from database information. Regardless of the number of updates, pages, or locations of updated information, LinkUP! performs pricing changes, version or zone changes, specification changes or description changes this quickly and easily every time.

That's all that is required for a full database update using LinkUP!. As you have seen, this is an extremely fast process - typically 2000 page elements can be updated in under 2 minutes.

# Tutorial 2 - Template Building

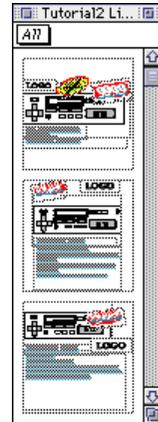
## Overview

In Tutorial 1 you were able to see LinkUP! updating a QuarkXPress document from our supplied database file. Next we will take this a stage further by allowing you to decide which products go where and in what format. We are going to achieve this through a simple building block approach using some pre-marked library items. This will enable you to see just how quickly a page can be assembled using the library and LinkUP! - saving operators countless hours of data input.

## Making it Happen

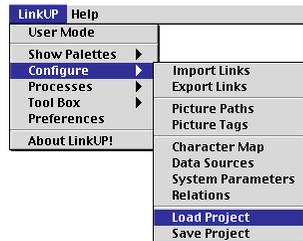
All the files necessary for the successful completion of Tutorial2 are located in the **Tutorial2** Folder that you copied across to your hard disk during the installation. The only exception to this are the pictures that are located in the **Pictures** Folder.

1. Launch QuarkXPress with the **LinkUP! 7 LinkDB XTension** properly loaded as above.
2. Create a new blank QuarkXPress document, A4 or letter size will do.
3. Locate and open the Tutorial2 Library that you copied across during the installation (it's located in the **Tutorial2/Documents** Folder)
4. Switch LinkUP! on by selecting **LinkUP!** from the **Utilities Menu**.
5. Now drag four of the library elements from the **Tutorial2 Library** onto the blank QuarkXPress page you created



What you now have on the page are some marked up items denoted by the fact that they are underlined. Currently LinkUP! knows what each marked item is. It knows which are product pictures, logos, prices, feature lists. LinkUP! also knows which items are grouped together on the page through its LinkUP! Record Number (see *Concepts* on page 34). What LinkUP! doesn't know is what specific products are to fill each of the templates you dragged onto the page. So the next task is to assign products to the templates. (How to prepare your document for use with LinkUP! is contained in Tutorial 3 - Preparing a document for LinkUP!).

6. From the LinkUP! menu select **Configure Load Project**
7. Locate and open the **Tutorial2 Project**

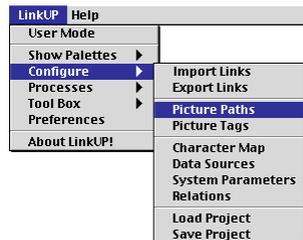


Before we update the page it is necessary to tell LinkUP! where both the pictures and the data you wish to work with are located. LinkUP! works with absolute paths so when you copied the files from the LinkUP! CD-ROM to your hard disk the paths changed. We'll start with the picture paths.

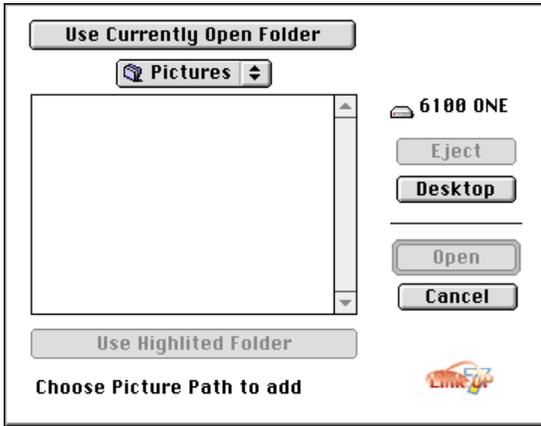
An unlimited number of picture paths can be set for any one document so you can point LinkUP! to use Low Res. pictures during the design stage and then point it to the High Res. images on your OPI server when the time comes to output the film.

To configure your picture paths;

8. From the LinkUP! Menu, select **Configure Picture Paths**
9. Click the **Add** button
10. Browse your computer to find the location of the pictures. (They will be located in the **Tutorial Files/Pictures** Folder copied to your hard disk during the installation)
11. Click either the **Use Currently Open Folder** Button if you are inside the Pictures folder, otherwise highlight the Pictures folder and click the **Use Highlighted Folder** Button.



Either way you will see that the new picture path has been added.

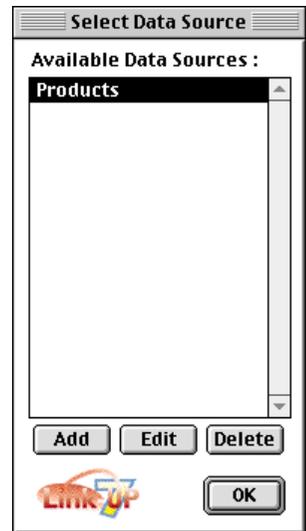


12. Click the **OK** button

The next process is to tell LinkUP! where to find the data needed for updating the document. As you are working with LinkDB files it is simply a case of letting LinkUP! know which Folder contains the LinkDB file. LinkUP! comes with a 'learn' feature that gives it the ability to learn all of the necessary information about a data source without you having to type it all in. We'll use this function now.

To select your data source;

13. From the LinkUP! menu select **Configure Data Sources**
14. Select the **Products** Data Source
15. Click the **Edit** Button
16. The data for this tutorial is held in the **Tutorial2/Data** Folder
17. Click the **Learn** Button
18. You will now need to locate your **Data** Folder



(located in the **Tutorial2/Data** Folder)

19. Click either the **Use Currently Open Folder** Button if you are inside the **Data** folder, otherwise highlight the data folder and click the **Use Highlighted Folder** Button

Providing you have chosen the correct location, you are now presented with a list of LinkDB files located in this folder. In this case there is just the one, Products.LNK.DB

20. Select this and click the **Open** button

LinkUP! has now automatically re-configured itself to the new location of the data source. Your dialogue will look similar to this apart from the Login Database will contain your local path to the data.

Add New LinkUP! Datasource	
LinkUP! Alias Name:	Products
System Data Source Name:	LinkDB
User Name:	NTS
User Password:	
Login Database:	Products.LNK.DB
Table/View:	Products.LNK.DB
Where:	
Order By:	

Buttons: Learn, Cancel, OK

21. Click **OK** on the Add New LinkUP! Data Source dialogue
22. Click **OK** on the available Data Sources dialogue to finish

All of the initial configuration is complete. LinkUP! knows where the pictures are stored and also where the Data Source is located. These two actions are one-off actions. You need only perform these two configurations once. You are now ready to update the QuarkXPress document.

The next task is to tell LinkUP! which products you would like to appear in which templates. We do this by using the Data Viewer to look directly at the data within QuarkXPress and then assigning database records to the page.

23. From the LinkUP! menu select **Show Palettes Data Viewer**

Use the controls and the **Requery** button to select the fields that will be most helpful for you in deciding which products to use. Probably Title and description (For more information on the Data Viewer see page 50).

All you have to do to assign a product to one of the templates is;

24. Place the cursor somewhere in the middle of the price on the template
25. Double-click on the product listed in the Data Viewer that you wish to assign to that template

You will notice that the cursor has moved from the middle of the price field to the beginning. By opening the Configuration Tools Palette and clicking in the price field again, you should be able to see that the stock number of the product you selected is now visible in the hidden key section of the Configuration Tools Palette. If this is not the case you can either try the above again or alternatively you can simply type the stock number of the product into the hidden key box.

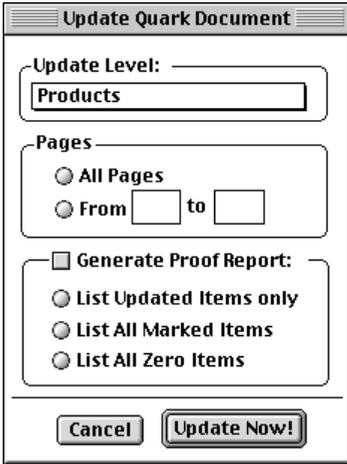
The reason we chose the price field and not any of the other elements to assign the product to is that the price field in these templates is a special one. The price field in this example is what is called a hidden key. That means that although to the user the price field just looks like any other price, it also had embedded within it the stock number for that particular group of elements. So by clicking in it and double-clicking on the product in the data viewer, you are transferring the stock number from the Data Viewer to the price. This could be done as we have just mentioned by simply typing the desired stock number into the hidden key section on the Markup Configuration Tools Palette (Marking Up on page 117 for more details).

Now repeat this process for the other items you dragged onto the page using either the same product or some different products from the Data Viewer.

All that now remains is to update the page to reflect the products that you assigned to it.

26. Click the Update button on the Data Viewer Palette 

A dialogue will appear;



We are about to see LinkUP! use this database to update the QuarkXPress page to reflect the information held in our database for each product on the page.

27. Click the **Update Now!** Button

LinkUP! will present its **LinkUP! is about to Update** windows, allowing you to observe and confirm each individual update. For now we will just proceed with the process automatically.

28. Click the **Update without further confirmation** button and watch LinkUP! bring the real products and pictures onto the page.

It is important to realise that although you were working with grouped templates from the Library this was only done for convenience in this tutorial. LinkUP! places no restrictions on how QuarkXPress operators layout the pages. Items do not have to be grouped, in the same text box or even on the same page.

# Tutorial 3 - Preparing a Document for LinkUP!

## Overview

So far you have seen LinkUP! updating pages and how easily operators can construct pages using pre-marked library items. What this section will do is show you how to achieve this with your own documents. By the end of this section you should have a good feel for how to connect to your data source, define fields, configure links and markup the pages.

## Making it happen

All the files necessary for the successful completion of Tutorial2 are located in the **Tutorial3** Folder that you copied across to your hard disk during the installation. The only exception to this are the pictures that are located in the **Pictures** Folder.

1. Launch QuarkXPress with the **LinkUP! 7 LinkDB** XTension properly loaded as above.
2. Locate and open the Tutorial3 Document that you copied across during the installation (it's located in the LinkUP! **Tutorial3/Documents** Folder)

Before you go any further we would suggest you make sure you have read *Concepts* on page 34. It explains some key terms that we will be referring to throughout this tutorial.

The main topics that will be covered in this tutorial are;

- **Defining LinkUP! fields**
- **Marking up the QuarkXPress page**
- **Database links and configuration**
- **Updating your publication**

## DEFINING LINKUP! FIELDS

Before you can do anything in your QuarkXPress document, the first task is to obtain a list of database fields to work with. Defining LinkUP! fields is the process of describing the textual or numerical elements on the QuarkXPress page e.g. SKUs, part numbers, prices, pictures etc. There are two ways to obtain a list of fields to work with in LinkUP! you can either manually create them or alternatively you can import them straight from your data source.

The fields defined for any document are listed in the Markup Tools Palette. Display this now as follows;

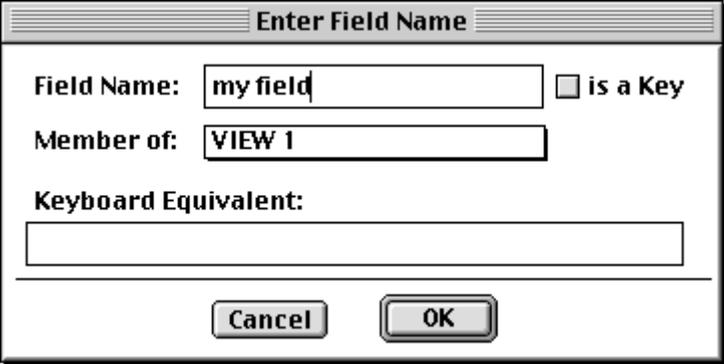
1. From the LinkUP! menu select **Show Palettes Markup Tools**

The list should be empty as there are no fields in this document. In fact this document has been completely untouched as far as LinkUP! is concerned.

To define a field with LinkUP!

2. Click the Define Field Button on the Markup Tools Palette 

The following dialogue appears



**Enter Field Name**

Field Name:   is a Key

Member of:

Keyboard Equivalent:

You can enter a name for the field, specify if it is a Key Field (see read *Concepts* on page 34), specify what relational level this field is at (default Level 1) and even define a Keyboard Equivalent.

3. Type the words **my field** in the Field Name box

4. Click **OK**

You will now see the field *my field* appear in the Markup Tools Palette

This was just to illustrate that the LinkUP! field names defined in QuarkXPress can be anything at all. Really whatever names best describe the fields to the XPress operators.

As we already have a database to work with, a field list can be obtained directly from the data file. To do this you will need the Data Viewer Palette open.

5. From the LinkUP! menu select **Show Palettes Data Viewer**
6. On the Data Viewer Palette, click the **View:** pull down menu and select the **<New View>** option.

You are immediately confronted with a search box so you can locate the data file, in this case a LinkDB file.

7. Browse your computer and locate and open the **Tutorial3/Data** Folder
8. Click either the **Use Currently Open Folder** Button if you are inside the **Data** folder, otherwise highlight the data folder and click the **Use Highlighted Folder** Button

You should have one available table, **Products.LNK.DB**

9. Select this Table and click the **Open** Button

You now have the Import Fields from Data Source dialogue on screen. On the left is a list of available fields in the data source and on the right a list of fields to be imported. For now we will import all field names. You can also prefix each field name, the default being L1\_. This is essential when you are working with relational database structures and need to differentiate between the different levels. You can name the data source and finally you must select a field to use as the Key Field.

10. Click **Add All**
11. Change the View name to **Products**
12. Select **CODE** as the Key Field

13. Click the **OK** Button

The Markup Tools Palette now contains all of the field names you have imported. Now you have a list of fields to work with, the next stage is to markup the document.

## **MARKING UP THE QUARKXPRESS PAGE**

You will now learn how to markup the page. This is where you assign the field names just imported to specific items that are to be linked to the database. The markup process, like the define field names process is only done once for the entire publication. Future updates to the publication can then be done automatically!

1. Highlight the Stock Number **48632084** located under the word FEATURES
2. Double click on the **L1\_CODE** (the one with a key next to it) in the Markup Tools Palette

This establishes the record to which the items (prices, pictures etc) will be linked. Notice that the number is now underlined in red, indicating that it is the active LinkUP! Record Number (LRN). Subsequent markups will also be underlined in red, providing a visual confirmation that you are linking the appropriate text and pictures to the appropriate record.

3. Now highlight the word **BODY** and then double-click on the **L1\_BODY** field in the Markup Tools Palette

We have now instructed LinkUP! to apply the **BODY** Field from the database to Stock Number **48632084**. Notice that the marked up field is underlined in red.

4. Highlight the word **FEATURES** and then double-click on the **L1\_FEATURES** field in the Markup Tools Palette
5. Highlight the word **CALLOUT** and then double-click on the **L1\_CALLOUT1** field in the Markup Tools Palette
6. Highlight the large **999.99** price and then double-click on the **L1\_UNIT\_PRICE\_1** field in the Markup Tools Palette

Pictures are marked up in exactly the same way.

7. Select the large picture and then double-click on the **L1\_PICTURE1** field in the Markup Tools Palette
8. Select the **LOGO** picture and then double-click on the **L1\_LOGO1** field in the Markup Tools Palette

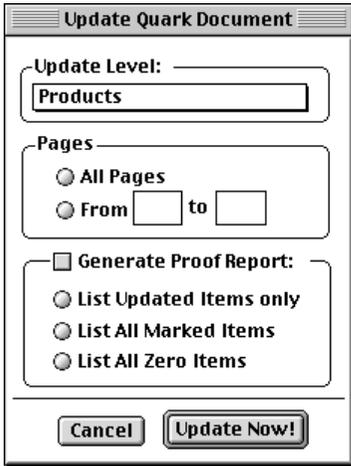
This item has now been marked up. We do however need to perform a couple of configurations before we actually update the page and bring the latest information and pictures for the product 48632084. First of all you must tell LinkUP! where to find the pictures as in the previous tutorials.

9. From the LinkUP! Menu, select **Configure Picture Paths**
10. Click the **Add** button
11. Browse your computer to find the location of the pictures. (They will be located in the **Tutorial Files/Pictures** Folder copied to your hard disk during the installation)
12. Click either the **Use Currently Open Folder** Button if you are inside the Pictures folder, otherwise highlight the Pictures folder and click the **Use Highlighted Folder** Button. Either way you will see that the new picture path has been added.
13. Click the **OK** button

Now lets update the page;

14. Click the Update button on the Data Viewer Palette 

A dialogue will appear as below;



We are about to see LinkUP! use the Products database to update the QuarkXPress page to reflect the information held in our database for the product on the page. The process is very fast...

15. Click the **Update Now!** Button

LinkUP! will present its **LinkUP! is about to Update** windows, allowing you to observe and confirm each individual update. For now we will just proceed with the process automatically.

16. Click the **Update without further confirmation** button and watch LinkUP! do the hard work!!

On your page you should now see the product as below with just a few differences that we will address next.



**WaRRioR**

**New**

**\$79<sup>95</sup>**

**Warrior Virtual  
Pilot PC Controller**

*Fly your way around the world with this ultra-realistic flight control stick.*

- 6 Button Access
- Includes Windows 95 Software
- Desk mounting kit

48632084

You will probably notice first of all that your 79.95 doesn't look quite like the one above. Effects such as this superscript can be achieved on an individual field basis. These are setup in the Configure Import Links dialogue.

17. From the LinkUP! menu select **Configure Import Links**

You are presented with the following configuration screen:

**Configure Import Links**

View:

**Quark Field**

- L1\_CODE**
- L1\_BODY
- L1\_CALLOUT1
- L1\_CALLOUT2
- L1\_CALLOUT3
- L1\_CATEGORY
- L1\_DESCRIPTION
- L1\_FEATURES

**Currently Linked To:**

Datasource: Products

Table: Products.LNK.DB

Field: CODE

Allow update from database

Delete if Empty

Add Prefix:

Add Suffix:

If Absent:

Find:   Replace  Before  After

Change:

Use Tags:  None  Quark  LinkUP!

Previous:

Next:

**Change Link for the selected Quark Field**

NONE SELECTED

Select Field:

On the left-hand-side of the screen is a list of the available LinkUP! fields. As you click on each one of these, the rest of the screen will change to reflect the individual configuration for each field, also showing which field in the data source the LinkUP! field is linked to.

The links for this document were setup for us automatically because we actually imported the field names. The mapping to the database fields can also be achieved within the Configure Import Links screen by clicking the Select Database button and choosing the relevant field from the Select Field list.

To achieve the superscript effect on the price and the \$ symbol you will need to setup a prefix for the field and also set some find options. It is important to note that all of these effects are taking place without having to make any changes in your data

18. Click on the **L1\_UNIT\_PRICE1** field on the left-hand-side of the screen
19. Add a **\$** symbol into the **Add Prefix** box
20. Add a full stop **'.'** into the **Find** box
21. Type **<V>** into the **Change** box
22. Click the **Quark Use Tags** Button

What you have told LinkUP! to do is add a \$ symbol in front of the Unit Price1. You have also instructed LinkUP! to look for a full stop in the price, which will obviously fall before the units, and replace it with the Quark Tag <V> which is the Superscript Quark Tag.

This will illustrate to you the power of Tag processing during an update. For reference the screen should now look like this:

**Allow update from database**

<p><b>Add Prefix:</b> <input style="width: 100%;" type="text" value="\$"/></p> <p><b>Add Suffix:</b> <input style="width: 100%;" type="text"/></p> <p><b>If Absent:</b> <input style="width: 100%;" type="text"/></p> <p><b>Find:</b> <input style="width: 50%;" type="text" value="."/> <input checked="" type="radio"/> <b>Replace</b> <input type="radio"/> <b>Before</b> <input type="radio"/> <b>After</b></p> <p><b>Change:</b> <input style="width: 100%;" type="text" value="&lt;V&gt;"/></p> <p><b>Use Tags:</b> <input type="radio"/> <b>None</b> <input checked="" type="radio"/> <b>Quark</b> <input type="radio"/> <b>LinkUP!</b></p>	<p><input type="checkbox"/> <b>Delete if Empty</b></p> <p><b>Previous:</b> <input style="width: 50px;" type="text" value="0"/></p> <p><b>Next:</b> <input style="width: 50px;" type="text" value="0"/></p>
--	--

23. Click the **OK** button
24. Click the Update Button  on the Data Viewer Palette and click through the confirmation screens as before

If you setup the Import Links correctly, LinkUP! will have updated the page and this time the price field will be in superscript with a \$ symbol in front of it.

For the more astute of you, you will notice that the list of features is also different in the sample page above from your updated page. The features are actually listed with bullets in front of them.

Again you will need to use the Configure Import Links Screen to achieve this effect.

25. Select Configure...Import Links from the LinkUP! Menu
26. Select the L1\_FEATURES field from the field list on the left-hand-side

This is currently all blank, so you will need to change it to appear as in the example below.

**Allow update from database**

<p><b>Add Prefix:</b> <input type="text" value="•"/></p> <p><b>Add Suffix:</b> <input type="text"/></p> <p><b>If Absent:</b> <input type="text"/></p> <p><b>Find:</b> <input type="text" value="*"/> <input checked="" type="radio"/> <b>Replace</b> <input type="radio"/> <b>Before</b> <input type="radio"/> <b>After</b></p> <p><b>Change:</b> <input type="text" value="&lt;\n&gt;•"/></p> <p><b>Use Tags:</b> <input type="radio"/> <b>None</b> <input checked="" type="radio"/> <b>Quark</b> <input type="radio"/> <b>LinkUP!</b></p>	<p><input type="checkbox"/> <b>Delete if Empty</b></p> <p><b>Previous:</b> <input type="text" value="0"/></p> <p><b>Next:</b> <input type="text" value="0"/></p>
---	--

The data in the Features field in the database contains a list of features separated by a \* symbol. What the above screen does is look for the \* symbol and replace it with a bullet and carriage return. Also, as there is no 8 symbol before the first feature, we will insert a bullet point here as well.

27. Click the Update Button  on the Data Viewer Palette and click through the confirmation screens as before

The end result is as above with a formatted list of features each preceded by a bullet.

There is no limit to number of effects that can be achieved automatically this way on the XPress page without having to place any tags or changes in your data. A full list of Quark Tags is given on page 214.

# REFERENCE

## ODBC

### Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Installation - if you would like to work with an ODBC Compliant Data Source	26
Data Viewer Palette	50

Open Database Connectivity (ODBC) is a uniform interface standard used to access databases. It does just about the same thing as CodeBase, the Borland Paradox Engine, Informix, or dozens of other libraries. It allows your application (in this case QuarkXPress) to manipulate data in a database.

So why use ODBC instead of any of these other libraries? The reason is because ODBC can manipulate almost any database, whether it be DB2 on an AS/400 or bTrieve files on a laptop. Even applications that you may not consider to be databases such as spreadsheets like Excel.

You might be curious as to how ODBC can manipulate such diverse databases. Well that's actually the easy part as the vast majority of databases conform, in whole or in part, to the relational database concept of E.F. Codd. So any ODBC compliant database, i.e. one that has been written to understand the ODBC commands, can be manipulated using ODBC.

### How does ODBC work?

ODBC provides a considerable amount of database independence through the use of drivers. Drivers are specific to a database. You use a FoxPro ODBC driver to manipulate a FoxPro database. A driver is a module (usually a DLL) written to support ODBC function calls. LinkUP! manipulates the database by calling these functions inside the driver. If you want to manipulate a different type of database, you dynamically link to a different driver.

These ODBC Drivers are available from various vendors, including INTERSOLV, Sybase and OpenLink. Contact details are available on the MC Research Website or you local reseller should you be unsuccessful.

## **So why should you use ODBC?**

There are many good reasons to use ODBC. Some are technical, while others are economic. Some are common sense. It's just hard to find a good reason not to use it.

Much of the motivation for using ODBC stems from its relative database independence. Changing databases is usually easy, and often trivial. It's often just a matter of changing drivers. This means that you can incorporate new products and technology with existing I.T. investments - hardware, operating environments, and training, with minimum effort.

Using ODBC allows you to mix and match the 'best of breed' products. Thus databases and development tools, and connectivity software become totally independent.

Using ODBC also means that you can use LinkUP! without any processes falling apart because the network has been re-configured (e.g. change of topology, or transport protocol etc.), or the graphical user interface standard has changed, or a component operating environment has changed.

## **LinkUP! and ODBC**

LinkUP! ODBC embraces this powerful technology letting you exchange information with any ODBC compliant data source. For some companies working with an interim ASCII file and LinkUP! is an adequate solution for publishing data in QuarkXPress. In some cases there may be very specific reasons for doing so. However this can be a tedious process and also runs the risk of someone using a file that is not the most up-to-date one. By using ODBC LinkUP! you are guaranteed that you are always working with the current data.

Here are just some of the data sources that you are likely to be able to use with LinkUP!

