

InCatalog InCatalog Pro User's Guide

Em Software, Inc.

www.emsoftware.com

July, 2004

InCatalog™ is a plug-in to Adobe InDesign® that enables you to link document elements—both text and graphics—to corresponding database elements, and gives you the power to update in both directions (document from database, database from document), make locale-sensitive versions (e.g., changing prices or swapping languages), and more. InCatalog can help you automate the publishing, updating, and versioning of projects ranging from one-page advertising flyers displaying a handful of products, to thousand-page industrial products catalogs detailing tens of thousands of products.

Using InCatalog, you can work with off-line data snapshot files (delimited ASCII text files created from any database, spreadsheet, or other application), on-line using FileMaker Pro under Mac OS, or on-line using any ODBC-accessible database under Mac OS or Windows (Pro version only).

As a plug-in to InDesign, InCatalog becomes a seamless part of the InDesign publishing system, once installed. InCatalog runs inside InDesign, under its own menu.

InCatalog comes in two versions, normal and Pro. The only difference is that the Pro version is ODBC-capable, able to talk to any database server that is ODBC-accessible.

Overview of this Guide

Installation The next chapter, “Installation,” covers software installation.

The subsequent chapters cover the four basic aspects of using InCatalog, as follows.

- Data** Dealing with (and preparing, in some cases) your source and destination data is covered in the chapter “Data.”
- Data Descriptors** Preparing and using sets of data descriptors and price styles to inform InCatalog about your data structure and pricing formats is covered in the chapter “Data Descriptors.”
- Linking** Linking the elements in your InDesign documents is covered in the chapter “Linking.” In particular, this covers using the InCatalog data linker palette.
- Updating** Updating the linked document content with new data, or updating your data with linked document content, is covered in the chapter “Updating.”
- Menu, Preferences, Automation** Filling in the rest of the story, chapter “Menu” covers InCatalog’s main menu, chapter “Preferences” covers InCatalog’s preferences, and chapter “Automation” covers automating InCatalog’s updating operations.

Host and System Requirements

To work properly, InCatalog 2.x requires InDesign 2.x, and InCatalog 3.x requires InDesign CS.

Your system will need to comfortably support the particular version of InDesign that you’re using, and InCatalog has no serious system level or resource requirements beyond that of its host application.

Consult the InDesign documentation for detailed system requirements.

What You Should Know

- About Your Computer** You should be familiar with basic Macintosh or Windows concepts and procedures, such as using the mouse, selecting items from

menus, entering information in dialogs, navigating among folders, and manipulating files (e.g. copying, renaming, and deleting).

About InDesign

You should be comfortable with basic InDesign tasks, including working with text and pictures. This manual assumes you have adequate knowledge of InDesign to set up and manipulate documents as required by your particular projects.

About Your Data

This manual assumes you are familiar with using and manipulating the database, spreadsheet, or other data source or destination you will employ in your InCatalog publishing project.

You should understand basic database concepts including fields (a column of data), records or rows (a collection of data cells which comprise all the information for a single data unit or key), and keys (a column containing a value—i.e., part number, SKU, company name, serial number, etc.—that uniquely identifies each record or row, i.e., there is no other record or row with that particular value in the key column).

If you will be using an off-line data snapshot, then you need to know how to export (or import) data in tab- or comma-delimited text format out of (or into) the particular application.

InCatalog+Data for data publishing synergy

You can use Em's InData plug-in in concert with InCatalog to achieve greater data-based publishing automation.

InData is the original InDesign data publishing plug-in. It automates the production of documents such as catalogs, directories, price lists, and schedules which contain repeating data units. InData uses prototype statements—user created scripts, including conditionals and loops—written directly in the InDesign document. Once the prototype is written, data is imported into the InDesign document while the prototype evaluates and formats the data, on the fly, according to the rules outlined in the prototype. InCatalog links can be embedded inside InData prototypes to automatically produce documents with links already in place for later update or extraction.

About this Manual

Terminology In this manual, we'll refer to both product versions as simply "InCatalog," except when we need to distinguish them, and then we'll use the "Pro" and "non-Pro" modifiers (or identify a given feature as Pro only).

Platform-Specific Issues InCatalog has been designed to work as the same (as closely as humanly possible) across operating systems. Differences between versions will be noted in this manual, where appropriate. In selecting illustrations of dialogs, we will usually show Mac OS X versions with Mac OS 9 and Windows dialogs sometimes throws in for good measure (sometimes two or three together.) Note, however, that the logical content of each dialog should be identical under Mac OS 9/X and Windows. Only the visual appearance will differ.

In this manual, we will consistently refer to directories and subdirectories as "folders," a term which is used in both Mac OS and Windows parlance.

Typographic Conventions The names of InDesign and InCatalog menu items, as well as dialog fields, are set in this font. For example:

Select Open... from the File menu. Click the Cancel button to change your mind.

The following syntax represents paths through a series of nested menus: InCatalog / Select Data Source / Snapshot File.

File and folder names within text are set in quotes. For example: ...this file should be placed in the "DD_Sets" folder.

Finally, important definitions will be set in *bold italic and the word being defined will be underlined*.

Which Plug-ins to Install?

Before installing InCatalog, you have to be aware of the fact that it comes in two flavors—one ODBC-capable (the “Pro” product) and one normal (non-“Pro”)—both of which are present in the distribution folder, since everything else about the distribution is identical.

The plug-in files InCatalog Pro 2.x and InCatalog Pro 3.x are ODBC capable, the files InCatalog 2.x and InCatalog 3.x are not.

◆ *It’s important to note that the “Pro”-labelled versions require ODBC libraries to be installed. For Mac OS, the DataDirect libraries are available from our web site, and we support the built-in iODBC manager (OpenLink Software). Under Windows versions later than 98, such libraries are standard.*

Furthermore, each plug-in comes in two versions, one for InDesign 2.x (InCatalog 2.x, Pro and non-Pro), and one for InDesign CS (InCatalog 3.x, Pro and non-Pro).

Installation

Once you know which file you want to install, based on the above, installing InCatalog is quite simple:

1. Download the InCatalog distribution from our web site at <http://www.emsoftware.com/products/incatalog/download> (or ask us to send it via e-mail, if you can’t download it), expand the appropriate StuffIt or WinZip archive to find the InCatalog distribution folder, called InCatalog 2-3.x in the current version. Open that folder to see two sub-folders Plug-ins for InDesign 2.x and Plug-ins for InDesign CS. Open the appropriate sub-folder for your version of InDesign.
2. Drag the appropriate Pro or non-Pro InCatalog plug-in to the Plug-ins sub-folder of the InDesign application folder, and drag

the InFlow plug-in to the same folder, if it's not already there for InData's use. (Note that without help from the InFlow plug-in, InCatalog won't be able to flow in the contents of error log documents longer than one page.) We recommend creating a sub-folder of the InDesign Plug-ins sub-folder called Em Software where you can put the InCatalog or InCatalog Pro plug-in, along with any InData and InFlow plug-ins, and where any related auxiliary files can live undisturbed.

3. If you're using Mac OS 9/X and are installing the "Pro" version of InCatalog, run the Em-customized DataDirect ODBC installer called "Em ODBC Installer"—downloadable from the URL above—to install the appropriate drivers for your system. Or, if you'd rather use the iODBC manager from OpenLink, visit their web site (www.openlinksw.com) to purchase specific drivers (such as for MySQL or PostgreSQL). Under Windows, you should already have appropriate ODBC drivers installed as part of your system.
4. Launch (or, if running, quit and re-launch) InDesign.
5. When the InCatalog serialization dialog appears, enter your name, company affiliation and the serial number you purchased (see sample below). Or, if you have no serial number and you're installing for demonstrating or testing only, press the "Demo" button in the serialization dialog, which puts InCatalog in demonstration mode (see next section). Note that if you have installed the "Pro" version, only a "Pro" serial number will work.

Please personalize your Em product, entering the serial number you were given when you purchased it. If you don't have a serial number, press the Demo button to proceed.

Product: InCatalog™

Name: Esmeralda Quakenbush

Affiliation: Q Associates

Serial #: ASDF-ASDF-ASDF

Demo OK

6. If you're using InCatalog in demonstration mode, you'll also get an InFlow serialization dialog, in which you should press the "Demo" button to proceed. (If you're using a serialized

copy of InCatalog, InFlow will automatically work without serialization, since it's bundled.)

◆ *You can find your name, affiliation and serial number any time by selecting the InCatalog / About... menu.*

Demonstration mode

When in demonstration mode, InCatalog does not cripple itself nor InDesign, beyond limiting the number of updated elements to 50 (in either direction), for each update action, batch or interactive.

Copy Protection

InCatalog Pro is serialized but, like InDesign, not network-copy-protected, so it will never complain about too many copies running at once.

However, if you fail to purchase enough copies to cover the maximum simultaneous usage of InCatalog at your site, you are legally in violation of your license agreement, and more importantly, ethically guilty of stealing from Em Software.

We have some of the most aggressive multi-pack discounts in the industry to make multiple copies affordable, so we encourage you to stay legal as well as ethical.

When updating a document from a source database (what we'll at times call "updating" for short) or updating a destination database from a document ("extracting" for short), InCatalog can address three different types of data: off-line data "snapshots" (delimited ASCII text files), FileMaker Pro on-line (Mac OS only, including FileMaker Pro runtime databases), and ODBC-accessible databases on-line such as Oracle or SQL Server (Mac OS or Windows).

This chapter discusses dealing with these different types of data.

Data Snapshot Files

If the data is from a typical database, spreadsheet or manually prepared file, it must be saved as a tab- or comma-delimited text file. This file is called a data "snapshot" because it reflects a consistent state of the database at a specific point in time. Consult the documentation of your particular database or spreadsheet application for instructions on how to save a file as delimited text.

InCatalog preferences allow you to choose tab- or comma-delimited data. The FoxBase Pro tab-delimited format (a quirky variation on the basic tab-delimited format) is also supported.

When you export your data for use with InCatalog, it must be in the same column order as the columns (what we call "fields") in the data descriptor you intend to use (see chapter "Data Descriptors").

When you export data from your document with InCatalog, it will be in the column ("field") order given by the current data descriptor.

To select a data snapshot file as source or destination, use the InCatalog / Select Data Source / Snapshot File or the InCatalog / Select Data Destination / Snapshot File menu.

If you want to use a snapshot file for interactive update (or just keep one open for multiple updates in either direction), you can choose InCatalog / Open Data File... or Create Data File... to open an existing file or create a new one (for extraction only). When you're finished, use InCatalog / Close Data File.

You can choose various character sets for the snapshot data in both directions, using the Preferences / Data Snapshot... dialog's Character set menu: Windows, Macintosh and Unicode (encoded as UTF-8 on output, and with UTF-8, UCS-2BE/LE recognized on input).

FileMaker Pro On-Line (Mac OS only)

If the source or destination is a FileMaker Pro database or FileMaker Pro Runtime application, open the databases or applications that contain the records to be used for updating or extraction, and be sure that all such records are all in the current set of found records.

You can use multiple open databases simultaneously in FileMaker Pro by employing the [**D**"database name"] qualifier in your data descriptor (see the chapter "Data Descriptors" for details). Note that all that such referenced databases must be open when doing updates in either direction.

You must qualify each field in your data descriptor with its related FileMaker field name (cf. the [**F**"field name"] qualifier), if the data descriptor field name doesn't match the database field name.

◆ *One current problem with using FileMaker Pro (or a Runtime application) is that the keys in your document have to be completely unique across the whole found set in order to work reliably. E.g., if you have a key "100" and there are two records with key fields "100" and "1000", then an InCatalog reference to key value "100" will randomly find one or the other record (or both). If this is a serious problem, one work-around is to create a computed key field in your database that wraps the real key field with special characters such as <>, and to use the computed key field as your key as far as InCatalog is concerned, linking with key values such as <100>, etc.*

To select FileMaker Pro databases on-line as your source or destination, use the InCatalog / Select Data Source / FileMaker Pro Online or

the InCatalog / Select Data Destination / FileMaker Pro Online menu (ditto for FileMaker Pro Runtime on-line).

Once you have FileMaker Pro selected for your source or destination, be sure FileMaker Pro or a Runtime application is running when you're doing updates in the chosen direction(s).

ODBC-Accessible Databases On-Line (Pro only)

ODBC (an acronym for “open database connectivity” coined at Microsoft many years ago) is a technology enabling a unified software interface to any number of database back-ends (SQL and otherwise) on-line, using TCP/IP (the Internet's transport medium) as the connecting pipe.

Under Windows, ODBC is generally available built-in (or at least installable from system discs), while, under Mac OS, we supply an Em-customized set of Data Direct drivers with the Pro product. Under Mac OS, you can also use generic Data Direct drivers purchased from another source such as www.datavations.com's online store or those installed as part of the extended Microsoft Office or FileMaker Pro packages.

Multiple ODBC managers are supported, selectable from the Open ODBC Connection... dialog, the data descriptor (using the “^” qualifier – see chapter “Data Descriptor”), or via a script parameter. Support for Data Direct and OpenLink driver managers is built-in. When only one of these managers is available, it will be used as the default manager. When both a Data Direct and an OpenLink manager are available, the user will be prompted to select one in the Open ODBC Connection... dialog if a manager hasn't been specified in the DD. (Mac OS only)

To use ODBC-accessible databases with InCatalog Pro, just like any other application, you'll have to configure either a system-wide or user-specific DSN (data set name) that points to your real database. In InCatalog, you refer to any database via its DSN's name, using the [&“dsn”] data descriptor qualifier on the key field (see chapter “Data Descriptor”), which must be present (there are no interactive facilities for choosing a DSN).

Your data descriptor may contain a specification for the user name and password for the ODBC connection, but they can also be prompted for interactively, to avoid compromising database security. (You can also supply the user name in the data descriptor, having InCatalog prompt only for the password.)

You can use multiple tables simultaneously in an ODBC database by employing the [D“table”] qualifier in your data descriptor (again, see the chapter “Data Descriptors” for details). And, in fact, you must qualify each field in your data descriptor with its related table name (and column name, if it doesn’t match the data descriptor’s field name).

To select an ODBC-accessible database as source or destination, use the InCatalog / Select Data Source / ODBC Online or the InCatalog / Select Data Destination / ODBC Online menu.

When you want to use an ODBC database interactively or for multiple updates in either direction, use the InCatalog / Open ODBC Connection... menu, and the InCatalog / Close ODBC Connection menu when you’re finished.

Otherwise, when you start an update in either direction, InCatalog will automatically open the connection to the database, prompting you if you haven’t supplied your username and password in the data descriptor.

Before InCatalog can link your documents to your data, it needs to know about the general structure of each database (set of data) to be linked. InCatalog supports multiple databases linked to a single document, as well as any number of documents linked to any particular database. Each database (really, each potential set of data, not a particular database) is described by a data descriptor file, which lives in a common file system folder—called a data descriptor set—with all other related utility files you might need, such as a price style file and any picture-bearing folders or aliases to them.

This chapter describes the format of data descriptor files, the concept of data descriptor sets, the format of price style files, and the concept of picture folders. But first, we'll give you a “quick start” in case you're in a hurry and have very simple needs.

Quick Start

In the simplest and most common case, with only one source or destination data set, no price styles, and no picture folder aliases, you can do the following to set up your data descriptor.

- Create a special folder with a short name suggestive of the overall project, followed by “DD Set” (e.g., “WCQ DD Set” if this is a women's clothing quarterly catalog project).
- In that folder, create a text file with a name suggestive of the database source or destination followed by “DD” (e.g., “Prices DD.txt” or just “Prices DD” under Mac OS). You can create this file with any editor such as SimpleText, or use Word or InDesign to create a new story, saving it as plain text.
- In that text file, list your database's field names on the first line, preceded by a tab, each field separated by a tab from the previous, in the proper order (i.e., the export or import field order, if you're using a data snapshot file). For example, for a pricing

database containing just two fields, a SKU (stock-keeping unit) and Price, the file would be a single line as follows. (The arrows are actually tabs, and the [K] denotes the key field and the [P] tells InCatalog that this is a price field.)

→SKU [K]→Price [P]¶

- Choose InCatalog / Select Data Descriptor... and, in the ensuing open file dialog, pick the data descriptor file you just created.
- Go ahead with linking and updating your documents. (See the following chapters “Linking” and “Updating.”)

You can read further in this chapter if you want to know more about the details, or have more complex needs.

Data Descriptor Sets

Definition *A data descriptor set (hereafter referred to as the “DD Set”) is simply a folder containing the component files InCatalog needs to operate.* These components are the data descriptor file(s) for your InCatalog project, along with any optional files like the price styles file. Additionally, you can place folders (or, perhaps more commonly, aliases/shortcuts to folders) containing the graphics or pictures for a project in the DD Set folder, to make picture importing simpler.

Creating and Naming a DD Set

To create a DD Set, simply create a new folder in your file system and give it a short name suggestive of the overall project purpose. It is a good idea to add “DD Set” to the folder name to distinguish its purpose, but this isn’t necessary. For example, “WCQ DD Set” might be a useful name for a women’s clothing quarterly catalog project.

