

# EASYCATALOG

## FOR ADOBE® INDESIGN®



# RELATIONAL MODULE USER GUIDE

65bit Software Ltd



### *Revision History*

Version	Date	Remarks
1.0.0	02 May 2008	First draft.
1.0.1	08 August 2008	First release.

© **Copyright 2008 65bit Software Limited.** All Rights reserved. Reproduction or copying prohibited.

Adobe and InDesign are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

All other trademarks and copyrights are the property of their respective owners.

# CONTENTS

<i>OVERVIEW .....</i>	<i>5</i>
Welcome.....	5
What is it?.....	5
Benefits .....	5
Installation .....	6
Assumptions.....	6
Further assistance .....	6
 <i>INTEGRATION .....</i>	 <i>7</i>
Overview.....	7
Prototyping .....	7
Seamless Integration.....	7
Multiple client workflows .....	7
Browser-based data maintenance .....	7
 <i>HOW IT WORKS.....</i>	 <i>8</i>
Built-in database engine .....	8
Child Data Sources .....	8
Virtual tables.....	8
Groups.....	8
Queries .....	8
Workspace contents.....	9
Sqlite Data Sources.....	9
Further references .....	9
 <i>RELATIONAL DATA SOURCES .....</i>	 <i>10</i>
Example Data .....	10
Creating an empty relational panel.....	11
Creating a Child Data source .....	11
Creating a Relationship .....	13
Panel edit Mode .....	17
Editing Relational Data sources .....	18
Refining the demo .....	20
Multiple Groups .....	21
Menu .....	24
Links .....	25
Adoption.....	26
Storage.....	26
Further assistance .....	26

---

*MENU REFERENCE.....27*

The Relational Panel Menu..... 27

    Synchronize with Data Source ..... 27

    Show/Hide Fields ..... 27

    Field Formatting Tags..... 27

    Field Specifiers..... 27

    New Panel ..... 27

    Table → Row Options ..... 28

    Reports..... 29

    Adopt Fields ..... 29

## CHAPTER 1

# OVERVIEW

---

**WELCOME** Thank you for downloading the Relational Module for EasyCatalog. 65bit Software are committed to providing high quality software for Adobe InDesign, and appreciate the time you take to evaluate our products.

All feedback is welcome, good or bad. Please email any comments you have to [support@65bit.com](mailto:support@65bit.com).

If you have any questions relating to this or any of our other products, please use the [support form](#) on our website.

**WHAT IS IT?** The Relational Module allows EasyCatalog to combine data from multiple data sources which can then be linked hierarchically in a panel within InDesign.

The Relational Module also offers the ability to use SQLite databases as a data source.

The Relational Module is an additional module that must be purchased in addition to EasyCatalog and the optional data providers. Pricing information for the Relational Module and EasyCatalog can be found on our web-store at the following link: [65bit Software Web Store](#)

- BENEFITS**
- Working with complex data becomes intuitive for non-technical users.
  - Tighter integration with relational databases, such as PIM systems, without the need for custom development.
  - System integrators and resellers can easily create tailor-made custom database publishing solutions.
  - Complex data structures can be modelled from various sources of data, including databases, delimited files, XML files and HTTP portals.
  - Panels representing the underlying physical model of the data can be created without the need to create a 'flattened view' of the data.
  - Simplified menu structure. Many facilities of regular EasyCatalog panels become unnecessary.
  - Updating relational databases is made easy.
  - Data of virtually any complexity can be paginated, such as products with multiple sets of attributes, or records that appear in multiple sets of data.

- Where a direct connection to your database is not available or allowed, instead of creating a delimited file from your database an SQLite database file could be used.

## **INSTALLATION**

The Relational Module is an optional module for EasyCatalog, and is distributed as part of the main EasyCatalog installer. During the installation process, please ensure that you select the 'Relational Module' plus any of the data providers that you need to access your data.

## **ASSUMPTIONS**

- This manual assumes that you have a working knowledge of EasyCatalog.
- A basic understanding of SQL is also desirable. Although SQLite can process much of the SQL language, there are some minor differences and omissions. Details of these can be found on the [SQLite website](#).

## **FURTHER ASSISTANCE**

We'd be happy to configure a demo of the relational module. Ideally we need an extract from each of the data sources involved, a brief description or schematic of how they relate, and an example of the kind of output to be produced.

Using this data we can set a demo up which we send back. There is no obligation by using service, as we'd be happy to show you the benefits of using our software. Feel free to contact us using the [support form](#) on our web-site.

We are more than happy to help configure the software for your evaluation and to answer any questions you may have.

## CHAPTER 3

# HOW IT WORKS

---

**BUILT-IN DATABASE ENGINE**

The Relational Module uses a built-in relational database engine, SQLite, to locally store your data. This is the most widely deployed SQL engine in the world, used in desktop computer applications as well as consumer electronic devices. There is no installation, other than installing the plug-ins and once a data source is configured the user is not aware of its use.

**CHILD DATA SOURCES**

Each relational data source maintains a collection of child data sources. These are similar to regular data sources, except they are managed and stored as part of a relational data source.

Child data sources are physically stored inside the parent data source's workspace folder structure.

**VIRTUAL TABLES**

Whatever its original source, the module treats each child data source as a separate SQL virtual table. These tables mirror the contents of the child data sources exactly. SQL Joins between tables ultimately define how each data source relates to each other.

**GROUPS**

Groups are displayed in a relational data source panel, and each relational panel must contain at least one (top-level) group. The user navigates between groups using user interface provided by the panel. A group is simply a named collection of records, retrieved using an SQL query against one or more of the virtual tables. Groups can be children of other groups and even placed alongside one another - useful when a product has multiple sets of data. Groups are given a unique name that is used to reference the data they contain.