- Oracle
- Ingres
- Informix
- Sybase
- SQL Server
- Microsoft Excel for Mac
- Microsoft Excel for PC
- FoxPro
- dBase x
- Clipper
- and many more besides

## The Standard

ODBC is an abbreviation that stands for "Open Database Connectivity", and is the standard for inter-database communications as defined by Microsoft and others. By complying to this standard, information can be exchanged directly between data sources and applications without either having knowledge of how the other works.

Within LinkUP! this simply means that we can access database files created by various ODBC compliant database applications directly, without the use of the LinkDB application.

## ODBC Requirements

To be able to utilise the ODBC standard the following conditions must be met:

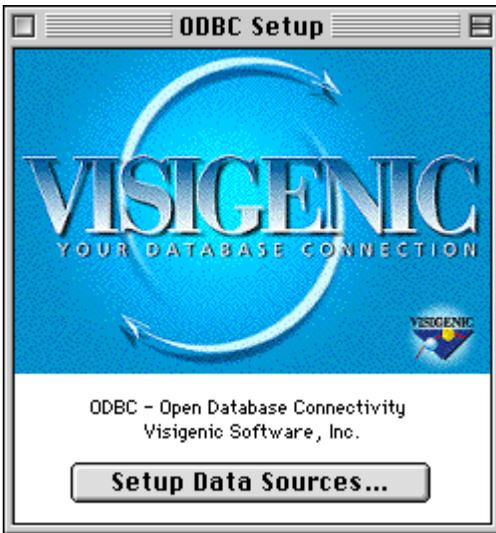
1. The Database you are trying to communicate with must be ODBC compliant, and hence have an ODBC Server driver running on the platform it resides on.
2. A physical network connection must exist between the Macintosh running LinkUP! and the above mentioned system, unless actual data files are copied and residing on the local Mac disk.
3. The Macintosh running LinkUP! must also run:
  - a) Apple Shared Libraries Manager (ASLM)

b) ODBC driver software to connect to the above mentioned system.

The Apple Shared Library manager is available on the LinkUP! CD-ROM. The ODBC Driver software you use depends on the choice of the database you are attempting to connect to. This choice remains with your database/network manager. The list of drivers available is constantly being updated and released by database vendors.

Before attempting to run LinkUP! 7 68k or PPC please ensure that your ODBC drivers are fully installed on your Mac, as per the suppliers instructions. You will also need to configure an ODBC Data Source via the drivers control program, that will allow LinkUP! to communicate directly with your chosen database.

As far as the use and functionality of LinkUP! is concerned, the 68k and PPC versions differs from the LinkDB version only in the setting up of the data sources you intend to use. Instead of locating a LinkDB processed database file, you choose an ODBC source from the list presented of available ODBC applications.



The actual screen you may see in this stage of configuration depends on your choice of ODBC driver, and may differ from the one shown above.

With ODBC connectivity, data exchange can occur in both directions. This means that you are able to update your data source directly from your XPress document. When writing data from LinkUP! to a data source, LinkUP! will only write the existing records but not create new ones.

## Configuring a Data Source (LinkDB Driver)

To configure a data source for use with the LinkDB driver supplied, perform the following steps:

1. Run the ODBC Setup Control Panel from your Apple Menu
2. Click on “Add” from the list of available ODBC drivers and select “LinkDB” as the driver
3. In the Data Source Field Name type in “LinkDB”
4. You can leave the remaining fields blank (Database Folder and Data File Name) as these will be specified within QuarkXPress.
5. Click **OK** and then exit the Control Panel

You can now run LinkUP! and access LinkDB database files through ODBC.

## Configuring a Data Source (Other Drivers)

To fully configure a data source, you will have to provide information about the location of the database that you want to work with, and possibly further parameters such as login information. The information that is required varies from driver to driver, and so you will need to refer to documentation that is supplied with the driver for the necessary details.

## Installing the correct ODBC version of LinkUP!

The final configuration that you need to do to gain access to your data from within Quark is to install the correct version of LinkUP!. If you have already been working with the LinkDB version of LinkUP! for the purpose of running the tutorials, then you will need to remove it. This will mean taking it out of either your main QuarkXPress folder, or the XTension folder that is contained within the QuarkXPress folder.

If you are working with a 68K driver (including the LinkDB driver supplied) you should copy the **LinkUP! 7 68K XTension** into your Quark (or XTension) folder. Note that this is the correct XTension for accessing 68K drivers REGARDLESS of whether you are running the 68K or PowerPC version of QuarkXPress. (3.32r5 or 4.02).

If you are working with a PowerPC driver you should copy the **LinkUP! 7 PPC XTension** into your Quark (or XTension) folder. Note that this XTension can ONLY be used with the PowerPC version of QuarkXPress (3.32r5 or 4.02).

## Establishing a Connection

To check that your ODBC setup is functioning, you should;

1. Start up QuarkXPress and activate LinkUP!
2. Next, from the LinkUP! menu, choose **Show Palettes Data Viewer**.
3. From the Data Viewer choose the **<New View>** option from the View: Popup list.

You will now be presented with a list of data sources that you have defined in the ODBC setup.

4. Choose the correct data source, and you will then be prompted to connect to your database.
5. Finally, you will be able to select a list of fields to work with. (see page 105)

If all has worked correctly, you will now be able to view data from your database within the database viewer window, exactly as you were able to with the Tutorial data.

## ODBC Connectivity Issues:

1. Make sure network connection between the Mac and the Data Server is present and reliable.
2. Make sure Data Server has networking TCP/IP option installed and functioning properly.
3. Make sure MAC TCP is configured correctly to TCP/IP (under normal circumstances) and not to Mac PPP.
4. Make sure IP address and the Subnet mask of the Data Server is entered correctly in the ODBC Setup parameters.
5. Confirm TCP/IP link by using "Ping" or similar utility between the Mac and the Data Server.
6. Make sure that login to the Data Server is possible under the name and password you intend to use.
7. Make sure that database you are trying to log into exists on the Data Server and that appropriate access for the type of user is allowed.

If all of the above is functioning correctly, the following 3 dialogues would be the only difference from ASCII version of LinkUP!

# Field Names & Data Setup

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Working with Simple Databases and LinkUP!	40

## Defining Fields

One of the first steps to using your documents with LinkUP! is to define a list of LinkUP! fields. LinkUP! fields are labels that you will use to describe the elements of the QuarkXPress page. A typical catalogue page for instance might need fields called, description, price, picture, logo etc. By identifying to LinkUP! which elements are which on the XPress page, you can then connect these LinkUP! fields to the corresponding fields in your database.

As the linking process is separate to the field definition process, the LinkUP! field names do not have to be identical to the field names in the database. This can be especially useful in situations where the database field names are not particularly meaningful in the context of the page being created.

There are two ways to obtain a list of fields to work with in LinkUP! you can either manually create them or alternatively you can import them straight from your data source.

### DEFINING FIELDS MANUALLY

This is the most basic method of defining LinkUP! fields and is appropriate when either the fields you are defining are not currently in the database, or when the database is not available to you during the markup process. This is most likely to be the case when you are using LinkUP! to export data from the XPress page to create a database, but might be necessary if you are working offline.

The fields defined for any document are listed in the Markup Tools Palette. Display this now as follows;

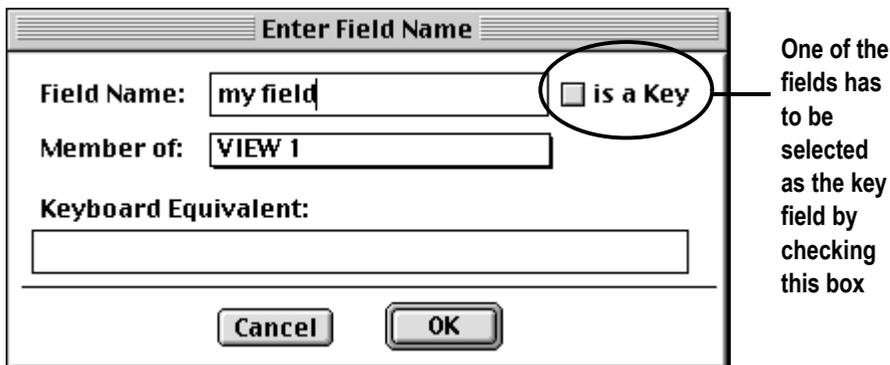
14. From the LinkUP! menu select **Show Palettes Markup Tools**

The list should be empty if you are starting with an XPress document unused with LinkUP! before as there will have been no fields defined.

To define a field with LinkUP!

15. Click the Define Field Button on the Markup Tools Palette 

The following dialogue appears



You can enter a name for the field, specify if it is a Key Field (see LinkUP! concepts p41), specify what relational level this field is at (default Level 1) and define a Keyboard Equivalent if the field name is one that you expect to be using very often. You can use any of the function keys on your keyboard as Keyboard Equivalents. They can be used either in isolation or in combination with the Ctrl, Option and Command Keys. When you are marking up, the Keyboard Equivalent performs the same function as double-clicking on the field in the Markup Tools Palette.

16. Type the words **my field** in the Field Name box
17. Click **OK**

You will now see the field **my field** appear in the Markup Tools Palette

This was just to illustrate that the LinkUP! field names defined in QuarkXPress can be anything at all. Really whatever names best describe the fields to the XPress operators.

You can define up to 200 fields for any one document. The field list is specific to each document you use. Of course once you have a field list defined you don't want to have to go through this process for every new document for your publication. What you can therefore do is save a project for a document that contains all of the field definitions. When you work on another XPress document that requires the same or a similar field list, simply load this project in (see Projects on page 159).

## **AUTOMATIC FIELD IMPORT**

If you already have a data source available to work with you can import a field list directly from your data source. This has two advantages over the manual method. Firstly you save the time of inputting field names. Secondly, LinkUP! configures the links from the data source to the LinkUP! fields automatically, cutting out a step.

Importing a field list takes place as part of the defining a New View process. This is discussed in more detail on page 105.

To import a field list from your data source

1. Make sure the Data Viewer Palette is visible
2. On the Data Viewer Palette, click the View pull down menu and select the **<New View>** option.

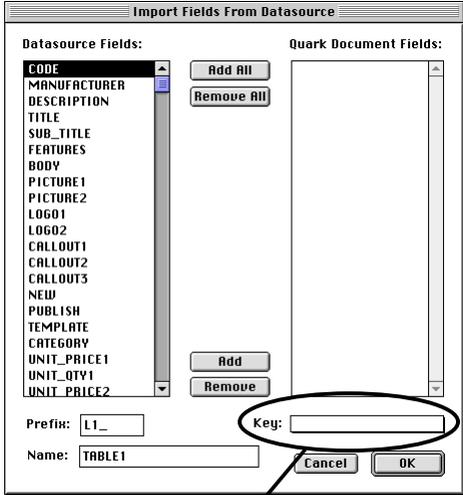
You are immediately confronted with a search box so you can locate your data source. This might be an ODBC data source or just a LinkDB file on your hard disk. If you are using the LinkDB LinkUP! XTension and connecting to a local LinkDB file all you have to do is locate the folder that contains the LinkDB file.

3. Select the desired Data Source

You are now be presented with a list of available tables in that data source. If you are using the LinkUP! 7 LinkDB XTension, LinkUP! presents all of the available LinkDB files in that folder.

4. Select the table or LinkDB file required and click the **Open** Button

You now have the Import Fields from Data Source dialogue on screen. It will look similar to the one below but containing your list of fields instead.



One of the fields has to be selected as the key field by selecting it from the drop down list of available fields in this box before clicking OK

On the left is a list of available fields in the data source and on the right a list of fields to be imported. For a field name to be imported simply highlight it and click the **Add** button. To remove a field name, highlight it in the right-hand window and click the **Remove** button. Alternatively you can use the **Add All** or **Remove All** buttons.

There are some other important options on this screen. You can prefix each field name so that when the LinkUP! field is created it contains this prefix. The most important use of this feature is when you are working with relational database structures. The prefix will make it obvious which level a field is defined at. The default setting is L1\_.

You can also name the view you are creating, the default name is View 1 so we would suggest you change it to something more meaningful that reflects the data you are viewing.

The final and arguably most important option on this screen is the Key Field. This is the field to which all the other fields will relate, such as the SKU or stock number (see p41 for more details). By clicking on the drop down list of fields, you can select the one that is to be your key field for this level.



Remember that the field you choose to be your Key Field must contain a unique value for each database record.

Once you are happy with the field list to be imported along with the other settings;

5. Click the **OK** button to continue.

The Markup Tools Palette will now contain all of the field names you selected to import. These fields have been automatically linked to your data source.

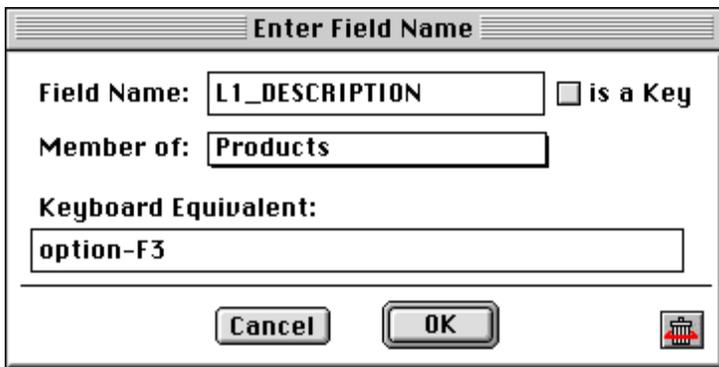
## Editing Fields

The field names defined for the currently open XPress document are listed in the Markup Tools Palette. New fields can be added to this list at any time by simple clicking the Define Field Button on the Markup Tools Palette.  These fields then become available for the marking up process.

You can also edit an existing field that is listed in the Markup Tools Palette. To edit the settings for a field;

1. Hold down the **Option Key** and double-click on the required field in the Markup Tools Palette

The Enter Field Name dialogue will appear. This is the same dialogue you will have used if you defined the fields manually (see page 98 for details).



You can make changes to any of the parameters in the dialogue box including changing the field name, setting the field to be a Key Field (remember you can only have one Key Field per level), changing the View it is a member of and the Keyboard Equivalent.

Be sure to exercise **extreme caution** when making changes to field names or the View that the field belongs to. If you have already marked up your document using any of the fields in the Markup Tools Palette, changing those field names **will result in the markup being lost**. For this reason we would recommend that you only change field names either before you

begin the markup process or if you are absolutely sure that the field is not in use anywhere in the document.

## Deleting Field Names

This is achieved on the Enter Field Name Dialogue and is accessed as follow;

1. Hold down the **Option Key** and double-click on the required field in the Markup Tools Palette.

You will notice the button at the bottom right of the dialogue. This is the Delete Field Button. To delete the field completely;

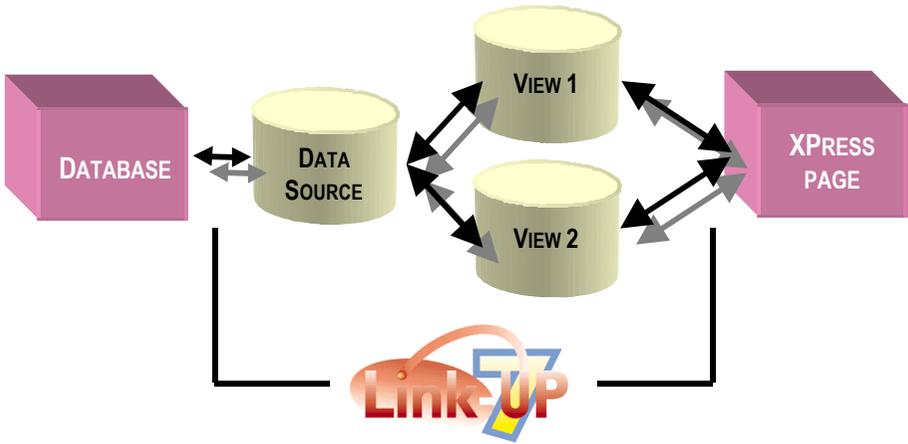
2. Click the Delete Field Button 

Be sure to exercise **extreme caution** when deleting a field from the Markup Tools Palette list. If you have already marked up your document using a field that you wish to delete, deleting the field name **will result in any markups using this field being lost**. For this reason we would recommend that you only delete field names either before you begin the markup process or if you are absolutely sure that the field is not in use anywhere in the document. It is often safer to create another field rather than deleting an existing one.

# Views

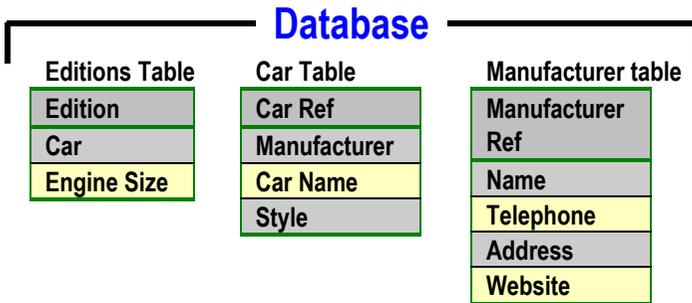
## WHAT IS A VIEW

A view is in essence exactly what the name suggests. It is a view of the data held in a data source. The following diagram should make this clearer.



Let us now look in more detail at how this would break down by examining each of the components in the above diagram and applying an example to them.

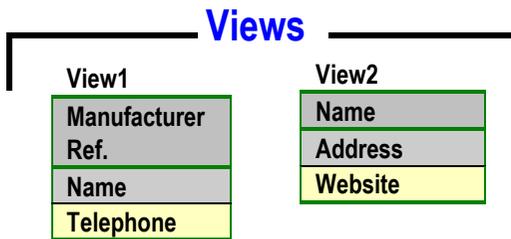
The database could be a database in the usual sense or it could be a spreadsheet or text file. Therefore imagine a database with the following tables of data;



Each of these 3 tables has its own set of fields, but collectively the 3 tables constitute the database. If we were to create a data source for this database we would have to use one of the above tables to be our data source. For this example we will choose the Manufacturer Table as our Data Source. So our Data Source would contain the fields opposite.

Data Source	
Manufacturer Ref	
Name	
Telephone	
Address	
Website	

Once you have setup a Data Source that in this case is the Manufacture Table from the Database you can now use LinkUP! to create a **View(s)** of this Data Source. A View can either be a partial or a complete view of any Data Source. You can therefore have multiple views of a single data source.



If we create a View called View1 and include the first three fields of the data source in it, our view would be as in the above illustration. We could then setup a second View, very unimaginatively, called View2 containing the fields, Name, Address and Website. So each view uses the same Data Source but contains different fields from the database. As was stated earlier, you can have an almost unlimited number of Views of each Data Source.

### WHY DO I NEED A VIEW

The main reason for working with Views of your Data Source rather than the actual Data Source itself is because a View can be far more manageable. The above example was a database with only a few fields in, but if you imagine a corporate database containing several hundred fields, trying to work with all of these within XPress would be a very unmanageable process. By using a View, you can work with just the fields required for your XPress publication. Alternatively you could break the fields in the database into manageable chunks.

As a View is only a 'view' of your data, you can use different naming conventions for a view to the field names in the database. This is especially important if the field names in the database are not immediately obvious to the XPress operators.

By working with Views LinkUP! is able to provide you with better performance in both displaying data in the Data Viewer and in the update and export processes.

## How do I create a New View

The process of creating a new view is performed on the Data Viewer Palette. The process varies depending on whether or not you are using LinkUP! 7 668k or PPC XTension or the LinkUP! 7 LinkDB XTension. These two scenarios will therefore be treated separately.

### DEFINING A NEW VIEW WITH LINKUP! 7 68K OR PPC XTENSION

1. Click on the View pull down menu on the Data Viewer Palette.
2. Click the **<New View>** option

You are immediately presented with a list of available ODBC Sources (defined in the ODBC Control Panel on your computer).

3. Select the desired ODBC Source.

You are now presented with a list of tables that are available in that ODBC Source.

4. Select the Table required (there may only be one if you are not working with a relational database)

This takes you to the Import Fields from Data Source Dialogue. On this screen you can give a name to the view you are creating, as well as specifying which fields to use in this view. Details on this dialogue are given on page 100.

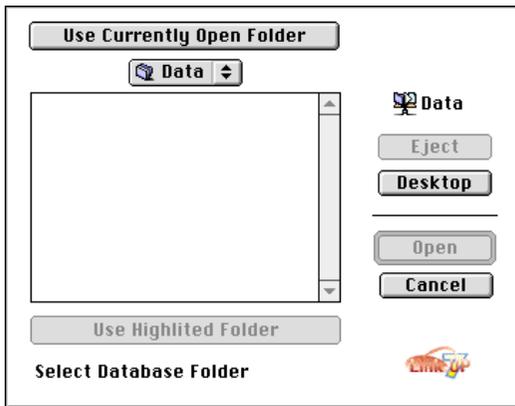
5. Once the view and field names have been configured on this dialogue, click the **OK** button.

Your View has now been created and is available for viewing in the Data Viewer Palette. See page 50 for details on using the Data Viewer Palette.

## DEFINING A NEW VIEW WITH LINKUP! 7 LINKDB XTENSION

1. Click on the View pull down menu on the Data View Palette.
2. Click the <New View> option

You are immediately presented with the Mac File Finder Dialogue. This allows you to browse your computers disk drives and locate the LinkDB file you wish to use (see page 39 for information on LinkDB).



3. Locate the Folder containing the LinkDB File(s)
4. Click the **Use Currently Open Folder** or **Use Highlighted Folder** option depending on your location.

You are now presented with a list of LinkDB Files that exist within this chosen folder. This is the equivalent to a list of the Tables available in an ODBC Data Source, proving you with the same functionality but using individual LinkDB files.

5. Select the LinkDB file required

This takes you to the **Import Fields from Data Source Dialogue**.

On this screen you can give a name to the view you are creating, as well as specifying which fields to use in this View. This dialogue is explained in detail on page 100.

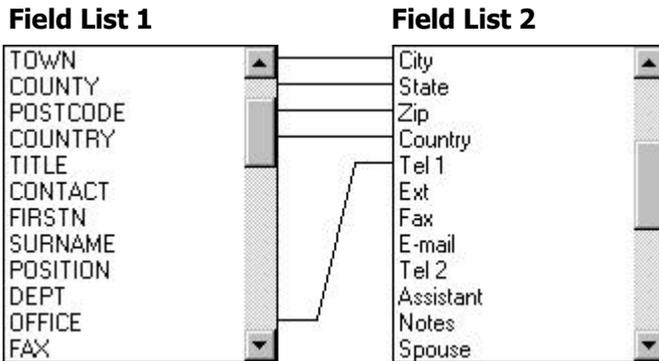
6. Once the View and field names have been configured on the Import Fields from Data Source Dialogue, click the **OK** button.

Your View has now been created and is available for viewing in the Data Viewer Palette. See page 50 for details on using the Data Viewer Palette.

# Configuring Import Links

## LINKING FIELDS

An Import Link is a link between the fields that have been used in the XPress document and the fields contained in the actual database. It is in essence the mapping of one set of fields to another. The following diagram should illustrate this more clearly.



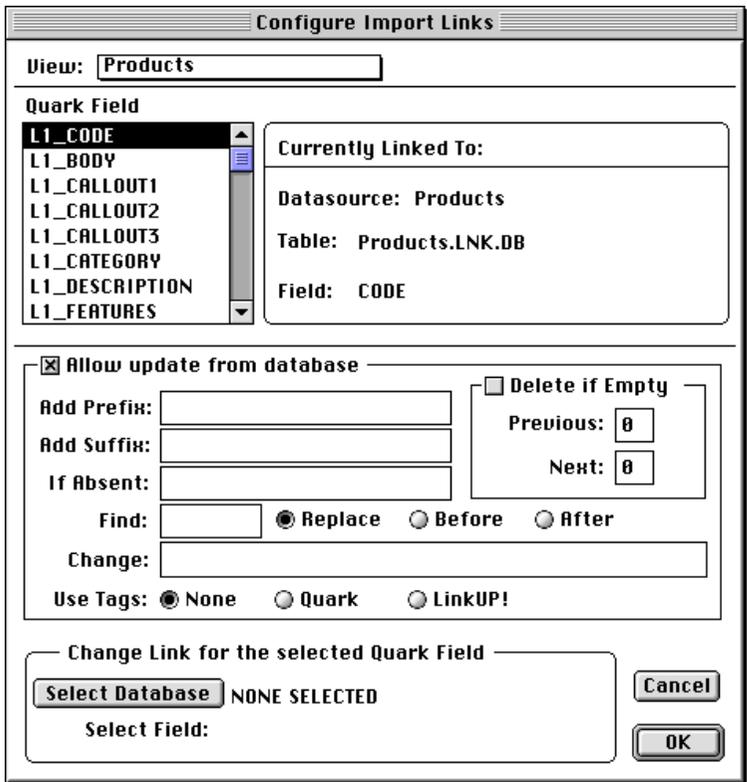
We have two lists of fields, **Field List 1** and **Field List 2**. For now these two lists simply represent field lists available in two different applications, such as LinkUP! and your Database. In the above example, the fields TOWN, COUNTY, POSTCODE, COUNTRY and OFFICE are linked the fields in the Field List 2, City, State, Zip, Country and Tel 1. This process of linking fields in one list to fields in another is called field mapping and is a common process in any computer application that works with external data.