All data descriptor files pertaining to your project must be placed in this folder, along with any price style file or picture folders/aliases.

InCatalog is also capable of handling links from and to more than one data source or destination structure in a single document. In this case, you would create one data descriptor file for each data source or destination.

Moving a DD Set

If you have to move a DD Set folder to a new place on your system, or move it to another system (or share it from a server), you should

have no problems as long as you keep its name unchanged. You'll have to re-select any contained data descriptor that was selected as the current (remembered) data descriptor in any copy of InCatalog, using Select Data Descriptor... to re-select the data descriptor in its new location.

(And don't forget to change any automation scripts that refer to the DD set path explicitly.)

Renaming a DD Set

If you have to rename a DD Set (or you move a data descriptor to a DD Set with a new name), you'll have to manually adjust each document that was linked using the original DD set.

Do this by opening each affected document, and holding down the Option (Mac OS) or Alt (Windows) key while choosing Select Data Descriptor... from the InCatalog menu, then saving the document (even though you haven't explicitly changed it). This will re-set the DD Set remembered by InCatalog for the document.

Sharing a DD Set on a File Server

InCatalog is designed to share DD Set folders on a file server. When you select a data descriptor, the file is opened and read into memory (waiting for access if necessary), and then the file is closed almost immediately. (The same thing is true of price style files.) As a result, multiple users can access the shared data descriptor set (and all the components of the set) simultaneously.

Data Descriptors

InCatalog needs to know something about the structure of the data which is to be linked with your InDesign documents. The most basic information InCatalog requires is the name of each database field that could be linked, and which field is the key (uniquely identifying a record or row in the database or spreadsheet by its content). It must also know in what order those fields appear in any data snapshot file (but not for on-line databases, where the access is by name).

Definition

The database field information required by InCatalog is given by a file called a data descriptor. We also refer to the data descriptor as a DD file or even just a DD, for short.

Creating a DD

In its usual form, a data descriptor is a text file containing a first line that lists the field names, in the order in which they are arranged in any data source or destination snapshot file(s) (but if you're only using on-line data, the order isn't important). *The line begins with a tab* and is followed by the first field name. Each subsequent field name is also preceded by a tab. Field qualifiers (properties of the field) are given in square brackets after the field name.

Here are some sample DD file contents (one physical line, though it may wrap here; the tabs and ending paragraph mark are shown here but are of course normally invisible). The "SKU" field is the key field, the prices are qualified as same, and the "description" field is qualified as a tagged text field.

```
→SKU[K]→retail price[P]→sale price[P]→description[T]→picture path¶
```

This paragraph can be typed in a InDesign document and then saved as a plain text file, or typed in a word processor or text editor. *However it's created, it must be saved as a plain text file.* This text file must be placed in the DD set folder (see above).

Naming a DD

There are no restrictions on how the DD file is named. However, it makes sense to use a short name, suggestive of what database structure it describes, and with some indication that it's a data descriptor file—"Pricing DD" under Mac OS, and "Pricing.DD" or "Pricing DD.txt" under Windows.

◆ *Note that it is virtually impossible to change the name of a data descriptor file once you have used it to link elements in a InDesign document; you would have to start over to change it. Therefore, before you name your DD file, think about your choice fairly carefully.*

Using a DD

Once you've created a data descriptor, use InCatalog / Select Data Descriptor..., and pick this data descriptor file in the ensuing file selection dialog. InCatalog will always open the file selection dialog to the data descriptor set folder currently in use (if any) to make it easy to choose among multiple data descriptors in your project.

Naming Fields in the DD

The field names in your data set are the basic information you supply to InCatalog in a data descriptor file. Field names are listed one after the

other, each preceded by tabs (including the first field name, very importantly), all contained on the first line of the file.

Each field you specify in the data descriptor should have a short but descriptive name; these field names show up in the linker palette's field pop-up menu. There are no real restrictions on what characters you can use in a field name, though we recommend sticking to the basic upper and lower case alphabets and the numerics. Alphabetic case doesn't usually matter, so "Retail Price" and "retail price" are the same field in nearly all situations.

For snapshot file-based data, the field names themselves are completely arbitrary, though it's good to make them match the names in the originating or target database for sanity's sake. However, the field order is crucial—you must list the fields in the exact order in which they'll be found (or created) in the snapshot file.

If you're using an on-line database (e.g., FileMaker or ODBC), you should probably take care to match the actual field names in the database, to avoid extra work, but even that's not required if you use the **[F"field name"]** field qualifier (see below). If you are using an on-line database exclusively, then field order doesn't matter—the fields will be accessed by name.

(To determine the field names in a FileMaker application, open the FileMaker database. From the File menu choose Define Fields. A dialog will appear listing every field in the database. Only the fields being accessed by the InCatalog linking process need to be defined.)

Qualifying Fields in the DD

Field qualifiers, or just qualifiers, tell InCatalog how to treat each field being declared in your data descriptor file.

Field qualifiers consist of “[]”-bracketed information following a field name but before the following tab or line ending, with the actual qualifier consisting of a single, case-insensitive character, and with optional data following (surrounded by straight quotes). For example, **SKU[K]**, the “SKU” being the field name, [K] being the qualifier character in brackets. As a longer example, **Retail Price[P"normal"]** is a whole price field specification with a qualifier, “Retail Price” being the field name, the open bracket “[” starting the qualifier, “P” being the qualifier character denoting a

price field, “normal” being the optional default price style specified), and the closing bracket “]” ending the qualifier.

If you don’t use a field qualifier for a given field, then no special treatment will be given to that field beyond the basic update and extraction of the field’s content at update time (and, thus, it can’t be a key field, a price field, a tagged text field, etc.).

Note that many qualifiers provide a default setting for a given field’s properties during the linking process. You can override some of those defaults in the actual link itself. (E.g., a field may be denoted a price field with a particular price style by default, or a tagged text field, but you can turn that treatment on or off anytime for a particular link, using the linking palette.) Providing accurate field qualifiers (when a field is linked using the same properties across a whole document or series of documents, generally) can save immense amounts of time in the linking process, to say nothing of avoiding mistakes in that same process.

Basic Field Qualifiers

The three most common qualifiers for a field are key, price, and style-tagged data. Each of these tells InCatalog how to treat (and not treat) the given field when it appears in a document link. (Or, more accurately for the latter two, how to default the link properties, when it’s created.)

[K] — Key field This qualifier marks the field whose contents uniquely identify each record or row in a data set. Only one field can be designated as the key in any one data descriptor file.

The key field is the linchpin in the InCatalog process. It is through the value in the key field or column that InCatalog knows which record or row to access in the database, when updating a particular link, given that link’s key value.

In merchandise catalog settings a key field is usually a part number field or SKU field in the database. For list rankings—as in top 40 hits or top 50 industries, the rank field would be the key field. In real estate books, it might be the property number.

Examples for these cases: **SKU[k]**, **rank # [K]**, **MLS number[k]**.



Being a key field is incompatible with being a price or tagged text or other content field. Document contents linked to a key field (i.e., where the actual

field being linked is a key field) are normally not updated nor extracted, but only serve as an indirect reference for other links.

[P] or [P"price style"] — Price field This qualifier marks a field which is to be treated as a price, by default. Optionally a default *price style* name can be given as a further qualifier. See the “Price Styles” section of this chapter, below, for details on price style handling.

Examples: **retail price[p]**, **price1[P"discounted"]** (the second declares a “price1” price field with default price style “discounted”).

[L] — This qualifier assigns a default key type to fields specified in a DD, using the specifier **L"type"**, where type can be **L** for **Key from link**, **G** for **Key from group**, **B** for **Key from text <<**, **F** for **Key from text >>**, and **C** for **Key from contents**.

E.g., if you specify a field **productName[L"B"]**, choosing the field **productName** will default to make the current link a link with a **Key from text <<** key type.

This can greatly decrease the amount of manual linking work required; e.g., if you’re building modular product groups, and specify all fields as Key from group except for the key field, which could be Key from contents, then you can link all fields with a single hot key press with all link properties set the way you need them, including the key type, whether it’s a price or XPress Tagged text, and a price style in the case of a price.

[T] — InDesign Tagged field This qualifier tells InCatalog to interpret this field as containing tagged text by default, including converting double and single quotes from straight (" ") to printer’s (“ ‘ or ” ’) quotes as appropriate, in context.

See the **[B]** and **[I]** qualifiers (below) for information about optional **line break** and **independent of context** specification.

[U] — InDesign Tagged but Unquoted field The **[U]** qualifier is like the **[T]** qualifier but without any quote conversion during tag interpretation.

See the **[B]** and **[I]** qualifiers (below) for information about optional **line break** and **independent of context** specification.

The following qualifiers are used with on-line FileMaker Pro and FileMaker Pro Runtime databases. (They are ignored when using data snapshot files.)

[D"database"] — **FileMaker database** Use this qualifier with the actual **database** name in instances where information is coming from or going to more than one FileMaker database (otherwise, InCatalog simply uses the currently-open database).

This qualifier can be used as a precaution against updating information in the wrong FileMaker database.

Note that you can specify different databases for different fields in a data descriptor, but the key field name must be common to all the databases accessed.

[F"field"] — **FileMaker field** Use this qualifier with the actual database **field** name in instances where the field name you're describing in the data descriptor doesn't match the field name in the database. If they do match, then this qualifier isn't needed.

For example, if the data descriptor contains the field name "retail price" but the corresponding database field is "retail.price.1", then you should use **retail price[f"retail.price.1"]** in your data descriptor.

As a combined example, the data descriptor field specification **Part #[K D"Products" F"SKU"]** indicates that the field "Part #" is the key field, comes from the FileMaker database named "Products" and the field is named "SKU" in the database.

[R] or [R"separator"] — **Repeating field** Qualifies the current field as a FileMaker repeating field, with subfields separated by the FileMaker repeating subfield "group separator" (ASCII decimal code 29) character, or the given (optional) separator character.

Fields marked with this repeating qualifier cause InCatalog to insist on a **#subfield** suffix for each use of this field. E.g., if you've declared a field **feature[R]** in your data descriptor, then all linking palette usage of the field "feature" must be of the form **feature#1**, **feature#5**, etc. (meaning the first and fifth subfields of the field "feature", respectively).

The separator character, if given, can be either a single printable character or a multiple-digit decimal ASCII character code. For example, the two qualifiers [**R**",**,**] and [**R**"**44**"] are equivalent, in that they qualify a repeating field that uses a comma (ASCII decimal code 44) as a delimiter between the fields.

[Y"key field database name"] This qualifier specifies (for on-line databases/tables only) by what name to reference this field's keydatabase name.

With this qualifier, you can now specify the database/table containing the field, the field's name in the database/table, and the field's key name in the database/table, so you can effectively have multiple DDs in a single DD, with each group of fields referencing the same database/table and key a "logical sub-DD".

ODBC-Specific Field Qualifiers

The following qualifiers are used with on-line ODBC databases. (They are ignored when using a data snapshot file.)

[^"ODBCmanager"] Specifies an ODBC driver manager library. For example, [**^"iODBC CFM Bridge"**] would specify the Open-Link manager, [**^"ODBC Driver Manager"**] would specify Data Direct's MacOS 9 manager, and [**^"ODBC DriverMgr PPC"**] would specify Data Direct's Mac OS X manager. This qualifier is only useful when multiple ODBC managers are present, in order to avoid having to choose one in the **Open ODBC Connection...** dialog. (*Mac OS only*)

Note that, in theory, any provider's manager library may be specified, so long as it exports the ODBC functions required by InCatalog.

This qualifier may also be used to specify a connection string which is passed directly to the ODBC manager's `SQLDriverConnect()` service. The DSN qualifier will be treated as a connection string if (and only if) it doesn't name an existing user or system DSN and it contains at least one equals sign (=). For example, a file DSN named "foo.dsn" may be specified using the qualifier [**&"FILEDSN=foo"**]. For more information on `SQLDriverConnect()`-compatible connection strings, see this document on Microsoft's site. (Microsoft defined the ODBC standard.)

[&"dsn"] Specifies the ODBC data source name for the whole data descriptor as **dsn**, regardless of which field it qualifies (normally the

key field). Note that this is case-sensitive on some systems such as Mac OS, and is required when using ODBC.

["\$user"] Specifies the ODBC user name for the whole data descriptor as **user**, regardless of which field it qualifies (normally the key field). If **user** is “?” (a question mark), then whenever InCatalog attempts to open an ODBC connection, it will prompt interactively for the user name (and possibly the password—see below).

["password"] Specifies the ODBC password for the whole data descriptor as **password**, regardless of which field it qualifies (normally the key field). If **password** is “?” (a question mark), then whenever InCatalog attempts to open an ODBC connection, it will prompt interactively for the password (and possibly the user name—see above).

[D"table"] — ODBC table You must provide the **[D"table"]** qualifier for every field you want to access in an ODBC database. (Otherwise, InCatalog won't know which table to access when handling links for a given field.)

Note that you can specify different database tables for different fields in a data descriptor, but the key column name (the field marked with a **[K]**) must be common to all the tables accessed.

NB: When linking to a Microsoft Excel spreadsheet, it is necessary to: 1. add a “\$” suffix to the Data Sheet name specified in the **[D"..."]** qualifier, and 2. enclose the sheet name in the **[D"..."]** qualifier within brackets. For example:

[D"[sheetname\$]"]

[F"column"] — ODBC column Use this qualifier with the actual database **column** name in instances where the qualified field name in the data descriptor doesn't match the column name in the database. If they do match, then this qualifier isn't needed.

For example, if the data descriptor contains the field name “retail price” but the corresponding “Prices” table column is “retail_price_1”, then you would use **retail price[d"prices" f"retail_price_1"]** in your data descriptor (specifying both the table and the column name). Alternately, you could use the actual

column name as the data descriptor's field name, viz. **retail_price_1[d"prices"]** as the qualified field name.

If you have database table column names with special characters in them (such as spaces), you could use brackets in the column name qualifier to satisfy some SQL dialects, which would otherwise fail on seeing the special characters in the statements used to access the column in question. E.g., if the “Prices” table actual column name is “retail price” (includes a space), you could use **retail price[d"prices" f"[retail price"]**].

NB: When linking to a Microsoft Excel spreadsheet, it is necessary to enclose the column name within the **[F"..."]** qualifier in brackets. For example: **[F"[fieldname]"]**

[Q] — quoted character data This qualifier tells InCatalog that the qualified field represent a column with character data of some sort (**char**, **varchar**, **longvarchar**, etc.—i.e., requires quoting in the actual SQL constructs that transfer data back and forth).

This qualifier is automatically implied (and therefore not needed) in the case that a field is marked as tagged text (field modifiers **[T]** or **[U]**) or multiple-valued of any sort (field modifiers **[M]**, **[G]**, **[R]**, **[#]**). You only need to use it when you're using a character-typed column with no other qualifiers that imply character data.

You may need to use this qualifier with price columns that include non-numeric characters such as \$, %, etc.

[W] — “where clause” key value Used with key fields, this qualifier tells InCatalog that the key value in the link is to be used directly as an SQL “where” clause, and is not a normal key value.

E.g., if you have a key field “id” declared as **id[K W D"products"]**, then every key value must be a valid SQL “where” clause that identifies a unique row in the table in question, such as **id = 1023** or **id = 1023 and page < 23**.

Note that you can't use the Key from contents linking palette option in conjunction with this feature, that you're limited to 255 characters for the whole clause, that editing a long clause in the small linking palette key field edit area is painful (this feature is present primarily for keys that are being entered externally in tags), and that InCatalog can't attempt record creation when updating data in

the presence of this modifier. (Also note that this feature is a stop-gap until we have a full user interface for involving multiple fields in the row selection.)

[Y“key field database name”] This qualifier specifies (for on-line databases/tables only) by what name to reference this field’s keydatabase name.

With this qualifier, you can now specify the database/table containing the field, the field’s name in the database/table, and the field’s key name in the database/table, so you can effectively have multiple DDs in a single DD, with each group of fields referencing the same database/table and key a “logical sub-DD”.

Sample Data Descriptor Here’s a sample data descriptor file that might be used with an ODBC database (this is a single physical line with tabs before each field description and a return at the end of the file, suggested by the paragraph mark):

```
→product_id[K D"product_table" &"Test" $"scott" !"tiger"]  
→product_name[Q D"product_table"]  
→product_name2[Q D"product_table"]  
→product_description[T D"product_table" F"description"]  
→product_points[P D"product_table"]  
→product_price[P D"product_table"]  
→product_price2[D"product_table"] ¶
```

In the above example,

- the DSN “Test” (case is potentially significant), user name “scott,” and password “tiger” are supplied as key field modifiers;
- all columns are from a table named “product_table”;
- all column names are exactly their field names except for the “product_description” column, which is named “description” in the product table;
- “product_name” and “product_name2” are two columns that InCatalog couldn’t otherwise tell are character-based columns (so they have an explicit [Q] qualifier).
- “product_description” is automatically character-based by virtue of being a tagged text field, etc.