**QUERIES**

SQLite provides a rich SQL syntax, including the ability to create ad-hoc result columns (for example, the calculation of a collection of fields).

The relational module uses a mechanism to differentiate between this type of column and those from the child data sources. If a query result contains key fields these are used to link to the original source data. When no key fields are found, columns are treated as 'ad-hoc' and will be displayed in red when in edit mode. These kinds of fields can be paginated, but not updated after pagination.

Another method is to include a 'rowid' column in the result set. This contains an internal reference that identifies which record the result came from. The advantage of using rowid is that this is a hidden column.

### **WORKSPACE CONTENTS**

Relational data sources are stored as regular data sources in the EasyCatalog workspace folder. All configuration information is stored in the 'Datasource.xml' file, which describes the child data sources, the links between groups and field definitions for any fields that are created on an ad-hoc basis.

Child data sources are stored inside a relational data sources folder structure.

### **SQLITE DATA SOURCES**

The module also provides the ability to create regular data sources from SQLite files (accessed from the File→New→EasyCatalog Panel→New SQLite Data Source menu). These data sources can be used independently or combined by the relational data provider. Typically each table in an SQLite database file be be created as a separate child data source and combined using the relational module.

All updates to SQLite data sources are transactional. If a relational data source uses multiple tables in single SQLite file, all updates are executed using a single transaction.

Default update statements are provided when creating SQLite data sources.

### **FURTHER REFERENCES**

A wealth of information on SQLite can be found on their website:

<http://www.sqlite.org>



## CHAPTER 2

# INTEGRATION

---

**OVERVIEW**

The Relational module enables EasyCatalog to work seamlessly with data held in a relational form, such as tables in a database. It facilitates the creation and navigation of multiple EasyCatalog data sources, linked by the data they contain.

Once the plug-in is configured the user works with a single view of data and can easily navigate through complex hierarchies.

**PROTOTYPING**

Because the Relational module is easy to setup and use it's ideal for the rapid prototyping of new and experimental database-based systems.

Data can be sourced from anywhere, including a combination of delimited files, XML files or ODBC databases. Once configured, the entire data set can be exported as a single (SQLite) database file, which is useful when prototyping or when making a transition from a flat file-based workflow to a client server solution. The exported file can be used to populate external databases such as MySQL, SQL Server or used directly within a browser-based environment.

**SEAMLESS  
INTEGRATION**

Since a datasource is defined using a collection of XML configuration files, these could potentially be generated automatically from third-party product information systems.

**MULTIPLE CLIENT  
WORKFLOWS**

In an environment where multiple users work from a single database, the relational module could also be employed to reduce synchronization times. This involves extracting information from the database into an SQLite file that is then copied to each user's computer or a common location. Users who synchronize against this file will see a significant improvement in performance when compared to an ODBC connection. This will also reduce the load on the host database, and the transactional nature of updates to the file also ensures the integrity of the data.

**BROWSER-BASED  
DATA MAINTENANCE**

While SQLite is not a client server database in the traditional sense, it's ideal for use in a hosted solution with fewer than 100K hits per day. Using the module it's easy to build solutions that allow complex data structures to be maintained via a browser interface. For example a design agency could allow their customers to maintain product information. If you are interested in us or one of our partners building such a solution, then feel free to email us.

## CHAPTER 4

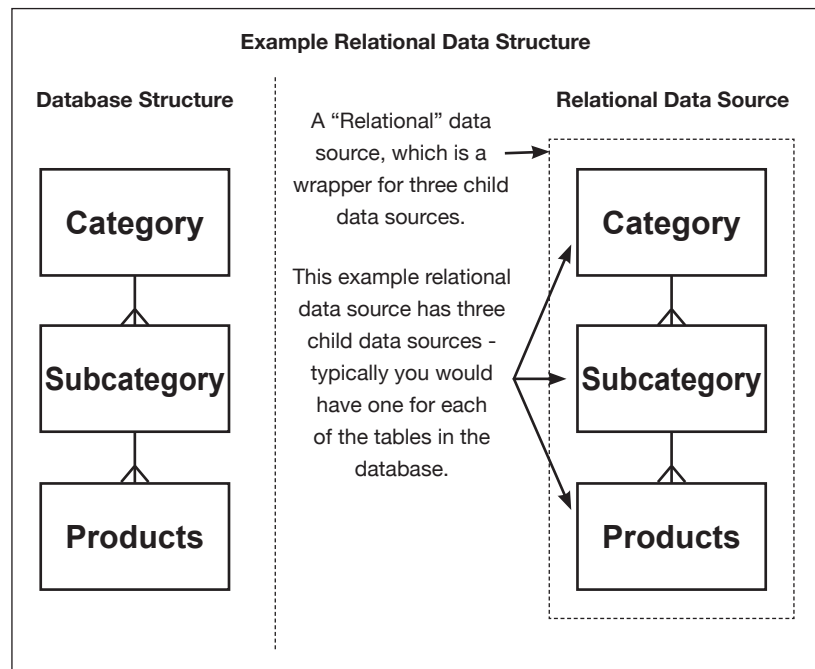
# RELATIONAL DATA SOURCES

**EXAMPLE DATA** To illustrate a typical configuration of the module we'll look at example data and work through the steps involved with setting up a relational data source from scratch.

The demo data represents a hierarchy of "Products", stored in a hierarchy of "Categories" and "Subcategories".

A simplified relational database structure would contain three tables to represent this; "category", "subcategory" and "products".

