



Xdata

Version 4.2 / 6.2

Read Me and User Guide Addendum

January, 2004
Em Software, Inc.
www.emsoftware.com

Welcome to the world of data publishing with Xdata™ for QuarkXPress®!

Introduction

This user guide addendum details what's changed in the Xdata 4.2 and 6.2 releases (January, 2004) since the Xdata 3.01/4.01 release, and acts as a companion to the Xdata 3/4.01 *Users Guide*, which is still quite accurate, as far as it goes.

The general idea is that Xdata 4.2—for QuarkXPress 4.11—and Xdata 6.2—for QuarkXPress 6.x—are identical in function, except where noted below (see the Xdata 6.2-specific changes below). They ship as a pair, so you can use the appropriate XTension for your preferred version(s) of QuarkXPress.

All documentation applies as well to both Mac OS and Windows versions of Xdata, except where noted.

Instant installation guide

To install Xdata, drag the appropriate XTension file from the top-level Xdata installation folder (Xdata 4.2 for QuarkXPress 4.11, Xdata 6.2 for QuarkXPress 6.0 and later) into the XTension sub-folder of your installed QuarkXPress folder, and (re-)start QuarkXPress.

Then, when prompted by Xdata, enter your name, affiliation and the serial number you were given when you purchased Xdata, and press **OK**. If you are only trying out a demonstration version, press the **Demo** button.

(This is spelled out in some detail in the *Users Guide*.)

Note that Xdata 6.2 will not accept an old-style serial from Xdata versions up through 4.1.

Xdata-supported QuarkXPress versions

We fully support Xdata 4.2 under QuarkXPress 4.11, Mac OS (Classic) and Windows.

We **do not** support Xdata 4.2 under QuarkXPress 5.0. It might work in your case, but you have to use it at your own risk: if you run into problems, we probably can't help.

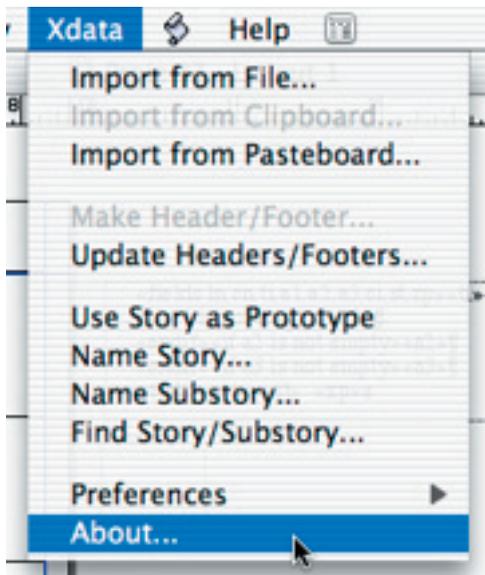
We fully support Xdata 6.2 under QuarkXPress 6.0 and later, Mac OS X and Windows.

As a side note, starting with these releases of Xdata, we no longer support QuarkXPress 3.3 on any platform.

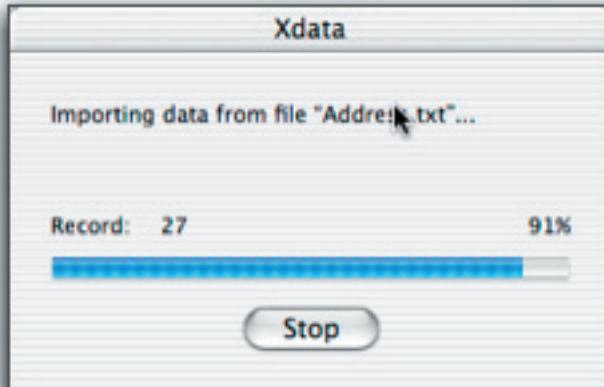
Xdata 6.2 changes vs Xdata 4.1

Xdata 6.2 now native under Mac OS X

The biggest change for Xdata 6.2 under Mac OS (involving massive internal re-engineering) is that it is a fully native Carbonized XTension, running under QuarkXPress 6.x for Macintosh, as seen in the screenshot below.