[M] or [M"separator"] — Multiple-value This qualifier indicates that information extracted to this field (data destination update only) may have more than one value, each of which is preserved. (Rarely used on its own; usually implied by one of the specifiers below.) See the **[R]** qualifier (above) for information about the optional **separator** character specification.

[G] or [G"separator"] — Global This qualifier indicates that this field might only appear in a link once or a few times on a spread, but will be extracted (in a data destination update) for all records on that spread. If there are more than one links with this field in a spread, then the “closest” linked value to the record in question is used.

This can be used for extracting containing section and subsection information for each record, for example.

See the **[R]** qualifier (above) for information about the optional **separator** character specification.

[N] — Numbered Defines a sequence numbered, extraction-only field, which records, for each key in the output, the ordinal (sequence number) of the appearance of the given key during whole-document data extraction (that is, during Update/Create Data... or its equivalent), but only for text links (since text is more likely to be in flow sequence).

Note that two extracted sequence numbers aren't defined relative to each other unless the two keys are in the same story, where a lower number means a link with the the given key (and any field) appears before any link with the second key in the same story.

If there are multiple links for a given key, then the highest sequence number is used (though see below if you want a list of all sequence numbers for a given key).

Note that, if you want a list of all relative output sequence numbers for a given key (e.g., if there are multiple links involving the same

key), you can use the “multiple valued” field attribute (e.g., [N M]) to get the entire list.

The intent of this feature is to allow you to extract a field on which the resulting data snapshot file (or whole table/database for on-line updates) could be sorted, to order the file based on the relative appearance of each keyed item in the printed document, within the limits of the same-story requirement noted above.

You can't link explicitly to this type of field, and it doesn't have much meaning when doing incremental, interactive updates of your data from a document.

[#] or [#"separator"] — Page number This qualifier indicates that this (virtual) field will be extracted with page numbers for all pages on which its corresponding key field appears.

See the **[R]** qualifier (above) for information about the optional **separator** character specification.

Miscellaneous Field Qualifiers

[?"..."] — Default key value Allows the specification of a field's default value in the event of a missing key. For text, the qualifier's value is unquoted tagged text which is inserted in place of the missing data. For pictures, the value is the path of an alternative image to be loaded or a hyphen ("?-") to specify that the picture frame be cleared. Note that missing key errors are logged regardless of this qualifier's setting.

[B] — Transform line breaks Instructs InCatalog to turn all forms of line breaks (paragraph returns, new box characters, new column characters, “hard” returns) into spaces, on input, for the field in question. (Note that this applies to input only, which is the only direction where it makes sense.)

E.g., a DD field declaration description[**tb**] tells InCatalog that the given field description is InDesign Tagged and should have any line breaks turned into spaces on input.

[I] — Independent of context Modifies a **T** or **U** (as in **[TI]** or **[UI]**) tagged text/unquoted tagged text attribute, which directs

InCatalog to, on field extraction, output full character setting tags, independently of context. Note that the I modifier is thus not useful if you're expecting to import the same data again, because the face codes (such as) are toggles, not settings, and thus would alternate faces at each update. This modifier is designed, rather, to aid any data post-processing that needs to know the full character settings for a given field. that a field is “read-only” or “input only,” and will not be modified when updating the data destination from a document

[O] — Read-only field Indicates that a field is “read-only” or “input only,” and will not be modified when updating the data destination from a document.

Obscure or Unlikely Field Qualifiers

[S] — Space-relevant key field By default, InCatalog deletes leading and trailing spaces in its document links' key values, and collapses internal multiple spaces automatically. The **[S]** qualifier indicates that beginning and ending spaces in the associated key field are relevant and should be kept intact. If leading/trailing or multiple spaces are relevant in your database keys, use this qualifier with the K qualifier (e.g., **SKU[KS]**). *We recommend you only use this qualifier in extraordinary cases.*

[A] — Case-sensitive key field By default key values are case-insensitive (e.g., the keys “A100b” and “a100B” are considered identical). If case sensitivity is critical, use this qualifier with the K qualifier (e.g., **[KC]**). *We recommend you only use this qualifier in extraordinary cases.*

◆ *Note: This refers only to InCatalog's ability to locate keys in the data source and doesn't affect the case of data flowing in or out.*

Price Styles

Because InCatalog is often used to link prices to a database, it provides some powerful features for formatting linked prices. (If your projects don't require any special price formatting, you can safely ignore this section.)

InCatalog can format prices in two basic ways. It can either use a default formatting method (when (None) is chosen as a price style in a link's properties, the default case), or follow a named, specific price style that you supply, when you have specific formatting needs beyond InCatalog's default handling.

InCatalog's price style definitions give precise control over how a price appears, even if it changes value significantly. As an example, in US currency, prices below one dollar are often denoted with a cent sign (e.g., 99¢). Prices over one dollar are denoted with a dollar sign and a fraction point (e.g., \$10.99). The fractional amount (or decimal) separator is the period. Sometimes prices which are whole dollar amounts would appear as \$25 without the fractional zeros. And prices over \$999 often use a comma as the thousands separator. Other currencies have similar variability.

(Note that you can specify a default price style for a given field in the data descriptor with the [**P"price style"**] field qualifier, described earlier in this chapter.)

Price Styling with no Price Style

When a frame or selection text link is marked as containing price data but has no related price style (more accurately, has price style "None"), then when InCatalog updates the linked price from a data source, it follows this procedure:

- The raw input data from the data source is scanned for a whole number and a fractional number, separated by a decimal point. The scan ignores other characters (e.g., \$ prefixes, () parenthesis, or anything else).
- If either part is missing, a zero is used in its place; the fractional part is right-padded with zeros to two digits (and truncated to two digits *without rounding*, if there are more than two digits).
- Any thousands-separating commas in the input data triggers thousands separators in the whole part.
- For example, with the rules so far:
 - “(\$10.33)” would result in a whole part of “10” and a fractional part of “33”;
 - “5.” or “5” would result in a whole part of “5” and a fractional part of “00”;
 - “.3” would result in a whole part of “0” and a fractional part of “30”;

- “1,999” would result in a whole part of “1,999” and a fractional part of “00”, etc.
- The decimal point and thousands separator to be used in the formatted price are taken from the system’s “Numbers” control panel (Mac OS) or “Locale” information (Windows), so they adapt to local conventions.
- When InCatalog is updating an existing price in a document, it picks out the first two numeric (whole and fraction) parts it can find in the existing price, separated by either a decimal point or by a change in formatting, and replaces just those parts with the corresponding part obtained from the input (whole or fractional), retaining the formatting of each part. E.g., a linked text selection whose contents are “Sale price \$333”, when updated with the input data “10.00” (or “10” or “10.0”) would end up with contents “Sale price \$10”.
- If InCatalog finds no existing numeric data in the price being updated, it simply inserts the new price at the start of the linked text selection as *whole.fraction*.

The net result of this default price formatting approach is that price field updates generally do what you’d expect, following the “principle of least surprise.” For example, a document price \$9⁹⁹ being updated with a data source value of 10.00 would end up as \$10⁰⁰, 9.99 would end up as 10.00, \$9 would end up as \$10, etc.

Creating a (Price Styles) File in Your DD Set

To define named price styles you create a plain text file named “(Price Styles)” (Mac OS) or “(Price Styles).txt” (Mac OS or Windows)—what we’ll call simply a “price styles” file—and place it in your data descriptor set folder. (The parentheses around the name keep it from being considered one of the data descriptor files in a data descriptor set.)

Using and Modifying a (Price Styles) File

Each time you select a data descriptor from your DD Set folder (either implicitly at application start-up using the remembered data descriptor, or explicitly using InCatalog / Select Data Descriptor...), InCatalog loads (or re-loads) any price styles file. If you add, delete or modify any price styles, select the current data descriptor again to re-load (and re-define) your price styles.

Be careful, though, not to orphan any price styles used in links; only delete a price style or change its name if you’re sure it’s unused in

all documents you care about (otherwise, you'll get an error when updating linked contents using any such orphaned price styles). It's better to add new price styles and leave the old price styles in place until you're sure the latter are no longer used.

Price Style Definitions in a (Price Style) File

Each price style definition consists of a single paragraph (line) with up to eight elements separated by a tab. You can create as many price style definitions in your price styles file as you need, one per paragraph (line), each with a unique name.

These elements are the name, flags, and the InDesign-Tagged text coding for up to six different cases of whole and fractional price element values, viz. (1) both whole and fraction non-zero, (2) non-zero whole but zero fraction, (3) zero whole but non-zero fraction, (4) zero whole and fraction, (5) non-numeric, and (6) entirely empty. In the six coded text element cases, the placeholders **^W**, **^F**, and **^.** can be inserted anywhere, and stand for the whole portion of the price value to be formatted, the fractional portion of the price value, and a locale-appropriate decimal point, respectively.

This table provides a simple (if not particularly useful) US currency example used in the following descriptions of each element.

Name	Flags	1.	2.	3.	4.	5.	6.
Normal	T	^W^.^F	^W	^F¢	\$0		Free!

Price style name. This element is the price style name as it appears in the data linker palette's Price w/style pop-up menu.

The above sample style is named "Normal." Selecting Normal in the InCatalog linker palette's price style will render prices according to the remaining example price style elements.

Price style flags. One or more of **T** (for thousands separator inclusion in the whole part of the price—e.g., a whole price part of 10000 would format as 10,000), **U** (for zero-suppression—e.g., a whole price part of 00030 would format as 00030 instead of 30), or **-** (a dash meaning nothing, as a visible placeholder), and **1**, **2**, **3**, **4** or **5** (for 1 through 5 fractional digits in the final price—any more digits after the given number are simply dropped; the default is 2 digits).

In the example above, the appropriate local thousands separator will appear (**T**) in any prices greater than 999.00.

1. Non-zero dollars and cents. Prices with both non-zero dollars and non-zero cents use this tagged text format.

The example above will place dollar signs in front of whole dollars (**^W**) and fractional dollars (**^F**) separated by the appropriate local fraction separator (**^.**) (e.g., \$10.85).

2. Non-zero dollars and zero cents. Prices with non-zero dollars but with zero cents (i.e., whole dollars) use this tagged text format.

The example above will have dollar signs placed in front of whole dollars (**^W**) and no fractional dollars (\$25 instead of \$25.00).

3. Zero dollars and non-zero cents. Prices with zero dollars but with non-zero cents (i.e., fractions of a dollar) use this tagged text format.

In the example above, non-zero prices less than a dollar will appear as the fractional portion (**^F**) followed by a cent sign (e.g., 0.99 will format as 99¢).

4. Zero dollars and zero cents. Prices with zero dollars and zero cents use this tagged text format.

In the example above, completely zero prices will format as \$0.

5. Non-numeric prices. Price values with no numerics use this tagged text format.

In the example above, prices that have no numerics will format as nothing (an empty price).

6. Empty prices. Price values that are completely empty (no characters at all) use this tagged text format.

In the example above, prices that are completely empty will appear as the literal text **Free!**.

Any empty case (nothing between tabs) in a price style will cause InCatalog to treat the case it as if it were case 3 (non-zero dollars and cents).

Picture Folder Aliases/Shortcuts

When InCatalog is importing graphics through a link on a picture frame and has a relative picture path name, it will search any picture folders contained in the data descriptor set folder (but not in the data descriptor set folder itself).

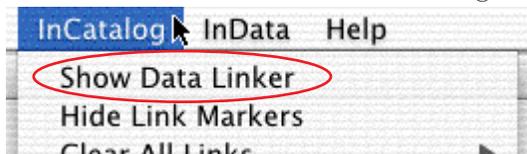
Even more usefully, it will search any picture folders referenced from the data descriptor set folder via aliases (Mac OS) or shortcuts (Windows). Thus, you can store your actual picture folders anywhere on your system or file servers, logically “gathering” all project-related picture folders in one place by pointing to them from your data descriptor set folder.

5 Linking

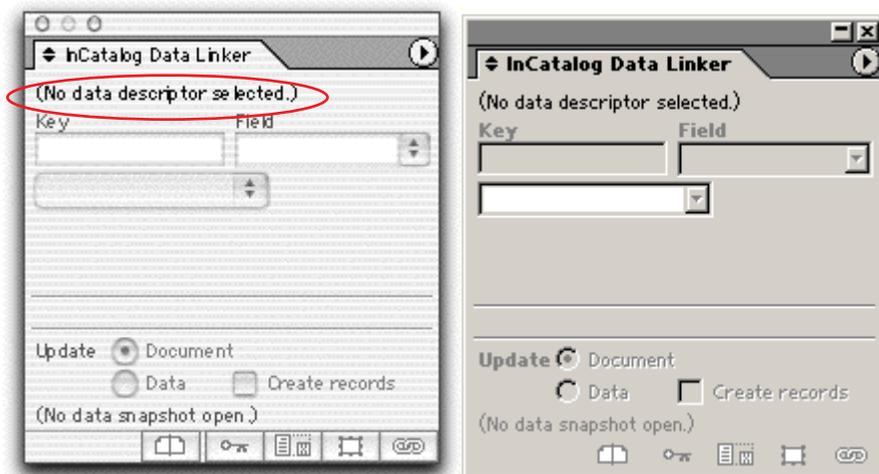
This chapter describes how to link your documents to external data with InCatalog.

Getting Started

Begin by opening the InDesign document to which you want to add links (or modify or delete them later). If you look toward the right of InDesign’s top-level menu, you will see a menu heading for InCatalog. All major InCatalog actions are invoked from this menu. To open InCatalog’s data linker palette (if it’s not already open), select Show Data Linker under the InCatalog menu.



Since no data descriptor has been selected, an empty data linker palette is displayed with “(No data descriptor selected)” in the DD name area.

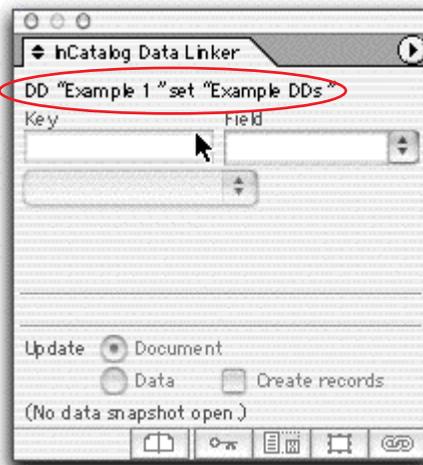


The data linker palette acts like most other InDesign palettes in that its visibility, placement and size can be controlled by the user in the usual ways.

The data linker palette's contents are only active when a document is open, a data descriptor selected, and the type or direct selection tool is active.

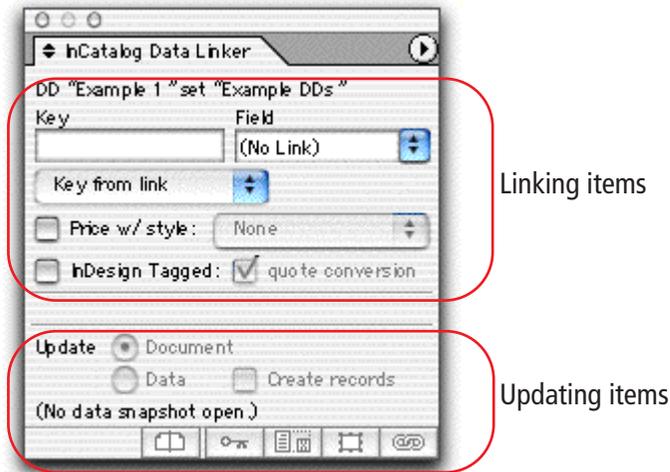
Selecting the Data Descriptor

Before linking can begin, you must select a data descriptor. Choose Select Data Descriptor... from the InCatalog menu, and a file selection dialog will be displayed. Locate and open the data descriptor file. Notice that the data linker changes slightly.



The title area (circled) of the data linker palette displays the active DD file (in this case, "Example1") and further identifies the name of the data descriptor set folder where the DD resides (set "Example DDs").

The rest of the linker palette contents will not become active until a text frame or picture frame is selected on the InDesign page with the type or direct selection tool in force. If the cursor is placed inside a text frame, the data linker palette will then look something like the following:



The linking items in the data linker palette will be discussed in the sections below, and the updating items in the next chapter, “Updating.”