While you may need to access several database files in order to import all the data needed for a single publication, you can only have one link between XPress and a database file active at a time. Each link to a database file needs to be configured and set up within LinkUP! (This may be done for you by an administrator).

If you chose the Automatic Field Import option (see page 100), then the links to your data source will have been setup for you automatically. If on the other hand you created a field list manually (see page 98) then you will not have any links currently setup to your database. What you will therefore have to do is go through this process of mapping fields defined in XPress (using LinkUP!) to the fields in your Data Source. The Configure Import Links Dialogue is accessed as follows;

1. Select **Configure Import Links** from the LinkUP! Menu.

The following dialogue will then appear.

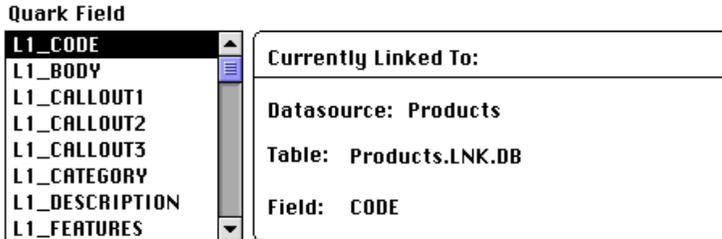


There are several components to this dialogue, but we will begin with the actual linking components.

At the very top left you will be able to see a **View:** dropdown list. By clicking on different Views in the list you can switch the list of fields below it to show the fields in each View. These are the fields available in the current XPress document. These are the fields that you defined or imported.

The area immediately to the right of this list displays the current linking status of each field as you click on it in the field list. It shows which Data Source (or folder) it is linked to, the table (or LinkDB File) within this Data Source (or folder) and the actual field within the Table (or LinkDB File) to which it is linked. If you have not yet linked any of the fields, then each of these entries will be blank.

In the dialogue below, you can see that the **L1\_CODE** field in the XPress document is currently linked to the **CODE** Field in the **Products.LNK.DB** Table in the Products Data Source.



Each link is established by first selecting a Database to work with as follows;

2. Click the Select Database Button at the bottom left of the dialogue

You are immediately presented with a list of available ODBC Sources (defined in ODBC Control Panel on your computer) or a the Mac File Finder Dialogue if you using the LinkUP! 7 LinkDB XTension

3. Select the desired ODBC Source or Folder

You are now presented with a list of available tables (or LinkDB Files) that are available in that ODBC Source (or folder).

4. Select the Table (or file) required.

Instead of saying None Selected it should now give the name of the Database Table or file and next to the **Select Field** Label a drop down list of available fields is now present. To link a field in the XPress document to a field in your database;

5. Highlight the field in the list of fields at the top left of the dialogue
6. Select the field in the database to which it should be linked

The Currently Linked To: portion of the dialogue should now change to show that the highlighted field is linked to a particular field in the database.

## CONFIGURING IMPORT PARAMETER

LinkUP! provides a whole lot more than just the ability to link a field in XPress to a field in your database(s). You can set up conditional updating, the addition of prefix and suffix and tell LinkUP! what to do if a field contains no data.

### *Add Prefix*

**Add Prefix:**

This instructs LinkUP! to place the character(s) contained within the prefix box, in front of the data extracted from the database for this field. If you were importing prices for example, you could place the currency symbol in the prefix box, so all prices would automatically have the currency symbol in front of them. This way you do not need to store the symbol either on the XPress page or in the database.

### *Add Suffix*

**Add Suffix:**

This instructs LinkUP! to place the character(s) contained within the suffix box, after the data extracted from the database for this field.

### *If Absent*

**If Absent:**

If when LinkUP! updates the XPress page, it finds no record in the database for a particular field, LinkUP! can place the character(s) contained within the Absent box onto the XPress page instead. This feature can be especially helpful for locating missing data for items in the XPress pages, ensuring you don't go to print with missing prices etc.

## Find / Change

Find:   Replace  Before  After

Change:

Use Tags:  None  Quark  LinkUP!

It is often a requirement in a publication that a particular piece of text appears in a different style, such as superscript or a different colour or font etc. This and more can be easily achieved using the Find / Change function.

When LinkUP! performs an Update of the QuarkXPress page from a database, it looks for the character(s) contained within the Find box and can then act on what it finds. The action LinkUP! takes will depend on which of the three options next to the find box is selected. LinkUP! can either replace what is found with the character(s) contained within the Change box or it can place the character(s) contained within the Change box **Before** or **After** the found character(s). This is a very powerful feature, especially when used in conjunction with Quark or LinkUP! Tags. Take for example the situation of a list of features held in the database.

Code	Product	Features
42154686	PCMCIA Card	Short Message Service*Phone Book Management*Includes Mic*temperature management

The features are all contained within one field in the database, separated by a star symbol. However, if in XPress document you would like them formatted in a list with a bullet at the beginning of each point, you can setup the find parameter to search for the \* and replace it with a bullet character and a carriage return so each feature appears on a new line. As there is no \* character before the first feature, this can be added in the Prefix Box. You will be able to see that the Replace option and the **Quark** Use Tags Options are selected.

Allow update from database

Add Prefix:

Add Suffix:

If Absent:

Find:   Replace  Before  After

Change:

Use Tags:  None  Quark  LinkUP!

Delete if Empty  
 Previous:   
 Next:

The above settings would give the following results in the XPress document.

- Short Message Service
- Phone Book Management
- Includes Mic
- Temperature management

NB. The <\n> is the Quark Tag for a carriage return.

There is no limit to number of effects that can be achieved automatically in this way on the XPress page without having to place any tags or changes in your data. A list of Quark Tags is given on page 214.

**Delete if Empty**

Delete if Empty  
 Previous:   
 Next:

If a field in the database contains no data, then you can instruct LinkUP! to delete it from the page along with any number of preceding and following characters. This is of most use when using LinkUP! to build documents. If you think of a situation where a directory is being assembled but no fax number is present in the database for a particular company, then LinkUP! could delete the fax field along with the fax label next to it.

For the parameter to take effect on the XPress document it is necessary for you to perform an update (see *Updating* on page 162). Once a link between the current XPress document and a database file has been set up, the configuration can be saved as a Project File. A Project File does not actually make connections between your XPress pages and your data sources - instead, it acts as a map or plan that LinkUP! can use in order to set up connections and links correctly next time or in subsequent XPress documents (see *Projects* on page 159).

# Marking Up

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Index Tools Palette	49
Markup Configuration Palette	47
Concepts	34

There are two important palettes you will need to use in the markup process, the Markup Tools Palette and the Markup Configuration Palette. For details on both of these please refer back to page 46 of the Navigation Chapter of this manual.

## Overview

### WHAT THE MARKUP PROCESS IS - AND WHY YOU NEED TO DO IT

LinkUP! allows you to establish connections between elements on the pages that make up any XPress document and data that is held in one or more database tables that exist totally separately to and independently from the XPress document.

LinkUP! allows you to configure relationships between elements on your XPress pages and items of data. As a result of the marking up process, you set up links between your XPress document and one or more database tables held outside XPress. These links can be two-way: you can read data into your XPress document from individual fields within individual records in a database table; and you can write data out from your XPress document into a database table.

When you markup an XPress document in LinkUP!, using LinkUP!'s Markup tools, you work with text, text boxes and picture boxes within the document. Usually you will be setting up links between a 'family' of individual elements or items in a page and individual database fields that are part of the same database record. Links are established as markups that instruct LinkUP! how data needs to be imported into the document in order to update your pages. When LinkUP! updates your marked up XPress document, the data contained within individual fields in the database will either appear directly in your XPress document, or will be used to govern the content of a text or picture box.

Why you are marking up a document will vary, depending on the publishing project you are undertaking; how you achieve the markup will remain broadly similar. When you markup an XPress document you perform a sequence of operations by which you assign the contents of some or all of the fields of a single database record to one or more elements in the XPress document.

The process of marking up an XPress document does not change the way that the document looks when it is printed out (if it is printed out before you update it using LinkUP!). Nor does it need to effect the way that the document is designed in terms of typography and layout. Marking up an XPress document will affect the way that it appears on screen – the process adds blue underlines to elements that have been marked up.

While you are working with a record from the database, assigning fields from that record to elements in the XPress document, you are marking up that record. Red underlines are used in the XPress document to indicate that you have not yet finished working with a record – elements on your pages that are underlined in red will relate to the current database record which you are marking up.

You will use three palettes to achieve such markups: The Markup Tools palette, the Markup Configuration palette and the Data Viewer palette. The *Guide to Floating Palettes* section of this manual contain more details on these palettes.

## **DIFFERENT KINDS OF MARKUP**

### ***Text Markups***

You can markup a section of text contained within a text box on the page so that it will be replaced with the text contained in a field in your database table when your XPress document is updated by LinkUP!. By default, the typographic styling of the 'placeholder' text you select during the markup will be preserved when it is replaced by 'live' data from a database field. This markup involves creating database links.

You can also markup text boxes so that their text content is generated dynamically after the XPress document has been updated in a way that reflects the content of individual pages – a useful facility if you want to have headers and footers generated for pages, automatically. One application for this facility would be to provide the headers in a telephone directory, where each page is headed up by first and last name that appears as a directory entry on that page. This markup involves creating Header/Footer (Page Reference) links.

### ***Picture Markups***

Although some databases are capable of storing pictures as the content of fields, LinkUP! only works with filenames of pictures. If your source database contains the names of picture files, then you can markup a picture box on your XPress page so that a database field containing the name of the picture file is associated with that picture box. When the XPress document is updated by LinkUP! you need to specify the physical path on your hard drive or network where LinkUP! should 'look' for the picture files it will need. LinkUP! then uses the picture filename data imported to specify which picture file should be collected and placed into a given picture box. This markup involves creating database links.

### ***Marking Up Database Links***

If you are marking up elements of an XPress page in order to establish the relationship between those elements and a database table held outside XPress, then you will be marking up database links. This task involves marking each element either as a Field, as a Picture or as a Key Field.

### ***Marking Up Header/Footer (Page Reference) Links***

LinkUP! also allows you to markup elements on XPress pages so that they automatically generate information for headers, footers, section heads or other items of page referencing dynamically after the XPress document is updated by importing data.

Once you have completed a document markup in order to set up database links and have updated that document from the database source(s) you are using, you can use the Add Special Markup tool within the Markup Tools palette to create markups for header/footer information that will appear in your final document. You need to open a new textbox which will hold the information that a special markup delivers – and you specify whether the content of such a textbox will be generated based on the TOC, Key or Field – and at what level.

### ***Marking Up Table of Contents (TOC) and Index Links***

If you are Marking Up elements of an XPress page in order to identify them as entries that need to be included in a Table of Contents (TOC) or Index then you use the Index palette to create your markup rather than the Markup Palette. Descriptions of the Index palette and LinkUP!'s LinkIndex utility appear in the *Creating Table of Contents (TOC)* from your QuarkXPress documents and *Creating Indexes From your QuarkXPress Documents* sections of this manual on pages 167 and 175 respectively.

## Marking Up Database Links

A sound understanding of the role and function of the LinkUP! Record Number (LRN) is vital if you are to understand why and how to markup a document – the Concepts chapter that starts on page 34 of this manual contains a concise description.

Each collection of XPress elements that, between them, are used to display the content of some or all of the fields from a single record in the data source, will be assigned the same LRN so that you and LinkUP! can keep track of which data items will be placed where on your pages when you perform an update. LRNs are assigned for you by LinkUP! while you are performing a markup. If you like, the LRN assigned by LinkUP! is the equivalent of giving the family of page elements a surname so that members can be identified as belonging to the same virtual unit.

You also need to assign a Key Field to each collection of XPress elements that, between them, are used to display the content of some or all of the fields from a single record in the data source. LinkUP! uses the Key Field to determine which record in the data source will be used to supply the content when LinkUP! fetches the content of database fields and places them in your marked up XPress page elements. The database field in which LinkUP! will look for Key Field data during the update process appears in the Markup Tools palette with a key symbol next to its name.

The first step in preparing a document markup involves setting up the Markup Tools palette by providing the field names that you will be using inside your XPress document. These field names can be the same as the field names used to identify fields in the database table that will provide data for the update. Alternatively, if the field names in a source database are not particularly meaningful, you can make up your own names for use in the XPress document and then configure LinkUP! so that your meaningful names 'point' to the appropriate fields in the database table.

### **MARKING UP DATABASE LINKS - TWO BASIC APPROACHES**

There are two approaches to creating or amending a markup. You can either have a live link between your XPress document and the database table or database tables that will act as the data source when the document is finally updated, or you can work on your markup without direct access to the data source.

When you do have access to the database table, you are 'working online'; when you don't have access to the database, you are working 'offline'.

Generally, it is more straightforward to perform markup operations when you are 'online' to the file containing the data source ('online' in this context does not mean connected to the database via the Internet - it means that the file is accessible from your machine, either direct from disk or via a network connection to another machine or database).

You may be forced to work on a markup 'offline' because you cannot currently gain a live link to the database file that contains the data source. Alternatively, you may choose to work offline - probably because you are editing an existing markup and only plan to work with a few fields from a large database table that has many fields. You need to work 'offline' if you are marking up an existing XPress document that contains real data that you wish to export to a database table - to complete that process, you need to use the Export Links facility as the final step...

## **DEFINING FIELDS AN OVERVIEW**

### ***Offline Working – manual field definition***

The Markup Tools palette allows you to create field names by manually defining the field names within your XPress document - you can type them in. One field has to be assigned by you as the Key Field.

Clicking on the Define New Field button in the Markup Tools palette allows you to type in the name of a field that you can use in the XPress page world when you are marking up XPress page elements. Later, when you need to configure the link between your marked up XPress page and external database table(s), the Import Links tool can be used to make the association between fields in a database table and the field names you created in the Markup Tools palette. See *Defining Fields Manually* on page 98 for more details.

### ***Online Working – automatic field definition***

Using the Data Viewer palette allows you to set up a direct link with the contents of the database file that you will use as a data source once your markup has been completed. Using the facilities of the Data Viewer, you can select the database table with which you plan to work and automatically import all or a selection of field names to the Markup Tools palette.

This approach saves time by avoiding the need to create and type in field names, and by avoiding the need to use Import Links in order to configure the links between manually-created field names in the XPress document and field names in a database table. One field

has to be assigned by you as the Key Field. See *Automatic Field Import* on page 100 for more details.

## **STARTING TO MARKUP**

Once the Markup Tools palette contains a list of all the available database fieldnames you want to use in marking up the current document, and one of those fieldnames has been assigned as the Key Field, you are ready to start marking up an XPress document so that it can be updated by LinkUP!.

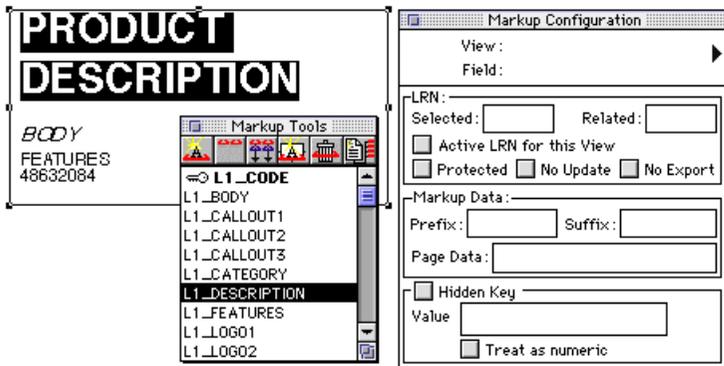
To begin a single markup you need to assign a Key Field to one of the text elements in a set of items in an XPress document. In assigning a Key Field, you markup a text element as the place where LinkUP! can find the value of the Key Field for that set of items – or family of XPress page elements.

The moment you markup a Key Field on the page, LinkUP! automatically assigns a unique LRN to the set of XPress page elements - in effect you have started a new family of items which has a 'surname' in the form of its unique LRN (by which its members can be identified). In marking up a Key Field, you have given your new family the address of the database record that LinkUP! will use to update the content of family members when your document is updated.

The set of XPress page elements that forms the family group that becomes a single markup in the document may or may not appear on the same page in the document, and may or may not be formed from an XPress Group of page elements. All family members could be contained in a single XPress page element – as tab separated columns in a text box, for instance. Picture boxes, text strings and text boxes can all be members of a markup family.

### ***Assigning Fields***

In essence, you highlight a text string inside an XPress page element to select it as a text string that you want to have replaced by data from the data source when your XPress document is updated by LinkUP!. Once you have selected some text on your page, select the field name of the field that contains the data you wish to use in the Markup Tools palette and double-click on it to assign that field with your chosen text item. For example;

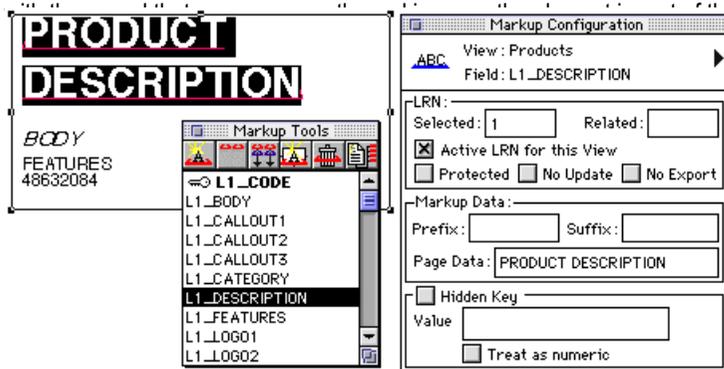


The picture above shows the text "Product Description" in an XPress text box about to be marked up so that it will be replaced by the content of the 'L1\_DESCRIPTION' field in the current database record when you update your marked up document with LinkUP!.

This markup is a two-step process:

1. Highlight the text **Product Description**
2. Double-click on the **L1\_DESCRIPTION** field in the Markup Tools Palette

The Highlighted text will now look as follows with a red underline signifying that it is marked up. The underline appears on screen in red to signify that you have not yet finished working



family that you performing this op left of the ing "Product

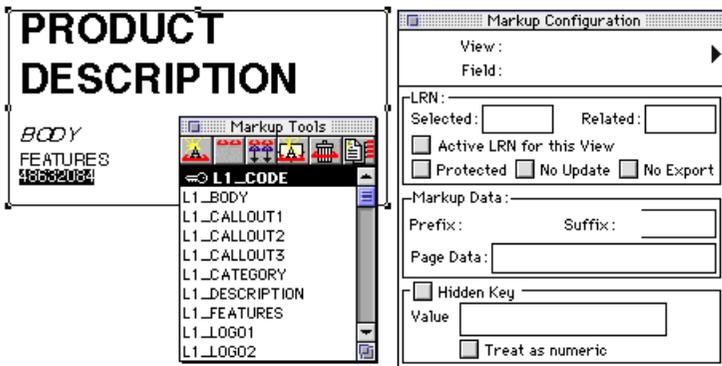
*This is how your XPress text box now looks on screen*

### ***'visible' Key Field - data in Key Field appears on the page***

If a field in the data source that contains 'useful' information that you want to appear on the pages of your document after it has been updated has been assigned as the Key Field, then you markup a Visible Key Field on your XPress page. The process is, in effect, identical to the Assigning Fields text markup in which we marked up the Product Description text, but the data in the text string you markup on your XPress page will be used by LinkUP! to identify which record in the data source will be used to provide information to update other members of the markup family you are working on.

The actual data that LinkUP! will use to identify the database record can be on the page as text before you markup a Visible Key Field, or you can markup a placeholder text string and change its content so that it represents the content of the Key Field in the database record that you wish to assign to this markup family. In the following example the record that contains the value 48632084 is the one that we want to use to update the other members of this markup family and the value "48632084" is already on the page as an item within a text box.

The picture above shows the text "48632084" in an XPress text box about to be marked up. As you can see by the key symbol next to the 'L1\_CODE' field in the Markup Tools window, the 'L1\_CODE' field is the Key Record in this database table.

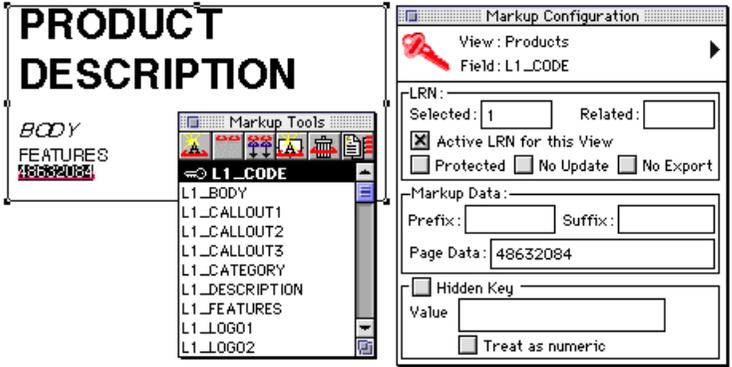


To markup the text so that it becomes the value by which LinkUP! will identify which database record to use when updating other members of this family of markups is a two-step process:

1. Highlight the text **48632084**

2. Double-click on the **L1\_CODE** field in the Markup Tools Palette

The Highlighted text will now look as follows with a red underline signifying that it is marked up. The underline appears in red to signify that you have not yet finished working with the record that you are currently marking up.



*You can see that it is a Key Field markup by the icon that appears at the top left of the markup configuration palette and that the data currently on the page for this markup is the text string 48632084 .*

**WHAT HAPPENS DURING THE UPDATE PROCESS**

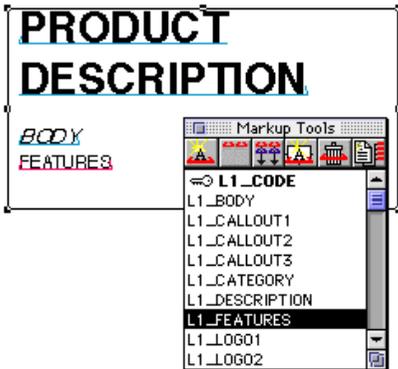
When LinkUP! updates your document it will encounter the marked up value 48632084 and, as a result of your markup, LinkUP! will recognise it as a Key Field value. LinkUP! will then look in the database table being used to update your document and try to find the value 48632084 in the field you specified as the Key Field – L1\_CODE in the case of our example.

Once LinkUP! has found a record in the database table which contains the value 48632084 in its L1\_CODE field, LinkUP! will then look at the markups in your document to find the items that belong to this markup family – items that share the same LRN (or 'surname') as this Key Field markup with the value 48632084. LinkUP! will then use the content of fields in the database record that has the value 48632084 in its L1\_CODE field and use that data to update the other members of the current markup family.

In our example, the text 'Product Description' which was marked up to associate it with the field L1\_DESCRIPTION will be replaced by the text held in the field L1\_DESCRIPTION in the record that has the value 48632084 in its L1\_CODE field.

### ***'hidden' Key Field - data in Key Field does not appear on the page***

If we didn't want the text string **48632084** to appear anywhere in our XPress Document, but still wanted to use the L1\_CODE field in the database as the Key Field, then we can hide the Key Field markup 'behind' any text markup we make in the XPress document for a member of the markup family of page items. Key Field markups cannot be hidden in picture box markups.



To markup our family of items on the page – “Product Description”, “Body” and “Features” involves four basic steps to markup the three items, and a further step to supply a value for the Hidden Key Field.

1. Select the text string FEATURES on the page
2. Holding down the CTRL key, double-click on the L2\_FEATURES fieldname in the Markup Tools palette. This tells LinkUP! that “FEATURES” has a Hidden Key Field...
3. Select the text string BODY on the page and double-click on the L2\_BODY fieldname in the Markup Tools palette.
4. Select the text string PRODUCT DESCRIPTION on the page and double-click on the L2\_DESCRIPTION fieldname in the Markup Tools palette.

You have now marked up a family of items with a Hidden Key Field. Just as with a Visible Key Field markup, LinkUP! needs to know the value of the Key Field that belongs to this family so that it can identify the appropriate database record to use when updating the content of family members. The next step is for you to supply the actual data, placing it in the Hidden Key Field 'behind' your markup of the text string FEATURES. There are two basic ways to do this – a third way of assigning values to Hidden Key Fields is covered in

the *Copying and Pasting Markups* section of this manual, but is not relevant to our current example markup.