Also, as noted below in more detail, Xdata has a simplified user interface during import, showing just a progress panel with a Stop button (instead of the more complex control panel), as seen in the following screen shot.



Xdata/XPress now scriptable in the background under Mac OS X

Under Xdata 6.2, QuarkXPress for Mac OS no longer has to be in the foreground (i.e., the currently-active application) to run Xdata scripts, including updating headers and footers (under Xdata 4.1, XPress 4.11 must still be the foreground process, or else scripting will hang at some point). Automating Xdata+XPress with scripting is now a much more system-friendly function and less error-prone.

Xdata 4.2 / 6.2 changes vs. Xdata 4.1

Control panel simplified to progress panel

Xdata no longer has a full-fledged control panel with buttons for setting preferences before starting, pausing in the middle of importing, etc. This user interface was overly complex for what it accomplished (and impossible to implement well under XPress 6.x), so now Xdata automatically starts importing as soon as you select an **Import from ...** menu item, providing feedback and control via a simple progress panel. The 4.x panel is shown below (see earlier for a screen shot of the progress panel under 6.x.).



This change means you'll have to set your preferences before you invoke the import. The only real inconvenience is that the **Range...** preferences must be set ahead of time, rather than during the import process.

Also, this means the general preference to start automatically is gone, since that's the only available behavior.

Preference dialogs now panels

Xdata now has a single preference dialog with four panels for Data, View, Range, and General settings, though there are still four **Preferences** sub-menus for accessing those preference panels directly. The preference panels behave exactly like their Xdata 4.1 counterparts. For an example of the preference dialog under Xdata 6.2, see the character set translation description below.

Picture clipping changes

When importing an image into a newly-created anchored picture box during an import, Xdata now honors all clipping settings applied to the image in the corresponding picture box in the prototype, as outlined below.

If the prototype picture box is empty, the image will load as if you used the **Get Picture...** dialog, namely, if the picture contains a default path, the default path will be used; otherwise, the box's clipping will remain **Item**.

If the prototype picture box is not empty, then any clipping settings applied to that image will be transferred to the imported image. Specifically, if its clipping setting is

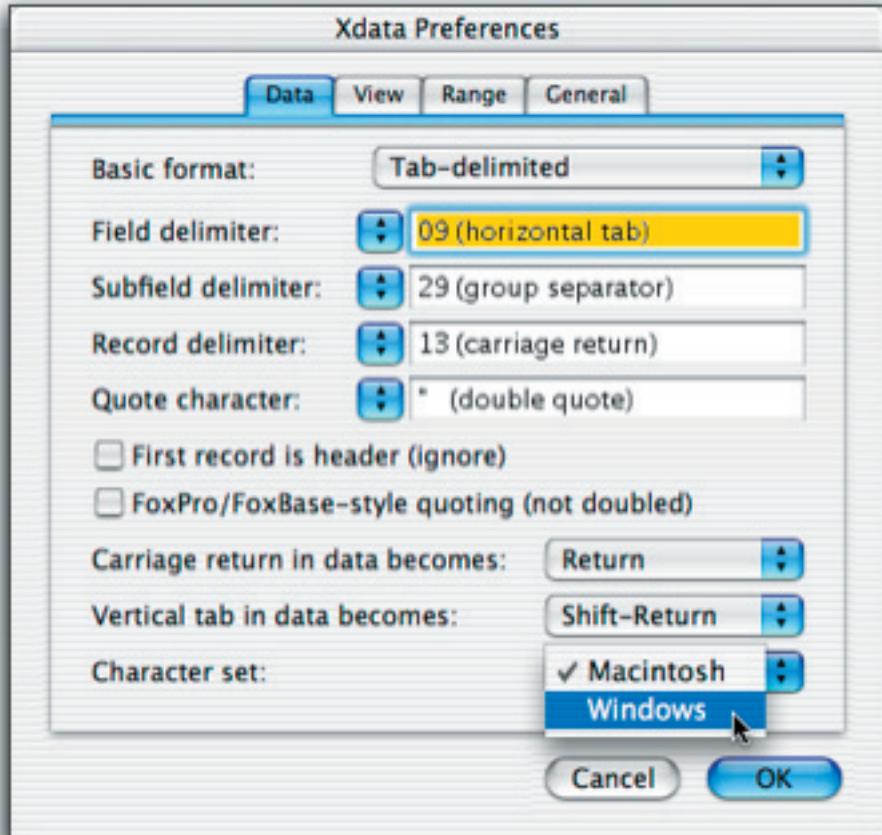
- **Item**, the image will always be set to **Item** clipping.
- **Embedded path/Alpha channel**, if it exists, a path with the same path index will be selected. For example, if the prototype picture's second path in the path list is selected, then the second path in the loaded image's path list