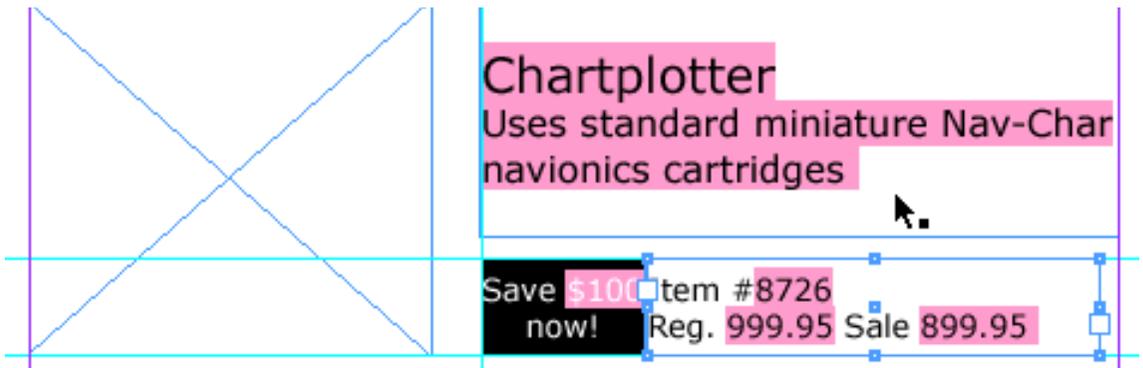
Linking InDesign Document Elements

You can insert links to data into existing InDesign documents or insert them while building new documents.

You can also insert “placeholder” links into library items, through which data can be injected semi-automatically when the library element is placed in a document.

A single logical presentation, or module, is composed of all the various InDesign document elements which are related to the same item or record in the data source or destination. A **module** can consist of a single text frame which lists several fields of information, or it can range over several frames that are grouped together—including combinations of picture and text frames and sub-groups. Links can also be established to anchored text and picture frames and to their contents.

In short, links can be established to any combination of document elements. The example illustrated below contains three text frames and one picture frame. The item number is the key value. That is, the item number is the value which identifies the particular record in the data source containing the information for this presentation.



Notice the highlights behind each linked text element in the example above, called **link markers**. They are non-printing markers which serve as visual reference marks for the InDesign interactive user.

(See chapter “Menu” for information about turning link markers on and off, and chapter “Preferences” for information about changing the color and shade of the link markers.)

Links can be set for any range of text within a frame, for an entire text frame, or for a picture frame.

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

Frame vs. Selection Links

InCatalog supports two kinds of links: whole-frame links and text-selection links.

Whole frame links connect an entire picture’s or text frame’s contents to a **single field**. No link markers are visible on whole-frame links.

However, when a whole-frame linked frame is selected, the linker palette will display information about the link.

- ◆ *When a picture frame is selected (using either the type or direct selection tool), or when a text frame is selected using the direct selection tool, any new link created is a whole-frame link—i.e., the whole frame is linked to a single field.*

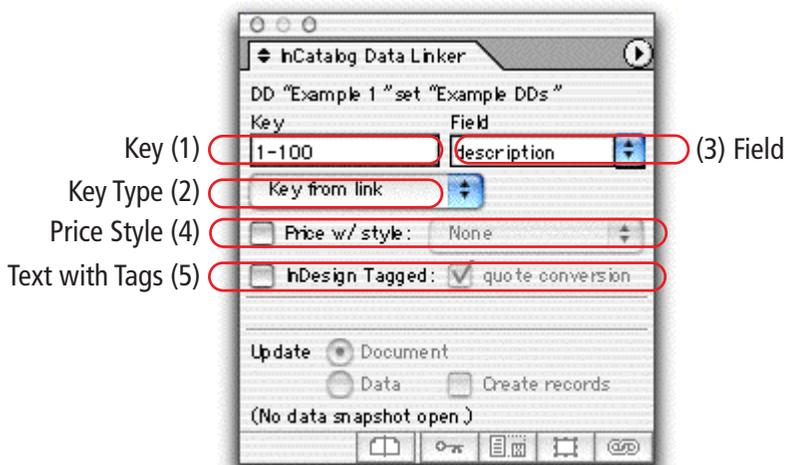
Text-selection links connect a contiguous range of text, within a text frame, to a single field. A single text-selection link can consist of any text within a text frame, from a single character up to and including all of the text in the frame (even a whole story through multiple linked text frames).

- ◆ *When text is selected, any new link is created as a text-selection link. Note that only the selected text is linked, and that there can be many text-selection links inside one text frame.*

Any time you select a text frame with the direct selection tool (indicating a whole-frame type link) and it contains text-selection links, the field pop-up menu will be disabled and you won't be able to add or delete links. Likewise, if you select a text frame that already has a whole-frame type link applied with the type tool, the field pop-up menu will be disabled.

To deal with the existing links, switch to the correct tool. This behavior prevents you from creating conflicting InCatalog links. If you really want to change the type of link, delete the existing link and change tools before re-linking.

Overall, you use the top part of the linker palette to create, edit and delete links; you use the bottom portion to interactively update your document or data.



The following items must be supplied for every link:

- the **key** (#1 in the illustration above), which identifies the record from which the data is to be read or to which copy is to be written;
- the **key type** (#2) for the link;
- the **field** (#3) to which the data should be linked.

(1) Key edit area The top left of the dialog contains the Key edit area. This is where the primary key is entered—that is, the value which identifies the record to which this link is to be made. This is often a part number, SKU, or other unique identifying number or other string.

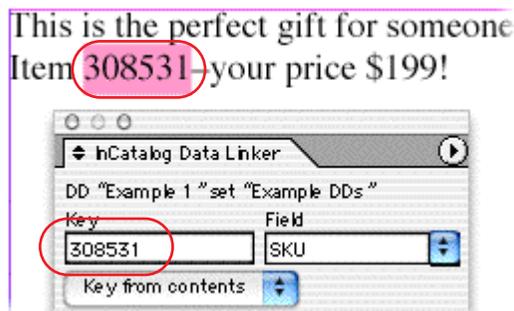
A key is made of a contiguous string of characters, up to 255 in length (63 if you're using a data snapshot). Keys should be made of alphanumeric characters, spaces, hyphens and periods. **Don't use special characters such as “, { } [] * / ; < > ”.**

The key value can be entered manually by placing the cursor into the key editing area and typing the value. More often, however, the value is established automatically as the rest of the link is defined (see below).

The key edit area is “sticky”—it retains the most recently displayed or entered key value, even when no link is being displayed. This stickiness makes it easier to link several elements with the same key, i.e., the key value only needs to be typed once for a group of fields from the same record.

(2) Key type The key type, chosen from the pop-up menu, has the following options.

- **Key from link** This key type tells InCatalog to get the key value from the link itself (the default), and renders the link “hard wired”—i.e., the key value does not change dynamically if the contents of the linked field change.
- **Key from contents** This key type should be used **only for the key field itself** (part number, SKU, etc.). The Key from contents key type dynamically picks up the current contents of the linked text itself as the key value for the link. This method creates accurate links by avoiding the need to actually type the key value into the key edit area of the linker palette (see illustration below).

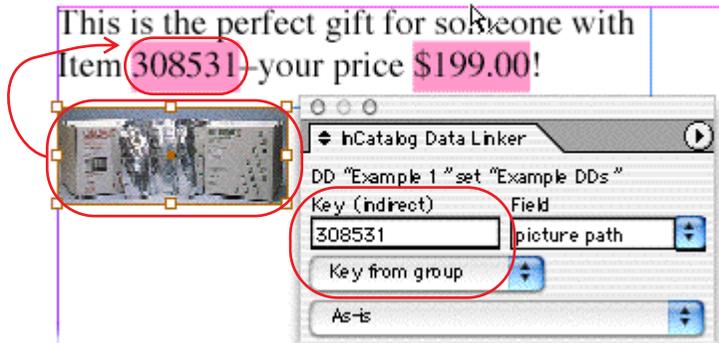


Both Key from link and Key from contents are **direct key types**: *their key values are established directly, either by entering the value into the key area of the linker palette (Key from link) or from the actual contents of the linked field (Key from contents).*

When used in conjunction with the other indirect link types discussed below, the direct key types can save much manual key entry and editing.

- **Key from group** This is one of InCatalog’s dynamic **indirect key types**. Key from group is used with picture frames and for fields where the text is placed in separate text frames, and which

are part of the same presentation or module, as in the sample below.



To use Key from group, first group the various elements of the presentation using the InDesign grouping commands. Then link the key field using the a Key from link or Key from contents key type. Finally, establish the other links within the group using Key from group.

Notice at the top left of the linker palette next to the header Key it says (indirect). This indicates that InCatalog is getting the key value “indirectly” from another (direct) key link in the group (in this case, the item number, which is a Key from contents link).

When InCatalog encounters a linked item with a Key from group link, or any of the other indirect link types, it will look for the primary (direct) key value it finds within the group and automatically use that value for the original item’s key. If the primary (direct) key value is changed for a group, then all the fields within the group that use Key from group are effectively changed automatically and immediately.

- Key from text « and Key from text ». A Key from text « key type instructs InCatalog to **look to the left** of a text selection link to find the nearest direct key-type link—a link whose key type is Key from link or Key from contents, or, that failing, a link somewhere else in the group whose key type is direct—and applies the value of the found direct key to the field being linked. Likewise, with Key from text », InCatalog **looks to the right** for the nearest direct key-type link and uses the key value from that link. Key from text « and Key from text » are also **indirect key types** whose value will change dynamically when the related direct link is altered.

These link types are ideal for items which are contained in a single text frame or table row. Key from text style links are also great for table-style tabbed lines of text with multiple fields which include the primary key field somewhere on the line. Even better, Key from text style links are great for links in real table cells, because InCatalog will search just the current row for a direct link, and then keep looking before or after the table from that point.



Just like the Key from group key type, if a key value is changed for a direct key field, then all the Key from text « and Key from text » links which reference the altered key value will be automatically and immediately effectively changed to use the new key value

(3) Field edit area The data field for the link is chosen here (or (No link) is chosen for link deletion), and can be selected in one of three ways:

1. Select the desired field from the menu, accessed by clicking and holding the arrow button to the left or right of the box. This menu lists all the available fields.
2. Use pre-defined hot keys (see chapter “Preferences”) to select the first 8 or 9 fields directly, using a keyboard shortcut.
3. Highlight the current information in the field editing area and begin typing the name of the field. The complete field name will be supplied by InCatalog as soon as enough characters have been entered to uniquely identify the field name you want and you can stop typing at that point.

(4) Price w/ style If you want the linked text element treated as a price, then you’d check this box.

If you select “None” for a price style, then you’re selecting InCatalog’s default price handling.

If your project includes a “price styles” file in its data descriptor set to automate the formatting of prices, you can select one of those those price styles here to get the special price formatting of that particular style (or leave “None” selected for default price style handling).

(See chapter “Data Descriptors,” section “Price Styles” for details.)

(5) InDesign Tagged Text which is imported through InCatalog links normally and automatically take on the style attributes of the first character (for character properties) and first paragraph (for paragraph properties) of the linked text. In order to automatically change InDesign sub-styling within the text block of a link as it is imported, InDesign Tags must be embedded in the incoming field value. Any fields which may have InDesign Tags embedded in them also need to have the InDesign Tagged interpretation **enabled**. If InDesign Tagged is not enabled, the tag (which is markup coding) will be read as regular text—in other words, it will introduce unwanted coding into your document.

Note that the key type, Price w/style and InDesign Tagged (along with the quote conversion) settings are automatically initialized from the relevant field’s default properties (as set in the data descriptor file) when a link is added or when a new field is selected for an existing link.

Adding Links

To establish a new link:

1. For a text-selection link, highlight the target text; for a whole-frame link, select the text or picture frame using the direct selection tool.
2. In the InCatalog linker palette, click in the key editing area and type a key value (a dummy value, in the Key from contents link type case, e.g., *).
3. Move to the field editing area and type the first few characters of the field name (or select a field from the adjacent pop-up menu). (Note that in the case of a Key from contents key type, it only makes sense to select the key field.) If the field you’re entering is a repeating field, enter a *#subfield* suffix to the field

name (InCatalog will insist on it), where *subfield* is the number of the subfield to which you want to link (#1, #2, etc.).

4. Select a key type from the pop-up menu beneath the key editing area if it's not what you need.
5. If you're linking a price, and it's not automatically set up, select Price w/style and, optionally, a price style from the pop-up menu.
6. If you're linking tagged text, and it's not automatically set up, select InDesign Tagged and whether the field needs quote conversion.
7. Exit the linker palette by typing a Return or Enter key, or by clicking outside the InCatalog linker palette.

◆ *Note that there are various shortcuts to adding links, discussed in chapter "Preferences."*

When you've entered a key value, but haven't selected or entered a field, before trying to leave the linker palette, InCatalog will sound an alert and select the field edit area contents for you to replace.

When you've selected a field in the linker palette by any method, and you haven't chosen a key value, before trying to leave the linker palette, InCatalog will sound an alert and select the key edit area contents for you to replace.

If you start to edit a key value or a field name and change your mind, you can use the Escape key (and Command-period under Mac OS) to leave the InCatalog linker palette without making any changes.

Modifying/ Changing Links

To modify an existing link:

1. For a text-selection link, select any part of the already-linked text; for a whole-frame link, select the text or picture frame using the direct selection tool.
2. Make any desired changes in the linker palette—key value, associated field, key type or any other link property;
3. If you've edited the key or field value (or both), exit the linker palette by typing a Return or Enter key, or clicking outside the InCatalog linker palette.

If you start to edit a key value or a field name and change your mind, you can use the Escape key (and Command-period under Mac OS) to leave the InCatalog linker palette without making any changes.

Deleting Links To break an existing link, select the linked item, and choose (No Link) in the field drop-down menu.

◆ *Note that for text selection links, you can have more than one link selected, but only the first link in the selection will be deleted. You can keep selecting No Link until all of your links are deleted, or you can choose the InCatalog / Clear All Links / In Selection menu item.*

Exiting the Linker Palette Once a link definition has been completed, just move the cursor to any place outside the linker palette and click; if you're typing in the palette, type a Return or Enter to leave it.

To exit the linker palette without completing a link, use the Escape key (or Command-period under Mac OS).

Indirect Key Feature (Root + prefix/suffix)

InCatalog has a root+prefix/suffix key feature which facilitates setting links for groups with several related keys in them. For example, if you have a number of items which all belong to a main part category (e.g., "100") but the individual items all have a suffix added to the end to distinguish color of the product, you can use the root +prefix/suffix method as a shortcut. This feature changes all of the key roots in the group dynamically when the main root number is changed.

Roots in a key are enclosing using curly braces { } which are ignored for the purposes of actual data lookup. Any part of the key before the open brace is called the key prefix, any part of the key inside the braces is called the "root," and any part of the key after the close brace is called the key suffix.

Let's explore a list of parts in a group.

100R	part# 100 in red
100Y	part# 100 in yellow
100K	part# 100 in black

Using the root + key method the first part could be linked with a **key from link** key type with key value “{100}R”, the second with an **indirect key type** and key value of “{*}Y”, and the third with an **indirect key type** and key value of “{*}K”. When you view the second or third keys with the InCatalog linker palette, you'll see “{100}Y” and “{100}K” as the key values, because everything in their root is replaced with the direct key's root.

If you changed the first part's direct key to “{200}R”, then the second and third keys would automatically change to “{200}Y” and “{200}K”, respectively.

All these keys would be looked up in the data source or destination without the curly braces, e.g., as “100R”, “100Y” and “100K” in the first example.

Linking Graphics

InCatalog can import into and export from picture frames, with the path information being the linked value. All links to picture boxes are whole-frame links; therefore only one link per graphic is possible.

It is important to understand that InCatalog imports graphics by reading the data source, which contains the file name or path name of the graphic to be imported. The graphic itself must live somewhere in the accessible file system: **InCatalog does not import the actual graphic contents from the data source.** Similarly, when exporting graphics, InCatalog exports the file or path name to the (external) graphic, not the graphic itself.

To link a picture frame, first select it and then choose the field which contains the name (simple filename, or partial or whole path name) of the picture to be imported. Then choose the key type you want. If the picture file “base name” (no path information and no file type) is to be the base key value for a group, use Key from contents. If the picture frame is grouped with the other elements of

its key, you can use a Key from group key type. If the picture frame is anchored, you can also use an indirect key type such as Key from text « or Key from text ». Otherwise Key from link is the appropriate link type.

InCatalog can place (position and scale) graphics in several different ways. To choose the placement handling you want, select the picture frame to be linked and then select the appropriate option from the pop-up menu located below the link style menu:

As-is InCatalog will import the linked graphic using all of the position and scaling attributes currently in force for the picture frame, including rotation, skew and flipping.

Upper Left This selection will position the newly updated graphic in the upper left of the picture frame at 100% scale. If the picture frame is smaller than the graphic then the graphic will be clipped on the right and bottom.

Centered The new graphic will be centered in the picture frame at 100% scale. If the graphic is larger than the picture frame, then the graphic will be clipped equally on the right and left and equally on the top and bottom.

Fit to frame The newly imported graphic will be scaled to fit the picture frame. InCatalog sets the x scale and y scale independently to exactly fill the frame. **This usually results in distorted graphics.**

Fit to frame maintaining aspect ratio **Probably the best option when automatically scaled graphics are desired.** This selection will cause InCatalog to choose a uniform x and y scale. If the picture frame is proportionate to the graphic, an exact fill will result. If the graphic is disproportionate to the picture frame, then InCatalog will use a percentage that will just fit the proportionately larger dimension and cause the proportionately smaller dimension to fit with white space “left over.”