In order to provide the value for your Hidden Record Key you either;

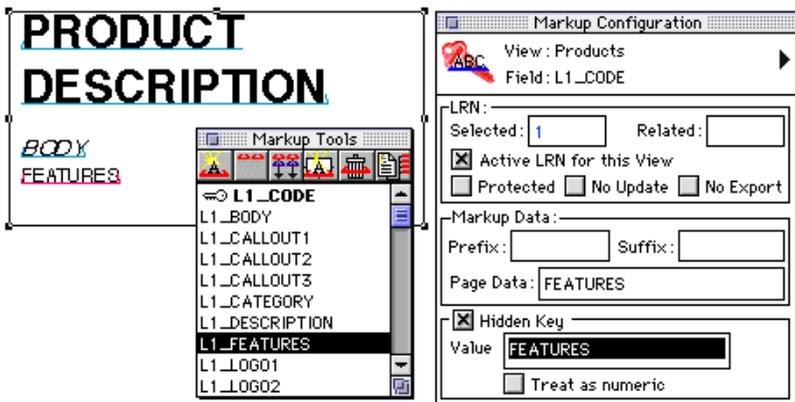
**A** Use the Markup Configuration palette to key in the data value for your hidden Key Field by hand

Or

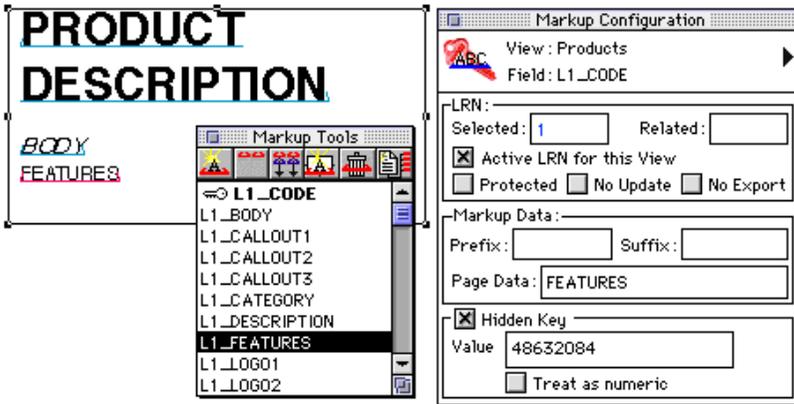
**B** Use the Markup Configuration palette in conjunction with the Data Viewer palette to make a direct selection after looking at the content of records in the database table that will be used to update the document.

### PROVIDING A HIDDEN KEY FIELD VALUE USING THE MARKUP CONFIGURATION PALETTE ONLY

1. Open the Markup Configuration palette
2. Click in the middle of the text string “FEATURES” on the page – note how the Hidden Key symbol appears in the top left of the Markup Configuration palette. The Hidden Key panel at the bottom of the palette window becomes active and the Value: box contains the text string FEATURES
3. Highlight the text FEATURES in the Value: box



4. Type in the value of the Key Field that you have Hidden behind the FEATURES markup – in our example, that value will be 48632084



## PROVIDING A HIDDEN KEY FIELD VALUE USING THE MARKUP CONFIGURATION AND DATA VIEWER PALETTES TOGETHER

- 1 Open the Markup Configuration palette and the Data Viewer palette
- 2 If you want to open additional fields, other than the default Key Field in the columns of the Data Viewer, this is the time to do so. They can help you find the record you want to assign to the current markup family, or check that you are about to assign the correct value to identify the database record you are about to use. Here, we have opened columns to display the contents of the DESCRIPTION, BODY and FEATURES fields of the database table
- 3 Select the row in the Data Viewer that contains the database record that you want to use to update the items in the current markup family. In our example, we need to find the row that has the value 48632084 in its Key Field column

The screenshot displays three windows from a software application:

- PRODUCT DESCRIPTION:** Shows a document with the text "BODY FEATURES" where "FEATURES" is highlighted in red.
- Markup Tools:** A palette with various icons and a list of markup fields: L1\_CODE, L1\_BODY, L1\_CALLOUT1, L1\_CALLOUT2, L1\_CALLOUT3, L1\_CATEGORY, L1\_DESCRIPTION, L1\_FEATURES (selected), L1\_LOGO1, and L1\_LOGO2.
- Markup Configuration:** A dialog box for "View: Products" and "Field: L1\_CODE". It includes:
  - LRN: Selected: 1, Related: (empty)
  - Active LRN for this View
  - Protected,  No Update,  No Export
  - Markup Data: Prefix: (empty), Suffix: (empty)
  - Page Data: FEATURES
  - Hidden Key
  - Value: FEATURES
  - Treat as numeric
- Data Viewer:** A table with columns: TITLE, FEATURES, UNIT\_, and PICTURE1. The table contains several rows of product data.

	TITLE	FEATURES	UNIT_	PICTURE1
11432602	Tai Wireless Keyboard	Advanced plug & play design*Innovative IR	49.45	KEYBOARD.EPS
24731100	Dictaphone Control Kit	Comes with software & user guide*Control all	345.95	DICTAPHONE.EPS
24731425	Derkson V33	P V3.4 (33.6Kbps) data	197.99	V33CARD.EPS
48632084	Virtual Pilot	6 Button Access*Includes Windows 95	79.95	VIRTUAL.EPS
48638911	TV in-a-Box	Composite & s-video output*181	89.49	TV.EPS
52431605	Saddle Bag	Padded notebook compartment*Expandable file	54.99	SADDLE.EPS
52433606	Platinum Nylon	Locaque notebook section*expandable folio	87.99	PLATINUM.EPS

- Place the cursor in the middle of the item on the page that has been marked up with a Hidden Key field – in this example, you would put the cursor in the middle of the 'FEATURES' text on the XPress page.
- Double-click on your selected record in the Data Viewer palette to update your Hidden Key Field markup with the appropriate Key Record value – in this example, 48632084.

## DEALING WITH TABLES

### Overview

Information that is presented on the page in a tabular form is a feature of many XPress documents produced with LinkUP! The next section of this manual deals with the powerful Copy Markup and Paste Markup tools accessed via the Markup Tools palette, and provides an overview of the way in which Key Field values can be assigned to a finished markup automatically – providing your data source has been prepared appropriately.

There are two basic ways in which you can implement tabular information in an XPress document: by using a single text box inside which the text that makes up columns is separated by TABs or other characters; and by using multiple text boxes so that the text for each column in a table is placed inside its own XPress text box.

Three detailed worked examples follow. The first two deal with marking up a table layout in XPress that has been prepared inside a single text box – one shows how to markup such a table with a Visible Key Field, the other shows how it should be marked up using a Hidden Key Field. The third worked example reveals how to markup a table that has been built in the XPress document using multiple text boxes – and shows how to create a markup using a Hidden Key

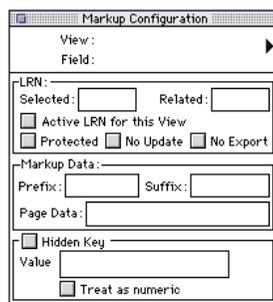
### ***Copy and Paste Markups: Single-Text-Box table, Visible Key Record***

Our example text box contains some heading text, and then three columns – Header, Column1 and Column2 – separated by TABs. The Markup Tools palette contains the names of the three fields that we will use to markup the placeholder text that makes up our table.

Each row in our table consists of three text elements: Header, Column1 and Column2. Each row in the table will be marked up so that 'Header' becomes the Key Field. Column1 will be updated from the 'field' field in the database table, and Column2 will be updated from the 'another field' field. Each row in the table, therefore, will become a marked up family of XPress items. We could perform the markup by hand, row by row by row... but there's an easier way. This is our starting point:

**CASE ONE**  
**Table in single text box**

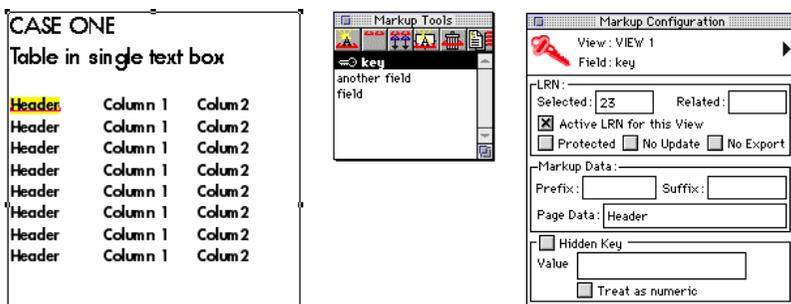
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2



The following is a step-by-step guide to saving time with the Copy Markups and Paste Markups tools...

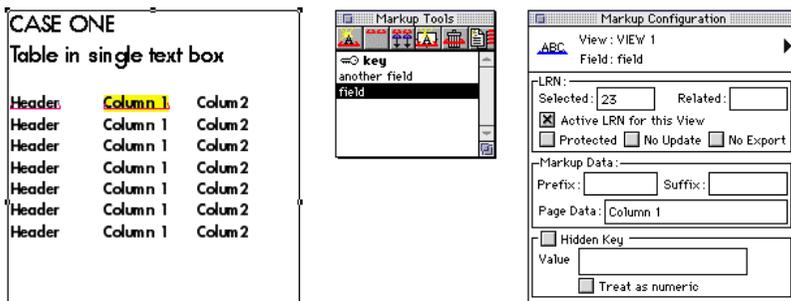
1. Select the text 'Header' in the first row of the first column and double-click on the 'key' field in the Markup Tools palette.

This marks up the first instance of 'Header' in our table as a Visible Key Field. A new family of XPress page items is created by beginning this new record markup, and LinkUP! assigns it the LRN (or 'surname') of 23 - as can be seen from the Markup Configuration palette. You will notice that the key icon appears at the top left of this palette to indicate that the current markup is a Key Field:



2. Select the text 'Column1' in the first row of the first column and double-click on the 'field' field in the Markup Tools palette.

This marks up the first instance of 'Column1' in our table so that it will be updated from the contents of the database field called 'field'. The LRN number of 23 still appears in the Markup Configuration palette, and the icon at the top left of the palette has changed to indicate that this new markup is a text markup.



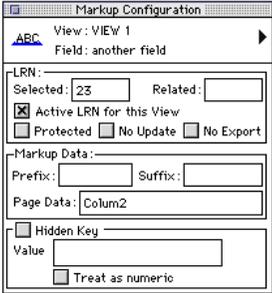
3. Select the text 'Column2' in the first row of the first column and double-click on the 'another field' field in the Markup Tools palette.

This marks up the first instance of 'Column2' in our table so that it will be updated from the contents of the database field called 'another field'. The LRN number of 23 still appears in the Markup Configuration palette, and the icon at the top left of the palette indicates that this new markup is a text markup:

**CASE ONE**  
Table in single text box

Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2





4. The first row of the table has now been marked up as a family of XPress page items. You could cycle round, repeating steps 1 to 3 seven more times to complete the markup of this table or you can select the marked up first row and click on the Copy Markup button in the Markup Tools palette.  This copies the markups you applied to the three text elements of the first row of the example table.

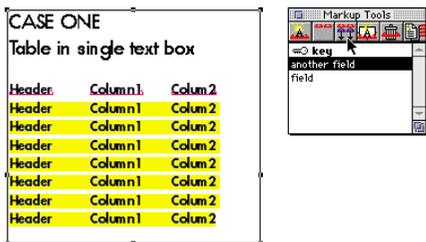
**CASE ONE**  
Table in single text box

Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2





- The next step is to select the seven rows below the first row and then click on the Paste Markup button in the Markup Tools palette...



- LinkUP! will then prompt you to accept the Start LRN value of 24 that it plans to assign as the 'family surname' to the three text items in the first row of your selection.



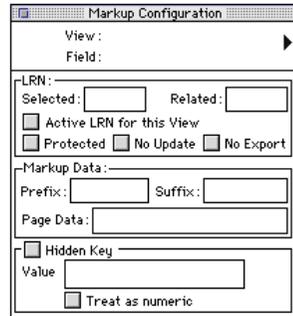
- As the first row in your table that you marked up manually had the LRN of 23, then you should click **OK** in this Paste Markups pop-up dialog box to accept the Start LRN value of 24.

When you do click OK, LinkUP! copies the markup you applied to the first row of the table to the second row and assigns the LRN of 24 to that 'family' of items; it then copies that markup to the third row of the table and assigns the LRN of 25 to that family of items... and so on, until your selected section of table has been marked up for you, automatically. When the process is complete, you can see that all the text placeholders have been marked up by the fact that they are now underlined:

CASE ONE

Table in single text box

Header:	Column 1.	Column 2
Header:	Column 1.	Column 2
Header:	Column 1.	Column 2
Header:	Column 1.	Column 2
Header:	Column 1.	Column 2
Header:	Column 1.	Column 2
Header:	Column 1.	Column 2



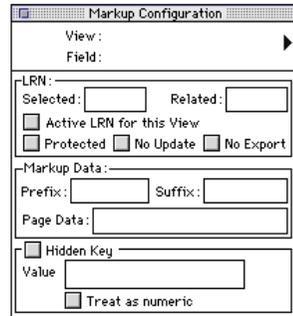
- To check that the markup of the whole table has taken place correctly, you could select the Proof By Field option.

This replaces the text strings you marked up with the names of the fields you assigned to those text strings. (Proof by Field does not change the text on the page that you marked up as a Visible Key Record.)

CASE ONE

Table in single text box

Header:	field	another field

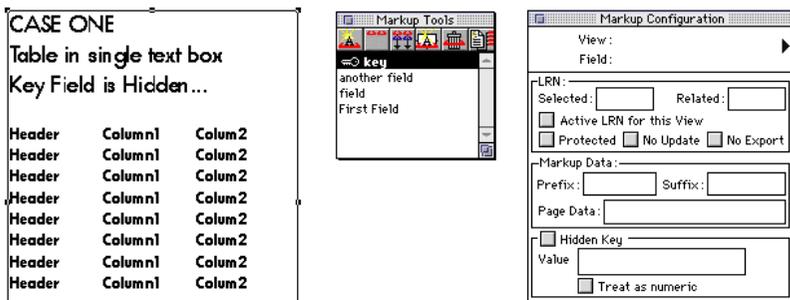


The final stage in our example markup would be to replace the placeholder text string 'Header' that is currently acting as the value of the Key Field for each row of the marked up table. A unique value has to be placed on the page, as a visible Key Field - you will recall that during an Update LinkUP! uses the value that it finds in a Visible Record Key on the page to identify which record in the database table should be used to update a markup. Further details on this process are included in the section 'visible' Key Field - data in Key Field appears on the page on page 124

### Copy and Paste Markups: Single-Text-Box table, Hidden Key Field

Our example text box contains some heading text, and then three columns – Header, Column1 and Column2 – separated by TABs. The Markup Tools palette contains the names of the four fields that we will be marking up the table with. (Four field names appear in the Markup Tools palette because we will be using a Hidden Key Field, rather than a visible Key Field when the table is marked up – as you will see.)

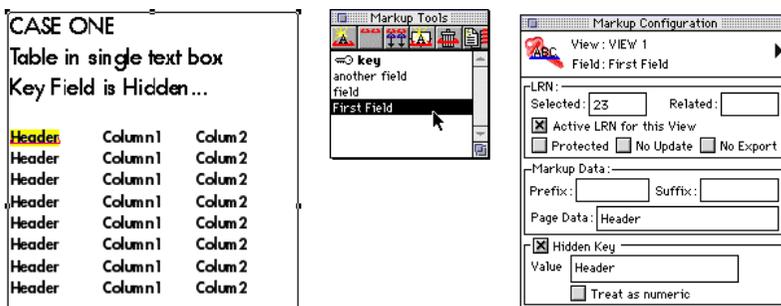
Each row in our table consists of three text elements: Header, Column1 and Column2. Each row in the table will be marked up so that the ‘Header’ becomes the First Field which ‘hides’ the Key Field. Column1 will be updated from the ‘field’ field in the database table, and Column2 will be updated from the ‘another field’ field. Each row in the table, therefore, will become a marked up family of XPress items. We could perform the markup by hand, row by row by row, but there’s an easier way. This is our starting point:



The following is a step-by-step guide to saving time with the Copy Markups and Paste Markups tools...

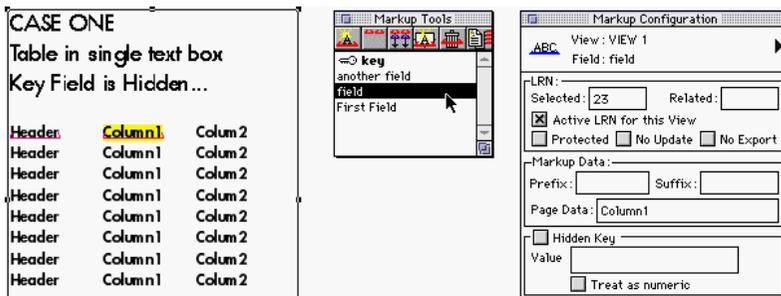
1. Select the text ‘Header’ in the first row of the first column, hold down the CTRL key and double-click on the ‘First Field’ field in the Markup Tools palette.

This marks up the first instance of ‘Header’ in our table as a text placeholder that will be updated by the contents of the field ‘First Field’ and tells LinkUP! that there is a Hidden Key Field associated with this markup – as you can see from the icon at the top left of the Markup Configuration palette. This palette also shows that the LRN number of 23 has been assigned to this new family of XPress page items you have started to mark. The Hidden Key section at the bottom of the palette shows that a Hidden Key is associated with this markup, and that the value of the Hidden Key is currently ‘Header’:



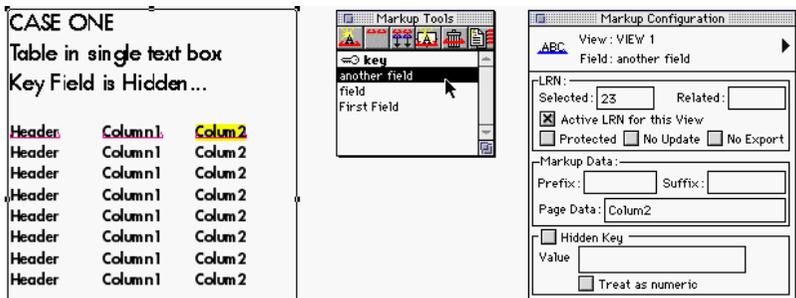
2. Select the text 'Column1' in the first row of the first column and double-click on the 'field' field in the Markup Tools palette.

This marks up the first instance of 'Column1' in our table so that it will be updated from the contents of the database field called 'field'. The LRN number of 23 still appears in the Markup Configuration palette, and the icon at the top left of the palette has changed to indicate that this new markup is a text markup:

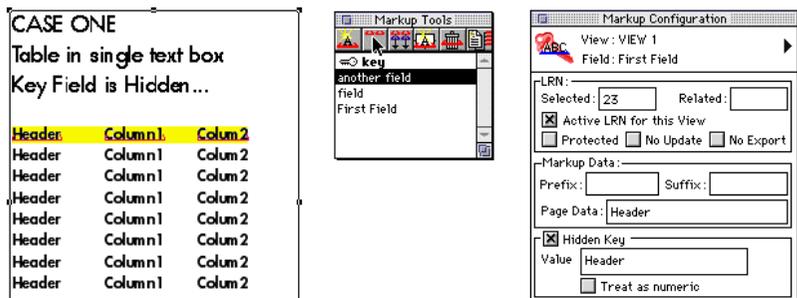


3. Select the text 'Column2' in the first row of the first column and double-click on the 'another field' field in the Markup Tools palette.

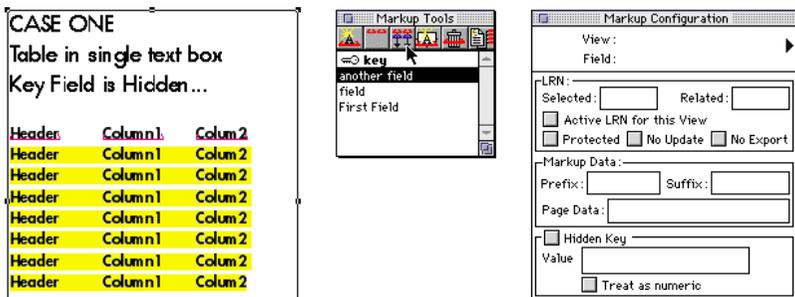
This marks up the first instance of 'Column2' in our table so that it will be updated from the contents of the database field called 'another field'. The LRN number of 23 still appears in the Markup Configuration palette, and the icon at the top left of the palette indicates that this new markup is a text markup:



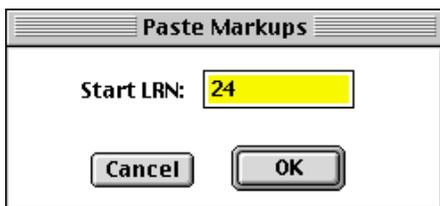
4. The first row of the table has now been marked up as a family of XPress page items. You could cycle round, repeating steps 1 to 3 seven more times to complete the markup of this table, or you can select the marked up first row and click on the Copy Markup button in the Markup Tools palette.  This copies the markups you applied to the three text elements of the first row of the example table.



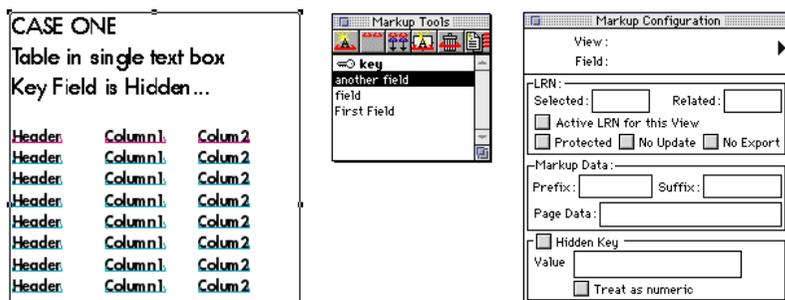
5. The next step is to select the seven rows below the first row and then click on the Paste Markup button in the Markup Tools palette. 



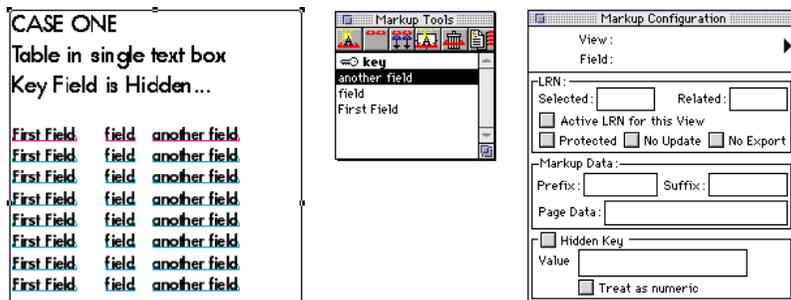
- LinkUP! will then prompt you to accept the Start LRN value of 24 that it plans to assign as the 'family surname' to the three text items in the first row of your selection.



As the first row in your table that you marked up manually had the LRN of 23, then you should click OK in this Paste Markups pop-up dialog to accept the Start LRN value of 24. When you do click OK, LinkUP! copies the markup you applied to the first row of the table to the second row and assigns the LRN 24, then copies that markup to the third row of the table and assigns the LRN of 25 to that family of items... and so on, until your selected section of table has been marked up for you, automatically. When the process is complete, you can see the markup by the underlines that appear under the text:



- To check that the markup of the whole table has taken place correctly, then you could select the Proof By Field option which replaces the text strings you marked up with the names of the fields you assigned to those text strings.



The final stage in our example markup would be to replace the placeholder text string 'Header' that is currently acting as the value of the Key Field for each row of the marked up table. A unique value has to be placed on the page, as a visible Key Field - you will recall that during an Update LinkUP! uses the value that it finds in a Visible Record Key on the to identify which record in the database table should be used to update a markup. Further details on this process appear at the end of *Text markups: 'hidden' Key Field - data in Key Field does not appear on the page* sub-section of this manual.

### ASSIGNING VALUES TO HIDDEN KEY FIELDS THE THIRD WAY!

However, there is a way that that LinkUP! can assign values automatically to a sequence of Hidden Key Field markups – such as the rows in the table we have just marked up using Copy Markups and Paste Markups. In order for LinkUP! to be able to provide a value for the Hidden Key automatically, your database needs to have sequential numbers as the data in the field that you have assigned as the Key Field in the database table.

You are most likely to have a sequence of numbers in the database field that LinkUP! uses as the Key Field if your data has been imported from a spreadsheet with row numbers preserved as one of the fields. You can use LinkDB to create a suitable file with sequential numbers in a field that can be used as Key Field if you are starting with an ASCII export file from a database. Or you may wish to arrange for your database table to contain sequential numbers in a field, so you can take advantage of this special, automatic way of assigning

values to a sequence of Hidden Key Fields in a markup. Some of these issues are covered in more detail elsewhere in this manual.

