The Helix Workbench

The Helix Workbench is an app for Helix database designers and has two purposes.

1. It contains a set of tools that speed up repetitive tasks and make them error-proof.
2. It contains a set of tools that implement design techniques.

Feedback

If you have questions about using The Helix Workbench or suggestions for additional tools, please write to **chuck@clhwares.com**.

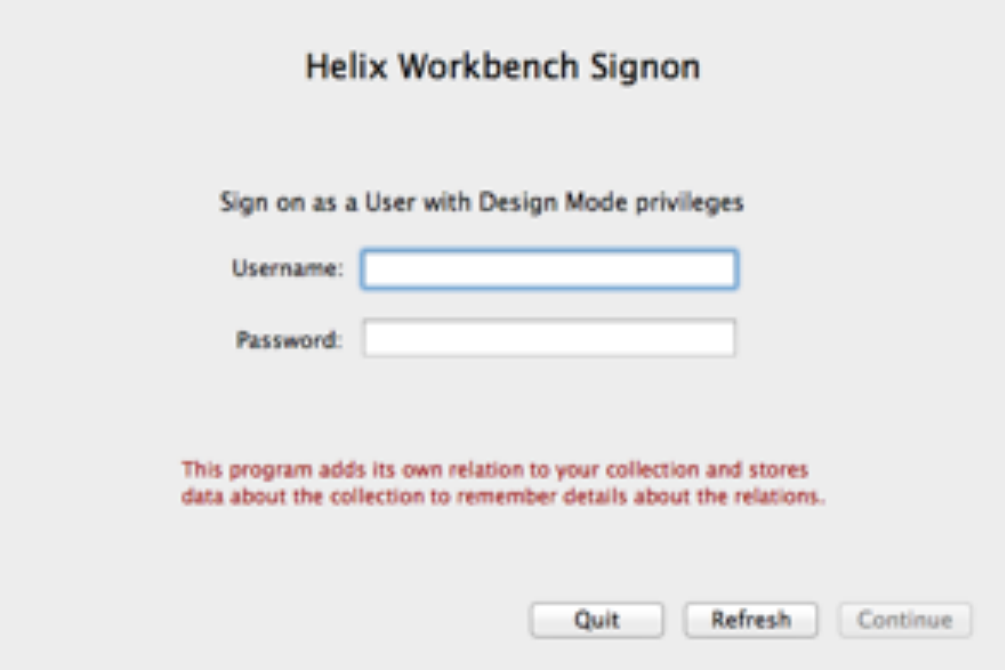
Starting Helix Workbench

Installation note:
When you run the Helix Workbench on a Macintosh running OS X with the default Gatekeeper settings enabled, Gatekeeper will not permit the Workbench to start the first time you run it. Gatekeeper is included with OS X Mountain Lion (Version 10.8) and later. Use the following procedure to run Helix Workbench for the first time. In the Finder, open the Applications folder (or wherever you store Helix Workbench) and right-click the Helix Workbench icon. From the shortcut menu, choose *Open*.

1. OS X will display a confirmation dialog box. Click *Open* to confirm that you want to start Helix Workbench.
2. When requested, enter your Mac OS X admin user name and password to complete the installation and to start the Helix Workbench.