To represent this data model in the Relational Module, three EasyCatalog data sources are required: a Relational data source that contains three child data sources (one for each of the tables).

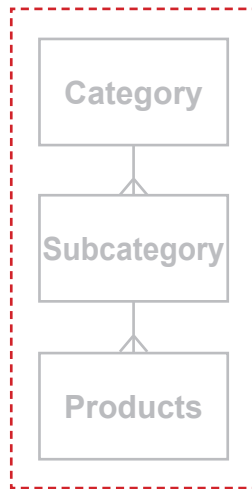


For the purposes of this illustration the data is held as a series of text files, which can be downloaded from our web-site:

<http://www.65bit.com/downloads/easycatalog/files/CS3/300/Relational/Relational Data.zip>

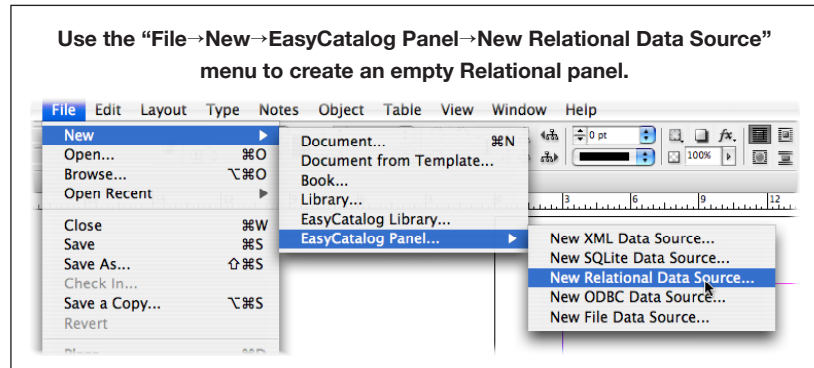
The contents of these files represents a typical relational database - each file should be considered to be a table in a database.

## CREATING AN EMPTY RELATIONAL PANEL



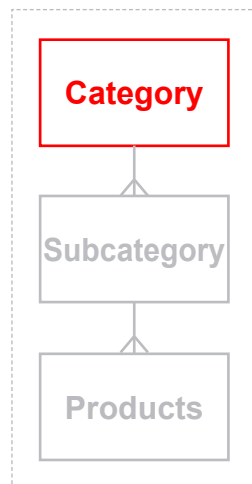
- 1 The first step is to create an empty relational panel. Consider the relational panel to be a container for all of the child data sources that represent tables in the database.

Create a Relational panel using the “File→New→EasyCatalog Panel→New Relational Data Source” menu. When prompted, give the data source any name, such as “Relational Test” and hit OK. An empty Relational panel should now be displayed:



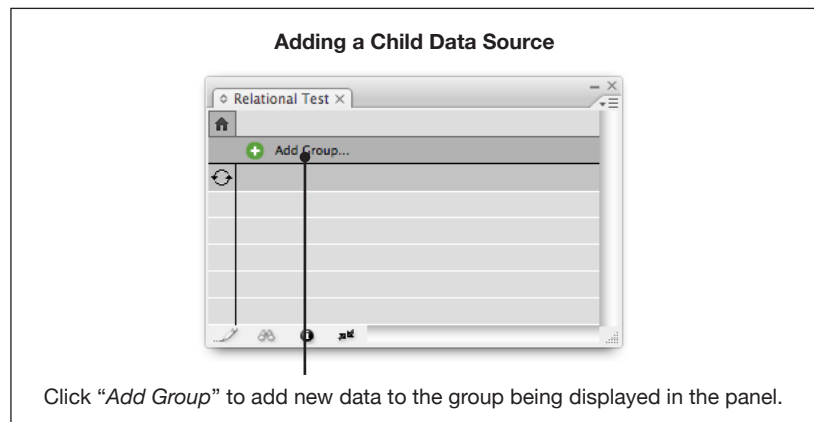
The panel is currently in “Panel Edit Mode”. Once setup is complete this can be disabled and a simpler user interface is presented.

## CREATING A CHILD DATA SOURCE



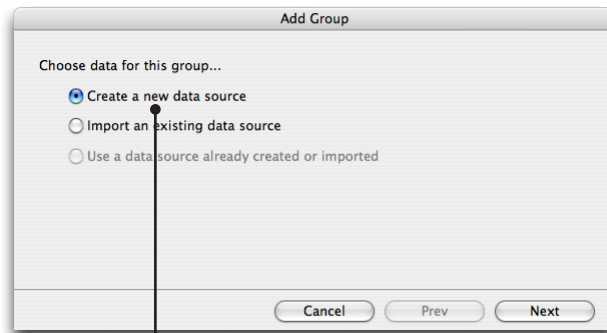
- 2 Now that we have a relational panel, we can go ahead and add the child data sources to the panel and define how each of these data sources relate to each other.

We'll start at the top of the data hierarchy by adding a list of categories:

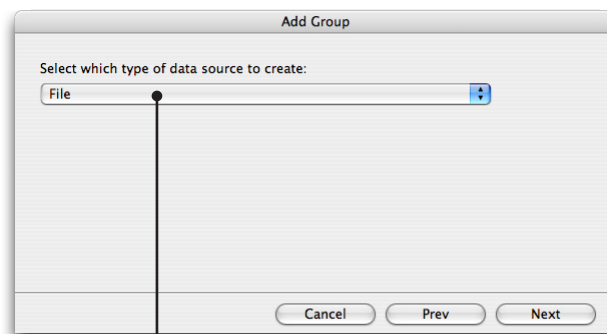


Click “Add Group” to add new data to the group being displayed in the panel.