will be used. The **Invert**, **Outside Edges Only**, and **Restrict To Box** settings will be preserved, as well as any tolerance settings that might apply to an alpha channel.

- **Non-white areas**, a non-white path will be generated from the new image using existing outset and tolerance settings. The **Invert**, **Outside Edges Only**, and **Restrict To Box** settings will be preserved.
- **Picture bounds**, a bounding path will be generated for the new image using the existing outset and **Restrict to Box** settings.
- **User-modified**, the existing user-modified path will be scaled appropriately and applied to the new image. The outset, **Invert**, **Outside Edges Only**, and **Restrict To Box** settings will be preserved.

New Mac OS/Windows character set translation

Xdata now has character set translation, where the character set of the source data (Macintosh or Windows) may be selected in the Data preferences dialog. This works in Xdata under both Mac OS and Windows (for full cross-platform translation). A screen shot of this feature is shown below, where the Windows character set is being chosen.



New AppleEvents for scripting

Xdata has a new **import data from** AppleEvent which accepts, as a direct parameter, either a file (alias, etc.) or a string containing the data to be imported. Otherwise, this event is exactly like **import data from file**.

Xdata also has a new **update headers** AppleEvent which triggers a headers update for the currently selected text story (just like the menu item). Optionally, a story name may be specified with the **using story** parameter, to update that story's headers.

Here's how Xdata's AppleEvent event suite now appears.

Xdata™ Suite: Em Software's Xdata data-importing events.

import data from file: Import (and format) raw data file into a target story (and optionally sub-story) using a prototype story.

```
import data from file alias
    into story plain text
    [substory plain text]
    using prototype story plain text
    [starting with record integer]
    [ending with record integer]
[Result: anything]
```

import data from: Import (and format) raw data (a file alias or a string) into a target story (and optionally sub-story) using a prototype story.

```
import data from anything
    into story plain text
    [substory plain text]
    using prototype story plain text
    [starting with record integer]
    [ending with record integer]
[Result: anything]
```

update headers: Update all headers associated with the current (or optionally specified) story.

```
update headers
    [using story plain text]
Result: anything
```

Miscellaneous changes

Xdata 4.2 and 6.2:

- changed, in the Data preferences panel, the **Basic format:** menu to update “live,” reflecting the current data settings as you change them.
- support negative runaround on anchored boxes, since XPress supports it. But be careful what you do with it—XPress can do some strange things in some cases of negative runaround; you’ll certainly get strange on-screen artifacts at times.

Bug fixes

Xdata 6.2 and 4.2:

- fix a bug where imported images would have incorrect offsets if the prototype’s picture box contained a rotated template image, and picture placement was set to **Top left**.
- fix a bug which caused crashes when using polygonal/bezier box shapes with anchored picture boxes

- fix a bug which crashed XPress 5.x when invoking Xdata’s **About...** dialog (Windows only).

Xdata 4.2 / 6.2 changes vs Xdata 4.01

pictureimportoptions Picture Property

Xdata has a new picture property **pictureimportoptions**, which controls how a given picture is imported. The options are the sum of the following independent values:

- 1 force creation of a high resolution preview (the default)—like using the shift key when importing manually;
- 2 force import of a picture as a 256-grey scale—like using the control key;
- 4 force import of an RGB picture as greyscale or a greyscale picture as line art—like using the command key;
- 8 force import of line art as greyscale—like using the option key.

E.g., **set pictureimportoptions of picture 1 to 2** would force grey scale import and produce a lower-resolution preview for the first picture in the prototype (**3** would force a grey scale import with a high-resolution preview).