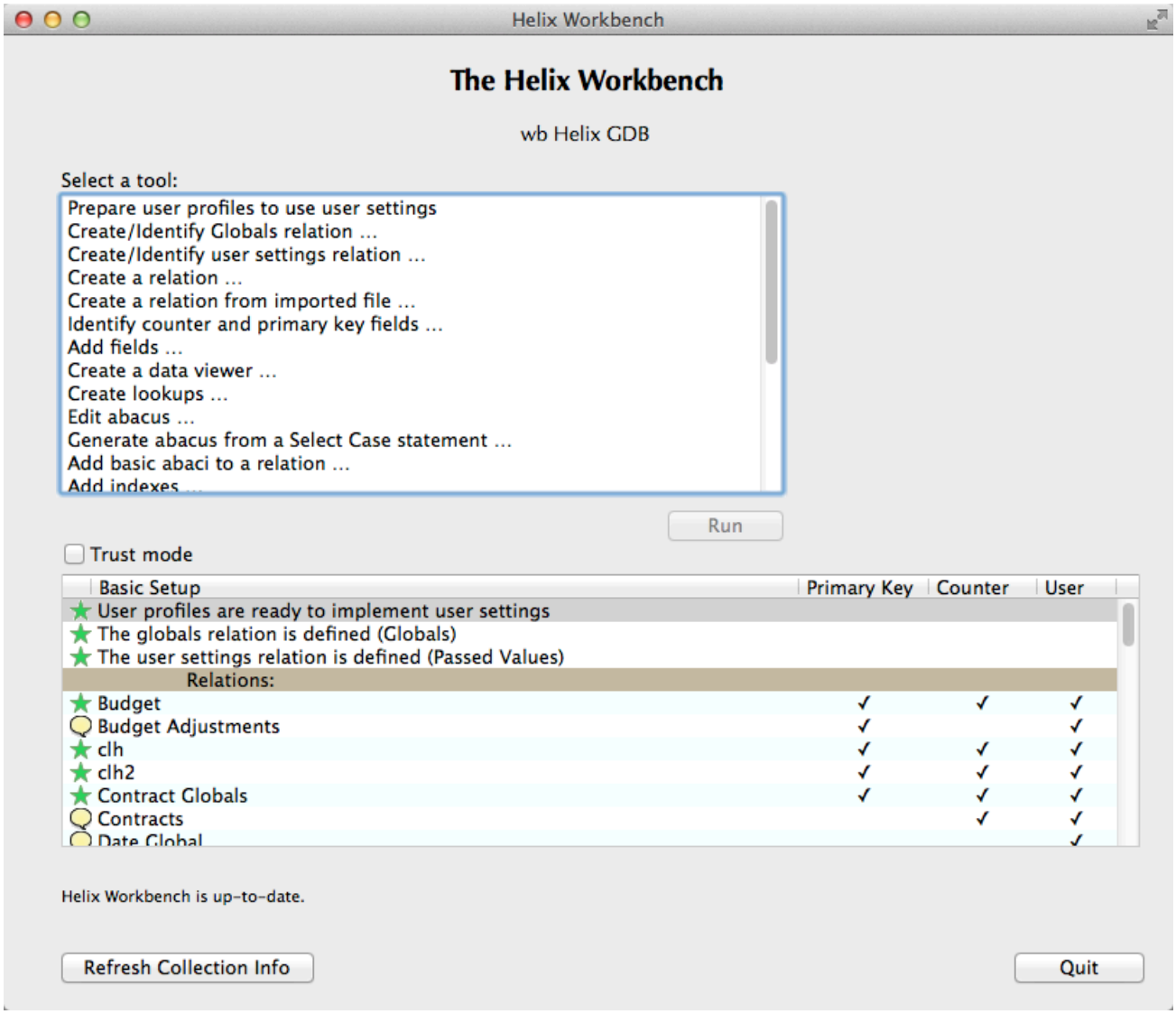
(This procedure is needed only the first time you start the Helix Workbench with Gatekeeper enabled. When you start the app again, double-click the application icon to start normally.)

To use the app, you must sign on with a Username that has Design access for the open collection to edit. That's because the Helix Workbench creates Helix objects for you. *It also creates its own relation in the collection that it uses to remember what it learns about the collection.* You can use the Helix Workbench to update your open Helix collection, even if you are not in Design Mode or if you are logged on to the collection as a different user.



The Dashboard

To use a tool, select the tool from the list, and press the *Run* button. If the tool is dimmed, the *Run* button won't be enabled when you select that tool (it means that this version of The Helix Workbench has not released that tool yet).



The bottom half of the main screen is the dashboard for the state of the database. The Helix Workbench analyzes the database and reports on various conditions.

1. To implement user settings, no user profile may permit more than one person to log on as that user simultaneously. The first test indicates if any User permits multiple logins at the same time.
2. The Globals relation test is whether a Globals relation has been set up and indicated. A Globals relation is used to store settings shared by all users.
3. The final test is whether a user settings relation is set up and defined in the workbench. A user settings relation is used to pass data among views.

4. The lower part of the workbench's scorecard lists each relation in the collection. If the workbench knows that the relation has primary keys defined, a counter, and a UserName abacus, then it will display a green star in front of the relation name to let you know that all of these items are defined. If any of those items are missing, a yellow conversation bubble appears. Hover over the bubble to see which items the Helix Workbench thinks are missing. Green arrows will also appear in front of the tools that you can use to create or point to those "missing" elements.

When you select a tool from the list, a more detailed description appears next to the menu.The Helix Workbench contains several useful tools.

- Prepare user profiles to enable user settings
- Create/Identify Globals relation
- Create/Identify user settings relation
- Create a relation
- Create a relation from imported file
- Identify counter and primary key fields
- Add fields
- Create a data viewer
- Create lookups
- Edit abacus
- Generate abacus from a Select Case statement
- Add basic abaci to a relation
- Add indexes
- Make quick entry form
- Make quick report
- Create shortcut popup
- Edit color palette
- Mass font changes
- Multi-user menu updates

It's possible to confuse the Workbench by making changes that it's not aware of. **If you have troubles with any tool, the first attempt to troubleshoot is to refresh the collection information by selecting the *Refresh Collection Info* button.** The workbench will keep itself up-to-date by refreshing the dashboard **unless you select *Trust mode***. When the *Trust mode* option is selected, the dashboard will refresh less frequently. Select the *Refresh Collection Info* button to force the dashboard to update.