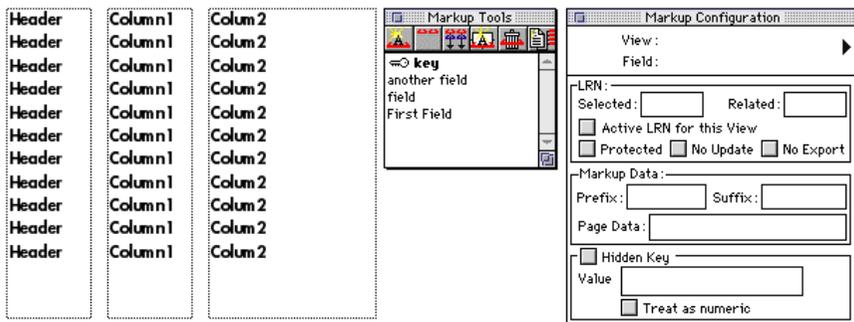
If the database table that will be used to update your XPress document contains sequential numbers in the field that LinkUP! will use as the Key Field, then you follow exactly the same procedure to markup the table as set out above – but at the end of Step 1 you check the Treat As Numeric button at the bottom of the Markup Configuration palette and enter a numeric value in the Value box. The numeric value that you need to enter is the numeric value that appears in the Key Field of the database record from which LinkUP! will source data to update the family of marked up text items in the first row of the table on the page.

When you move on to Paste Markups, LinkUP! will automatically update the Value: boxes for each of the Hidden Key Fields in your table markup, incrementing the numeric value by one for each row of the table on the page that is being marked up.

**Copy and Paste Markups: Multi-Text-Box table, Hidden Key Field**

This time, our example table on the page is built from four text boxes. A horizontally-oriented text box contains some heading text, and then three vertical text boxes hold the text for the three columns of our table – Header, Column1 and Column2. The Markup Tools palette contains the names of the four fields that we will be marking up the table with. (Four field names appear in the Markup Tools palette because we will be using a Hidden Key Field, rather than a visible Key Field when the table is marked up – as you will see.)

Each row in our table consists of three text elements: Header, Column1 and Column2. Each row in the table will be marked up so that the ‘Header’ becomes the First Field which ‘hides’ the Key Field. Column1 will be updated from the ‘field’ field in the database table, and Column2 will be updated from the ‘another field’ field. Each row in the table, therefore, will become a marked up family of XPress items. We could perform the markup by hand, row by row by row, but there is a quicker and easier way. This is our start point:



The following is a step-by-step guide to saving time with the Copy Markups and Paste Markups tools...

1. Select the text ‘Header’ in the first row of the first column, hold down the CTRL key and double-click on the ‘First Field’ field in the Markup Tools palette. This marks up the first instance of ‘Header’ in our table as a text placeholder that will be updated by the contents of the field ‘First Field’ and tells LinkUP! that there is a Hidden Key Field associated with this markup – as you can see from the icon at the top left of the Markup Configuration palette. This palette also shows that the LRN number of 31 has been assigned to this new family of XPress page items you have started to mark. The Hidden Key section at the bottom of the palette shows that a Hidden Key is associated with this markup, and that the value of the Hidden Key is currently ‘Header’:

Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2

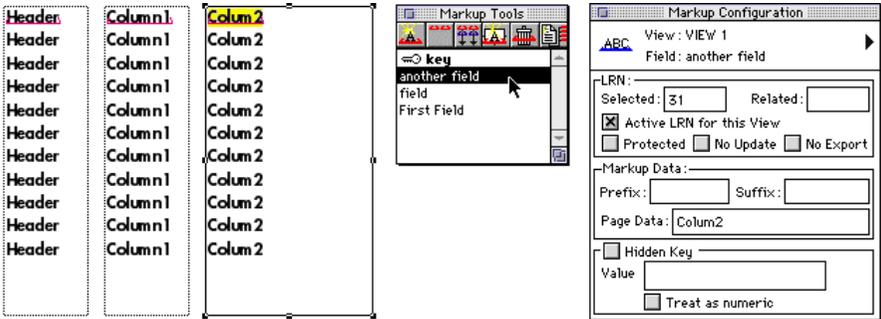


2. Select the text 'Column1' in the first row of the first column and double-click on the 'field' field in the Markup Tools palette. This marks up the first instance of 'Column1' in our table so that it will be updated from the contents of the database field called 'field'. The LRN number of 31 still appears in the Markup Configuration palette, and the icon at the top left of the palette has changed to indicate that this new markup is a text markup:

Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2
Header	Column 1	Column 2

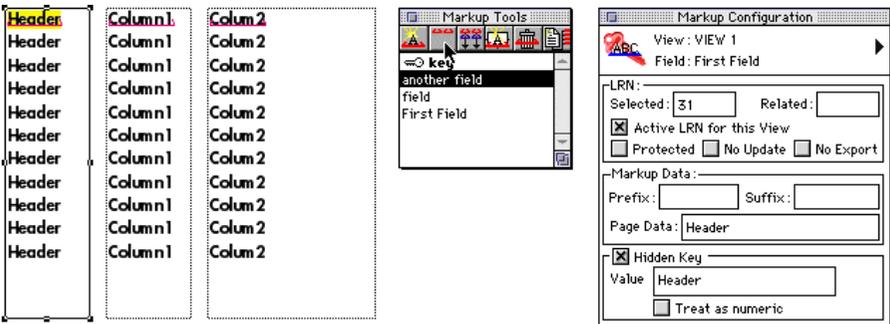


3. Select the text 'Column2' in the first row of the first column and double-click on the 'another field' field in the Markup Tools palette. This marks up the first instance of 'Column2' in our table so that it will be updated from the contents of the database field called 'another field'. The LRN number of 31 still appears in the Markup Configuration palette, and the icon at the top left of the palette indicates that this new markup is a text markup:

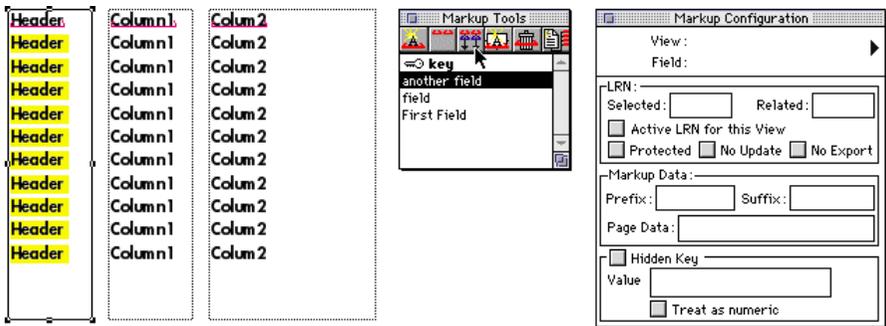


- The first row of the table has now been marked up as a family of XPress page items. You could cycle round, repeating steps 1 to 3 ten more times to complete the markup of this table, or you can use the Copy Markup and Paste Markup tools to speed things up...

Select the marked up text 'Header' at the top of the text box that forms the left-hand column of the table on your page and click on the Copy Markup Button in the Markup Tools palette. This copies the markup you applied to 'Header' in the first line of text in this text box.



- The next step is to select the nine rows below this first row and then click on the Paste Markup button in the Markup Tools palette...



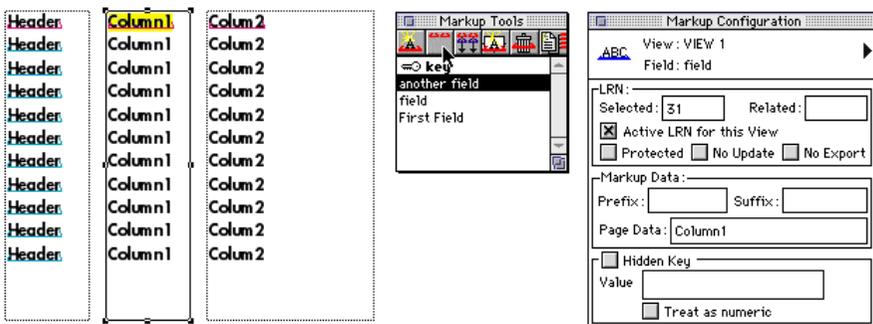
6. LinkUP! will then prompt you to accept the Start LRN value of 32 that it plans to assign as the 'family surname' to the first text item in your selection.



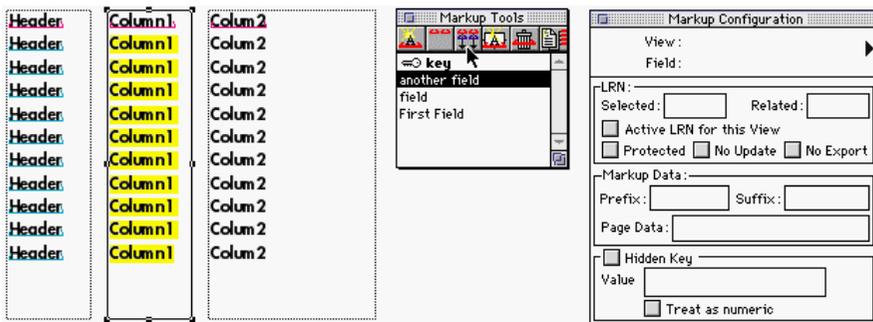
As the first instance of 'Header' in this text box that you marked up manually had the LRN of 31, then you should click OK in this Paste Markups pop-up dialog to accept the Start LRN value of 32.

When you do click OK, LinkUP! copies the markup you applied to the first instance of 'Header' to the second instance and assigns the LRN 32, then copies that markup to the third instance and assigns the LRN of 33 to that instance of 'Header'... and so on, until the contents of this text box – which is the leftmost column of the table on your XPress page - has been marked up for you, automatically.

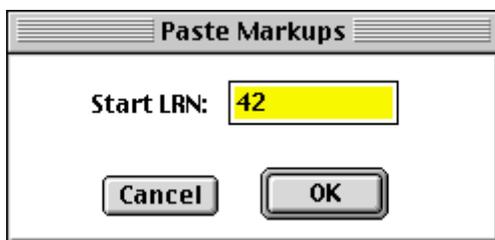
- 7 The next step is to copy and paste the markup to all the text in the middle text box. Select the marked up text 'Column1' at the top of the text box that forms the middle column of the table on your page and click on the Copy Markup Button in the Markup Tools palette. This copies the markup you applied to 'Column1' in the first line of text in this text box.



8. Then select the nine rows below this first row and then click on the Paste Markup button in the Markup Tools palette...



9. LinkUP! will then prompt you to accept the Start LRN value of 42 that it plans to assign as the 'family surname' to the first text item in your selection.



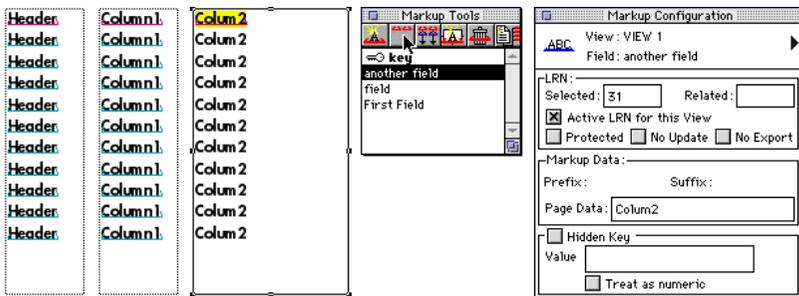
As the first instance of 'Column1' in this text box that you marked up manually had the LRN of 31, then you need to change the Start LRN value from LinkUP!'s suggestion of 42 to 32. In our example, when you copied and pasted the markup for 'Header' in the leftmost textbox

of the XPress table, the second instance of 'Header' was assigned the LRN value of 32 – so to preserve the horizontal relationship between instances of 'Header' and 'Column1' when you paste the copied markup, you need to ensure that the LRN numbers match up, 'row' by 'row'.

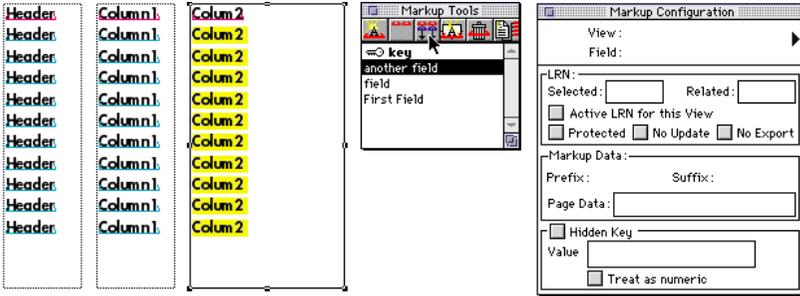


After changing LinkUP!'s suggested Start LRN to the Start LRN used during the Paste Markup operation you performed on the textbox table column to the left, click OK. LinkUP! copies the markup you applied to the first instance of 'Column1' to the second instance and assigns the LRN 32, then copies that markup to the third instance and assigns the LRN of 33 to that markup of 'Column1'... and so on, until the contents of this text box – which is the middle column of our example table on the XPress page - has been marked up for you, automatically.

- The next step is to copy and paste the markup to all the text in the right-hand text box. Select the marked up text 'Column2' at the top of the text box that forms the rightmost column of the table on your page and click on the Copy Markup Button in the Markup Tools palette. This copies the markup you applied to 'Column2' in the first line of text in this text box.



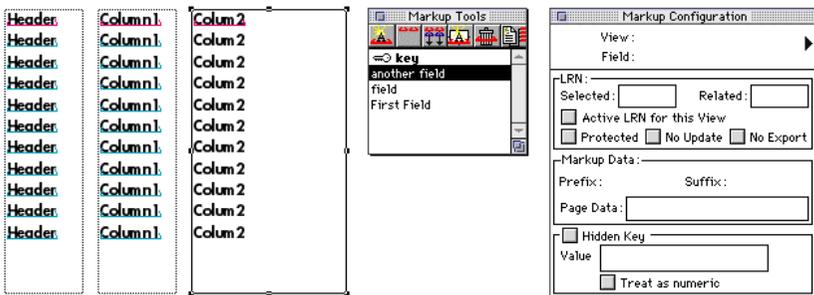
- Then select the nine rows below this first row and then click on the Paste Markup button in the Markup Tools palette...



12. LinkUP! will then prompt you to accept the Start LRN value of 42 that it plans to assign as the 'family surname' to the first text item in your selection on the page.



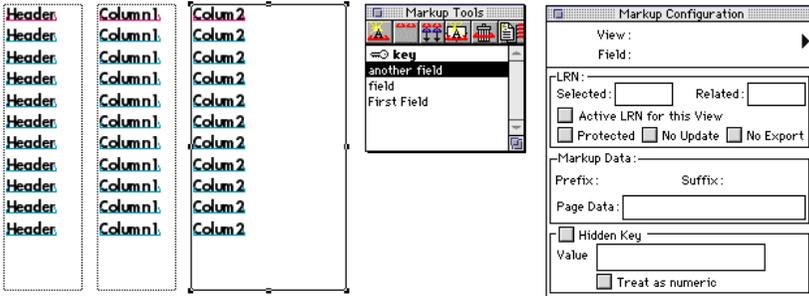
As the first instance of 'Column2' in this text box that you marked up manually had the LRN of 31, then you need to change the Start LRN value from LinkUP!'s suggestion of 42 to 32. In our example, when you copied and pasted the markups for the text in the two textboxes to the left in the XPress table, the second instance of the text in each case was assigned the LRN value of 32. To preserve the horizontal relationship between instances of 'Header', 'Column1' and 'Column2' when you paste the copied markup, you need to ensure that the LRN numbers match up, 'row' by 'row'.



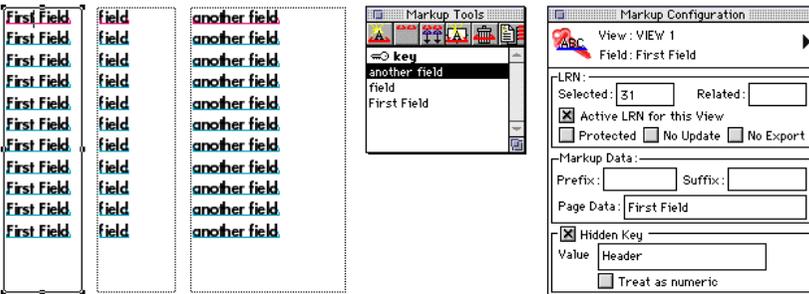
After changing LinkUP!'s suggested Start LRN to the Start LRN used during the Paste Markup operation you performed on the textbox table columns to the left, click OK. LinkUP! copies the markup you applied to the first instance of 'Column2' to the second instance and

assigns the LRN 32, then copies that markup to the third instance and assigns the LRN of 33 to that markup of 'Column2'... and so on, until the contents of this text box – which is the middle column of our example table on the XPress page - has been marked up for you, automatically.

Your markup of the XPress table is complete – as can be seen by the underlined text items in the three textboxes used to hold text for the columns...



A quick check using LinkUP! Proof By Field tool shows that the markup has been achieved correctly...



...but you will still need to assign real values to the Hidden keys behind your markups in the first column of your XPress table. Details on how to do this can be found after Step 7 in the **Copy and Paste Markups: Single-Text-Box table, Hidden Key Field** section of this manual.

**Copy and Paste Markups: Multi-Text-Box table, Visible Key Field**

Essentially you need to follow all the steps in our worked example for marking up a multi-text-box table with a Hidden Key, but you will only use three field names in the Markup Tools palette and will markup the 'Header' text directly as a field rather than as a hidden key field.

If you have any difficulty, compare the worked example of marking up a single-text-box-table with a Hidden Key field with the worked example of marking up a single-text-box table with a Visible Key Field.

## Picture markups

LinkUP! references pictures based on the picture file name. By having a field in the database that holds the name of the file to be used, LinkUP! can place the picture onto the page. Essentially, marking up a picture involves the same process as marking up text.

1. Select the target XPress picture box
2. Double-click on the field name in the Markup Tools palette that contains the picture name data

It is worth noting that you cannot assign a Hidden Key Field to a picture box. When the picture is brought onto the XPress page during an update (see page 162) it is sized to fit the picture box. If a picture is not found, the picture box is blanked. There are various options that can be configured for the picture fields such as delete if empty. Details on these are found in the section of this manual on page 110.

## Editing Markups

Once you have marked up key fields and elements on the QuarkXPress page you should, where possible, use the Markup Configuration palette in conjunction with the Markup Tools palette to alter them if necessary. Care is needed when you edit markups on the XPress page: it is all too easy to accidentally delete a markup by deleting its end or beginning character.

The Markup Configuration palette allows you to check markups on the page once you have made them, allows changes to be made to existing markups, and gives access to more advanced markup facilities required for creating relational markups, which are essential for template building. See *Relational Markups* for a guide to these more advanced markup features.

If you have the Markup Configuration palette open, you can click the cursor on any markup in your XPress document and the Markup Configuration palette will display all the attributes of that single markup. If you then wish to edit that markup, you first need to make the family of markups that the markup belongs to active, i.e. you need to make the correct LRN active. Checking the Active LRN for this View button in the Markup Configuration Palette achieves this making the family of marked up items than belongs to the markup you selected on the page the active family that you are currently marking up – markups that belong to this family will now be highlighted on your XPress page by red underlines.

## Marking up an item out of sequence

When you markup your document all of the XPress elements on the page that, between them, will hold the content of fields from a single record in the database, will share a single LRN and a single Key Field, hidden or visible.

If you need to return to your marked up XPress page to re-markup an element you have already marked up or add a new XPress element to hold the contents of another field from a record in the database, you need start working with the appropriate Key Field and therefore have the correct LRN active.

1. With the Markup Tools and the Markup Configuration palettes open, click on one of the marked up XPress items that will be used to hold one of the fields from the record that you want to be working with.
2. Check the Active LRN for this View button in the Markup Configuration palette.

You will notice that a set of markups on the XPress page will now be underlined in red.

3. You can now either re-mark one of the marked up elements or add a new element and mark it up so that it will be updated by a field from the currently active record in the database file.

The record that LinkUP! will use in a future update is, of course, identified by the Key Record now highlighted in your XPress document with a red underline.

## Deleting a Markup

There are two ways of deleting items that you have marked up. You can just delete XPress page elements from your document, but in most cases you are more likely to want to remove a markup from an item in your document without deleting the page element itself.

If you select a section of text that has been marked up, a picture box that has been marked up, or a Header/Footer box that has been marked up and then click on the Clear Markup button in the Markup Tools palette, then the selected markup is removed from the item you selected.

The clear markup button can be found on both the Markup Tools Palette and the Index Tools Palette.

To achieve this:

1. On your XPress page, select and highlight the marked up item (or items) for which you want to delete the LinkUP! markup(s)
2. Click the Clear Markup Button in the Markup Tools Palette 

# Relational Markups

## WHAT ARE THEY?

The fundamental difference between marking up a non-relational structure and marking up a relational structure is the ordered way in which it must take place and the rules that must be followed. The actual process of highlighting the text and double-clicking on the field in the Markup Tools palette remains exactly the same (see page 117 for details).

## WHO NEEDS TO DO IT?

The answer is anyone who wants to gain the benefits of connecting XPress pages to a relational database structure. Of course even if your publication can be approximated to a relational structure this doesn't mean you have to mark it up relationally. You proceed to markup the pages in the standard way where every element on the page must use the same field in the database as the Key Field. However if you already have a relational database to connect to, you would be far better advised to markup relational structures and make the productivity gains that LinkUP! 7 can bring you.

## HOW DO I DO IT?

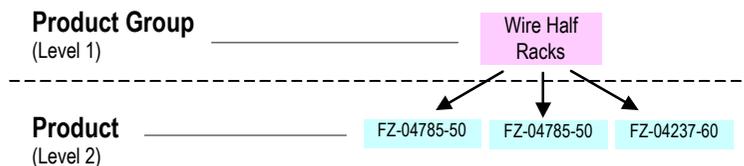
The overall premise of marking up relational structures is that you must Markup the parent before you Markup up the child. If you look at the portion of an XPress page below, you may be able to see that we have a relational structure, with a group level and a product level. We can see this because the Group information i.e. the heading and descriptive text is common to all of the products in the table below it.

### F • Wire Half Racks

Half racks save space—they're half as long as full-size WIRE racks with the same durable, glass-reinforced acetal construction. Choose from four test tube sizes. Autoclavable. NALGENE. **8 ea/cs\***.

Catalog number	Tube size	Tubes held (array)	Dimensions L x W x H (mm)	Color	Price/ea
<b>FZ-04785-50</b>	14 mm	36 (6 x 6)	102 x 102 x 56	Green	<b>£21.99</b>
<b>FZ-04456-55</b>	14 mm	36 (6 x 6)	102 x 102 x 56	Blue	<b>£4.99</b>
<b>FZ-04237-60</b>	16 mm	36 (6 x 6)	126 x 126 x 68	Green	<b>£35.95</b>
<b>FZ-04347-65</b>	16 mm	36 (6 x 6)	126 x 126 x 68	Blue	<b>£7.49</b>
<b>FZ-04754-70</b>	21 mm	21 (4 x 5)	128 x 103 x 83	Green	<b>£32.05</b>
<b>FZ-04258-75</b>	21 mm	201 (4 x 5)	128 x 103 x 83	Blue	<b>6.95</b>
<b>FZ-04578-80</b>	26 mm	16 (4 x 4)	122 x 122 x 75	Green	<b>£32.95</b>
<b>FZ-04415-83</b>	26 mm	16 (4 x 4)	122 x 122 x 75	Red	<b>£7.49</b>

A good way to enable you to analyse your XPress pages and decide what the structure is, is to sketch out a diagram showing how it fits together. For instance, for the page above, you will see we have illustrated the XPress page as a two level relational structure below.



To markup this catalogue page you would therefore need to define two views, one view for the group information and another view for the product information (see Views on page 105).

Where as with a non-relational structure you have only a single Key Field (see page 41), for a relational structure, such as the one above, you have to define a Key Field for each View. The Key Field must contain unique value for each record in the database. The Key Field for the Group Level would probably be the letter **F** found at the beginning of the group. The Key Field for the Product level would be the stock number found at the beginning of each row. To markup this example you would need to markup the Key Field for the group level (i.e. **F**) followed by the fields for the group level (description, title etc). Once all of the group level fields have been marked up, you then markup the Key Field for the first product (FZ-04785-50), followed by the fields for that product (Tube Size, Tubes Held etc) then the Key Field for the next product (FZ04456-55) followed by the fields for that product and so on until all of the products for Group **F** have been marked up. You would then be ready to move onto Group **G**, followed by all of the products in Group **G**.

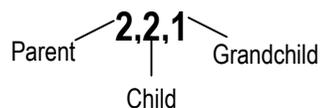
This is obviously a relatively simple example with only two levels, but the same principal applies if there are more levels. You always start with the highest level and work your way down from parent to child to grandchild. i.e. from level 1 to level 2 to level 3 etc.

The following page has been designed as a guide to the procedure for marking up relational structures.