## Adding a Child Data Source (continued)

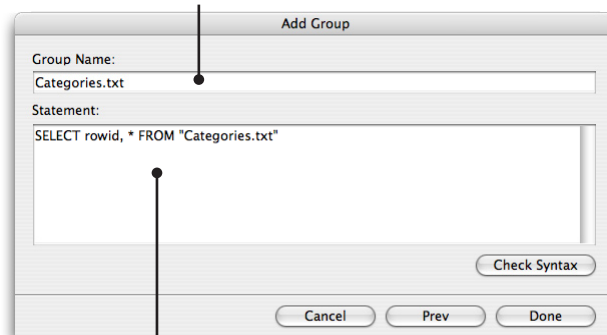


We need to create a new data source for the “Categories.txt” data, so select “Create a new data source” and click “Next”.



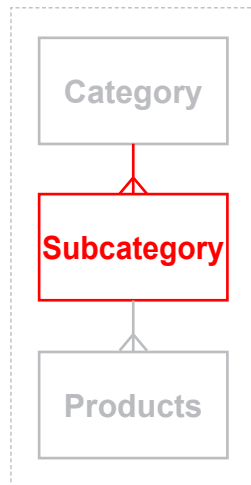
A list of data types that can be imported is shown here. The demo data is being imported from a file, so select “File” and click “Next”. Choose the “Categories.txt” file and OK File Data Provider configuration dialog that appears. The default settings are sufficient for this example, so click “OK”.

The name of the group that will be displayed in the Relational panel



The final screen allows you to tailor the results shown in the panel. The default SQL statement will select all data from the data source you've just imported. Again, this is sufficient for this example so click “Done”.

## CREATING A RELATIONSHIP



3

Once the “*Categories.txt*” group is in place we need to add the next level in the hierarchy, the “*Subcategories.txt*” group. To do this, we first need to navigate to where we want the new group to be added in the panel. Click on any of the categories in the list to go down a level in the hierarchy:

**Navigating around the grouping hierarchy**

Delete this level      Add another group at this level

Category ID	Category
1	AV Accessories
2	Audio Separates
4	Batteries & Chargers
5	Books

The fields in a relational panel behave in a similar way to a hyperlink in a web-browser. Click on one of the “Category” names in the list to go into the group:

The navigation bar at the top of the panel shows the group that is currently being displayed.

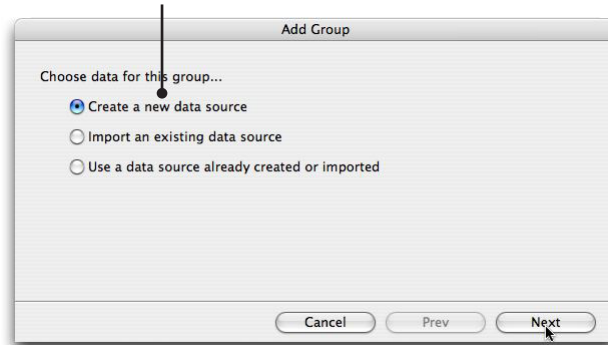
Use the “Add Group” button to add data from a new file based data source.

**Adding a child group**

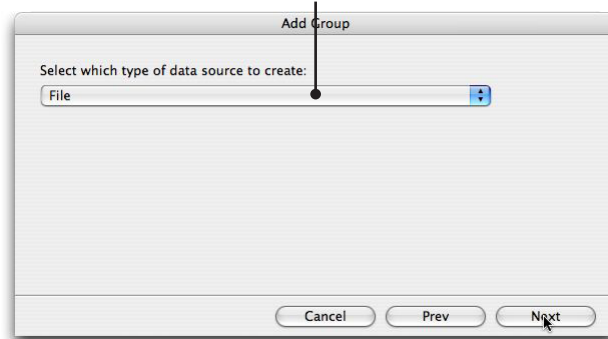
Navigate into any of the categories, and use the “Add Group” button to add a new group.

## Adding a child group (continued)

The “*Subcategory.txt*” data source has not been imported yet, so select “*Create a new data source*”



We’re importing from a file (as before), so select “*File*” and import the “*Subcategory.txt*” file.



As we’re adding a sub-group, EasyCatalog will now allow you to configure the relationship between this new group (*Subcategories.txt*) and the parent group (*Categories.txt*).



## Adding a child group (continued)

From the database schema, we can see that the two tables/files are linked by *Category ID*.

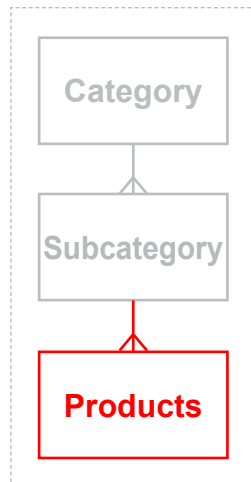
The diagram illustrates the relationship between two tables: **Categories.txt** and **Subcategories.txt**. **Categories.txt** has fields **Category ID** and **Category**. **Subcategories.txt** has fields **Subcategory ID**, **Subcategory**, and **Category ID**. A line labeled *Category ID* connects the **Category ID** field in **Categories.txt** to the **Category ID** field in **Subcategories.txt**. Below this, the **Add Group** dialog box is shown. It has a title bar "Add Group" and a subtitle "Choose the criteria for the selection...". It contains two pop-up menus: "Where field:" and "Matches parent:". Both menus have "Category ID" selected. The "Where field:" menu has a blue arrow pointing right, and the "Matches parent:" menu has a blue arrow pointing left. At the bottom are buttons for "Cancel", "Prev", and "Next".