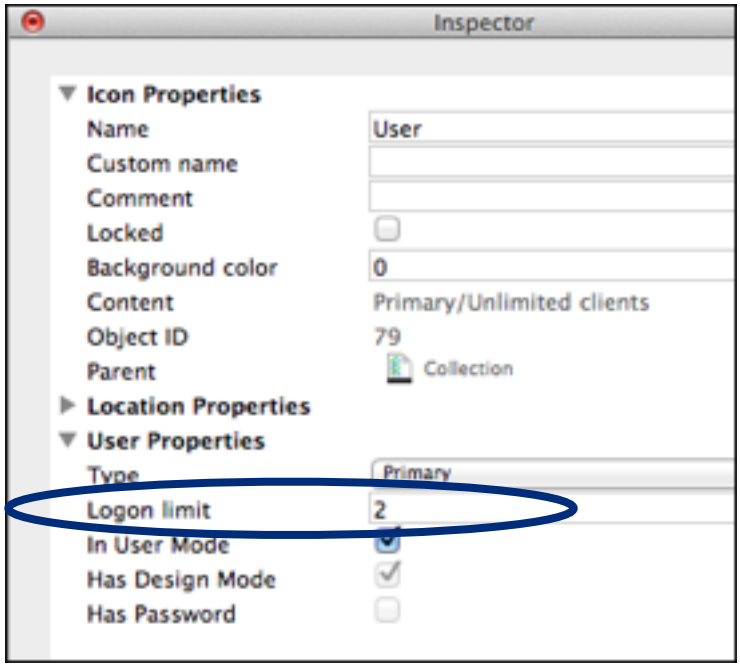
Prepare user profiles to use user settings

This tool sets the logon limit for all users in this collection to one.

When users are set up in Helix, you can specify how many Helix clients can sign on to the database as this user at the same time. This number can be anywhere from 0 to infinity.

However, to implement user settings and many other useful design techniques, no more than one person may be logged on as the same user at the same time.

It can be a chore to go back through all of the users and change all of the Logon limits to one. Select this tool and press the *Run* button to automatically set all of the user profiles to have a logon limit of one.

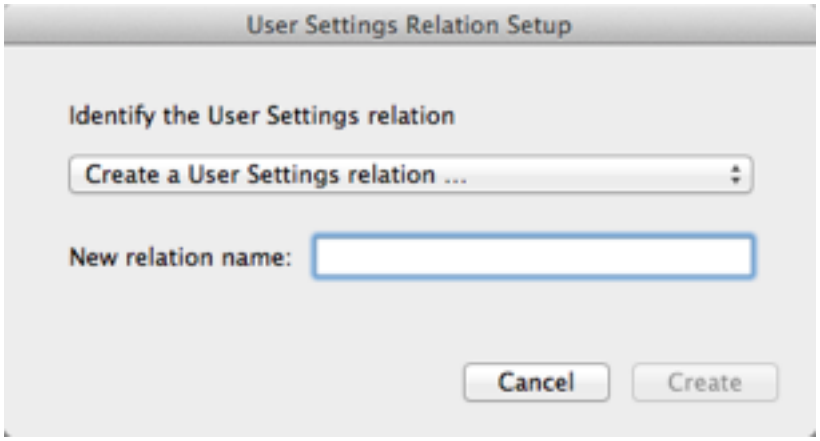


Create/Identify Globals relation

This tool sets up a Globals relation. A Globals relation is used to store information/defaults that are shared by all users. Examples might be the company address, the company logo, the corporate tax discount rate to use for economic calculations, and the current business year.

If you already have a relation in use in your collection that's set up as the Globals relation, then use this tool to designate that relation to the Helix Workbench. You can also use this tool to generate a new relation that will act as the Globals relation. If you create a new relation as the Globals relation, a *True* abacus will be created in the relation and an index created for the abacus.

Then, to use the Globals relation, you will add fields with values to share. Other relations will use these values in their own abaci and templates by using the *Lookup* tile (and matching the *True* abacus where the value = *True)*. You can also use the Helix Workbench tool *Create Lookups* to create abaci that retrieve the values stored in the Globals relation.