Anchored Boxes inside Repeat Loops

Xdata now properly handles anchored picture boxes inside repeat loops. The only restriction is that any **set** statement referencing a given picture box (e.g., **set the filename of picture...**) must be at the same loop level as the anchored picture box. (The “same loop level” includes being outside any repeat loops altogether, as in previous versions.)

There are no restrictions on the depth of loop nesting nor the number of anchored boxes inside each loop.

E.g., the prototype

```
<<set filename of picture 1 to “x.tiff”
[anchored picture box 1]
<<repeat with ix=1 to 3
[anchored picture box 2]
<< set filename of picture 2 to “y.tiff”
<<end repeat
```

will, for each record, anchor a picture box containing the file x.tiff, and then anchor three picture boxes each containing the file y.tiff.

Note that it doesn't matter whether you set the filename of the picture box before or after the actual anchored box (as in previous versions). And, as before, picture boxes are numbered statically, in order of their appearance in the prototype, regardless of conditionals and repeat loops.

fileinfo() function

Xdata has a new function **fileinfo(which, kind)** that returns kind information about file which, where which is 0 for the current document and 1 for the current input file, and where kind is 1 for the file's base name and 2 for the file's full path name.

If which is 1 and the input source is the system clipboard or the pasteboard, **fileinfo()** returns **clipboard** or **pasteboard**, respectively.

E.g., **fileinfo(0, 2)** might return `em:engineering:Xdata test files:test doc.qxd`, and **fileinfo(1, 1)** might return `input.txt`, under Mac OS.

boxinfo() function

Xdata has a new function **boxinfo(n)** that returns various information about the current box containing ("of") the current insertion point.

boxinfo(1) returns a unique identifying number for the current box. This identifier is unique only for this particular document session in this particular session of XPress: if you close and re-open the document or exit and re-start XPress, no uniqueness is guaranteed. I.e., this number is only good for ephemeral comparisons, e.g., to detect whether you're in the same box as an earlier test point, etc.

boxinfo(2) returns the 1-based index of the current box in the text chain on the current page. E.g., if you have three boxes linked on the current page, **boxinfo(2)** will return **1** when the insertion point is in the first box on the page, **2** in the second box on the page, and **3** in the third box. And, to further this example, in the next (first) box on the next page, **boxinfo(2)** would return **1**.

boxinfo(3) returns the 1-based index of the current box in the text chain on the current spread. E.g., if you have the example above, on the first page of the spread, **boxinfo(3)** would return the same values as **boxinfo(2)**. On the second page of the spread (assuming the same number of boxes on this second page), **boxinfo(3)** would return **4, 5, and 6** in the first, second and third boxes, respectively, of this second page, while **boxinfo(2)** would return **1, 2, and 3**.

Pageinfo(5) sub-function

Xdata has a new sub-function, **pageinfo(5)**, which returns the name of the master spread applied to the current page at the current insertion point. Note that you can have different master spreads applied to each page in a single document spread, and this function will reflect such a situation accurately.

fieldvalue() function

Xdata has a new function **fieldvalue(index)** or **fieldvalue(index, which)** which returns the value of the current record's indexth field in the first case, and returns the value of the previous, current, or next record's indexth field, as which is **1**, **2**, or **3**, in the second case.

E.g., **fieldvalue(1)** returns the value of the first field, **fieldvalue(25)** returns the value of the 25th field, **fieldvalue(fieldindex(name) - 1 + idx)** returns the value of the idxth field after the name field (see **fieldindex()** description below), etc.

A field index which is out of range (less than 1 or more than the number of fields in a given record) causes **fieldvalue()** to return an empty value. A which value out of range (not **1**, **2**, or **3**) also causes **fieldvalue()** to return an empty value.

Note that you can access fields that aren't even named; e.g., if you had three named fields **name**, **address**, and **phone**, and there were several more unnamed fields after those first three in your actual data, you could access them with **fieldvalue(4)**, **fieldvalue(5)**, etc.

fieldindex() function

Xdata has a new function **fieldindex(fieldname)** which returns the field index of the named field.