Stage	1 <sup>st</sup> Level (Parent Record)	2 <sup>nd</sup> Level (Child Record of 1 <sup>st</sup> Level)	3 <sup>rd</sup> Level (Child Record of 2 <sup>nd</sup> Level)	etc.
1	Key Field 1			
2	Fields 1			
3		Key Field 1,1		
4		Fields 1,1		
5			Key Field 1,1,1	
6			Fields 1,1,1	
7			Key Field 1,1,2	
8			Fields 1,1,2	
9		Key Field 1,2		
10		Fields 1,2		
11			Key Field 1,2,1	
12			Fields 1,2,1	
13			Key Field 1,2,2	
14			Fields 1,2,2	
15	Key Field 2			
16	Fields 2			
17		Key Field 2,1		
18		Fields 2,1		
19			Key Field 2,1,1	
20			Fields 2,1,1	
21			Key Field 2,1,2	
22			Fields 2,1,2	
23		Key Field 2,2		
24		Fields 2,2		
25			Key Field 2,2,1	
26			Fields 2,2,1	
27			Key Field 2,2,2	
28			Fields 2,2,2	

**KEY**

**Key Field** is an abbreviation for Markup Key Field  
**Fields** is an abbreviation for Markup Fields



In the above table you will see that the process involves starting at the first level for a record and moving down the chain to the second and third level. This example can be extended for as many levels as are present in the publication and for as many records are in the publication. For instance after stage 14 you could have another 10 child records belonging to parent in the 1<sup>st</sup> Level. These would therefore all be marked up first, along with their child records etc. before moving onto the next parent.

### ADDING A NEW FIELD TO AN EXISTING MARKUP LEVEL

To add a new field to an existing markup level you first need to make the LRN (see page 42) for the record in that level to which the field needs to be added to, the Active LRN.

To do this;

1. Click on any of the existing marked fields for the record in the level to which the field needs to be added
2. In the Markup Configuration Palette click the **Active LRN for this View** option
3. Now markup the element in the usual way by highlighting the piece of text or picture to be marked up, and double-clicking on the field in the Markup Tools Palette

### ADDING A NEW CHILD TO ITS PARENT

In this case you are adding a new child record to a parent record, i.e. to a record in the level above in the hierarchical structure. If we relate this back to the example below, we are adding a new product into the Wire Half Racks group.

## F • Wire Half Racks

Half racks save space—they're half as long as full-size WIRE racks with the same durable, glass-reinforced acetal construction. Choose from four test tube sizes. Autoclavable. NALGENE. **8 ea/cs\***.

Catalog number	Tube size	Tubes held (array)	Dimensions L x W x H (mm)	Color	Price/ea
<b>FZ-04785-50</b>	14 mm	36 (6 x 6)	102 x 102 x 56	Green	<b>£21.99</b>
<b>FZ-04456-55</b>	14 mm	36 (6 x 6)	102 x 102 x 56	Blue	<b>£4.99</b>
<b>FZ-04237-60</b>	16 mm	36 (6 x 6)	126 x 126 x 68	Green	<b>£35.95</b>
<b>FZ-04347-65</b>	16 mm	36 (6 x 6)	126 x 126 x 68	Blue	<b>£7.49</b>
<b>FZ-04754-70</b>	21 mm	21 (4 x 5)	128 x 103 x 83	Green	<b>£32.05</b>
<b>FZ-04258-75</b>	21 mm	201 (4 x 5)	128 x 103 x 83	Blue	<b>6.95</b>
<b>FZ-04578-80</b>	26 mm	16 (4 x 4)	122 x 122 x 75	Green	<b>£32.95</b>
<b>FZ-04415-83</b>	26 mm	16 (4 x 4)	122 x 122 x 75	Red	<b>£7.49</b>

In this case you must make the parent the Active LRN before marking up the new record. In the above example you would make the Wire Half Racks the Active LRN and then markup the new product being added. So to achieve this;

1. Click on any of the fields in the Parent Record to which the child record is to be added
2. In the Markup Configuration Palette click the **Active LRN for this View** option
3. Now markup the element in the usual way by highlighting the piece of text or picture to be marked up, and double-clicking on the field in the Markup Tools Palette. Remember to markup the Key Field first.

For further information see *Relational Reference* on page 184

# Projects

## Working with Project Files

### OPERATOR PERSPECTIVE

A Project File contains all the information LinkUP! needs in order to work with a particular database file in a particular way. By loading a Project File you setup a link between the current document and the database file specified in the project file.

Several Project Files might be set up for one document. For example to produce a catalogue, you might need to access two different database files. One might contain the pricing and descriptive information and another the picture references. You would therefore use two Project Files to change between the two database files. A third Project File might be set up so that a link was made to a single field, the price field perhaps, in one of the two database files used to produce the document. This third Project File, which gives access to price data only, could be used to make last-minute price updates once the remainder of the document had been signed off as ready for film or plate.

One of the most common uses of multiple project files is where you want to produce a different version of the same publication but using different data for instance a foreign language version or a no pricing version. This is simply a case of pointing LinkUP! to a different project and re-updating the publication.

### *Loading a Project File*

This is a simple process and just involves selecting the appropriate project file from your hard disk or network drive. To load a project

1. Click the load project button on the Data Viewer Palette 

You are now presented with the file selection dialogue

2. Locate and select the project file
3. Click OK

Your project file has now been loaded, over-writing any previous configuration setting for you current document.

To load a different project simply repeat the above procedure.

## **ADMINISTRATOR PERSPECTIVE**

A Project File contains all the information that LinkUP! needs in order to make a link to the content of a single database file. It contains a list of Field Names together with definitions of all the connections for each item in that field list, details of the path on which picture files will be found, details of any relationships between fields and the field configuration parameters which include what happens during an import when a field is empty, whether there should be a prefix or suffix to the data, whether XPress Tags should be imported from a field, whether there should be a static label before or after a field, and whether (and how) data can be written back from the XPress document to a database file.

### ***Setting Up A New Project File***

A project file is simply a snapshot of the configuration of your current document at the point you save a project. Therefore configuring a project is actually a case of configuring all of the various aspects of the XPress document that are saved in a project. The following is a list of all of the information that is saved to a project file.

- Database Import Links
- Database Export Links
- Import Link Configuration
- Export Link Configuration
- Data Source Configuration
- Relationship Configuration
- Picture Paths
- Character Maps

You find details on how to configure all of the above components elsewhere in the manual. For instance if you need to know about configuring database links you should look at on page 110.

To create a new project file, once you are satisfied that all of the above configuration is correct;

1. From the LinkUP! Menu select **Configure Save Project**

You are now presented with the file selection dialogue

2. Type in a name for your new project file
3. Click the **OK** Button

### ***Editing an Existing Project File***

To edit an existing project you must first load that project by clicking the load project button or from the LinkUP! Menu selecting **Configure Load Project** 

Once the project has been loaded you can make changes to all of the configuration for that particular document. Once you have completed changing the configuration, save the project again.

### ***Advice and hints***

There are several things you should bear in mind when create project files for workgroup use. The first of these is the role that paths play within the project. This applies specifically to data source paths and picture paths. As LinkUP! works with absolute paths, if you use a project created on one Macintosh, on a different Macintosh, you will need reconfigure the picture paths to allow for this. Otherwise LinkUP! will fail to find the pictures on update.

If you are working with the LinkUP! 7 LinkDB XTension, and your data files are stored on a network drive, data source setups will work no matter which computer the project is used on (providing the path to the network drive is the same on each machine).

When using the LinkUP! 7 68k or PPC XTension, make sure that the data source setup on each machine that needs to use the project, is identical. This way, no reconfiguration will be necessary.

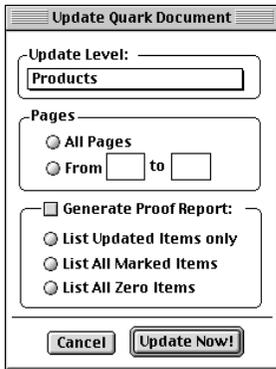
# Updating

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Data Viewer Palette	50
Markup Configuration Palette	47

LinkUP! does not alter any information within your document until you instruct it to do so. When you are ready to have the latest data placed onto your page, choose the Update Button on the Data Viewer Palette, at which point you will be presented with the following dialogue:



Before performing the update, certain parameters can be specified to define LinkUP!'s behaviour during the update process. The first selection that must be made is to identify the pre-processed database that LinkUP! is to use for this update. This is achieved by pressing the Select Database button, and then using the dialogue to identify the database required for the update. Note that no other update options are available until the update database has been successfully chosen.

If your document is linked to multiple databases, then a separate update must be performed for each of these databases. LinkUP! only concerns itself with the data relevant to the chosen database on each update.

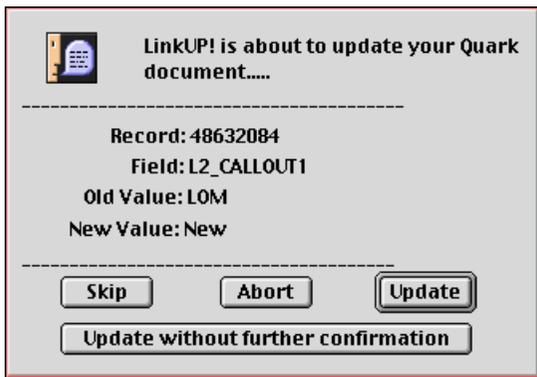
## Restricting the update range

By default, LinkUP! will always attempt to update the entire document that you are working in. If you wish, however, the action can be narrowed to work only with a certain page range.

Note: The page identifiers that you identify must correspond to the page numbering scheme that you have chosen to use within Quark. In other words, if you have defined a section to follow a particular page numbering style, then you must use this style to refer to the pages within your document.

The final button on the Database Update dialogue is the one that actually causes LinkUP! to process the document in the required manner, which is the Update Now button. When you press this button, LinkUP! will firstly prepare the information that it is required to update. If you have asked for a Proof report, then you will be prompted for the name of a file to hold this report. Note that this file is a pure ASCII format, suitable for reloading into Quark using the standard Get Text option from Quark's File Menu.

When LinkUP! is ready to make the first data change in the document it will then present you with the following confirmation dialogue:



In the Dialogue, LinkUP! informs you of the name of the field and Key field that it is working with. It will also show both the current content of the field and the proposed new value for that field. Note that for long text fields only the first part of the text field and its replacement is displayed.

If you choose the update button, LinkUP! will make the replacement, and then step through to the next point and redisplay the confirmation dialogue. If you do not want LinkUP! to make

the replacement, then asking it to skip will leave the current value intact, before moving onto the next and asking again for confirmation.

The update can be cancelled by choosing the ABORT button, or if you are happy to allow the update to continue without manual intervention, you can ask LinkUP! to proceed to update without further confirmation.

As the update takes place, LinkUP! maintains a counter and progress bar, indicating its progress through the document.

You can cancel the update whilst it is in process by pressing the CANCEL button on the progress bar. If you cancel an update using either of these methods, or by choosing the Abort key in the confirmation dialogue, then LinkUP! would stop further updates. Any updates that have taken place before abort point would now be part of your document.

# Working with Pictures

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Data Viewer Palette	50
Markup Configuration Palette	47

To mark a picture box for LinkUP! use, you define a field that will refer to the picture file name, and then highlight a picture box and double-click the field in the LinkUP! palette for Markup in the usual way. The picture field is then configured to a field in the database that will contain the picture file name.

If we have a record in the database that has the following fields:

Name: Fred

Age: 33

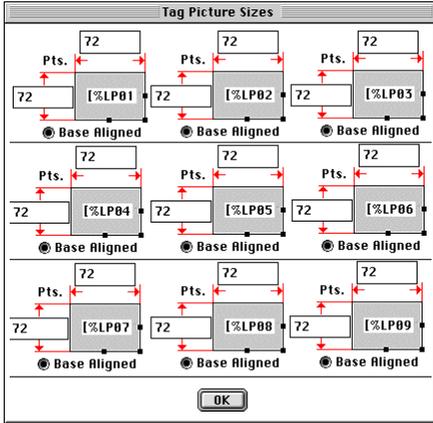
Photo: FRED1.TIF

you could then define a field in LinkUP! called 'Picture1' and apply this to a picture box. If the Picture1 field is then configured to link to the 'Photo' field in the database, then on update we would search for FRED1.TIF in the designated picture folder and perform a best fit to place it in the box. Note that the picture box is linked to a Key field in exactly the same way as a normal textual piece of information.

# Configuring Picture Tags

The Dialogue for each of the nine configurable picture parameters looks like this:

This particular example would produce base aligned embedded picture with the dimensions of 72x72 points (1 inch=72pts). Note that the significance of the base alignment of an embedded box is identical to that for a standard anchored picture box.



# Creating Indexes From your QuarkXPress Documents

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Index Tools Palette	49
Marking Up	117

## Indexing Fundamentals

LinkUP! allows you to generate an Index for either a single document, or for a group of documents that might constitute an entire publication. The Index entries can be inserted at any point in the document, and can either draw on text that is already present on the page or can be made up of text that is provided especially for the purpose.

Index entries can be specified at two levels and direct provision has been made for three independent entries to be defined with each Markup. This means that with a single Markup, you can classify an entry to appear in up to three different places in the overall index - this contrasts with the traditional index generation utilities which rely solely on the text that appears in the document. Consider a catalogue of DIY tools in which you have a section for electric drills. Using LinkUP!'s indexing facility, you can indicate that the drills should appear as Drills, Power; Power Drills; and Power Tools, Drills. If further entries are required, just include a second mark-up at the same location in the document - you could now add Electric Drills; Rotary Tools and Drills, Electric.

Every index entry is considered to have a main classification, and a sub classification. This means that entries that all have the same main classification can then be sorted according to their sub classification. Taking the example above, a main classification might be Drills, with sub classification of Power, Manual and Electric. A sample section of the eventual index, might then have the following layout:

## Drill Bits

Masonry	45
Metal	45,60
Wood	45

## Drills

Electric	33,34,40-42
Manual	39,46
Power	33,34,40-42

**Drop Stays** 50

**Duplex Routers** 75

Unlike other indexing applications, LinkUP! does not require that your entire publication is maintained as a single document to create a global index. Instead, index entries are extracted from all relevant documents and then combined using a separate utility.

## Marking and Altering Index Entries

To mark an index entry in a document,

1. Display the Index Tools Palette (see p65)
2. Highlight any piece of text that is on the page and is close to the point to which the entry will refer
3. Click the add index entry button on the Index Tools Palette 

The Index Tools Palette now automatically expands to give a visual representation of the index entry. By default LinkUP! inserts into the main heading the actual text that you have highlighted. To change this simply overwrite the text in the Index Tool Palette. As mentioned above, you can also ask LinkUP! to add the entry under a sub heading by typing one into the next box down on the Index Tools Palette.

If you want this item to appear in more than one place in the index, you are now able to add an additional 2 Index classifications, each of which has a main heading and a sub heading as described above. To achieve this use the up and down arrows to move between each of the three entries. The title bar of the palette will assist you by displaying which entry, 1,2 or 3 that you are editing.

You can review and optionally alter an index Markup at any time, simply by clicking on the Markup with the text tool selected. The Index Tools Palette will, if displayed, refresh to show the actual index entries for the text you have clicked into.

NB. The exact location of the index Markup is not important. LinkUP! only uses the physical position to determine the page number that will appear in the index. In most cases this will mean marking the actual entry you would like referencing in the index but if this is not the case it is important to ensure that the Markup is sufficiently close to the intended text that it remains on the same page. Almost always this will mean marking a piece of text in the same text box.

## Generating an Index

Once you have marked up the QuarkXPress document(s), there are three stages that need to be completed to create your Index as follows;

- Creation of Index Data File(s) for the document(s)
- Use the LinkIndex utility to Import and process the Index Data File(s)
- Load the index file that has been created into a QuarkXPress document for publication and formatting

### CREATING INDEX DATA FILES

Once you have completed the process of marking up and naming all of the index entries for your document(s), you are ready to create Index Data File(s). This is achieved using LinkUP! as follows

1. From the LinkUP! menu choose **Create Index**
2. Input a name for the Index Data File you are about to create
3. Click **Save**

NB: Whilst you can choose to use the default name that is suggested, it is recommended that you create the file under a name that you will be able to recognise at a later date as belonging to the document you are using. It is also a good idea to create all Index Data Files in a common folder - this makes the generation of indexes across multiple documents much easier.

For the second stage in the index process you need to use the LinkIndex/TOC utility that was installed during the initial installation.

## LINKINDEX UTILITY

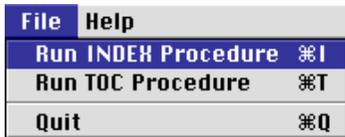
The LinkIndex Utility is used to generate both Indexes and Table of Contents for your QuarkXPress document. For now we will concentrate only on the Indexing part of the utility.

The LinkIndex Utility serves three main purposes in terms of Indexing;

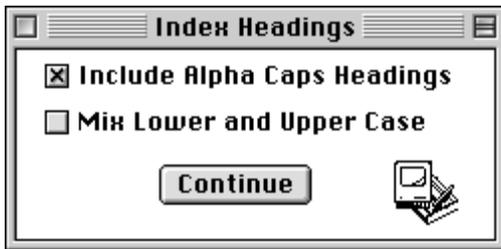
- Process the Index Data Files created using LinkUP!
- If there are several Index Data Files, to collate all of them into a single Index
- Create a master Index File including QuarkXPress styling information

We will now proceed to create our index.

1. From the File menu select Run Index Procedure



You are now presented with a dialogue box entitled Index Headings containing two options as follow:



This box allows the user to specify whether the LinkIndex utility should automatically add Alpha capitals into the index where appropriate. If this is selected then a sample index section would look as follows:

<b>A</b>	
<b>Aardvarks</b>	<b>5,10-11,20</b>
<b>Armadillos</b>	<b>4,18</b>

<b>B</b>	
<b>Bison</b>	<b>17,23</b>
<b>Buffalo</b>	<b>22</b>

If you choose not to Include Alpha Headings, then the index would be created as follows:

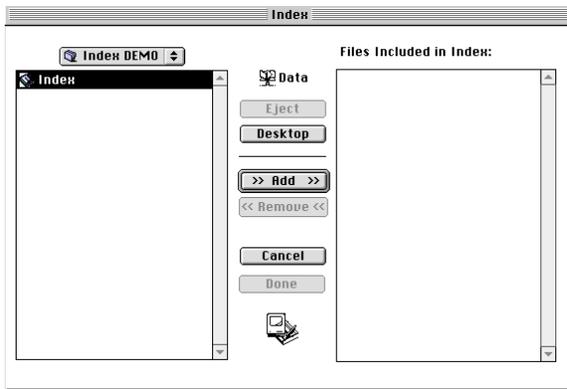
<b>Aardvarks</b>	<b>5,10-11,20</b>
<b>Armadillos</b>	<b>4,18</b>
<b>Bison</b>	<b>17,23</b>
<b>Buffalo</b>	<b>22</b>

The Mix Lower and Upper Case option tells LinkUP! whether to be case sensitive in sorting the index entries.

- Once you have selected the desired option click the continue button

There are now two separate file lists presented to you. The left hand side will show a list of Index Data Files that are available for inclusion in the index you wish to build. The right hand side will display a list of files that are currently selected. This is the Files Included in Index Window.

Initially the list on the left hand side of the dialogue may be empty until you have selected the folder containing the Index Data Files. The right hand list box will obviously be empty until you have added Index Data Files from the left hand side.



You must now use the drop down folder option at the top of the left hand window to locate the folder containing your Index Data Files.

Once you have located the Index Data Files you can add them as follows;

1. Highlight the Index Data File in the left hand windows
2. Click the Add Button or Double-click the actual Index Data File itself

You will now be able to observe that the Index Data File that was highlighted and added, now appears in the Files included in Index window.

3. Now add all of the Index Data Files needed in the same way

You can remove any Index Data Files from the Right hand window by highlighting the file and clicking the remove button

NB. For a single document index, only one file will appear in the inclusion list, whereas an index for a large publication might collate the Index Data Files from many documents. You can choose Index Data Files from multiple drives or folders, where appropriate.

4. Having selected as many Index Data Files as you need to include in the Index you are generating click the Done button.

You are now prompted to specify a name for the text file that will hold the index that is to be created.

5. Input an appropriate name

The LinkIndex Utility has now created an index text file that is in a suitable format to bring into QuarkXPress.

## **USING THE INDEX FILE**

The last stage in creating your Index is to read the newly created text file into QuarkXPress. To do this you must do the following;

1. Create a new document to your own specifications or use an existing one
2. Create a text box that will contain the index (dependent on the size of the index it may be necessary to create a chained text box spanning several pages)
3. Select the Content Tool from the XPress Toolbar
4. Click into the text box you created for the Index
5. From the File Menu select Get Text
6. Select the Index text file generated by the LinkIndex utility, and check the box stating that you wish to "Include Style Sheets".

The index will now be read in, with the following styles automatically generated:

Alpha Caps	Used to mark the Alpha Headings, if you chose that these would be included
MainIndex	Used to mark Main index entries
SubIndex	Used to mark Sub index entries.

If an index entry appears in multiple locations in your publication, then it will have several page numbers alongside it (providing you marked each individual entry). Only identical entries are treated in this way, so it is important to ensure that the entries are added consistently for the index to be successfully created.

You are now free to use Quark's style definition and editing abilities to modify the appearance of each of these styles. For more information on using QuarkXPress Styles see your QuarkXPress Manual.

NB. The Get Text command is only available when the Content tool is selected and a text box is active. When a picture box is active, Get Picture replaces Get Text in the File menu.

# Creating Table of Contents (TOC) from your QuarkXPress documents

## Navigation

For this section of the manual you may want to familiarise yourself with the following:

Topic	Page
Index Tools Palette	49
Marking Up	117

## TOC Fundamentals

LinkUP! allows you to generate a TOC for either a single document, or for a group of documents that might constitute an entire publication. The Index entries can be inserted at any point in the document, and can either draw on text that is already present on the page or the entry can be made up of text that is provided especially for the purpose. You also have to indicate one of nine Levels that the entry should assume.

You are free to choose how these levels should be allocated according to the nature of the publication that you are working with. A novel, for example, would only use a single level, which would correspond to a chapter heading. A technical document might use a scheme of numbered sections and sub sections as below

**Chapter 1 - The Bennets of Longbourne**

**Chapter 2 - Mr Bennet visits Bingley**

**Chapter 3 - Mr Darcy snubs Elizabeth**

**Chapter 4 - Bingley & Darcy Discussed**

**Chapter 5 - The assembly discussed**

**Chapter 6 - The Lucases visit Longbourne**

<b>Section 1</b>	<b>Installation of equipment</b>
1.1	Preparation of site
1.2	Delivery of Equipment
1.3	Installation of equipment
1.4	Commissioning equipment
<b>Section 2</b>	<b>Testing of equipment</b>
2.1	Safety tests
2.2	Functional tests

In the above example, the two section headings would be marked as Level 1, with all the remaining entries marked as Level 2. A maximum of nine hierarchical levels are available for use in this manner.

## Marking and Altering TOC Entries

To mark a TOC entry in a document,

1. Display the Index Palette (see p65)
2. Highlight any piece of text that is on the page and is close to the point to which the entry will refer
3. Click the add TOC entry button on the Index Tools Palette 

The Index Tools Palette now automatically expands to display a box that will contain the text to appear in the TOC and a drop down list displaying the level. By default LinkUP! inserts into the text entry box the actual text that you have highlighted. To change this simply overtype the text in the Index Tool Palette.

4. Enter the text you wish to appear in the TOC by editing the entry in the Index Palette
5. Select the hierarchical level you wish the entry to appear at

You can review and optionally alter a TOC Markup at any time, simply by clicking on the Markup with the text tool selected. The Index Tools Palette will, if displayed, refresh to show the actual TOC entries for the text you have clicked into.

NB. The exact location of the TOC Markup is not important. LinkUP! only uses the physical position to determine the page number that will appear in the TOC. In most cases this will mean marking the actual entry you would like referencing in the TOC but if this is not the

case it is important to ensure that the Markup is sufficiently close to the intended text that it remains on the same page. Almost always this will mean marking a piece of text in the same text box.

NB. It is essential if you are marking a number of documents to be combined into a single publication that the same level numbering scheme is applied consistently to all documents. In other words, you should not mark Section headings with Level 3 in one document and Level 4 in another. This is, of course, just common sense.

## Generating a Table of Content

Once you have marked up the QuarkXPress document(s), there are three stages that need to be completed to create your Table of Contents as follows;

- Creation of TOC Data File(s) for the document(s)
- Use the LinkIndex utility to Import and process the TOC Data File(s)
- Load the TOC file that has been created into a QuarkXPress document for publication and formatting

### CREATING TOC DATA FILES

Once you have completed the process of marking up and naming all of the index entries for your document(s), you are ready to create TOC Data File(s). This is achieved using LinkUP! as follows

1. From the LinkUP! menu choose **Create TOC**
2. Input a name for the TOC Data File you are about to create
3. Click **Save**

NB: Whilst you can choose to use the default name that is suggested, it is recommended that you create the file under a name that you will be able to recognise at a later date as belonging to the document you are using. It is also a good idea to create all TOC Data Files in a common folder - this makes the generation of TOCs across multiple documents much easier.