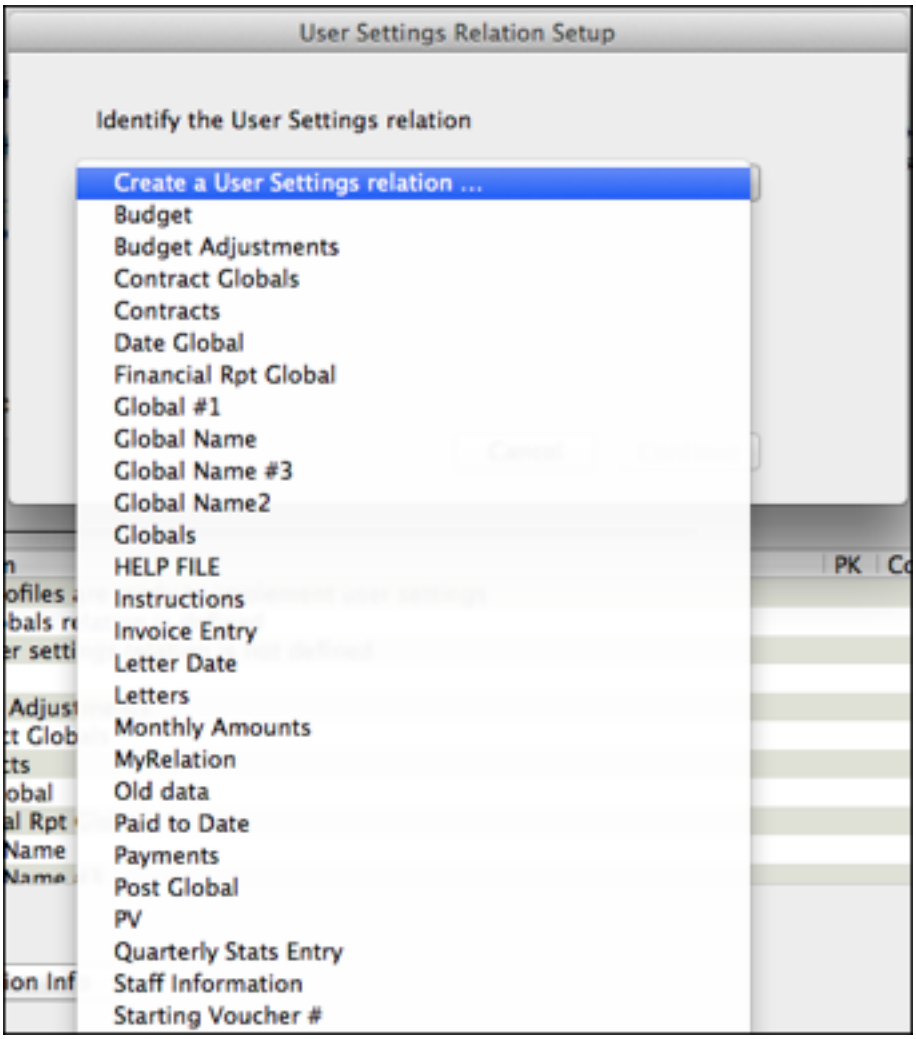


Create/Identify User Settings Relation

This tool sets up a user settings relation. The user settings relation's purpose to store information/defaults that are particular to a specific user. Examples might be the pricing authority that this user has or the invoice number that this user is editing. This relation is also used in several design techniques that rely on sequences that open a series of views and pass values from one view to the next.

If you already have a relation in use in your collection as the User Settings relation, then use this tool to indicate to the Helix Workbench which relation is the User Settings relation. You can also use this tool to generate a new relation that will act as the User settings relation. If you generate a new relation, a *User* field will be created in the relation and an index for the abacus.

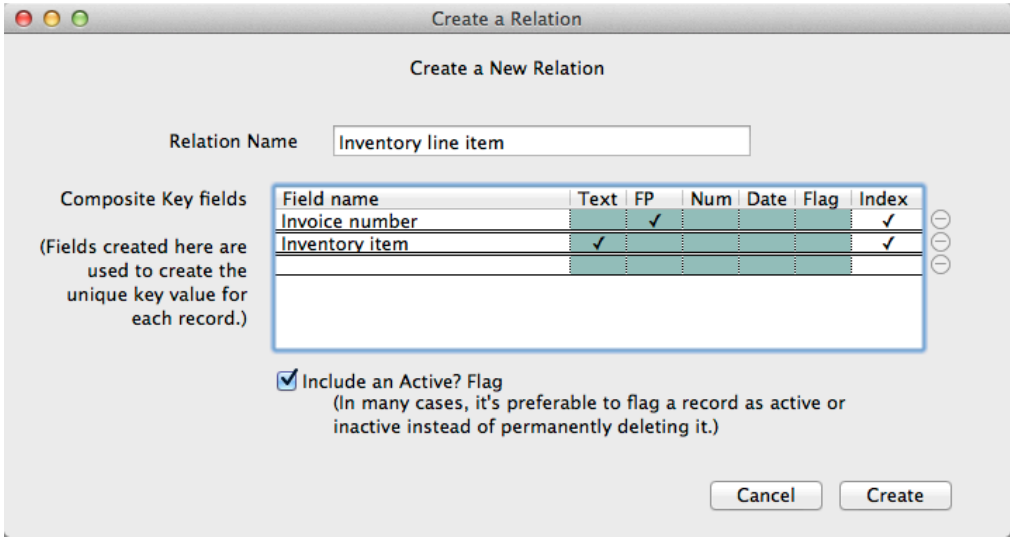
To apply a user settings relation, you will add fields with values to share in it. Other relations will use these values in their own abaci and templates by using the *Lookup* tile (matching the *User* field with an abacus that contains the *Username* tile). You can also use the Helix Workbench tool *Create Lookups* to create abaci that retrieve the values stored in the User settings relation.