If the given argument to **fieldindex()** isn't a bona fide field name, then **fieldindex()** returns 0.

Miscellaneous changes and additions

Xdata 4.2 and 6.2:

- make all tagged text state global for the entire import, so that, e.g., any translation table entries, macro definitions, etc., only need be defined once to be effective for all **put styled** statements in all records.
- support the new **&tt[2]e** tag of Xtags 3/4.1, which can be very useful for doing string substitution. E.g., a **put styled unquoted "<&tt2e(-,--)" & phone** would output the field **phone**, translating any single dash to a double (em) dash. Note that, in conjunction with the previous change, entering a translation once enters it for all records.
- capture any prototype's picture contrast setting (on a sample picture) and applies it to like-kind pictures at import time. E.g., an anchored picture box with grayscale contrast applied (to a sample grayscale picture) will set the same contrast on any grayscale picture imported into the given picture box (and will ignore it unless the picture type is the same, i.e., grayscale vs. color vs. line art).

- support Xcatalog tags on anchored boxes in prototypes, just as the 4.01 release supported them in text in prototypes. (You would need to use this in conjunction with Xcatalog 4.1 or later, since the latter now supports **Key from text <<** and **Key from text >>** on anchored boxes.)
- force selection of the XPress document preference **Edit > Preferences > Document > Master Page Items: Delete Changes** when starting an import, to ensure that master page changes work as expected in Xdata. (This seems to be required in XPress 4.11 to work reliably in all cases, due to some unexplained XPress change.)
- support spline anchored box shapes.
- support the larger keys (255 characters maximum) of Xcatalog 4.1 and later.
- support AppleEvent-based box names as well as Xdata-based box names in the **import data** AppleEvent [Mac OS only]. If there is more than one box with a given AppleEvent name, it will choose one box at random. (I.e., don't use the same name for multiple boxes if you want to avoid confusion.)

Bug fixes

Xdata 4.2 and 6.2:

- fix a bug whereby master page application could fail because of what happened with the previous import (in the same XPress session).
- import pictures with a high-resolution screen preview.
- work around a QuarkXPress 4.1 bug whereby other XTensions' activities could make Xdata hang during import. (QuarkLink, in particular, was an obvious trigger under QuarkXPress 4.1.) [4.x only]
- fix a bug whereby automatic anchored box resizing (e.g., shrink box to fit image) during importing could cause an XPress crash. [4.x only, Windows only]
- fix a bug whereby anchored box creation could fail if the original (unanchored) box was created on a spread larger than the current spread at data import time. This could cause an Xdata panic or even an XPress crash.
- fix a bug with repeat loops in the previous beta (7) that would cause Xdata to terminate an import prematurely.
- work around a serious XPress bug whereby using the **put ... marked ...** or **put ... hidden marked ...** statements could cause document corruption or crashes.
- work properly when an ASCII line feed (decimal 10) is used as the record separator. Before, it would consume all blank records in this case as a side-effect

of “eating” line feeds after carriage returns (in its handling of CR-LF DOS-style line endings).

- no longer interfere with document opening (e.g., previous versions of Xdata would conflict with alap’s XPert Open XTension).
- fix a bug whereby Xdata’s resources could have been confused with another XTension’s resources, resulting in strange dialog and error messages. [Windows only]
- fix a bug whereby some of the window “decorations” (e.g., scroll bars, grow handle) might not be properly redrawn after importing, when using the “hide document during import” option.
- fix a bug whereby the use of **put styled** with tabs could cause memory corruption and lead to a crash. [4.x only]
- support the use of style-default references (e.g., **f\$** to switch back to the current style’s default font) in XPress Tags in a **put styled** statement. (It’s a bug that it didn’t before, to be honest.)
- fix a bug whereby sometimes Xdata 4.0 would hang after importing (until the user switched in and out of XPress). [Windows only]

Manual Errata

The following error has been discovered in the *Xdata 3/4 User Guide*:

On page 182, after the reference entry for “next”, pencil in an entry for a missing “the number of <chunks> in <expression>” entry, as in:

the number of [chars | words | lines | items] in <expression> -- see page 117.