The “*Where field*” pop-up menu lists the fields in the data source we’ve just added; the “*Matches parent*” pop-up lists the fields from the parent group.

On the following screen, the SQL that is generated includes the necessary “where” clause to link the “*Subcategory.txt*” table to the “*Category.txt*” table.

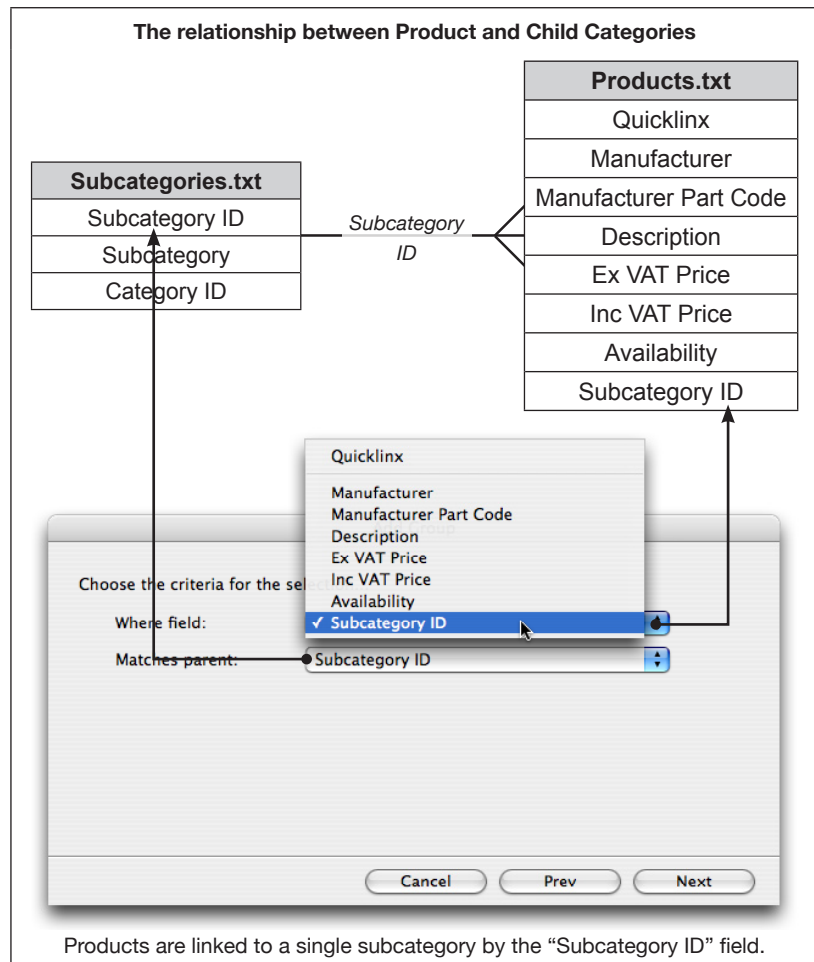
The **Add Group** dialog box is shown again, but with different content. The title bar is "Add Group". It has a "Group Name:" field with "Subcategories.txt" entered. Below it is a "Statement:" field containing the SQL query: `SELECT rowid, * FROM "Subcategories.txt" WHERE "Category ID" = {Category ID}`. At the bottom right is a "Check Syntax" button. At the bottom are buttons for "Cancel", "Prev", and "Done".

Click “Done” to complete the configuration of the “*Subcategory.txt*” group. You should now be able to click on any of the category groups and see a list of subcategories below it.



4 The final stage is to link product data underneath subcategories.

- Click one of the subcategory names to drop down a level in the hierarchy.
- Use the “Add Group...” button to add a new data source from the “products.txt” file.
- In this instance matching the “Subcategory ID” in the product data source to “Subcategory ID” in the parent:



The panel now provides the ability to navigate through the entire three tier hierarchy.



## Navigating through the grouping hierarchy

Category ID	Category
1	AV Accessories
2	Audio Separates
4	Batteries & Chargers
5	Books
6	Business and Finance
7	Cabinets & Accessories

Navigate into a group by clicking on the text of any of the fields

Subcategory ID	Subcategory	Category ID
3	AV Amplifiers & Receivers	2
4	Analogue Radio Tuners	2
5	Audio Hard Disk Recorders	2
6	CD Players/Recorders	2
7	Cassette Decks	2
8	DAB Radio Tuners	2

The navigation bar updates to show your current position in the hierarchy

Quicklink	Manufacturer	Manufacturer Part...	Description	Ex VAT Price
11SS	Sony	ST-SE370S	ST-SE370S Silver...	\$76.60
11T2	Sony	ST-SA3ESB	ST-SA3ESB Black...	\$237.00
11T4	Sony	ST-SA3ESN	ST-SA3ESN Gold ...	\$237.00

Click on any of the levels in the navigation bar to go directly to that group


Use the Home button to go to the top-level group

## PANEL EDIT MODE

Once configuration of the panel is complete, "Panel Edit Mode" should be turned off to prevent accidental adjustment of the configuration.

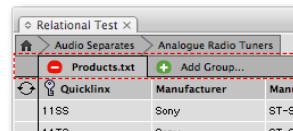
## Turning Panel Edit Mode off



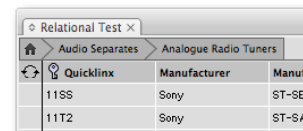
Press the  icon at the bottom of the panel to display the “Information” dialog.



“Panel Edit Mode” allows you to add new data to the Relational Panel. This option should be turned off to prevent accidental adjustment of the configuration.