Create a Relation

There's lots of work to include when you create a new relation.

- I recommend each relation contains a counter which is used as a physical key. That means that every record will have a unique value in this field. A common method is to make it a fixed point field and to add 1 to the largest value every time a new record is added to the relation. The Helix Workbench does this for you.



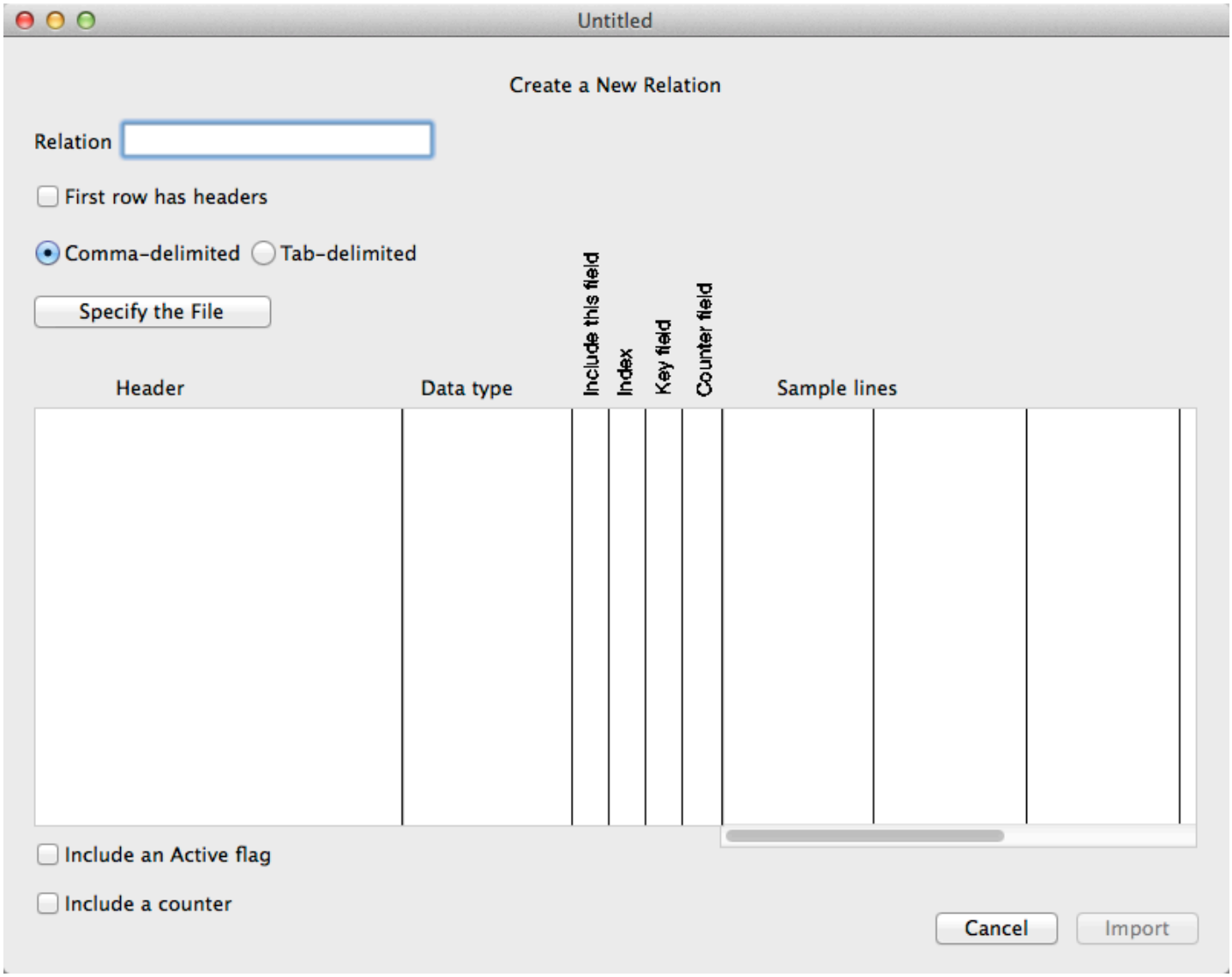
- To set up a normal relational database, create fields that act as a "composite key." That means that the combination of values for these fields must be unique for every record in the relation. For example, invoice numbers are unique, but every line item on an invoice has the same invoice number. The combination of an invoice number and an inventory item, in this example, are unique.
- In many cases, you never want to permanently delete information from your database. This tool gives you an option to create a flag field named *Active?* that you can use to designate if this record is still an active record or not (think of active vs. inactive customers or inventory items).
- When you select the *Create* button, the tool:
  - creates a new relation with the you that you supply;
  - creates a field, using the selected datatype, for each line;
  - creates an index for each field*;
  - creates an abacus as the chain of all of the named fields;
  - creates a validation to ensure that the combination of fields is unique;
  - assigns the validation to the last field in the list;
  - creates an index field;
  - creates standard abaci: (UserName, constant True, constant 0, constant 1, calculation for the next ID); and,
  - creates a post named *clhAssign* that assigns the next counter# to the counter field when new records are added (this post must be attached to any entry forms that you create if you want to set the counter).