For the second stage in the TOC process you need to use the LinkIndex utility that was installed during the initial installation.

## LINKINDEX UTILITY

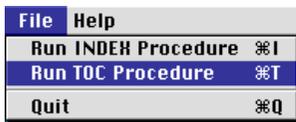
The LinkIndex Utility is used to generate both Indexes and Table of Contents for your QuarkXPress document. For now we will concentrate only on the Table of Contents part of the utility.

The LinkIndex Utility serves three main purposes in terms of TOCs;

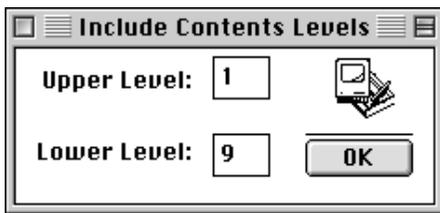
- Process the TOC Data Files created using LinkUP!
- If there are several TOC Data Files, to collate all of them into a single TOC
- Create a master TOC File including QuarkXPress styling information

We will now proceed to create our TOC.

1. From the File menu select Run TOC Procedure



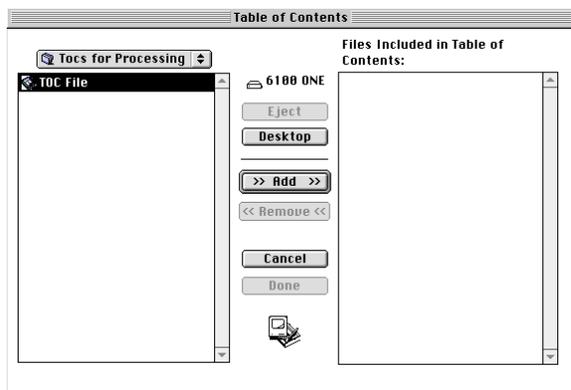
You are now presented with a dialogue box entitled Include Contents Levels. The are the Upper Level and the Lower Level Option Boxes as follows;



This dialogue is asking for the range of levels that you wish to incorporate into the Table of Contents. In most cases, you will be able to accept the default for all levels to be included, i.e. 1-9. If you plan to use the TOC feature of LinkUP! to generate other lists this will have to be reflected here. (see *Other uses of TOC* on page 181 for details).

2. Once you have selected the desired option click the OK button

There are now two separate file lists presented to you. The left hand side will show a list of TOC Data Files that are available for inclusion in the TOC you wish to build. The right hand side will display a list of files that are currently selected. This is the Files Included in Table of Contents Window.



Initially the list on the left hand side of the dialogue may be empty until you have selected the folder containing the TOC Data Files. The right hand list box will obviously be empty until you have added TOC Data Files from the left hand side.

You must now use the drop down folder option at the top of the left hand window to locate the folder containing your TOC Data Files.

Once you have located the TOC Data Files you can add them as follows;

3. Highlight the TOC Data File in the left hand windows
4. Click the Add Button or Double-click the actual TOC Data File itself

You will now be able to observe that the TOC Data File that was highlighted and added, now appears in the Files Included in Table of Contents Window.

5. Now add all of the TOC Data Files needed in the same way

You can remove any TOC Data Files from the Right hand window by highlighting the file and clicking the remove button

NB. For a single document TOC, only one file will appear in the inclusion list, whereas an index for a large publication might collate the TOC Data Files from many documents. You can choose TOC Data Files from multiple drives or folders, where appropriate.

6. Having selected as many TOC Data Files as you need to include in the TOC you are generating, click the Done button.

You are now prompted to specify a name for the text file that will hold the TOC that is to be created.

7. Input an appropriate name

The LinkIndex Utility has now created a TOC text file that is in a suitable format to bring into QuarkXPress.

## **USING THE TOC FILE**

The last stage in creating your TOC is to read the newly created text file into QuarkXPress. To do this you must do the following;

1. Create a new document to your own specifications or use an existing one
2. Create a text box that will contain the TOC (dependent on the size of the TOC it may be necessary to create a chained text box spanning several pages)
3. Select the Content Tool from the XPress Toolbar
4. Click into the text box you created for the TOC
5. From the File Menu select Get Text
6. Select the TOC text file generated by the LinkIndex utility, and check the box stating that you wish to "Include Style Sheets".

The TOC will now be read in, with the following styles automatically generated:

You are now free to use Quark's style definition and editing abilities to modify the appearance of each of these styles. For more information on using QuarkXPress Styles see your QuarkXPress Manual.

NB. The Get Text command is only available when the Content tool is selected and a text box is active. When a picture box is active, Get Picture replaces Get Text in the File menu.

## Other uses of TOC

Another way to use the TOC feature is to create a separate list that you want to appear in the document. A list of pictures in an illustrated publication would be an example of this or maps in a bible, scientific tables etc. This can be achieved by dedicating one particular level to mark this class of entries but not be embedded in the Table of Contents.

The Markup is achieved in an identical manner, with you choosing one of the later level numbers that have been set aside by you for this purpose. It is best to allocate the level numbers for separate items starting from Level 9 and working backwards towards Level 1, whilst assigning hierarchical Table of Contents entries in the opposite direction.

To generate these separate lists the process is exactly the same as generating a TOC above, the only difference comes when selecting the range of levels to be included in the TOC file. If for example you chose level 9 to be used to create a pictures list for the publication, when specifying the range for the TOC select the Upper level to be 9 and the lower level to be 9 also. The resulting TOC file will then only contain a list of marked pictures in the publication and not any of the real TOC entries.

You will gather from this that to generate different lists as well as your TOC you will have to run the LinkIndex process several times - the first time for the Table of Contents proper, when you might choose to include Levels 1-4, and then a further time for each separate item that you have additionally marked such as pictures or tables.

If you have chosen to mark Figures using Level 8 and Tables using Level 9, then you would run the utility two further times, specifying the range as Level 8 to Level 8 and then again for the tables specifying Level 9 to Level 9 respectively.

In every case, the resulting text file will then be available for importing into Quark using the process described above. In this instance, a different style is applied for each level that is exported.

# Header/Footer

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Markup Tools Palette	46
Marking Up	117

If you imagine for a minute a typical telephone directory, every page has a header that specifies the first and last directory entry on that particular page. Updating a header such as this manually is not only a long task but also prone to errors should the pagination of the document change at any point. Using LinkUP! these conditional headers can be created and updated automatically. This isn't only limited to headers or footers, conditional updating of text dependent on another item on the page can happen on any part of the page using LinkUP!'s automatic header/footer Creation facility.

Any piece of information containing LinkUP! markups can be used to form the conditional headers & footers. So in the case of the telephone directory example, the persons last name would be used as the header/footer text.

Headers & Footers are marked up on a text box basis as opposed to a character basis used for conventional markups.

## Creating a Header/Footer

Once you have marked up the items to be conditionally updated using the head/footer feature, to update them select the Create Header/Footer option on the LinkUP! menu.

You can also produce references based on information not displayed on the page, but contained in the Table of Content entries. To illustrate this feature, select and mark "Part No" as Level Nine Table of Content entry with "A starting hidden entry" string in the entry text field

If you now mark this box as "Header & Footer Markup" based on TOC Level 9 when processed, it would look something like this:

These features are very useful for publishing directory type publications. You can experiment with various settings to create the desired results. Detail specification of the parameters can be found in the Chapter 5.

Information marked with LinkUP! codes can be extracted, indexed, truncated (if necessary) and inserted into specifically designated text boxes. These text boxes are normally marked on the master page using 'Add Special Markup' from the LinkUP! menu. This way the marked box will automatically be brought onto each document page.

Contents can be based on any type of LinkUP! markups - Key fields, Fields, Index and TOC entries.

Parameters for Header/Footer boxes are specified under the 'Add Special Markup' button on the Markup Tools Palette. 

# Relational Reference

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Concepts	34

## What is a Relational Database?

Although in many instances data can be described using a single table, it is equally common for a database to incorporate methods of grouping similar records together. By incorporating this grouping information into the database, it is then possible to manipulate all of the grouped items together.

In database terminology, this grouping is referred to as a relational structure, where one table exists to hold the information that is common to the group as a whole and a second file is used to hold information that is specific to each item that belongs to the group. This is best considered with an example.

Consider a database that is used to hold information about cars. A simple level of grouping that can be applied to the cars can be based on the manufacturer that produces them.

In this case, we could setup two tables within our database, one that contains details of the manufacturers and another that contains details of the car models themselves.

### Manufacturer Table

Manuf_Ref	Name	Telephone No	Fax No	Website
FRD	Ford	0032 122 3333	00443434	www.ford.com
JP	Jeep	00123133	123123	www.jeep.com
ROV	Rover	0044122272	23232323	www.rover.co.uk

## Model Table

Model_Ref	Manufact	Model Name	Style
0001	FRD	Escort	Hatchback
0002	FRD	Toledo	Coupe
0003	JP	Off Roader	4x4
0004	ROV	214	Hatchback
0005	FRD	Fiesta	Hatchback
0006	ROV	832	Saloon
0007	FRD	Sierra	Estate

Note that in this example each car model has its own unique reference as well as a field that indicates the relevant manufacturer. It is now a simple matter to group the cars together according to the manufacturer code which is common to both tables.

Aside from the ability to rapidly select items on a group basis, there are other benefits that come from this kind of data structure. Firstly, there is a space saving, since information that would otherwise have to be repeated within the cars table is now centralised within the manufacturer table. This gives rise to the second benefit which is that should a manufacturer need to have their details changed, then the change needs to be administered only once, and all of the relevant car records will automatically be able to reflect that change. Finally, a database that uses this kind of data structure is faster to manipulate and easier to program.

Where a database contains information in more than one table as in the above example, regardless of the application you are using to view the information, there has to be a link that ties the information in these tables together. In database terminology this link is referred to as a foreign key.

In our example, there is a relationship between the **Manufact** field in the models table, and the **Manuf\_Ref** field from the manufacturer table. Every record in the model table must have an entry in the **manufact** field that corresponds to a record in the manufacturer table.

This is an example of a two-level relational hierarchy, with manufacturers occupying the first level and the models occupying the second level.

It is possible to extend this further, and introduce a third level, which shows the variation of each model of car that is available. Consider a third table called editions with the following data

### Editions Table

Edition	Model	Engine Size
ED1	0001	1.4L
ED2	0001	1.6L
ED3	0001	2.4L
ED4	0002	3.4L
ED5	0002	4.0L
ED6	0003	2.0L
ED7	0003	2.8L
ED8	0003	3.2L

This new table now has a link between the field **Model** and the **Model\_ Ref** field in the model table.

By following the link backwards, it is possible to work out both the model and the manufacturer that each of these editions refers to. This is illustrated below where we have taken data from each of the previous three tables.

Edition	Engine Size	Model Name	Manufacturer
ED1	1.4L	Escort	Ford
ED2	1.6L	Escort	Ford
ED3	2.4L	Escort	Ford
ED4	3.4L	Toledo	Ford
ED5	4.0L	Toledo	Ford
ED6	2/0L	Off Roader	Jeep
ED7	2.8L	Off Roader	Jeep
ED8	3.2L	Off Roader	Jeep

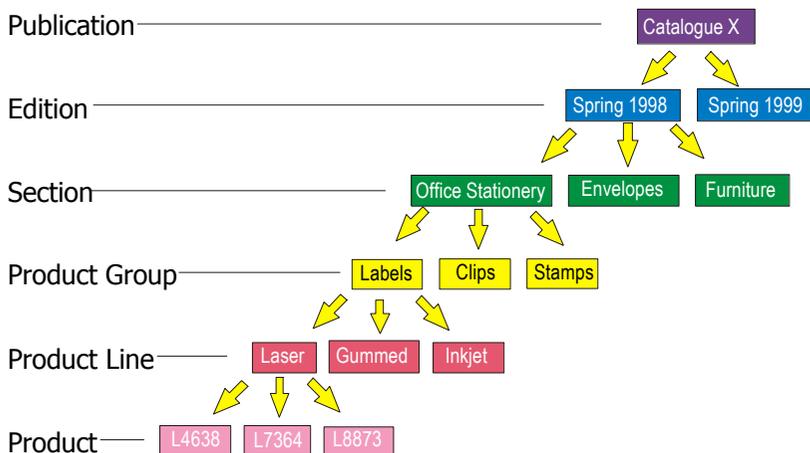
In this case, the data is now using a **three-level hierarchy** – Manufacturer, Model and Edition. LinkUP can support hierarchies of this nature that are up to a maximum of eight levels in depth.

## Why is being relational important to LinkUP!

By using a relational database model you are afforded far more flexibility in the way that you manage your data and in the way in which you can publish it. This is particularly important for any kind of automated publishing.

Suppose you had a catalogue where items are described in terms of product groups and each product group may then have one or more product lines. By using a relational model you simply have to specify the formatting for the product group and a single product line and LinkUP! is able to automatically publish all of the additional product lines within the same group.

This can be extended to the management of product groups within a section or sections within a catalogue as in the example below. This shows a relational structure for a stationery catalogue starting with the publication, then edition and ending up with the specific products.



A user could simply setup a template for a product group, product line and product and leave LinkUP! to search the database and bring the full set of records onto the page.

## Configuring Relations with LinkUP!

For the relational structure to work correctly, a number of conditions must be met.

1. Each level in the hierarchy must have a Key Field that is unique to each record at that level
2. Each record for any child level in the hierarchy must have a field that contains the Key Field of its parent
3. The Link from the child to the parent has to be to the parents Key Field
4. In both cases the Key Field data must be contained in a single field – LinkUP! does not support compound Key Fields.
5. The link is always made from the child to the parent

NB. It is possible to have keys that are not unique between levels. In our example, we could have used a straight numeric key for each set of data, in which case the number '1' would have been used at each level. It would only be used once for each level, however, which means that condition 1 would have still been met. Note also that it is possible to have multiple child tables for a single parent. You may, for example, have a separate database of dealerships which also relates directly to the manufacturer code.

Examples

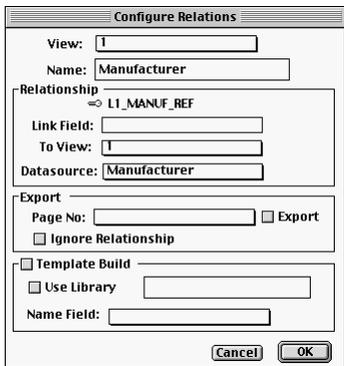
This means in LinkUP! to create a relationship between two tables the only additional piece of information that has to be supplied is to identify the link (parent) field in the child table .

Since we know that LinkUP! always links to the Key Field in the parent table, we always know that the Key Field must already be defined.

The Configure Relations Dialogue is located on the LinkUP! Menu

1. Select **Configure Relations** from the LinkUP! Menu

You should be presented with a dialogue similar to the following;



The main purpose of this dialogue is to tell LinkUP! which Views or Tables are linked to each other and which particular fields within each View is to be the link field i.e. the foreign key (as explained above). Remember rule 5. above, that that the link is always made from the child to the parent. So the first part of the linking process does not start with the first view but with the ones below.

You can switch between the configuration for the different views by clicking on the drop down list at the top of the dialogue.

To make this easier to grasp, we will use the example we looked at previously with the Car Manufacturers, Models and Editions.

In our example the Car Table is linked to the Manufacturer Table. The Key Field in the Manufacturer Table is **Manufacturer ref**. There is also therefore a field in the Car Table that contains this information that is called Manufacturer. We therefore need to tell LinkUP! that the View is linked to the View Manufacturer and the field in the Car View that contains the foreign key (the Link field) is called Manufacturer. The following would be the setup that would give LinkUP! this information.

View 1, which in this case is called Manufacturer, is not linked to any views at all, signified by the Link Field Box being empty. This is because in LinkUP! you always link from the child to the parent and not vice versa.

Configure Relations	
View:	<input type="text" value="1"/>
Name:	<input type="text" value="Manufacturer"/>
Relationship	<input type="text" value="L1_MANUF_REF"/>
Link Field:	<input type="text"/>
To View:	<input type="text" value="1"/>
Datasource:	<input type="text" value="Manufacturer"/>

View 2, which in this case is called Model, is linked to View 1 and the field that contains the foreign key (linking information) is the field L2\_Manufact.

Configure Relations	
View:	<input type="text" value="2"/>
Name:	<input type="text" value="Model"/>
Relationship	<input type="text" value="L2_MODEL_REF"/>
Link Field:	<input type="text" value="L2_MANUFACT"/>
To View:	<input type="text" value="1"/>
Datasource:	<input type="text" value="Model"/>

View 2, which in this case is called Edition, is linked to View 2 and the field that contains the foreign key is the field L3\_Model

Configure Relations	
View:	<input type="text" value="3"/>
Name:	<input type="text" value="Edition"/>
Relationship	<input type="text" value="L3_EDITION"/>
Link Field:	<input type="text" value="L3_MODEL"/>
To View:	<input type="text" value="2"/>
Datasource:	<input type="text" value="Edition"/>

With all of the views linked, the relationships are now configured.

# Proofing

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Data Viewer Palette	50
Updating	162

Whilst it is possible to confirm which Key field a Markup relates to by clicking onto it and viewing the dialogue that is displayed, this can be a time consuming process on a page that contains many markups. A quicker way of proofing the markups visually is to use either the Proof by Key or Proof by Field options, which changes the textual content of any marked field to indicate the key to which the field is related or its field name respectively.

NB: Because of the nature of the textual change that is made during a proof by key update, it is recommended that you do NOT save your document after the proof has taken place, as your original field contents will be overwritten after the proof. There is no way to undo this change once it has taken place. Therefore save a copy of the document before the proofing action is performed.

After choosing Proof by Key or Field, LinkUP! will step through the document making changes as it goes through. You can stop the Proof by choosing the Cancel button on the progress bar, but note that any changes made up to the point you cancel will remain.

## Proof by Field

The function of this command is to replace the contents of all marked fields with the name of the field that they are marked with. So if you have marked the price 195.00 as a **price1** field, after a proof by field the 195.00 would be replaced with the word **price1**. The description would be replaced with the word description etc.

The example below shows a sample XPress page before and after a Proof by Field.



**BEFORE  
PROOF BY  
FIELD**



**AFTER  
PROOF BY  
FIELD**

As all of the links back to the data source are still in present, to change the document back so that it reflects the actual data in the data source again, you must simply update the XPress page from the database(see p162).

## Proof by Key

The function of this command is to replace the contents of all marked fields with the value of the Key Field to which they are associated. So if you have marked the price 195.00 as a price field and associated it with product 12345AB, after a proof by field the 195.00 would be replaced with the word **12345AB**. The description would be replaced with **12345AB** etc.

The following shows an example of a page before and after a Proof by Key.



**BEFORE  
PROOF BY  
KEY**

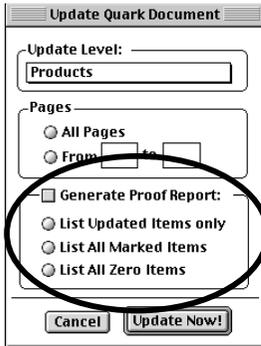


**AFTER  
PROOF BY  
KEY**

As all of the links back to the data source are still in present, to change the document back so that it reflects the actual data in the data source again, you must simply update the XPress page from the database(see p162).

## Proof Reports

The second proofing facility in LinkUP! is part of the update process (see p162). You can ask LinkUP! to generate various Proof Reports. There are 3 types in all. When you update an XPress document from your data source the following dialogue appears. You may be used to simply clicking on the **Update Now** button, but the lower section of the dialogue lets you select which proofing reports to generate.



If you have the Generate Proof Report option checked, you will be prompted for a file name for the proof report. The file LinkUP! creates is an ASCII text file so to read it you can open it up within a text box in XPress or in a text editor or word processor. The following are the three types of proof reports.

### LIST UPDATED ITEMS ONLY

This is the most useful of all the proof reports. LinkUP! will generate a report that contains only items whose contents have been changed during this update. Effectively this constitutes a list of all published values that have changed in the data source.

### LIST ALL MARKED ITEMS

This proof report causes LinkUP! to generate a list of EVERY item that it is controlling from within the document i.e. every marked item in the document. This can be used to verify that all of the anticipated items are, indeed, being updated from the data source.

### LISTING ALL ZERO ITEMS

This report is relevant for numeric fields only. When chosen, it creates a list of all values that are inserted with a numeric value of 0. It is intended to help identify items that are not priced

within the data source, to prevent the potentially disastrous consequence of publishing a price list with items listed as zero value.

Sample proof report:

```
Update to:           Learning to use LinkUP!
Updated from:       Learning LinkUP Database.LNK.DB
Date:              Monday, May 27, 1996    at:
5:33 am
Including ALL Marked Items

Page:              1    Record: 01-02212
Field:             Description
Old Value:         xx
New Value:         Small Hammer
```

# ADVANCED TOPICS

## Next Oversed Box

### Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Data Viewer Palette	50

It is a common occurrence when working in QuarkXPress that text boxes can end up not being large enough to display all of the text contained within them. This can be especially so when using LinkUP! to update XPress text boxes. If there is more data in the database than the text box can display, the textbox will become overset. LinkUP! will not act on this automatically as this is something that can only be controlled by an operator, who will have to decide whether to edit the story, reduce the leading or point size or perhaps increase the size of the box to ensure that all the textual material is correctly published. What LinkUP! can do, however, is to indicate the places that overset matter exists so that you can make the necessary changes.

Clicking this button will take you to the top left hand corner of the first encountered text box that has overset matter. Once this has been corrected, you can then click the button again and repeat the process. When there are no further instances of overset text, LinkUP! will display a dialogue box to inform you.

It is obviously sensible to use this feature before outputting your QuarkXPress document to ensure that no boxes in it are overset.

# Exporting

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Data Viewer Palette	50
Field Names & Data Setup	98
Marking Up	117

## Why you do it

There are two main scenarios for which exporting feature of LinkUP! might be appropriate. However, although the scenarios may differ, the process will be broadly similar.

### **CREATING A DATABASE FROM THE CONTENTS OF AN XPRESS DOCUMENT**

Have no database and would like to create one from QuarkXPress? Then read on.

This is an extremely powerful use of LinkUP!'s export feature. All too often a company's primary source of information is not stored in any database. The only reliable place where a complete set of data exists is in the QuarkXPress pages themselves. Companies are increasingly realising the importance of having the data in an accessible and generically structured format. Great as QuarkXPress is for laying out pages, it makes a lousy database. So you can use LinkUP! to export the information from your document into a database file that you can use in the future.

### **UPDATING AN EXISTING DATABASE**

If you have an existing database but the data in that database is not current with your XPress document, you can use LinkUP! to export the data that has changed on the XPress page back to the database. This is the reverse of the updating process (see page 162) and can be carried out as often as necessary.

## How you do it

No matter what your reasons for using the Export Features of LinkUP! there are common processes that you will have to go through.

Process	Required	Page
Field names & data setup	Yes	98
Marking up your document	Yes	117
Configuring Relationships	Only for relational database structures	184
Configuring Export Parameters	Yes	199

The list of processes above is in chronological order. You can see that there are several processes involved here and so what you need to do to achieve a successful output will depend on what stage you are at as you read this section. If you have not used LinkUP! on your documents before then you will need to go through the first 2 processes (the 3<sup>rd</sup> if you are working with relational database structures). You can then return to this section to continue. If you have already completed the first 2 or three processes then you should proceed.

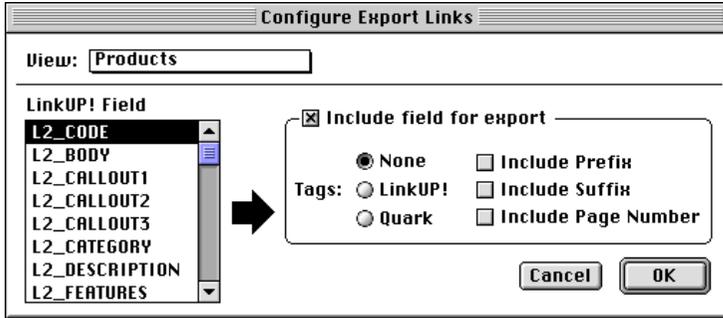
As has been stated, the first 3 of these processes are covered elsewhere in this manual (see above) so we will concentrate only on the last of these, the process that is specific to exporting, configuring Export Parameters.

### PLAN YOUR DATABASE

If you don't have a database to place all of the QuarkXPress data into yet, the first thing you should do is plan a structure for the new database. It will make life much easier if you know the structure before you start preparing the QuarkXPress document. Essentially this is a set of fields into which data exported from the XPress document can be stored. More information on appropriate structures for your database can be found in the planning and implementation section of this manual. Obviously if you already have an existing database then LinkUP! can work with it. You don't have to create a new structure just for LinkUP!