Fill frame maintaining aspect ratio This option maintains the aspect ratio of the picture. It determines the scale in each dimension that would just fit the frame, then uses the larger of the two (rather than Fit to frame..., which uses the smaller of the two) for both dimensions, then centers the picture in the frame. Chances

are good that some part of the graphic will be clipped, but there won't be any white space "left over."

6

Updating

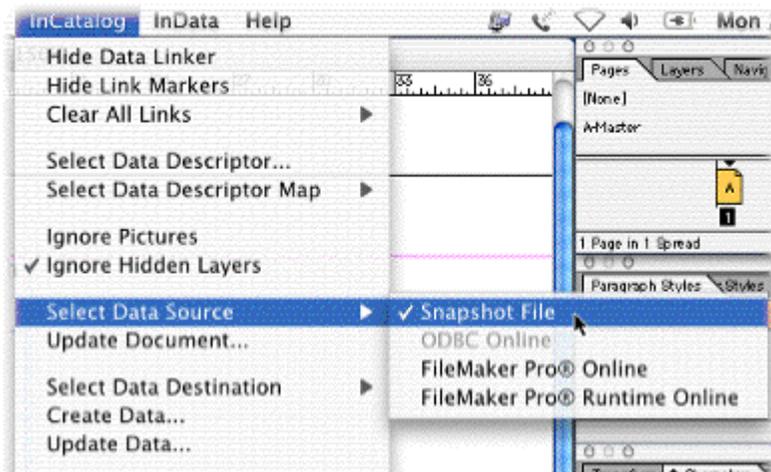
Updating Documents

Once some or all of your links are in place, document updating is possible. When updating linked document elements, you must first select the proper data source. See chapter “Data” for more details on dealing with various data sources and destinations.

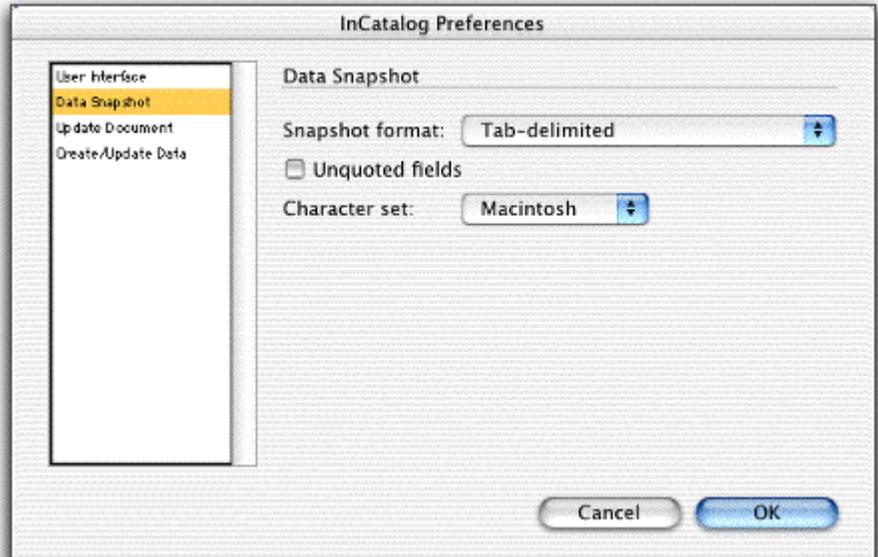
Select a Data Source

From the InCatalog / Select Data Source sub-menu, choose the appropriate data source:

1. A data snapshot file (Snapshot File);
2. FileMaker Pro online (FileMaker Pro® Online) [Mac OS only];
3. A FileMaker Pro Runtime application (FileMaker Pro Runtime Online) [Mac OS only];
4. An online ODBC connection (ODBC Online) [Pro version only].



Using a Data Snapshot If the data source is a delimited text snapshot file, you set the appropriate preferences in the InCatalog Preferences (InCatalog / Edit Preferences / Data Snapshot...).

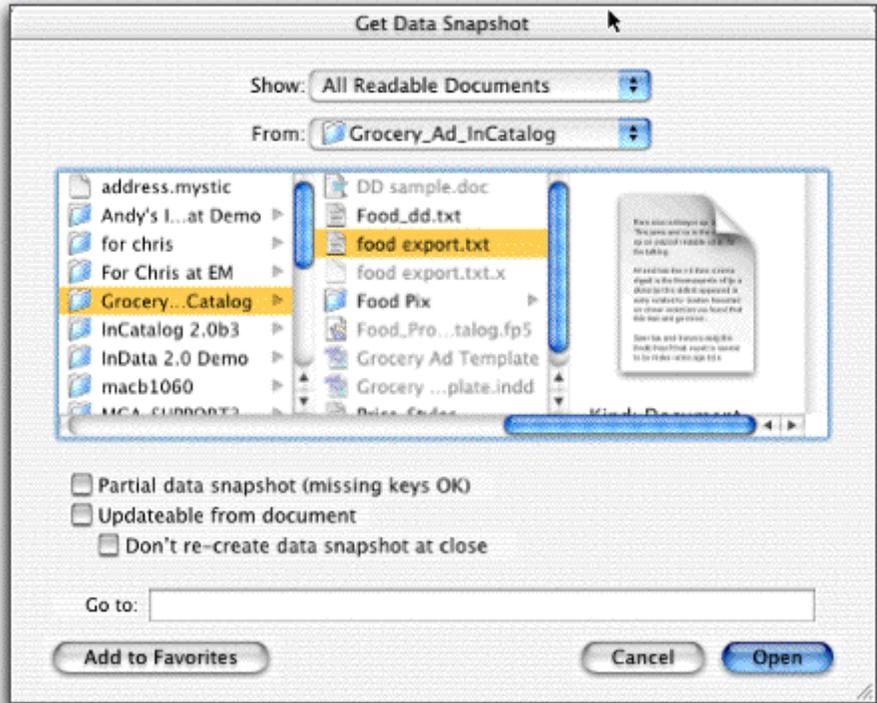


Pick the Snapshot format that matches your data's format:

- Comma delimited
- Tab delimited
- Tab delimited (FoxBase Pro)

You can also select a Character set for input and output: Macintosh, Windows, or Unicode (though you'll rarely need to use this).

Once you've selected the proper data format and character set, select Open Data File... in the InCatalog menu. A dialog like the following will appear:



Navigate to the desired file to open it. If the InDesign document contains key values not present in the data snapshot, and you want InCatalog to ignore those links, be sure to select Partial data snapshot (missing keys OK) when opening the data snapshot file. Otherwise, InCatalog will consider missing records an error and log each such missing key (though no other harm is done).

Select Updateable from document if you want the data snapshot to be updateable from the InDesign document as well (see the section “Create/Update Data” later).

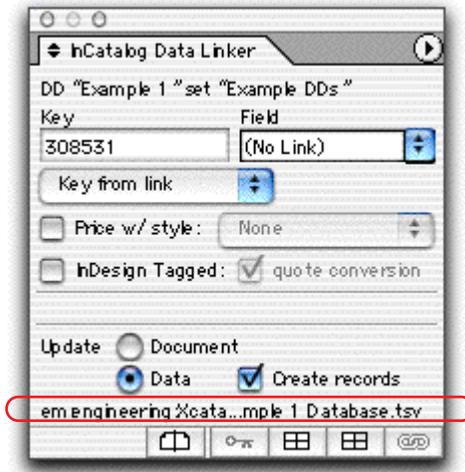
Using FileMaker Pro Online (Mac OS) If the data source is either flavor of FileMaker Pro Online (Mac OS only), simply open the FileMaker database that contains the desired data, making sure all the required data is in the current found set (e.g., perform a “Find All” function on the open database to make all records available). Also, make sure the active layout includes all the fields being used by the

current data descriptor (or at least those fields used in the current document).

Using ODBC Online (Pro version only) If the data source is ODBC Online, and if you want to do interactive updating, make sure the ODBC connection is open (using InCatalog / Open ODBC Connection...) before proceeding. For non-interactive (batch) updating, the connection will be opened automatically, if it's not already open, before updating, and closed automatically afterwards.

Linking Palette Update Status

The bottom portion of the linker palette is the area where interactive updating occurs. The middle line of this portion tells you the type of data source (for document updates) or destination (for data updates), and its current status.



When using a snapshot file, No data snapshot open is displayed until you open a snapshot file. When a data snapshot file is open, the name and path of the data source file is displayed (as above).

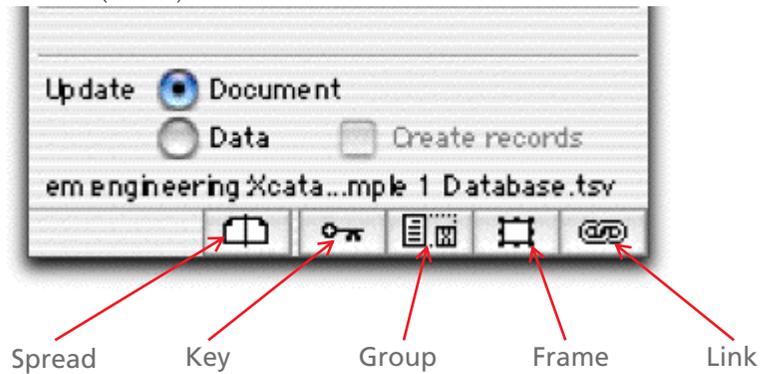
When FileMaker Online (either flavor) is the data source or destination, the Online with FileMaker message is displayed whether or not any FileMaker databases are open (it's up to you to make sure the right databases are open).

When ODBC Online is the data source or destination, Offline with ODBC is displayed when no connection is open, and Online with ODBC DSN "*dsn*" is displayed when a connection is open using the DSN named *dsn*.

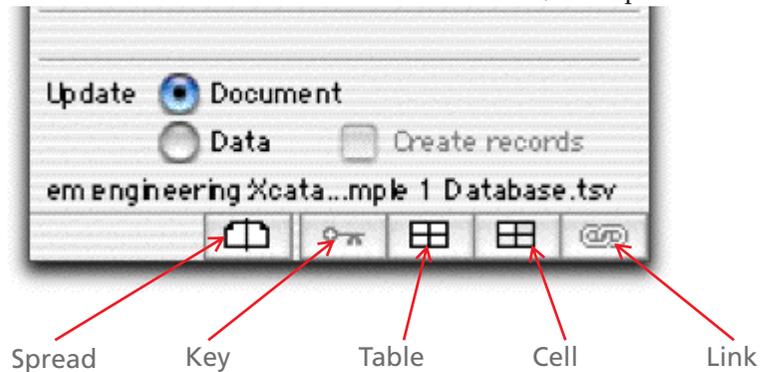
Interactive (Partial) Document Updates

When the correct data source type is selected, and the data source is active, you can interactively update portions of the document.

When anything but a table cell or its contents is selected, the update button (icons) are:



When a table cell or its contents is selected, the update buttons are:



- ◆ (You have to look closely to see that the Cell button icon has all but one cell shaded.)
- ◆ These buttons have tooltips associated with them: if you place the mouse over one for a couple of seconds, you'll see an explanation of the button's function.

To update the document interactively, make sure the (Update) Document radio button is activated in the interactive update portion of the InCatalog linker palette. Then, at any time, press the appropriate button at the bottom of the linking palette to perform the kind of update you need. Update scopes include Spread, Key, Group, Frame, Table, Cell, and Link. Depending on your current selection, they may not all be available (enabled and present).

The buttons are:

Spread Updates all linked elements on the current spread. (Note: the current spread is a tricky concept, because you can see multiple spreads in the document window at any point—look at the pages palette to determine the currently-selected spread.) If the document only has one page per spread, then only the current page will be updated.

Key Updates only linked document elements having the same key value as the currently selected link—i.e., all links that have the same key value as the key value currently showing in the linker palette.

Group If the current selection (picture or text) belongs to a group, this button will be enabled. Pressing this button will update all linked elements in the current group, no matter how complex it might be.

Frame Updates all the contents of the currently selected frame (all linked elements on or in the current frame).

Table Updates all the contents of the currently-selected table.

Cell Updates all the contents of the currently-selected table cell.

Link Only updates the contents of the currently-selected linked element.

Non-Interactive (Batch) Document Updates

To update the current document in its entirety:

1. Make sure the document you want to update is the current, active InDesign document (i.e., open it if necessary, and bring it to the front, if it is not currently the front-most document).
2. Check that the correct data descriptor is in force; if not, select it with **Select Data Descriptor...** from the InCatalog menu. (If this doesn't activate the linking palette, then hold down the Option (Mac OS) or Alt (Windows) key when choosing **Select Data Descriptor...** from the InCatalog menu; this forces InCatalog to update its idea of the document's DD set.)
3. Select **Update Document...** from the InCatalog menu.

4. If no data source is open, InCatalog will prompt you to open a file in the data snapshot case, or to provide a user name and password in the ODBC connection case (if the name or password are “?” in your data descriptor file).
5. InCatalog will then update all the document’s linked elements using the currently-selected data descriptor (links made with other data descriptors will be ignored). If you’re using a snapshot file, and if this is the first time InCatalog has used this particular database snapshot file (or if you’ve moved it or changed it since it was last used), InCatalog may take longer than normal to first build its indexed version of the snapshot (the “.x”-suffix file, always located in the same folder as the snapshot file).
6. If no data source was open before the update (and was opened in step 4 above), then InCatalog will close the current data source (file in the snapshot case, nothing in the FileMaker case, ODBC connection otherwise).
7. If InCatalog encounters any problems during the update, it will ask you whether you want to see a full report or not. (See section “Error Logging,” below.)

Create/Update Data

One of the most powerful features of InCatalog is its ability to send information from the InDesign document to the current data destination, whether a snapshot file, FileMaker database (Mac OS only) or ODBC-accessible database. InDesign can fully populate an empty data set or database, or simply update an existing data set or database.

(Note that InCatalog can’t tell which data has changed; it will export the contents of all links in the whole document with batch extracting, or only in those portions of the document you export with interactive extracting.)

With data snapshot files, the exported information is stored in the index file (the “.x”-suffix file) until the data snapshot is closed. At that time, as long as you didn’t select Don’t update snapshot at close when you opened the file, InCatalog will update the data snapshot file to reflect all exported information. (You might want to wait

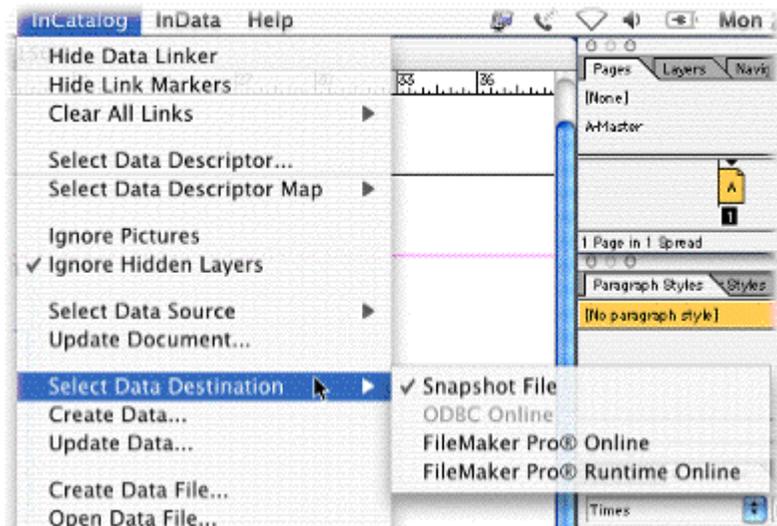
until a set of extractions is done before finally updating the actual snapshot file, if it's very large.)

Select a Data Destination

From the InCatalog / Select Data Destination sub-menu, choose a data destination:

1. A data snapshot file (Snapshot File);
2. FileMaker Pro online (FileMaker Pro® Online [Mac OS only]);
3. A FileMaker Pro Runtime application (FileMaker Pro Runtime Online [Mac OS only]);
4. An online ODBC connection (ODBC Online) [Pro version only].

(See sections “Using a Data Snapshot,” “Using a FileMaker Pro Online,” and “Using ODBC Online,” above, for details.)