Note that this tool only creates the relation and the composite key fields. The *Add Fields* tool (to be used next) adds additional fields.

* - note that you can suppress an automatic index for any field if you're concerned about taking up too much space or taking too long to update too many indexes.

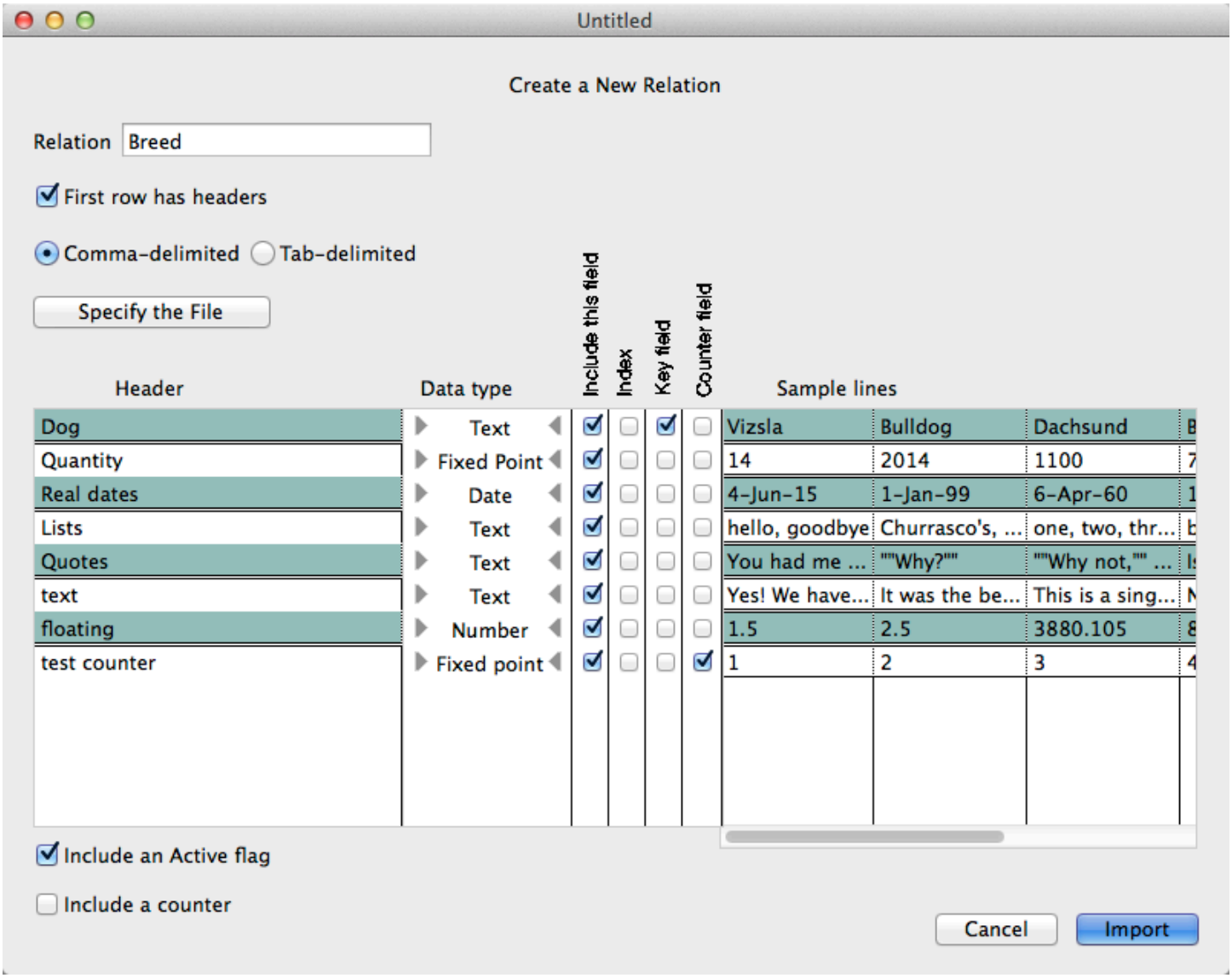
Create a Relation from an Imported File

This tool builds a new relation and populates it with data from a file. You can either use a CSV (comma separated values) formatted file or a tab-delimited file as the source. Both of these are options you can use in Excel to store your data.



Tell the Helix Workbench what the new relation's name is, specify whether the first row of the file has field names in it or not, and mark it as either a comma-delimited or a tab-delimited file. Then select the *Specify the File* button. The Workbench will display the first five lines of the file as samples. If the first row has headers, it will also populate the header names.

While the preview of the import is displayed, you can change the Field names in the Header. In the example below, 8 fields will be created: Dog, Quantity, Real dates, Lists, Quotes, text, floating, and test counter. The default data type is Text. You can change it to fixed point, number, date, or True/False. If there are any fields in the