## Configuring the Export Parameters

The configuration parameters for export allow you to have more control over what exactly will take place during the export. The Export Parameters Dialogue Box can be accessed on the LinkUP! Menu under configure export links. It will look similar to the dialogue below.



You can set LinkUP! to only export certain fields back to your database. If you had a catalogue, for instance, and you knew that the only thing that would change on the QuarkXPress page would be the product description, you can configure LinkUP! to only export the product description field, leaving all of the other fields in the database untouched.

Another powerful export feature is the ability to embed LinkUP! or Quark Styling Tags in your data stream. If you then update your QuarkXPress document with Tag Processing enabled, LinkUP! will activate the styling tags that were written back to recreate the same styling. For more information on Tagging see page 202. You could also use this styling information in the database to recreate some basic styling in the database itself.

Field Prefixes and Suffixes can also be included in the export by enabling the relevant options.

An important feature of exporting for many users is being able to export back to the database, the page number on which a certain element appeared. This is achieved by selecting a field in the database to export these page numbers to. If no field is selected the information will not be exported.



The export page number options are actually configured in the configure relations dialogue (**LinkUP! Menu Configure Relations**).

As you can see in the dialogue above, you simply select the database field to export the page number to and whether or not this information is actually exported. If you are not using relational structures within LinkUP! you should make sure that this box is checked.

## The Export

With all your document marked up, the database links and relationships configured and now also the export parameters configured you are ready to export the data.

There are two fundamentally different approaches to exporting data using LinkUP! for exporting and which one of these you choose will depend on if you will be exporting direct to your database using ODBC or to an interim file.

### ODBC EXPORT

If you have a direct ODBC connection to your database then to export the data from the XPress document to your data source;

1. Click the Export Button on the Data Viewer Palette 

The fields configured for export will be written back directly to the specified fields in your ODBC compliant database (provided you have the correct database Access Rights). This way, the changes in your QuarkXPress document will be reflected in your database. If your database was originally empty, then LinkUP! will populate your database. A progress bar is displayed whilst the export is taking place, and you can cancel the export at any point.

### ASCII EXPORT

If you do not have a direct ODBC connection to your database, then, when you perform an export, LinkUP! will actually export your data into a structured ASCII file for you rather than direct to your database.

To export the data from the XPress document to an ASCII file;

1. Click the Export Button on the Data Viewer Palette

LinkUP! will prompt you for a name for the ASCII file it's going to create.

2. Type in a name for your ASCII file
3. Click the **OK** Button

LinkUP! now performs the export. A progress bar is displayed whilst the export is taking place, and you can cancel the export at any point.

The exported ASCII file is structured in a format that lets it preserve the data that is in your QuarkXPress document. Items such as carriage returns will therefore not be lost in the process. If you have selected to export tagging information, this will also be embedded.

The file that is created is in the following format:-

```
F|01-02212|1|Description|Small Hammer
R|01-02212|1
F|01-02212|1|1+|0.900
F|01-02212|1|25+|0.820
F|01-02212|1|100+|0.700
F|01-03013|1|Description|Large Hammer
R|01-03013|1
F|01-03013|1|1+|0.900
F|01-03013|1|25+|0.820
F|01-03013|1|100+|0.700
F|01-03012|1|Description|Small Nails
R|01-03012|1
F|01-03012|1|1+|0.900
F|01-03012|1|25+|0.820
F|01-03012|1|100+|0.700
F|01-03015|1|Description|Large Nails
R|01-03015|1
F|01-03015|1|1+|0.900
F|01-03015|1|25+|0.820
F|01-03015|1|100+|0.700
F|01-03014|1|Description|Pliers
R|01-03014|1
```

F|01-03014|1|1+|0.900  
F|01-03014|1|25+|0.820  
F|01-03014|1|100+|0.700

There are two types of export format, one for Key fields and one for Field data.

The format of a Key field export is as follows:

R|key field|pageno

Here the R is a constant, and the key field and pageno values will be replaced appropriately.

The format of a field export is as follows:

F|key field|pageno|fieldname|data

Here the F is a constant, and this is followed by the actual key field that the field belongs to, the page number, the name of the field and finally its contents.

## **WHAT NEXT?**

If you are not a database expert you may be wondering what exactly you do with the structured ASCII file you have exported. You actually have two options. If you have the database skills either personally or within your organisation you may choose to write a customised import or conversion routine to allow you to place this data back into your chosen database.

Alternatively MC Research have a product called EasyDB that is freely available to Registered LinkUP! users and should have been included on your LinkUP! Install CD-ROM. It is also available for download from the MC Research Web Site.

EasyDB is a Windows Based Application for Windows 95 that has been designed to take any exported ASCII file from LinkUP! and convert it to a number of mainstream data formats including Comma Separated, Excel, Access, dBase and FoxPro. If your chosen database is not available in EasyDB, simply choose a database format that is compatible and use this instead.

# Tagging

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Data Viewer Palette	50
Updating	162

If you are familiar with the Internet to any extent, you will also be familiar with the words HTML, HyperText Markup Language. What HTML is no more than a text file that contains codes that your Internet Browser can act on. These codes are called tags and it is these that your Internet Browser act on to decide which pieces of text should be in bold, what character size it should be, which font to use etc. So if a piece of text has a bold tag around it, it will appear in bold as below.

`<B>Hello</B>`

The bits inside the greater and less than signs are not actually displayed, just the actual word 'Hello' itself. The `<B>` tells the Internet Browser to start using a Bold Font. The `</B>` tells the Internet Browser to stop using a Bold Font.

In the same way that Internet pages have these Tags embedded in them, it is possible to embed Tags in your database to act in a similar way. This is called Tagging. When the data comes into QuarkXPress, LinkUP! can read these tags and then act on them. These codes can make changes to the style of text, to allow certain aspects of the appearance to be controlled from the database automatically. If it sees a tag asking for a Helvetica Font, LinkUP! knows to turn that piece of text into a Helvetica Typeface. If it sees a tag specifying double space paragraph lines, then it will change the text to double space. I'm sure you get the picture.

There are two distinct types of tag that LinkUP! can act on, LinkUP! Tags and XPress Tags (both described below). These tags can be processed by LinkUP! in one of two ways. You can configure LinkUP! to look for and process tags during the update process. Alternatively you can get LinkUP! to search the entire active for any tags that are present.

# What are LinkUP! Tags

There are two types of LinkUP! Tags, styling Tags and Markup Tags. The styling tags are much simpler than the XPress Tags and designed for the user who only needs a low level of formatting capabilities from the tags. This is as opposed to the XPress Tags that can define every aspect of your pagination.

By way of example, suppose the following was stored in a database field:

This is <<BDon>>bold<<BDoff>>, and this is a new font - <<FC Tahoma >> Tahoma

When updated on the page, this would produce the following:

This is **bold**, and this is a new font - Tahoma

The other type of LinkUP! Tags of used for Marking up data before it is brought into QuarkXPress. The tags are placed around the data in the database or spreadsheet and hence the manual process of marking up elements on the QuarkXPress page is avoided.

# Using LinkUP! Style Tags

LinkUP! Style Tags are inserted into a standard piece of text, whenever you wish to change an element of the display style that they control. Every Tag starts with two mathematical less-than symbols (<<) and concludes with two greater than symbols (>>)

The Tags that are supported are as follows:

<<BDon>>	Switch bold on	<<BDoff>>	Switch bold off
<<ITon>>	Switch italics on	<<IToff>>	Switch italic off
<<UNon>>	Switch underline on	<<UNoff>>	Switch underline off
<<OUon>>	Switch outline on	<<OUoff>>	Switch outline off
<<SHon>>	Switch shadowing on	<<SHoff>>	Switch shadowing off
<<SPon>>	Switch superscript on	<<SPoff>>	Switch superscript off
<<SBon>>	Switch subscript on	<<SBoff>>	Switch subscript off
<<XXon>>	Revert to plain text	<<XXoff>>	Revert to plain text
<<SUon>>	Strike through on	<<SUoff>>	Strike through off
<<CAon>>	Switch caps on	<<CAoff>>	Switch caps off
<<SMon>>	Switch small caps on	<<SMoff>>	Switch small caps off

<<WUon>>	Word underline on	<<WUoff>>	Word underline off
----------	-------------------	-----------	--------------------

<<FCfontname>> will cause a change to the specified font name

e.g. <<FCGeneva>> will set the font to Geneva at that point

Codes can be combined into a single call, so that

<<BDonUNonIToff>>

will cause bold and underline to be switched on, and italics switched off. Alternatively, you can simple use a chain of codes such that

<<BDon>><<UNon>><<IToff>>

will achieve exactly the same effect.

Note that Font calls must always appear on their own.

## Using LinkUP! Markup Tags

Apart from the basic styling tags above, there is another type of LinkUP! Tag, the Markup Tag. The process of marking up a document can be a lengthy one and a way to get around this is to place LinkUP! Markup Tags in the data before it is brought into QuarkXPress. In some situations where the entire publication is built using tags this is also the most sensible way of getting the LinkUP! Markups onto the page. The codes that LinkUP! recognises for Markups are as follows:-

<b>[%LN]</b>	New Key field at this point
<b>[%LRkey field]</b>	Key field Markup where key field is visible on the page
<b>[%LFnfield]</b>	Field Markup for field nn. i.e. FIELD01, FIELD02 etc
<b>[%LPnpicturename]</b>	Markup for embedded picture box for picture group n
<b>[%LTn Table of Contents Data]</b>	Markup for Table of Contents entry
<b>[%LIndex Entry data]</b>	Markup for Index list entry – single level
<b>[%LMEntry I1 main  Entry for I1 sub   Entry I2 main   Entry I2 sub   Entry for I3 main</b>	Markup for Index list entry – multi-level

### | Entry for I3 sub]

Note that currently we only support fields that are defined as FIELD01, FIELD02, FIELD03 etc.

The following sequence:

```
[%LN][%LRABC] [%LF01My Field] [%LF02Second Field]
[%LN][%LRDEF] [%LF01Another field] [%LF02and more]
would result in the following
```

```
ABC My Field          Second Field
DEF Another field     and more
```

where ABC was marked as key field 1, 'My Field' was FIELD01 belonging to this key and 'Second Field' was FIELD02 belonging to this key. DEF is then marked as key field 2 and 'Another field' and 'and more' are FIELD01 and FIELD02 relating to it.

NB. For information on Marking up your document see page 98

## Picture Tags

Picture Tags allow embedded picture boxes to be automatically inserted at the relevant places in the text stream. Up to nine different styles and sizes of text boxes are supported, each of which could be linked to a separate database field using the standard configure links dialogue.

The following sequence is used to achieve this;

```
[%LP01]
```

In conjunction with the configure parameters found under :”Configure Picture Tags” you can configure picture size and alignment.

Once the boxes have been inserted, they can be altered using Quarks standard facilities for manipulating anchored picture boxes.

## XPress Tags

XPress Tags are in essence a complete tagging language that allow you to define all the QuarkXPress character attributes and paragraph formats, but using tagging instead of page layout tools. The use of XPress Tags is a powerful facility and for full explanation on using these codes to format text refer to your QuarkXPress Reference Manual that came with your copy of QuarkXPress.

If we use the same example that was used above for the LinkUP! Tags. Suppose the following was stored in a database field:

This is <B>bold<\$/>, and this is a new font - <\$/" Tahoma"> Tahoma

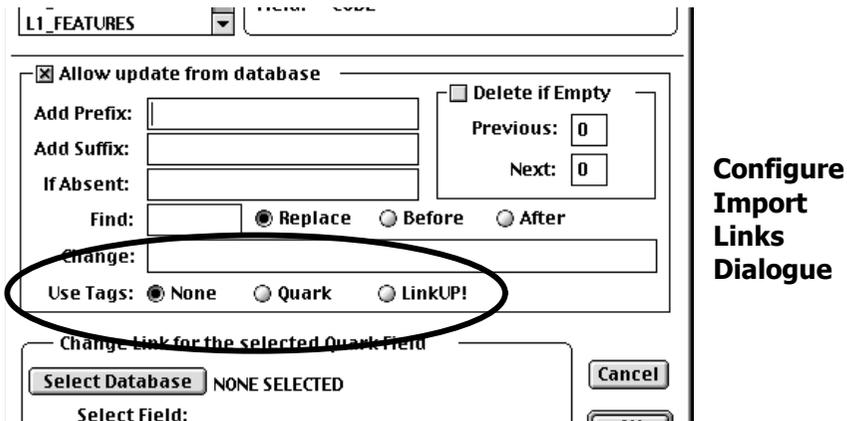
When updated on the page, this would produce the following:

This is **bold**, and this is a new font - Tahoma

A list of all of the currently defined XPress tags has been supplied in the Appendix to this manual on page 214.

## Processing Tags on Update

In the Configure Import dialogue, LinkUP! can be setup to process the LinkUP! Tags, XPress Tags or neither. When the document is updated, LinkUP! will process any tags present in the data. The default setting is that such codes are not used, and so Tags Processing is set to none.



**Configure  
Import  
Links  
Dialogue**

You may also want to write back the tagging information from the QuarkXPress page back to the database. Again, LinkUP! can embed LinkUP! Tags, XPress Tags or neither.

## Processing Tags in Entire Document

LinkUP! has a routine for activating all of the tags in an active QuarkXPress document. LinkUP! will search through all the text boxes in the document searching for the tags. When any are encountered they are translated into the desired typography. These tags can either be LinkUP! or XPress Tags. This function is called Convert Tags and located on the LinkUP! Menu under **Processes Convert Tags**. If this option is selected, LinkUP! will scan through the entire document in all XPress text boxes.

# Character Maps

## Navigation

For this section of the manual you may want to familiarise yourself with the following;

Topic	Page
Updating	162

ASCII is a universal definition of computer character sets. It works by assigning a number from 0 to 255 for every character that is used on the computer. For instance the space character is defined as number 32, A is defined as 65, a tab is 9 etc. The problem arises however that although the ASCII character set is supposed to be standardised, in reality there is a variation between different computer platforms. The numbers and letters are all the same but when you get into accented characters, fractions etc. the number from 0 to 255 that defines them varies from one system to another.

All this means is that if you are working with data coming from a platform other than Macintosh, then the translation of characters across the platforms will be an important consideration. Not if you only use numerals and letters, but definitely if you use other characters. It may only sound like a small thing, but this has the potential to turn into a DTP operators nightmare. What was a `ü` on your Windows 95 computer would turn out to be a `ú` on the Macintosh.

For this reason LinkUP! includes a character translation feature that in essence allows you to map characters from the platform your database is on to the equivalent Macintosh Characters and vice versa when you export data from the Macintosh back to your database.

The feature is very simple in how it works. You simply need to create a text file that contains two columns. One column contains the Macintosh character and the other contains the actual ASCII value of the equivalent character on the other platform.

To use a character map you need to do the following;

- Choose Load Character Map on the LinkUP! Menu
- Select the desired character map you have created
- Click the Open Button

From now on whenever LinkUP! either updates your QuarkXPress document from your database or the database from the QuarkXPress document, this character map will be used to ensure the correct translation of the characters between platforms.

The character map you have chosen will be saved with the QuarkXPress document. It will also be saved with any LinkUP! Projects you create.

If at any point you require a different character map, simply repeat the above sequence this time selecting the new character map instead.

To clear a character map all together, click Clear Character Map on the LinkUP! Menu.

The format of the text file containing any character maps you wish to create should be a two column text file with the columns separated with a tab. The first column holds the Macintosh Character. The second column holds the ASCII code (its unique number between 0 255) for the equivalent character on the platform the data is coming from. The example below illustrates this. You can also look at the sample Character Map that you will find on the LinkUP! Install Disk

0	\048	9	\057
1	\049	:	\058
2	\050	;	\059
3	\051	<	\060
4	\052	=	\061
5	\053	>	\062
6	\054	?	\063
7	\055	@	\064
8	\056	A	\065

A Sample text file containing a character map.

The sample character map included on your LinkUP! Install Disk one for translating characters between the PC and Macintosh Platforms. Character Maps for transferring data between the Macintosh and other platforms will become available from time to time on the MC Research Web Site. Should you create one and would like to share it with others, please feel free to send the text document to our support department at the MC Research UK address.

# Apple Scripting

## What is Apple Scripting?

Apple Scripting is a scripting language created by Apple Computer that gives you the power to automate your workflow. Labour intensive tasks can be performed automatically by the computer, usually in a fraction of the time. The Mac OS is of course Apple Scriptable but the real power of this language becomes apparent when you are able to control applications also.

AppleScript differs from conventional programming languages in that it is actually more like a conventional human language. It contains verbs, nouns and adjectives and is therefore very much like the way we think and speak. This brings it within the reach of many more Macintosh users because of its accessibility. Best of all, Apple Script comes free with your Mac OS so the only cost involved is your time.

## Where do I find it?

Apple Script is comprised of 8 components, all of which come with the Mac OS Operating System. These are as follows;

The AppleScript Extension is found in the Extensions folder in the System folder. It contains much of the code of the AppleScript language.

The AppleScriptLib is a shared library used by PowerPC native applications that incorporate AppleScript. It is also located in the Extensions folder in the System folder.

The ObjectSupportLib is a shared library used by PowerPC native applications that incorporate AppleScript. It is also located in the Extensions folder in the System folder.

Scripting Additions to OSAX are found in the Scripting Additions folder in the Extensions folder in the System folder. These are add-on modules to the AppleScript system. They provide special resources or commands to the AppleScript environment and can be very useful and powerful. There are hundreds of OSAX developed by companies and individuals,

some for sale, some for free and often found in online services and the Internet. AppleScript ships with a standard set of OSAX.

This is a Dialect file. It is found in the Dialects folder in the Scripting Additions folder in the Extensions folder in the System folder. AppleScript can be read or written in English, French and Japanese.

The Inline Filter extension provides resources to certain scriptable applications using WorldScript. Again, it is found in the Extensions folder in the System folder.

The Scriptable Text Editor is a small scriptable application that has basic text handling abilities such as applying type styles, colours, sizes and type faces to the text in its documents. It is a useful file for learning how to manipulate files with text and AppleScript.

OSA Menu is a system extension that adds a script menu to the right of your screen. You can add your compiled scripts to this menu and launch them by selecting them from the OSA Menu.

## LinkUP! and AppleScript

By combining both AppleScript and LinkUP! together you have the basis for an even more powerful data publishing solution. Most of the standard LinkUP! functions can be controlled by AppleScript without any need for User intervention. You can batch process the updating of documents, the creation of Indexes and table of Contents even build complete documents from scratch. The following is a list of available LinkUP! functions and their corresponding AppleScript command and Syntax.

<b>LinkUP! Command</b>	<b>AppleScript Command</b>
Update QuarkXPress	
Export to Database	
Batch Update QuarkXPress	
Batch Export	
Process Tags	
Template Build	
Master Page Build	

# APPENDICES

APPENDICES

# XPress Tags

XPress are in essence a complete tagging language that allow you to define all the QuarkXPress character attributes and paragraph formats, but in an ASCII text file format.

The use of XPress Tags is a powerful facility and for full explanation on using these codes to format text, refer to your QuarkXPress Reference Manual that came with your copy of QuarkXPress.

The following is a list of all the currently defined XPress Tags.

## Character Attributes

When you specify type styles using XPress Tag codes, <P> always sets the type style to Plain. When you specify any other type style (for example, <B> for Bold), that style is applied if it isn't already specified and removed if it has been specified. For example, the first time that you specify <B>, the Bold type style is applied to the text that follows. If you enter <B> again, the Bold type style is not applied to the text that follows. If you enter <\$>, the type style sheet is set to the one specified in the current style sheet. If a style sheet is not currently applied, the Normal style sheet is used. When entering XPress Tag codes, remember to use foot and inch marks ( ' and " ) instead of typesetter's quotation marks.

Plain	<P>
Bold	<B>
Italic	<I>
Outline	<O>
Shadow	<S>
Underline	<U>
Word Underline	<W>
Strike Thru	</>
All Caps	<K>
Small Caps	<H>
Superscript	<+>
Subscript	<->
Superior	<V>

Type style of current style sheet      <\$>

When any of these commands are followed by a \$ (for example, <f\$>), the attributes are set to the values specified in the current style sheet. If a style sheet is not currently applied, the Normal style sheet is used. (A # character in this list indicated a numeric value.)

Change text Font	<f"font name">
Change text Font Size	<z###.##> in points
Change text Colour	<c"color name"> or <cC, cM, cY, cK, and cW>
Change text Shade	<s###> in percentage of shade
Horizontally scale text	<h###> in percentage of scale
Vertically scale text	<y###> in percentage of scale
Kern the next two characters	<k###.##> in 1/200 em space
Track text	<t###.##> in 1/200 em space
Set Baseline Shift	<b###.##> in points

## Paragraph Formats

Left-align paragraph	<*L>
Centre-align paragraph	<*C>
Right-align paragraph	<*R>
Justify paragraph	<*J>
Force-justify last line in paragraph	<*F>

This is just a selection of the most commonly required XPress Tags for use with LinkUP! in configuring options such as Trapping. Should you require details of other XPress Tags please consult your QuarkXPress Reference Manual.

# Glossary

## **ASCII**

American Standard Code for Information Interchange. This is the type of file format that LinkDB works with.

## **Data Source**

This is usually the database that contains your data although doesn't have to be a database. The data source can equally be a spreadsheet or word processor.

## **Dongle**

A security device for combating software piracy. The computer usually checks for the existence of the dongle. If the dongle is not present the computer will refuse to start the application.

## **Hotkey**

These are shortcut keys that when pressed perform a predefined task such as copying text. Hotkeys are usually setup to save the time take to access menu commands.

## **Key fields**

A Key field identifies the information to which the pricing data is associated. This key field is usually a stock number or product code but can actually be anything you like as long as it is unique and appears on your Quark page.

## **Markups**

Markups are the extra pieces of information that Pricelt places on any of the text in your Quark document that you wish to be updated. This might be telling Pricelt which piece of text is the Key field or which prices are associated with each key field. All of these are called Pricelt markups.

## **ODBC**

Open DataBase Connectivity

## **LRN**

LinkUP! Record Number

# Index

- AppleScript, 16, 211, 212
- Clear Markups, 17, 55
- Copy Markups, 17, 56, 130, 135, 139, 141
- Data Source, 17, 18, 26, 33, 71, 72, 77, 78, 83, 91, 94, 95, 100, 106, 107, 108, 109, 110, 112, 160, 216
- Database, 17, 22, 29, 35, 39, 72, 78, 81, 88, 91, 93, 95, 106, 110, 112, 119, 120, 160, 162, 163, 197, 198, 212
- Dongle, 18, 24, 25, 27, 28, 216
- Floating Palettes, 9, 46, 118
- Footers, 182
- Headers, 182
- Index, 16, 17, 49, 53, 56, 65, 117, 119, 153, 167, 168, 169, 170, 171, 172, 173, 175, 176, 183, 205, 217
- Installing LinkUP!, 19, 21, 33, 202, 210
- Key Field, 41, 42, 43, 48, 50, 59, 63, 82, 83, 99, 101, 102, 103, 119, 120, 121, 122, 124, 126, 127, 128, 129, 130, 131, 134, 135, 139, 140, 141, 148, 149, 150, 152, 154, 155, 156, 158, 188, 189, 192
- LinkDB, 18, 23, 30, 31, 33, 39, 50, 67, 68, 71, 72, 75, 77, 78, 81, 83, 93, 94, 95, 96, 100, 107, 108, 112, 139, 161, 216
- LinkUP! Record Number, 16, 42, 44, 63, 76, 84, 120, 216
- Load Project, 59, 76, 161
- Markup, 16, 17, 46, 47, 48, 53, 58, 61, 62, 65, 66, 79, 82, 83, 84, 85, 98, 99, 102, 103, 104, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 135, 136, 137, 140, 141, 142, 143, 144, 145, 146, 147, 148, 150, 151, 152, 153, 154, 157, 158, 162, 165, 167, 169, 176, 181, 182, 183, 191, 203, 204, 205
- MS Excel, 91, 93, 202
- ODBC, 11, 18, 22, 26, 29, 30, 31, 32, 33, 39, 50, 91, 92, 93, 94, 95, 96, 97, 100, 107, 108, 112, 200, 216
- Paste Markups, 17, 61, 130, 133, 135, 138, 139, 140, 141, 144, 148
- Pictures, 61, 68, 70, 75, 76, 81, 84, 85, 165
- Prefix, 89, 113, 114
- Pricelt Reports, 194
- Proof, 62, 63, 134, 139, 148, 163, 191, 192, 193, 194
- Relational, 151, 154, 184
- Resequenece, 63
- Save Project, 64, 160
- Suffix, 113
- Support, 11, 12, 15, 16, 18
- Table of Contents, 4, 14, 17, 49, 54, 57, 65, 119, 170, 175, 176, 177, 178, 179, 180, 181, 182, 183, 205
- Tags, 61, 62, 89, 90, 114, 115, 160, 166, 199, 203, 204, 205, 206, 207, 208, 212, 214, 215
- Template, 75, 212