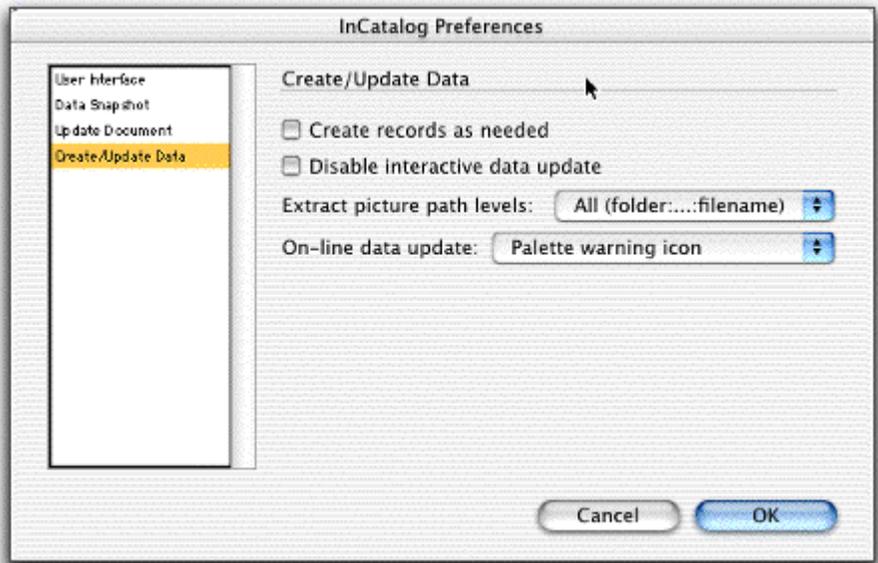


Data Update Preferences

There are several settings that control InCatalog's actions when updating data (i.e., extracting to the data destination).

InCatalog also gives you several preferences to control its actions during data updating and creating. Choosing InCatalog / Edit Pref-

erences / Create/Update Data... gives you the following dialog, from which you control automatic record creation, etc.:

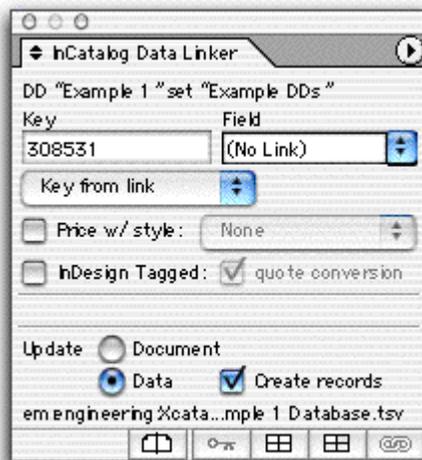


See chapter “Preferences” for more information on these settings.

Interactive (Partial) Data Updates

Interactively updating data from a InDesign document is parallel to interactively updating a document from data, as outlined above.

When the data destination type is selected and the chosen data destination file is open (with Updateable from document selected when opening the current data snapshot file), data updating can begin.



Make sure the Update Data radio button is activated in the interactive update portion of the linker palette (as above). If you wish to create new records in the data source file, then also check the Create records option. Then, at any time, press the appropriate button at the bottom of the palette to perform the type of update you need.

(See section “Interactive (Partial) Document Updates,” earlier, but with all updates going to the data destination instead of the document.)

Non-Interactive (Batch) Data Update or Creation

To update the data destination (or create the destination data set, if it’s currently empty) with the contents of every linked element in the document:

1. Make sure the document you want to use as the source is the current, active InDesign document (i.e., open it if necessary, and bring it to the front, if it is not currently the front-most document).
2. Check that the correct data descriptor is in force; if not, select it with Select Data Descriptor... from the InCatalog menu. (If this doesn’t activate the linking palette, then hold down the Option (Mac OS) or Alt (Windows) key when choosing Select Data Descriptor... from the InCatalog menu; this forces InCatalog to update its idea of the document’s DD set.)
3. Select Update Data... from the InCatalog menu (or Create Data... if you want to create a data snapshot file from scratch; for any other data destination, Create and Update are identical).
4. If no data destination is open, InCatalog will prompt you to open a file in the data snapshot Update Data... case (or to create one in the Create Data... case—be sure to select the file with Updateable from document selected), or to provide a user name and password in the ODBC connection case (if the name or password are “?” in your data descriptor file).
5. InCatalog will then extract all the document’s linked elements using the currently-selected data descriptor (links using other data descriptors will be ignored). If you’re using a snapshot file, and if this is the first time InCatalog has used this particular database snapshot file (or if you’ve moved it or changed it since it was last used), InCatalog may take longer than normal

to first build its indexed version of the snapshot (the “.x”-suffix file, always located in the same folder as the snapshot file).

6. If no data destination was open before the update (and was opened in step 4 above), then InCatalog will close the current data destination (file in the snapshot case, nothing in the FileMaker case, ODBC connection otherwise).
7. If InCatalog encounters any problems during the data update, it will ask you whether you want to see a full report or not. (See section “Error Logging,” below.)

When sending information directly to a FileMaker or ODBC database online, the information will be updated directly and you will be able to view the changes immediately in the database.

Picture Update and Extraction

How InCatalog Locates Pictures

During a document update, when looking for a picture by relative name (anything other than an absolute path name), the folder containing the document being updated is searched. Then any folders (or aliases to folders) in the data descriptor set folder will be searched for the given relative picture file name or path name.

The search order is:

- If an absolute path name is given, it is used without further search;
- the folder containing the document being updated is searched;
- any folders directly contained by, or aliased from, the data descriptor set folder are searched.

There is also a “fuzzy” picture lookup facility. When the “fuzzy” picture lookup preference is turned on (see chapter “Preferences” for details) and a simple file name is present (i.e. it does not contain an absolute or partial path), then InCatalog will look for and use the first file name fully containing the given picture file name as a sub-string (ignoring any character case differences).

This means, for example, that picture files could be named with the related product code or SKU and with any prefix or suffix you might need to track the picture file’s processing status (e.g., “1-101cc” for

“color corrected” will match the SKU “1-101”), or with any suffix denoting the picture type (e.g., “.eps”, “.jpg”, etc.).

Specifically, when searching folders other than the document folder, InCatalog looks for a given file name and then, if not found and the Do fuzzy picture lookup preference is selected, the picture file name is looked up in the “fuzzy” fashion described above.

The “fuzzy” picture lookup chooses the *shortest possible* file name in the first folder in which a match is found. While this facility can never be foolproof, in the sense that you can devise file naming schemes that defy handling, this refinement does distinguish and choose correctly between file names such as “111cc” and “1111cc”, given a SKU of “111”.

How InCatalog Extracts Pictures

When InCatalog extracts (updates the destination data) a linked picture element, it simply outputs all or part of the path name for the linked picture file. (So an embedded, unlinked picture will not be extracted.)

You can control how much of the path name for the picture file is extracted with the Edit Preferences / Create/Update Data... dialog’s Extract Picture Levels: pop-up menu. You can extract the whole path name or from one to five levels of the file system folder hierarchy above the final filename. (Note that one level simply extracts the picture file name without any path information at all.)

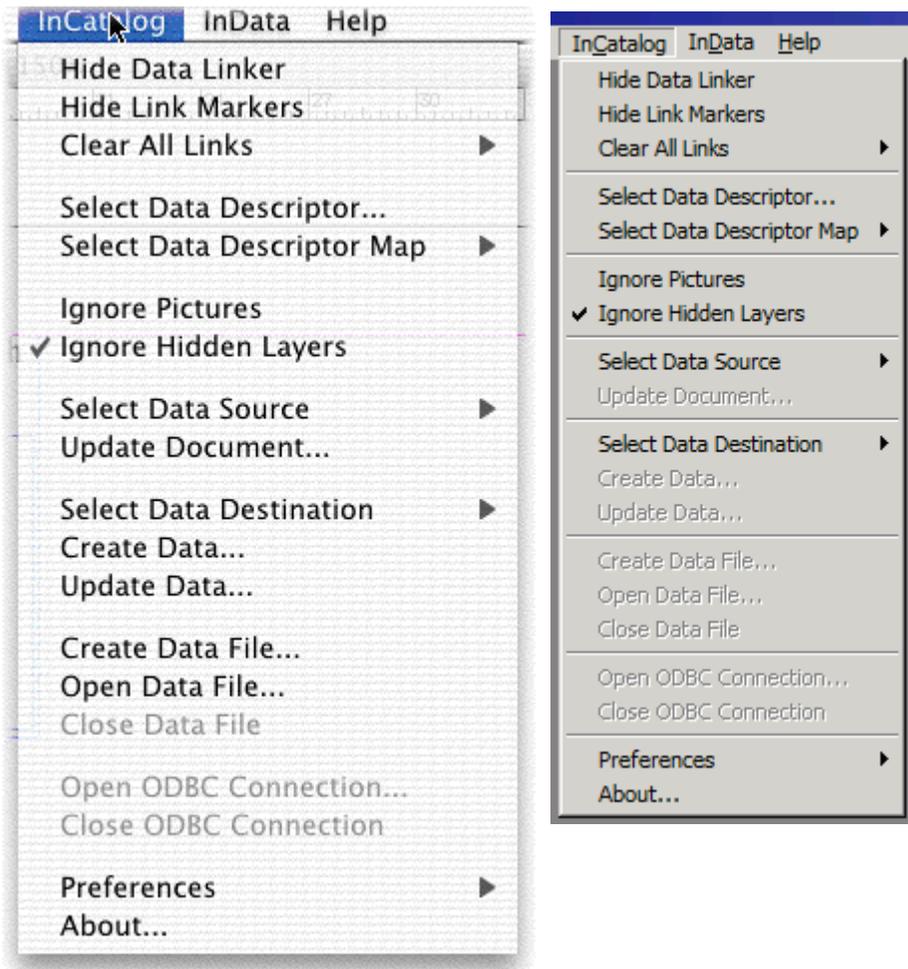
Error Logging

If InCatalog encounters any problems during any of its processing (batch or interactive updating in either direction, selecting a new data descriptor file, etc.), after finishing, it will prompt you by showing the first, most serious error or warning, and asking if you’d like to see the entire error log.

If you choose View Log... at this prompt, InCatalog generates a new, temporary InDesign document detailing all errors and warnings encountered during the session. You can review the errors and warnings, and either save or discard this log document when you’re finished with it.

(If only one page shows up in the new error log document, but you can see that there's more information, then you probably don't have InFlow installed.)

This chapter provides a detailed explanation for each of the menu items in the InCatalog menu and sub-menus. This menu appears toward the right of the InDesign top-level menus. You should see something like this:



Show (Hide) Data Linker

shows or hides the InCatalog data linker palette. You can also hide the palette by clicking in its close box. InCatalog remembers the state of its palette (open or closed, as well as its position) over InDesign sessions.

Show (Hide) Link Markers

turns the display of link markers on and off. Link markers are the colored highlights behind text-selection links. InCatalog remembers this setting between InDesign sessions.

Clear All Links

has one sub-menu, In Selection, which is enabled when text is selected, and which does exactly what it says: it clears all text links touched by the current text selection.

Select Data Descriptor...

is used to set or change the data descriptor (and, implicitly, the data descriptor set). Data descriptor you select will be active for all documents until another data descriptor (and thus its set) is selected. A standard file selection dialog appears when this menu item is selected; navigate in the usual manner to find and select the desired data descriptor file.

InCatalog remembers the selected data descriptor from session to session, unless the data descriptor file is moved. Even then, InCatalog uses an alias (Mac OS) or shortcut (Windows) to track the file. So it may well be able to find it without intervention; if it can't, reselect the data descriptor using Select Data Descriptor....

If you rename the data descriptor set folder or move the data descriptor file to a new folder, InCatalog will think there is a data descriptor set mismatch with any documents that were linked with the old data descriptor set, and will not work with the new set. To force InCatalog to reset its idea of the document's data descriptor set in this case, open the document and hold down the Option key (Mac OS) or Alt key (Windows) key while choosing Select Data Descriptor... for the new data descriptor file.

If a snapshot file is current open, it will be automatically closed (since the new data descriptor may not work with the current snapshot's structure).

Select Data Descriptor Map

is used to select the Data Descriptor "map" which is to be used for subsequent data updates, using a sub-menu selection for the actual

map (by name). A data descriptor map allows you to choose which data fields are used to update the links in your document.

DD maps are often used for versioning (e.g., multiple pricing in “zones,” as in the example, below).

A DD map is specified by a line in the data descriptor that names a set of relationships between the physical (actual) fields of the data source and the logical fields of the data descriptor. The first field of the map contains the map name (which is why the first field of the first line of the data descriptor is empty: as the “default” map, it does not have a name).

Here is an example of a data descriptor with three DD maps. In this example, there are 11 fields in the data: SKU, retailPrice, salePrice, percentSaved, retailPrice2, salePrice2, percentSaved2, retailPrice3, salePrice3, percentSaved3, and description. In the document, prices on the page are linked to the retailPrice, salePrice, and percentSaved fields. The data descriptor would be (`{tab}` and `{return}` would be actual tabs and paragraph returns in the actual data descriptor):

```
{tab}SKU{tab}retailPrice{tab}salePrice{tab}percentSaved{tab}
retailPrice2{tab}salePrice2{tab}percentSaved2{tab}retailPrice3{tab}
salePrice3{tab}percentSaved3{tab}description{return}
zone1{tab}1{tab}2{tab}3{tab}4{tab}11{return}
zone2{tab}1{tab}5{tab}6{tab}7{tab}11{return}
zone3{tab}1{tab}8{tab}9{tab}10{tab}11{return}
```

Notice that the first field position of the first line of the data descriptor is empty, as this is the position reserved for the name of the map. Notice that the first fields of the map lines contain the map names, which are selectable from the item in the InCatalog menu.

(for ease of reading, here is the DD represented as a table):

	SKU[k]	retail price[p]	sale price[p]	% saved[p]	description[t]
zone1	1	2	3	4	11
zone2	1	5	6	7	11
zone3	1	8	9	10	11

The net effect of these three sample DD maps (zone1, zone2, zone3) is that the retail price is one of three physical fields (2, 5, and 8), the sale price ditto (3, 6, and 9) and the percent saved as well (4, 7, and 10), and the choice of DD map controls which physical field is used for each of these three logical fields.

In other words, when a document is updated using the zone1 map, links to the retail price field will be updated with data from the second field in the data file, links to sale price updated with the third, and links to percent saved with the fourth. When the document is updated using the zone2 map, retail price links are updated with data from the 5th field, sale price links from the 6th, and percent saved links from the 7th. Using the zone3 map, the document is updated using fields 8, 9, and 10, respectively.

Note: The [D"..."] and [F"..."] field qualifiers may be used in a data descriptor map to further specify the field/column and database/table when linking to a FileMaker or ODBC data source (see the chapter "Data Descriptor").

Ignore Pictures

tells InCatalog, when checked, to completely ignore all picture frame contents when updating in either direction. It's off by default (i.e., pictures are processed by default.)

However, when a picture is selected and you're using interactive update from the linking palette, the current picture will be processed (updated in the appropriate direction), regardless of this setting.

Ignore Hidden Layers

tells InCatalog, when checked, to completely ignore all frames on currently-hidden layers when updating in either direction. It's on by default (i.e., hidden layers are ignored by default).

Select Data Source

selects the type of data source from a sub-menu:

- Snapshot File (a data snapshot file);
- FileMaker Pro Online (a FileMaker data snapshot, Mac OS only);
- FileMaker Pro Runtime Online (a FileMaker runtime application, Mac OS only);
- ODBC Online (an ODBC-accessible database, Pro version only).

◆ *Note that this setting merely specifies the data source type, but does not actually open anything or do anything else. Also note that you can have different data source and destination types chosen at any one time.*

Update Document...

updates, using the current data source, the contents of all linked elements in the current document that were linked using the current data descriptor. InCatalog will prompt you appropriately if the current data source isn't open, and automatically close any prompted-for data source, when finished.

If any errors or warnings are generated during this process, you will be given the option of viewing the entire log when the update is finished.

Select Data Destination

selects the type of data destination from a sub-menu:

- Snapshot File (a data snapshot file);
- FileMaker Pro Online (a FileMaker data snapshot, Mac OS only);
- FileMaker Pro Runtime Online (a FileMaker runtime application, Mac OS only);
- ODBC Online (an ODBC-accessible database, Pro version only).

◆ *Note that this setting merely specifies the data destination type, but does not actually open anything or do anything else. Also note that you can have different data source and destination types chosen at any one time.*

Create Data...

is exactly like Update Data... (see below), except that if the data destination type is Snapshot File, and none is open, it will prompt you for—and create—the new file, before updating.

Update Data...

updates the current data destination with the contents of all linked data elements in the current document that were linked using the current data descriptor. InCatalog will prompt you appropriately if the current data destination isn't open, and automatically close any prompted-for data destination, when finished.

If any errors or warnings are generated during this process, you will be given the option of viewing the entire log when the update is finished.

Create Data File... creates and opens a new empty data snapshot (delimited text) file, leaving it Updateable from document. The new file can be used for both batch and interactive update in either direction, though initially it'll be empty and use not useful for document update.

(Note that it will only be used in either direction if the selected data type for that direction is Snapshot File.)

Open Data File... opens a data snapshot file for subsequent use in updating either the document or the data itself. You will be prompted to select a snapshot file (which must be an ASCII file).

This menu item will be checked as long as a data snapshot file is open.