datafile that you want to skip and not store them as fields, uncheck the *Include this field* checkbox. Check any of the fields that you want an index to be created for. Check the fields that will act as the key fields. That means that the combination of those fields are unique for each record; no two records will ever



have the same values for all of their key fields. If one of the fields in the data file is a counter, then designate it with a checkmark. If none of these fields are counter fields, you can mark the checkbox to *Include a counter*, and then the Workbench will add a field and populate it as the counter. The last option is whether you want to add a True/False field to act as an active flag. When you import the records, every record will be imported as *Active*. Finally, select the *Import* button to being the creation of all of the elements in the new relation and to import the data from the data file into your new relation.

If there is an error while importing records (for example, if a format in the file doesn't match the data type you've given to the field), then the import will abort with an error, and no records past the record that caused the error will be imported.

Note: the Workbench will only recognize one field as the counter.

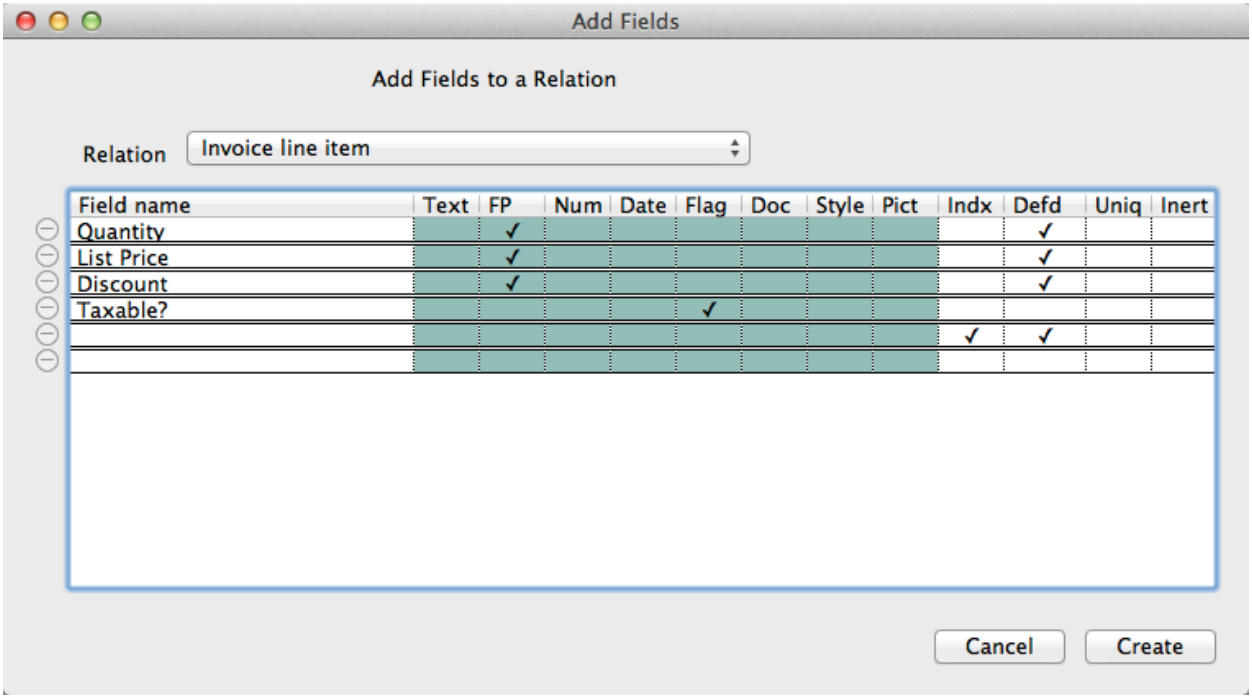
Designate Counter and Primary Key Fields