Panel Edit Mode On




Panel Edit Mode Off

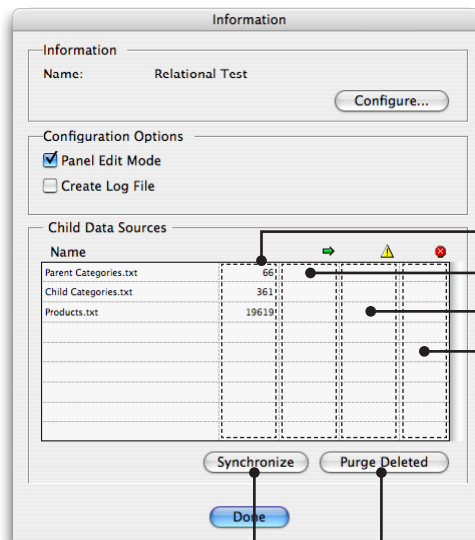


“Create Log File” is available for debugging the queries being executed and should be switched off during normal operation.

## EDITING RELATIONAL DATA SOURCES

To view general configuration information for a relational data source, press the  information icon at the bottom of the panel.

## Relational Data Source Information Dialog



Total number of records

➡ Number of inserted records

⚠ Number of updated records

✖ Number of deleted records

Retrieve the latest data from all child data sources

Remove all records that are shown as “deleted” in the panel

This dialog displays summary information for the relational data source and the child data sources added to it. Use the “*Synchronize*” button to retrieve the latest data for all child data sources.

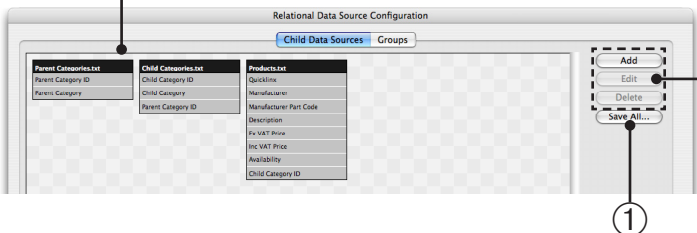
For more detailed information on the contents of the child data sources and the grouping configuration press the “*Configure*” button at the top of the dialog.

### Child Data Sources

The table configuration for each of the child data sources is shown on the “Child Data Sources” tab.

**Relational Data Source Configuration Dialog**

Each of the Relational data source's child data sources is shown here.



① Use the “Save All” button to create an SQLite database containing the child data source tables.

②

Button	Action
Add	Add a new child data source to the relational data source.
Edit	Edit the selected child data source. This button will present the configuration dialog for the selected data source.
Delete	Delete the selected child data source. Any groups that reference the data source will not be removed, but the queries used to create the group may become invalid due to the data source being removed.
Save As	Save all of the child data sources, and their content, to a single SQLite database file.



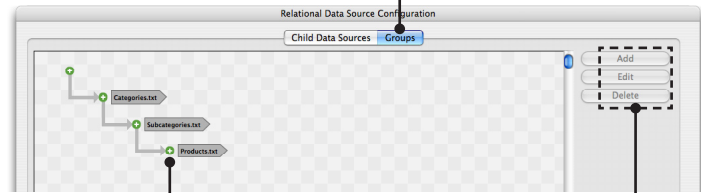
Always delete unused child data sources to keep synchronization times to a minimum.

### Groups

This dialog displays both the child data sources and the relationships between groups of data.

### Editing the Grouping Hierarchy

The grouping hierarchy is shown in this tab.

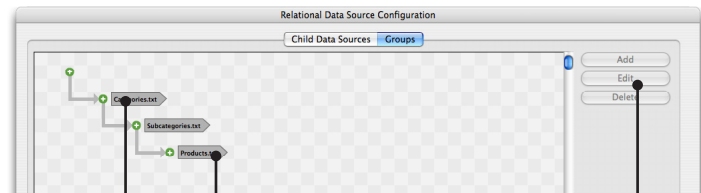


Click the + icon to add a child group

Add, Edit or Delete groups from the hierarchy. Deleting a group will also delete any sub-groups.

Groups can also be edited by double clicking the name in edit mode.

### Editing a group



Select a group and use the "Edit" button

Double click

Subcategory ID	Subcategory	Category ID
3	AV Amplifiers & Receivers	2
4	Analogue Radio Tuners	2
5	Audio Hard Disk Recorders	2
6	CD Players/Recorders	2
7	Cassette Decks	2
8	DAB Radio Tuners	2
9	DVD Audio Players/Recorders	2
10	HIFI Amplifiers & Receivers	2
12	Turntables	2

## REFINING THE DEMO

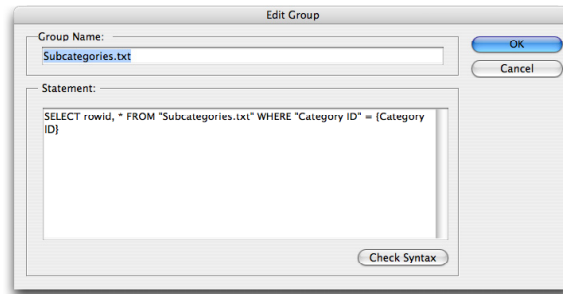
We'll refine the demo to prevent the display of any "ID" fields in the panel. First ensure the panel is in edit mode and click the home icon, then double click the "Categories.txt" group.