There are several options available as check boxes at the bottom of the dialog:

- **Partial data snapshot (missing keys OK)** Refers to InCatalog's behavior when a key value is not present in the data snapshot. InCatalog will normally not update the link (and issue an error to the log), if a link's key value is not found in the data source, when updating a document. To override this default select the Partial data snapshot option. With this option enabled, InCatalog will simply ignore any links using key values which are not found in the data source.
- **Updateable from document** Enables InCatalog's data extraction to the snapshot file being opened. Use this if you plan to update use this file as a data destination.
- **Don't recreate data snapshot at close** During the data extraction process to a snapshot file, InCatalog actually outputs all data to the data destination's associated index (".x") file. When the data destination snapshot file is closed, InCatalog then recreates the snapshot file from the index file's contents. For large amounts of extracted data, this procedure can take some time. If you will be performing multiple data update sessions on the same file, and don't require an updated snapshot file each time, you may select this option. Great caution should be exercised when using this option, however; if the snapshot is not recreated after the final session then newly-extracted data could be lost. We recommend you don't use this option unless you find updating unbearably slow without it.

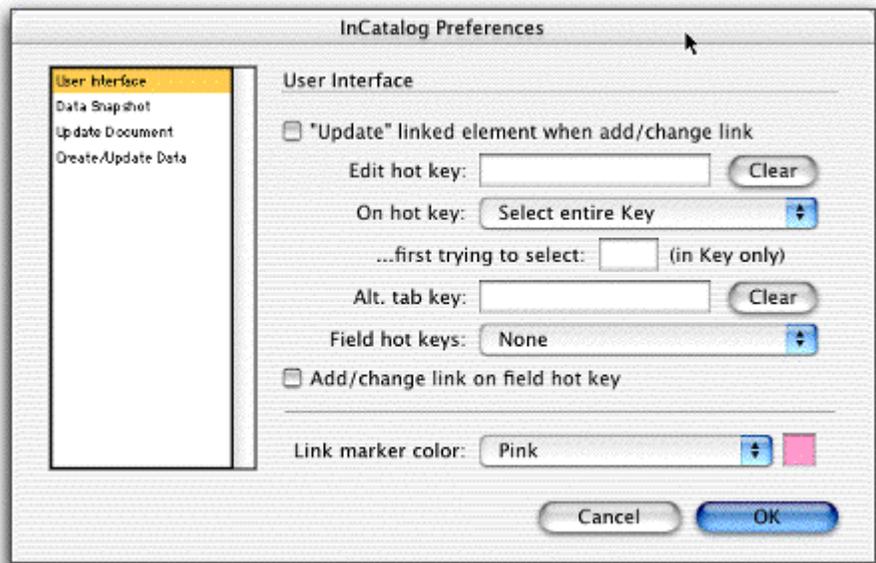
- Close Data File** closes the current data snapshot file.
- This menu item is enabled only when there is an open snapshot file.
- Open ODBC Connection...** opens an ODBC on-line connection using the DSN (data set name) specified in your data descriptor file, prompting you for a user name and password if your data descriptor requests it.
- This menu item is enabled only in the “Pro” version of InCatalog, and will be checked when there’s an open ODBC connection.
- Close ODBC Connection** closes any open ODBC connection.
- This menu item is enabled only when there’s an open ODBC connection.
- Edit Preference** This menu choices has several sub-menus to open the following preference dialogs:
- User Interface...
 - Data Snapshot...
 - Update Document
 - Create/Update Data
- These preferences are discussed in the next chapter, “Preferences.”
- About...** This menu choice opens a dialog displaying the serial number (including any multi-pack information), the name of the software licensee, the current version of InCatalog, and development credits.

InCatalog preferences are accessed via the InCatalog / Edit Preferences sub-menu. There are four sets of preferences, reflected in four different panels:

- User Interface
- Data Snapshot
- Update Document
- Create/Update Data

User Interface Preferences

The user interface preferences are divided into two major parts: hot keys and link marker display.



Hot Keys The "hot keys" portion has the following options.

“Update” linked element when add/change link When enabled, each time you add or change a link in a document and you have a data source selected (and open, where appropriate), InCatalog will attempt to update the contents of the current linked element, or of each element of the containing group, if the linked element is part of a group, or of each cell in the current table, if the linked element is in a cell.

This can be a real production time-saver. For example, with this option selected, if you drag a pre-linked group (with a dummy direct-style key value in one of its elements) from a library onto a page, and then change the dummy key value to a real key value, the entire group’s contents will be updated immediately with the appropriate data.

Edit hot key Allows the user to select a hot key that shifts the focus of the cursor from the InDesign document directly to the InCatalog linker palette. To change the hot key, select the hot key edit area and type the desired key or key combination (e.g., option-F5). The selected key will appear in the edit area, and anytime that key or key combination is typed (and the linking palette is active), the cursor will move from the InDesign document to an editing field in the linker palette (which field is controlled by the following option). To clear this field (leaving no edit hot key), press the Clear button next to it.

On hot key This hot key controls the cursor focus in the InCatalog linker palette when the edit hot key (see above) is activated. Its options are:

- **Select entire Key** When the hot key is typed, the cursor focus will move to the palette *and* select (highlight) the entire contents of the key edit area—making it possible to type a new key value without any further action.
- **Insert at start / end of key** When the hot key is typed, the cursor focus will move to the palette at the beginning or end of the key edit area, as selected. This option makes it easier to enter prefixes or suffixes on root keys.
- **Select entire Field** When the hot key is typed, the cursor focus will move to the palette and select the entire contents of the field edit area. You can then type enough of the desired field’s first characters to uniquely identify it.

Note that after you're done editing with a hot key, you can type the Return or Enter key to accept and apply the change, or the Escape key (Control-period under Mac OS as well) to cancel the change.

First trying to select ... (in key only) You can specify a single character that InCatalog will try to select, if present, in the key edit area when the hot key is used and one of the key field options (above) is selected for On hot key.

For example, if you commonly use root keys with a placeholder of “*”, then putting a * in this field will select just that placeholder when the hot key is depressed. Then you can very simply type the actual root followed by a Return, without any further keyboard or mouse action.

Alt. tab key With this option, you can select an alternate “hot key” to toggle keyboard focus between the key and field edit areas of the linker palette. To clear this field (leaving no alternate tab key), press the Clear button next to it.

Field hot keys With these choices—Function keys F1-F8 (Mac OS and Windows) or Numeric keypad keys 1-9 (Windows only)—you can have InCatalog automatically select one of the first 8 (or 9) fields listed in the field selection pop-up menu when the appropriate function key or numeric keypad key is typed, and then activate the appropriate edit area, based on the settings described above.

For example, if you select Function keys F1-F8 and press the F2 key, the second field on the field pop-up menu would be selected.

(The numeric keypad keys don't work under InDesign 1.x on Mac OS due to an InDesign limitation.)

Add/change link on field hot key With this option, in conjunction with the field hot key setting above, you can have InCatalog automatically complete link creation if a valid key is present.

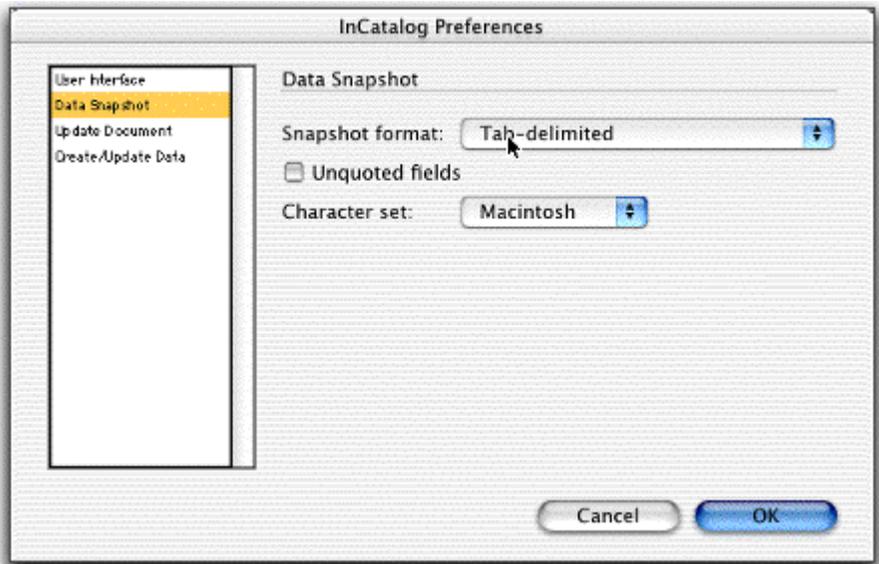
Using this option, and using the fact that the key field is “sticky,” you can have one-keypress linking, pretty much the ultimate in shortcuts: with a valid key setting already in the linking palette, select the text in the document to be linked to a particular field, press that field's corresponding hot key, and the link is made, with editing focus left in the document.

- ◆ *If you declare the fields in your data descriptor properly, when a given field is selected with a field hot key, all of its link properties (key type, price/tagged settings) should be set correctly and automatically, making this option even more useful.*

Link Marker The link marker display preference has the following option:

Link marker color To change the color of the link markers, select the Choose Color dialog and a standard color selection dialog is displayed. After making a color selection, the appropriate color swatch will appear in the box opposite Link marker color.

Data Snapshot Preferences



InCatalog accepts three different delimiter sets when using data snapshot text files as data source or destination files, as chosen with the Snapshot format pop-up:

- Comma-delimited
- Tab-delimited
- Tab-delimited (FoxBase Pro)

You can also indicate whether fields in the data snapshot file should have quoted-character handling with the `Unquoted fields` option. Normally, you should select this option, unless you know that quoted fields are being used in your data snapshot files (as produced by another application, usually).

When `Unquoted fields` is selected, InCatalog will treat double quote characters (") as normal data in a data snapshot file; otherwise, it will treat a double quote character at the start of a field as the start of a quoted field, which is ended only by another double quote character that isn't doubled.

A doubled (repeated) double quote character in a quoted field is treated as a literal single double quote. For example, the quoted field

```
"This is a ""quoted"" field."
```

will be read by InCatalog as

```
This is a "quoted" field.
```

The `Character set` menu allows you to specify which character set will be used in your data snapshot files, until you change it again. You can choose among `Macintosh`, `Windows`, or `Unicode`.

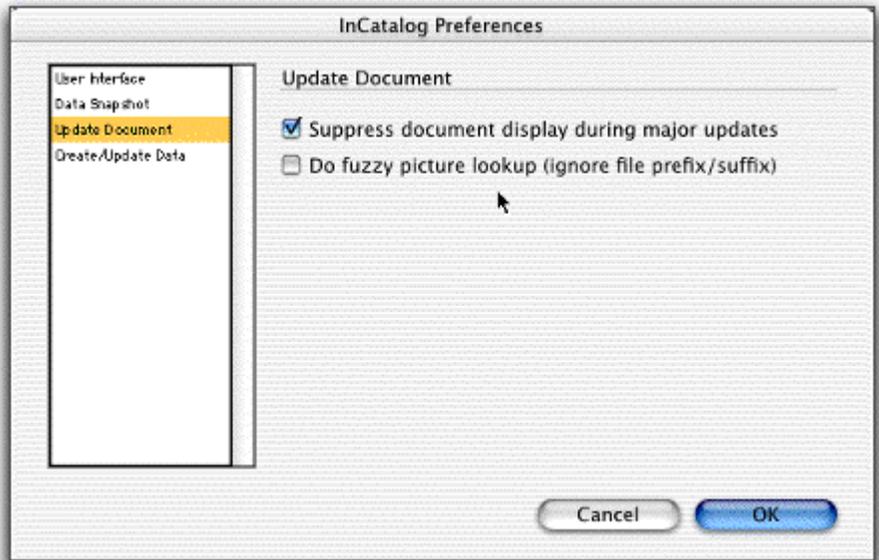
You can use `UTF-8` or `UCS (2-byte, big and little endian)` encodings in your `Unicode` files—the encoding used for input is auto-sensed based on the leading byte(s):

- an `0xFE` followed by an `0xFF` indicates `UCS-2 (big endian)`; else
- an `0xFF` followed by `0xFE` indicates `UCS-2 (little endian)`; else
- a `0x00` followed by a non-zero byte indicates `UCS-2 (big endian)`; else
- a non-zero byte followed by a `0x00` indicates `UCS-2 (little endian)`; else
- it's `UTF-8`.

If you choose the `Unicode` character set, then any output snapshot file's character set encoding will be `UTF-8` (which is almost universally supported in the `Unicode` world).

(If you don't know what any or all of this means, you can safely ignore it and let the character set default, which is dependent on your platform, `Macintosh` or `Windows`.)

Update Document Preferences

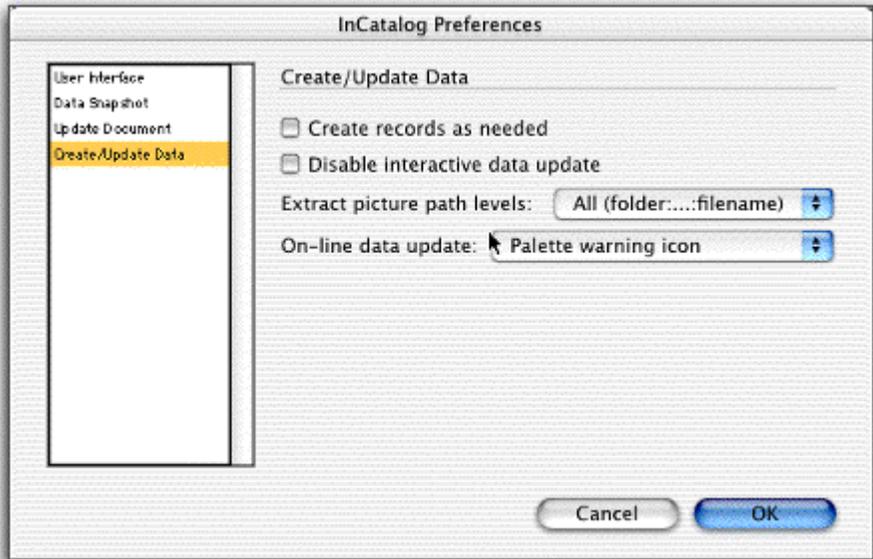


This set of preferences controls aspects of InCatalog’s operation when it is updating InDesign documents from a data source.

Suppress document display during major updates When this option is selected, InCatalog will suppress all document screen updates during its major (non-local interactive) data updates. Document update speed can be greatly improved when this option is selected, depending on the structure of the document.

Do “fuzzy” picture lookup (ignore file prefix/suffix) This preference tells InCatalog to find picture files with only approximate picture file names in the data source. When “fuzzy” picture lookup is activated, any simple picture file name lookups (i.e., lookups for file names containing neither an absolute nor a partial path, e.g., a SKU or part number) will cause InCatalog to use the first file found with the shortest file name fully containing the given picture file name (ignoring character case differences). See chapter “Updating,” section “Picture Update and Extraction” for full details on fuzzy picture lookup.

Create/Update Data Preferences



This set of preferences controls aspects of InCatalog’s operation when it is updating or creating data in the data destination from InDesign documents.

Create records as needed tells InCatalog to create new records when it encounters a key value that doesn’t currently exist in the data destination.

Disable interactive data update tells InCatalog to disable all interactive updating of data from the document. (This is a safety measure for those sites that don’t ever update data from a document and don’t want it to be possible. Note that this doesn’t stop a malicious user.)

Extract picture path levels controls how much of a picture path name is extracted when updating data:

- All (folder:...file name) extracts the complete (absolute) path-name.
- 1 (file name) extracts just the file name without any folder paths.
- 2 (folder:file name) extracts only the folder name in which the file resides plus the file name.

- 3 (folder:folder:file name) extracts up to two levels of containing folders plus the file name.
- 4 (folder:folder:folder:file name) extracts up to three levels of containing folders plus the file name.
- 5 (folder:folder:folder:folder:file name) extracts up to four levels of containing folders plus the file name.

Interactive warning When updating data interactively from a InDesign document, InCatalog provides two options for warning the user that the data destination is about to be altered: a palette warning icon which displays in the linker palette's update section (the default), or, for extreme caution, an alert at each on-line update, asking the user to confirm the update each time.

InCatalog was designed from the start to function as part of a larger automated workflow, and is thus fully scriptable. It also has a simple cross-platform batch facility for file-based automation.

Simple Batch Facility

InCatalog has a very simple batch facility of limited utility, intended as a low-technology path for another application (e.g., a database-driven tool), running in parallel with InDesign, to interactively populate an open document with data.

If a sub-folder “InCatalog:batch” (Mac OS) or “InCatalog\batch” (Windows) exists in the home application folder at InCatalog (InDesign) start-up time, then (and only then) InCatalog enables a simple folder-watching-driven batch capability as follows.