Alter the query to (ensuring correct use of quotes):

```
SELECT rowid, "Category" FROM "Categories.txt"
```

The same principal can be applied to the "Subcategory" statement:

```
SELECT rowid, "Subcategory" FROM "Subcategories.txt" WHERE  
"Category ID" = {Category ID}
```



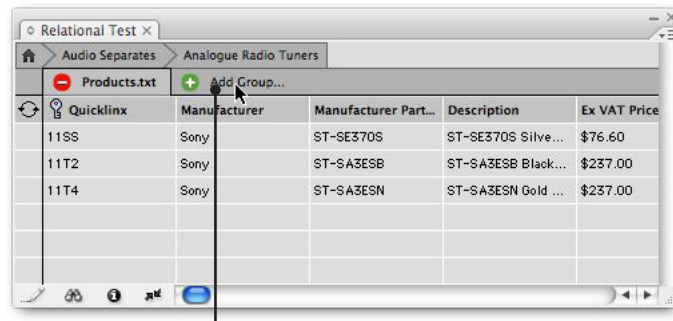
## MULTIPLE GROUPS

One of the most powerful features of the relational module is the ability to represent multiple groups of data alongside one another. A typical use would be a product that has multiple sets of data, such as price breaks, attributes, sub-components, etc.

For demonstration purposes we'll use the demo data to create a list of unique manufacturers under each "subcategory". This will be listed alongside products of a "subcategory".

### Multiple groups at the same level

In this example we'll add another group at the same level as "Products"



With any of the "Products" groups displayed, click the "Add Group" tab

### Multiple groups at the same level (continued)

Choose data for this group...

☐ Create a new data source

☐ Import an existing data source

☒ Use a data source already created or imported

Cancel Prev Next

We want to display additional information from the “products.txt” file/table, which has already been added as a child data source.

Choose the data source to select from:

Products.txt

Cancel Prev Next

Using the “Matching selection” option will generate a default SQL statement that we can subsequently modify to suit our requirements.

Select records to add to the group...

☐ All

☒ Matching selection

Choose the criteria for the selection...

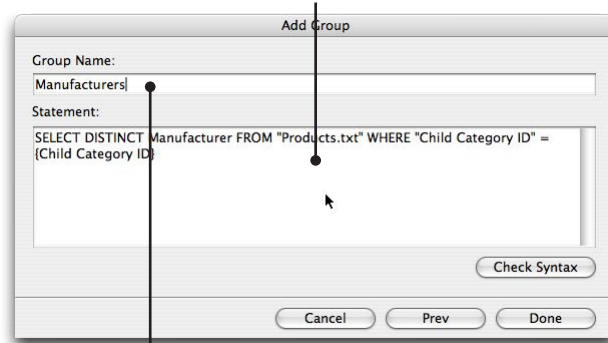
Where field: Subcategory ID

Matches parent: Subcategory ID

Cancel Prev Next

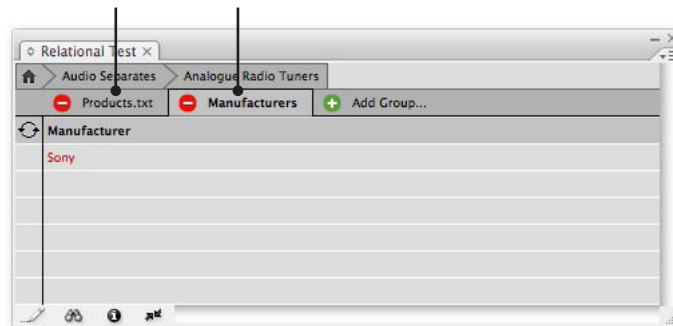
## Multiple groups at the same level (continued)

Modify the SQL to show a distinct (unique) list of all Manufacturers for the *Products* within each *Subcategory*:

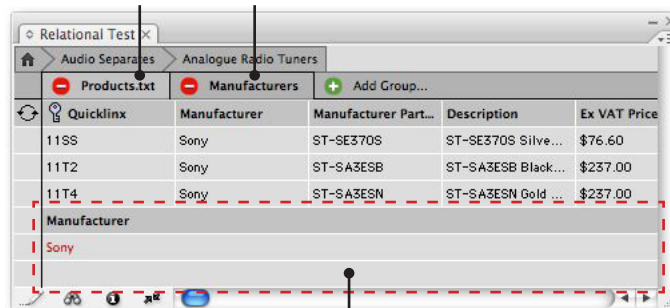


Give the group a new name

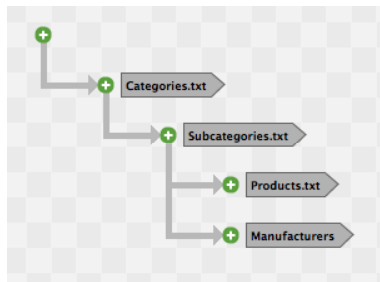
Both groups have the same “parent” and are shown at the same level in the grouping hierarchy. Switch between them by clicking on each tab.



Display both groups simultaneously by shift-clicking on each group tab



Multiple groups can be displayed at the same level in the hierarchy.



The panel now has the following grouping structure - the “*Child Categories.txt*” group has two sub-groups.

The panel's grouping configuration is shown on the Relational Data Source Configuration dialog.



Fields shown with red text in the panel can be inserted in the document but not updated. This will be the case where fields have been created as part of the SQL query and do not come directly from a child data source.

### LINKS

Linked content is identified by three components; Key field content, field name and the data source name. In relational panels the data source name contains both parent and child data source components. Using the previous example, a typical data source name would be “Relational Test/products.txt”.

In some instances (such as when using the “DISTINCT” keyword as shown above) there’s no way to link data from a query back to the source data. Links of this type can be paginated, but not updated. These links are shown in red when edit mode is enabled.

### ADOPTION

When adopting the child data source name remains unchanged. Only the parent data source component is altered.