- Every second, InCatalog checks in its batch folder for the appearance of a control file (any file whose name ends in “.xbc” for “Xcatalog batch control”—this predates InCatalog), and, if found, processes it as follows.
- InCatalog reads two lines of data from the file, which is a normal ASCII file with normal line endings.
- The first line should (for now) contain the four words “update document using file” (exactly as given, with no extra spaces and no quotes).
- The second line should contain the full path name (platform-appropriate) of a data snapshot file to be used for a document update (with no extra spaces, escape characters or punctuation in the name).
- If it gets this far successfully, and if a document is open, InCatalog updates the current document with the given snapshot data.
- The batch control file is deleted when InCatalog finishes the batch operation.

(There is interlocking code to avoid race conditions: it tries to open the file for exclusive access, and, if that fails, it backs off a second and tries again. Thus, if you're writing such batch control file, you should open it for exclusive access until you're finished writing it, and you should only write it after you've prepared any other files that control file references. Or, you could create it without the .xbc extension and rename it when you're done.)

◆ *NB: This batch facility is likely to be expanded and may well change; it's in a very nascent form right now.*

Mac OS Scripting

InCatalog provides complete access from AppleScript (or any OSA-compliant language such as Frontier, or even Javascript and Python) to every aspect of InCatalog's updating process in either direction, including all data source/destinations—file, string, FileMaker on-line, and ODBC (Pro version only) on-line—, options and preferences.

On-line terminology is available by inspecting the InDesign AppleEvent dictionary from your favorite AppleScript development tool. It will look roughly like the following.

InCatalog Suite: Em Software's InCatalog events for document update from data, data update from document, and (eventually) link creation and inspection.

update document using file: Update the contents of the current document's linked elements from the given database snapshot file.

```
update document using file alias -- data snapshot file
    [key updating boolean] -- keys in the data links themselves are to be
    updated, not the document contents--the DD must describe three fields: old
    key value (as the key), new key value, optional DD name (defaults to false)
    [DD set alias] -- alias to DD set folder (defaults to any current DD set)
    [DD string] -- name of data descriptor in DD set (defaults to any
current DD)
    [DD contents string] -- actual DD contents in lieu of DD set/DD
    [DD map string] -- which DD map to use
    [first page small integer] -- start of page range to update (defaults
to 1)
    [last page small integer] -- end of page range for update (defaults to
```

last page in document)
 [first spread small integer] -- start of spread range to update
 (defaults to 1)
 [last spread small integer] -- end of spread range to update (defaults
 to last spread in document)
 [pictures ignored boolean] -- don't process any pictures during
 update (defaults to global preference)
 [hidden layers ignored boolean] -- don't process any frames in
 hidden document layers (defaults to global preference)
 [document display suppressed boolean] -- don't show document
 changes during update (defaults to global preference)
 [fuzzy picture lookup boolean] -- import pictures using "fuzzy"
 picture lookup (ignoring file prefix/suffix; defaults to global preference)
 [partial data boolean] -- if a record isn't found, it's not an error
 (defaults to false)
 [data format tab delimited/Foxpro tab delimited/comma delimited] -
 integer -- the number of linked elements updated in the document- snapshot
 file's delimited ascii format (defaults from data preferences)
 [unquoted data boolean] -- whether delimited data uses quote
 marks around fields or not (defaults from data preferences)
 Result: integer -- the number of linked elements updated in the docu-
 ment

update data using file: Update/create the contents of the given database
 snapshot file from the current document's linked elements' contents.

update data using file alias -- data snapshot file (if it exists, its contents
 will be updated; if it doesn't exist (use a pathname in this case, not an alias),
 it will be created and filled)
 [DD set alias] -- alias to DD set folder (defaults to any current DD set)
 [DD string] -- name of data descriptor in DD set (defaults to any
 current DD)
 [DD contents string] -- actual DD contents in lieu of DD set/DD
 [DD map string] -- which DD map to use
 [first page small integer] -- start of page range to update (defaults
 to 1)
 [last page small integer] -- end of page range for update (defaults to
 last page in document)
 [first spread small integer] -- start of spread range to update
 (defaults to 1)
 [last spread small integer] -- end of spread range to update (defaults
 to last spread in document)
 [pictures ignored boolean] -- don't process any pictures during
 update (defaults to global preference)
 [hidden layers ignored boolean] -- don't process any frames in

hidden document layers (defaults to global preference)
 [picture path levels integer] -- number of picture path levels to extract--0 means "all" (defaults to global preference)
 [record creation boolean] -- create records as needed (defaults to true if snapshot file is being created; otherwise defaults to global preference)
 [snapshot creation boolean] -- the snapshot file should be fully created (fleshed out) from the .x file when finished; otherwise, the collected data will simply be left in the .x file for a later time--this is only useful in the case of huge snapshot files (defaults to true)
 [data format tab delimited/Foxpro tab delimited/comma delimited] -
 - delimited ascii snapshot string's format (defaults from data preferences)
 [unquoted data boolean] -- whether delimited data uses quote marks around fields or not (defaults from data preferences)
 Result: integer -- the number of linked elements extracted from the document

update document using string: Update the contents of the current document's linked elements from the given string.

update document using string string -- whole contents of data to be used for injection (comma- or tab-separated, return-terminated)
 [key updating boolean] -- keys in the data links themselves are to be updated, not the document contents--the DD must describe three fields: old key value (as the key), new key value, optional DD name (defaults to false)
 [DD set alias] -- alias to DD set folder (defaults to any current DD set)
 [DD string] -- name of data descriptor in DD set (defaults to current DD)
 [DD contents string] -- actual DD contents in lieu of DD set/DD
 [DD map string] -- which DD map to use
 [first page small integer] -- start of page range to update (defaults to 1)
 [last page small integer] -- end of page range for update (defaults to page page in document)
 [first spread small integer] -- start of spread range to update (defaults to 1)
 [last spread small integer] -- end of spread range to update (defaults to last spread in document)
 [pictures ignored boolean] -- don't process any pictures during update (defaults to global preference)
 [hidden layers ignored boolean] -- don't process any frames in hidden document layers (defaults to global preference)
 [document display suppressed boolean] -- don't show document changes during update (defaults to global preference)

[fuzzy picture lookup boolean] -- import pictures using "fuzzy" picture lookup (ignoring file prefix/suffix; defaults to global preference)
[partial data boolean] -- if a record isn't found, it's not an error (defaults to false)
[data format tab delimited/Foxpro tab delimited/comma delimited] -
- delimited ascii snapshot string's format
[unquoted data boolean] -- whether delimited data uses quote marks around fields or not (defaults to false)
Result: integer -- the number of linked elements updated in the document

update document using FileMaker: Update the contents of the current document's linked elements from one or more open FileMaker databases.

update document using FileMaker
[Runtime boolean] -- source is a FileMaker Pro Runtime application database (defaults to false)
[key updating boolean] -- keys in the data links themselves are to be updated, not the document contents--the DD must describe three fields: old key value (as the key), new key value, optional DD name (defaults to false)
[DD set alias] -- alias to DD set folder (defaults to any current DD set)
[DD string] -- name of data descriptor in DD set (defaults to any current DD)
[DD contents string] -- actual DD contents in lieu of DD set/DD
[DD map string] -- which DD map to use
[first page small integer] -- start of page range to update (defaults to 1)
[last page small integer] -- end of page range for update (defaults to last page in document)
[first spread small integer] -- start of spread range to update (defaults to 1)
[last spread small integer] -- end of spread range to update (defaults to last spread in document)
[pictures ignored boolean] -- don't process any pictures during update (defaults to global preference)
[hidden layers ignored boolean] -- don't process any frames in hidden document layers (defaults to global preference)
[document display suppressed boolean] -- don't show document changes during update (defaults to global preference)
[fuzzy picture lookup boolean] -- import pictures using "fuzzy" picture lookup (ignoring file prefix/suffix; defaults to global preference)
Result: integer -- the number of linked elements updated in the document

update data using FileMaker: Update/create the contents of one or more open FileMaker databases from the current document's linked elements' contents.

update data using FileMaker
[Runtime boolean] -- destination is a FileMaker Pro Runtime application database (defaults to false)
[DD set alias] -- alias to DD set folder (defaults to any current DD set)
[DD string] -- name of data descriptor in DD set (defaults to any current DD)
[DD contents string] -- actual DD contents in lieu of DD set/DD
[DD map string] -- which DD map to use
[first page small integer] -- start of page range to update (defaults to 1)
[last page small integer] -- end of page range for update (defaults to last page in document)
[first spread small integer] -- start of spread range to update (defaults to 1)
[last spread small integer] -- end of spread range to update (defaults to last spread in document)
[pictures ignored boolean] -- don't process any pictures during update (defaults to global preference)
[hidden layers ignored boolean] -- don't process any frames in hidden document layers (defaults to global preference)
[picture path levels integer] -- number of picture path levels to extract--0 means "all" (defaults to global preference)
[record creation boolean] -- create records as needed (defaults to global preference)
Result: integer -- the number of linked elements extracted from the document

update document using ODBC: Update the contents of the current document's linked elements from one or more ODBC tables.

update document using ODBC
[key updating boolean] -- keys in the data links themselves are to be updated, not the document contents--the DD must describe three fields: old key value (as the key), new key value, optional DD name (defaults to false)
[DSN string] -- system- or user-level data set name for ODBC connection (defaults to any DSN DD setting)
[user name string] -- user name for ODBC login (defaults to any DD setting)
[password string] -- password for ODBC login (defaults to any DD setting)

[DD set alias] -- alias to DD set folder (defaults to any current DD set)
 [DD string] -- name of data descriptor in DD set (defaults to any current DD)
 [DD contents string] -- actual DD contents in lieu of DD set/DD
 [DD map string] -- which DD map to use
 [first page small integer] -- start of page range to update (defaults to 1)
 [last page small integer] -- end of page range for update (defaults to last page in document)
 [first spread small integer] -- start of spread range to update (defaults to 1)
 [last spread small integer] -- end of spread range to update (defaults to last spread in document)
 [pictures ignored boolean] -- don't process any pictures during update (defaults to global preference)
 [hidden layers ignored boolean] -- don't process any frames in hidden document layers (defaults to global preference)
 [document display suppressed boolean] -- don't show document changes during update (defaults to global preference)
 [fuzzy picture lookup boolean] -- import pictures using "fuzzy" picture lookup (ignoring file prefix/suffix; defaults to global preference)
 Result: integer -- the number of linked elements updated in the document

update data using ODBC: Update/create the contents of one or more ODBC tables from the current document's linked elements' contents.

update data using ODBC
 [DSN string] -- system- or user-level data set name for ODBC connection (defaults to any set in the given or current DD)
 [user name string] -- user name for ODBC login (defaults to any DD setting)
 [password string] -- password for ODBC login (defaults to any DD setting)
 [DD set alias] -- alias to DD set folder (defaults to any current DD set)
 [DD string] -- name of data descriptor in DD set (defaults to any current DD)
 [DD contents string] -- actual DD contents in lieu of DD set/DD
 [DD map string] -- which DD map to use
 [first page small integer] -- start of page range to update (defaults to 1)
 [last page small integer] -- end of page range for update (defaults to last page in document)
 [first spread small integer] -- start of spread range to update (defaults to 1)
 [last spread small integer] -- end of spread range to update (defaults

to last spread in document)
 [pictures ignored boolean] -- don't process any pictures during update (defaults to global preference)
 [hidden layers ignored boolean] -- don't process any frames in hidden document layers (defaults to global preference)
 [picture path levels integer] -- number of picture path levels to extract--0 means "all" (defaults to global preference)
 [record creation boolean] -- create records as needed (defaults to global preference)
 Result: integer -- the number of linked elements extracted from the document

Windows Scripting

InCatalog provides complete access from any Windows Automation client such as VB or VBA to every aspect of InCatalog's updating process in either direction, including all data source/destinations—file, string, and ODBC (Pro version only) on-line.

InCatalog's type library looks like the following.

```
enum { EmTabDelimited, EmFoxProTabDelimited, EmCommaDelimited }
      EmDataFormatType;
```

```
long InCatalogUpdateDocUsingFile(
  [in] VARIANT dataFilePath,
  [optional, in] EmDataFormatType dataFormat,
  [optional, in] VARIANT_BOOL dataUnquoted,
  // Common update parameters:
  [optional, in] VARIANT ddSetFolder,
  [optional, in] BSTR ddName,
  [optional, in] BSTR ddContents,
  [optional, in] BSTR ddMap,
  [optional, in] short firstPage,
  [optional, in] short LastPage,
  [optional, in] short firstSpread,
  [optional, in] short lastSpread,
  [optional, in] VARIANT_BOOL ignorePictures,
  [optional, in] VARIANT_BOOL ignoreHiddenLayers,
  // Event-specific parameters:
  [optional, in] VARIANT_BOOL updateKeys,
  [optional, in] VARIANT_BOOL suppressDocDisplay,
  [optional, in] VARIANT_BOOL useFuzzyPictureLookup,
```

```

        [optional, in] VARIANT_BOOL allowPartialData
    );

    long InCatalogUpdateDataUsingFile(
        [in] VARIANT dataFilePath,
        [optional, in] EmDataFormatType dataFormat,
        [optional, in] VARIANT_BOOL dataUnquoted,
        // Common update parameters:
        [optional, in] VARIANT ddSetFolder,
        [optional, in] BSTR ddName,
        [optional, in] BSTR ddContents,
        [optional, in] BSTR ddMap,
        [optional, in] short firstPage,
        [optional, in] short LastPage,
        [optional, in] short firstSpread,
        [optional, in] short lastSpread,
        [optional, in] VARIANT_BOOL ignorePictures,
        [optional, in] VARIANT_BOOL ignoreHiddenLayers,
        // Event specific parameters:
        [optional, in] long numPicturePathLevels,
        [optional, in] VARIANT_BOOL createRecords,
        [optional, in] VARIANT_BOOL createSnapshot
    );

    long InCatalogUpdateDocUsingString(
        [in] BSTR dataString,
        [optional, in] EmDataFormatType dataFormat,
        [optional, in] VARIANT_BOOL dataUnquoted,
        // Common update parameters:
        [optional, in] VARIANT ddSetFolder,
        [optional, in] BSTR ddName,
        [optional, in] BSTR ddContents,
        [optional, in] BSTR ddMap,
        [optional, in] short firstPage,
        [optional, in] short LastPage,
        [optional, in] short firstSpread,
        [optional, in] short lastSpread,
        [optional, in] VARIANT_BOOL ignorePictures,
        [optional, in] VARIANT_BOOL ignoreHiddenLayers,
        // Event specific parameters:
        [optional, in] VARIANT_BOOL updateKeys,
        [optional, in] VARIANT_BOOL suppressDocDisplay,
        [optional, in] VARIANT_BOOL useFuzzyPictureLookup,
        [optional, in] VARIANT_BOOL allowPartialData
    );

```

```

);

long InCatalogUpdateDocUsingODBC(
    [optional, in] BSTR DSN,
    [optional, in] BSTR userName,
    [optional, in] BSTR userPassword,
    // Common update parameters:
    [optional, in] VARIANT ddSetFolder,
    [optional, in] BSTR ddName,
    [optional, in] BSTR ddContents,
    [optional, in] BSTR ddMap,
    [optional, in] short firstPage,
    [optional, in] short LastPage,
    [optional, in] short firstSpread,
    [optional, in] short lastSpread,
    [optional, in] VARIANT_BOOL ignorePictures,
    [optional, in] VARIANT_BOOL ignoreHiddenLayers,
    // Event specific parameters:
    [optional, in] VARIANT_BOOL updateKeys,
    [optional, in] VARIANT_BOOL suppressDocDisplay,
    [optional, in] VARIANT_BOOL useFuzzyPictureLookup
);

long InCatalogUpdateDataUsingODBC(
    [optional, in] BSTR DSN,
    [optional, in] BSTR userName,
    [optional, in] BSTR userPassword,
    // Common update parameters:
    [optional, in] VARIANT ddSetFolder,
    [optional, in] BSTR ddName,
    [optional, in] BSTR ddContents,
    [optional, in] BSTR ddMap,
    [optional, in] short firstPage,
    [optional, in] short LastPage,
    [optional, in] short firstSpread,
    [optional, in] short lastSpread,
    [optional, in] VARIANT_BOOL ignorePictures,
    [optional, in] VARIANT_BOOL ignoreHiddenLayers,
    // Event specific parameters:
    [optional, in] long numPicturePathLevels,
    [optional, in] VARIANT_BOOL createRecords
);

```

All these functions return the number of document elements actually updated (for UpdateDocument) or extracted (for UpdateData).

Here is a simple example of driving an InCatalog document update from Visual Basic:

```
Sub UpdateDoc(docFile as String, dataFile as String)
    Dim itemCount as long
    Set myInDesign = GetObject(" ", "InDesign.Application")
    myInDesign.Activate
    myInDesign.Open(docFile)
    On Error GoTo theerrorhandler
    itemCount = myInDesign.InCatalogUpdateDocUsingFile(dataFile,
        EmTabDelimited)
    MsgBox("InCatalog processed " & itemCount & " links.")
    Exit Sub
theerrorhandler:
    MsgBox ("InCatalog: " & Err.Description)
End Sub
```