### STORAGE

Relational data sources are stored in the EasyCatalog workspace folder. All configuration information is stored in the “Datasource.xml” file. This describes the child data sources, the links between groups and field definitions for any fields that are created on an ad-hoc basis in SQL.

Child data sources are stored inside the folder structure of a relational data source. Within the data folder there is a “Data Sources” folder in which they are contained.

### FURTHER ASSISTANCE

We'd be happy to configure a demo of the relational module. Ideally we need an extract from each of the data sources involved, a brief description or schematic of how they relate, and an example of the kind of output to be produced.

Using this data we can set a demo up which we send back. There is no obligation by using service, as we'd be happy to show you the benefits of using our software. Feel free to email us at:

[support@65bit.com](mailto:support@65bit.com)



## CHAPTER 5

# MENU REFERENCE

## THE RELATIONAL PANEL MENU

The menu structure of the relational module has been simplified from that of a standard EasyCatalog data panel.

Although the behaviour of the majority of options remains the same, there are some notable exceptions:

### Synchronize with Data Source

Selecting this option synchronizes all child data sources. A report will be offered should an error occur or the process is cancelled.

### Show/Hide Fields

### Field Formatting Tags

### Field Specifiers

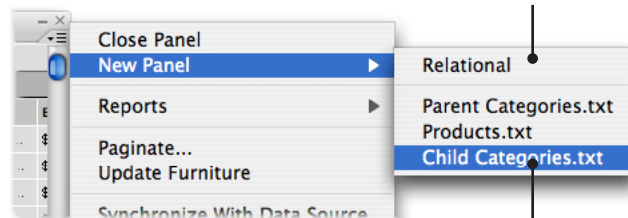
All menu options listing fields display only those that relevant to the data currently displayed.

### New Panel

The 'New Panel' menu has additional options to allow a copy of the relational panel to be created or a panel for any of the child data sources.

#### The 'New Panel' menu option

Create a new relational panel. All relational panels share the same grouping configuration.

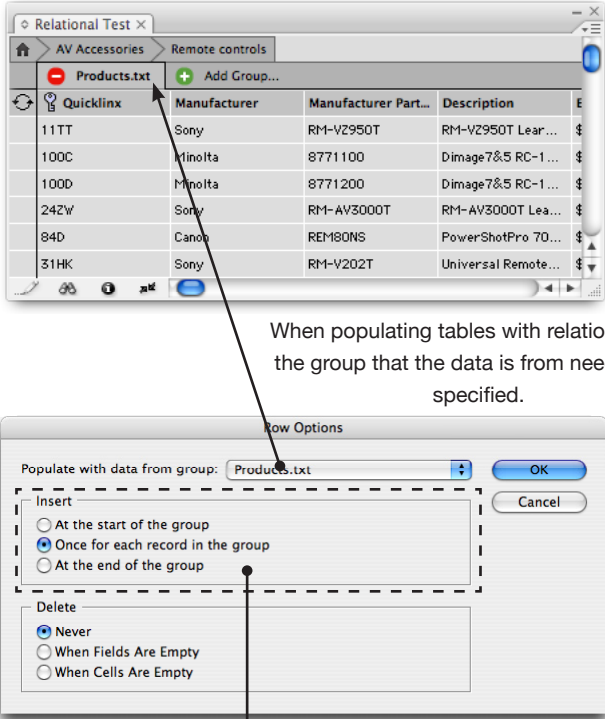


Create a new panel containing the data for one of the child data sources. This panel will behave in the same way as a non-relational data source panel.

Table → Row Options

The *Row Options* dialog has been modified to specify which group will be used to populate the row.

**The Row Options Dialog**

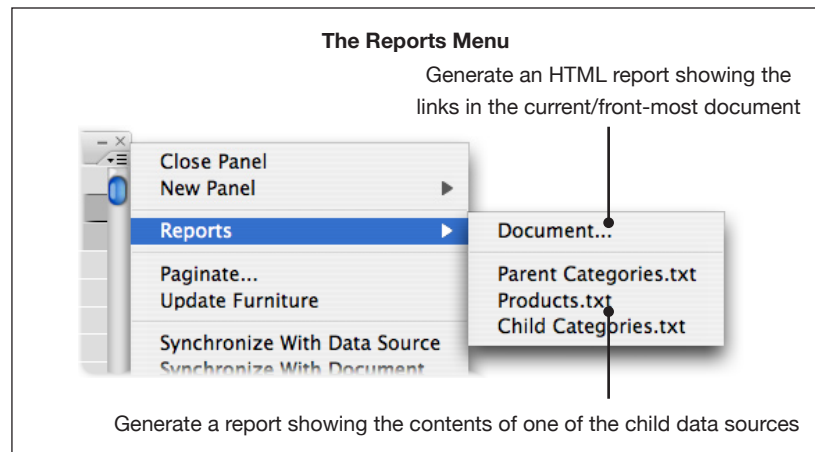


When populating tables with relational data, the group that the data is from needs to be specified.

Insert...	The table row will be inserted...
At the start of the group	At the beginning of the group, before any of the 'record' rows. The row will behave like a header row for the group, and will be populated with data from the parent record. In this example, the row will be populated with data for the 'Remote Controls' record in the parent group.
Once for each record in the group	Once for each record in the group.
At the end of the group	At the end of the group, after the 'record' rows. There row will be populated with data from the parent record.

This allows Product Styles to contain multiple tables, each of which is populated with data from different group. It also enables tables to be populated with data from multiple groups.

## Reports



The 'Panel' report is not currently available for Relational panels.

## Adopt Fields

Field adoption will only adopt links on the active document layer that were created with a relational data source. The names and configuration of the child data sources must be the same as those used to create the links.