This tool has two purposes. If an existing relation already has fields that are being used as primary keys and counters, you can designate them here for the dashboard. When you designate an existing field as the primary key, it means that the value for the field is unique to this record; no other record in the relation can have that same value. (If there's no single record that has a unique value for the record, the you can select multiple fields that, together, have a unique combination of values. That set of fields is referred to as a composite key.)
To help enforce the uniqueness of the primary keys, you can have a validation added to the fields to require the keys to be unique. If you select this option, the option to create an index for the primary key will also automatically be selected. This is a default because it keeps your collection from slowing down as it grows. You have the option to deselect this option, however.

If your relation doesn't have a counter field, you can have the workbench create one. It will select a name for the field, but you can override it. When you add a new counter field, you can also auto-populate it. That means if your relation has 5200 records in it, this counter field will be given the values 1 to 5200. The index is used if you have a preference for how to sort the records as you assign numbers to them. When you create a Counter field, the workbench also creates a post called *clhAssign*. Attach this post to all of your entry forms, and then the counter will be given the next value whenever any new records are entered.

Add Fields

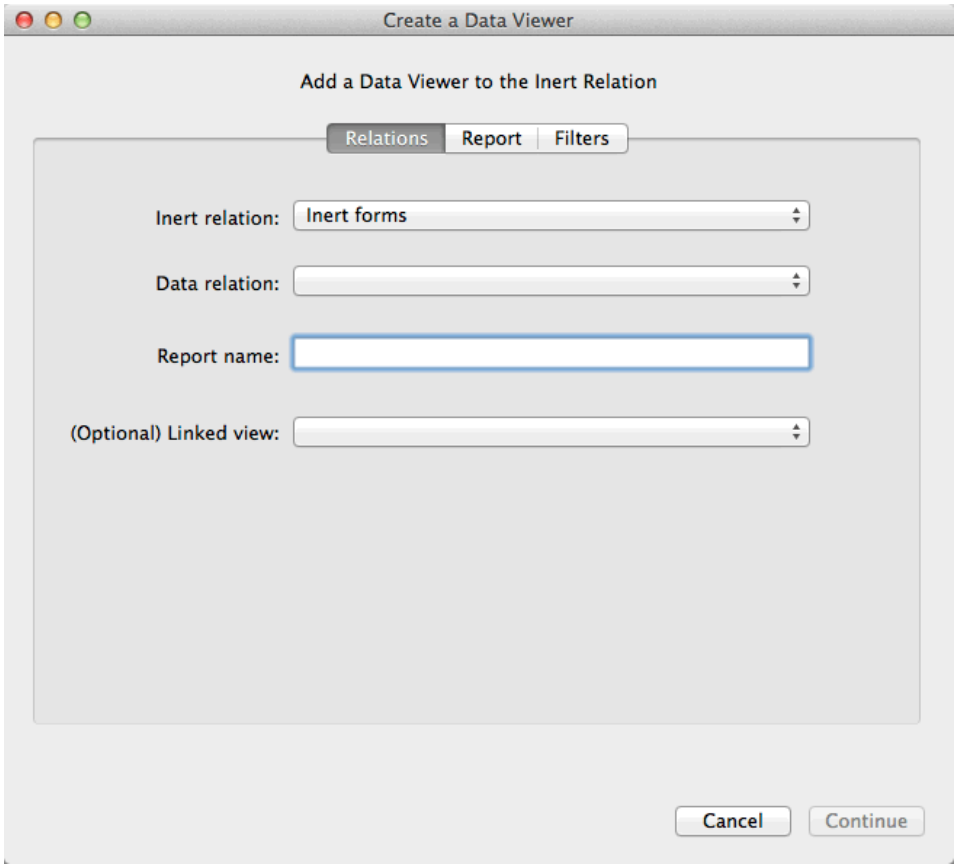
The Helix Workbench provides a tool to speed up adding fields (and supporting objects) to an existing relation. You can name multiple fields that you want to add to your relation. For each field, specify its datatype (text, fixed point, number, date, flag, document, styled text, or picture), whether you want an index created for this field, whether this field must have a value when a record is entered ("Defined"), whether it must have a unique value, and whether the field is inert. Note that if you specify that a field is unique, then the Workbench will schedule that field to automatically be indexed (you can override the setting, but your database performance will suffer). Note also that inert fields cannot be indexed and cannot be set up to require a unique value.



Create a Data Viewer

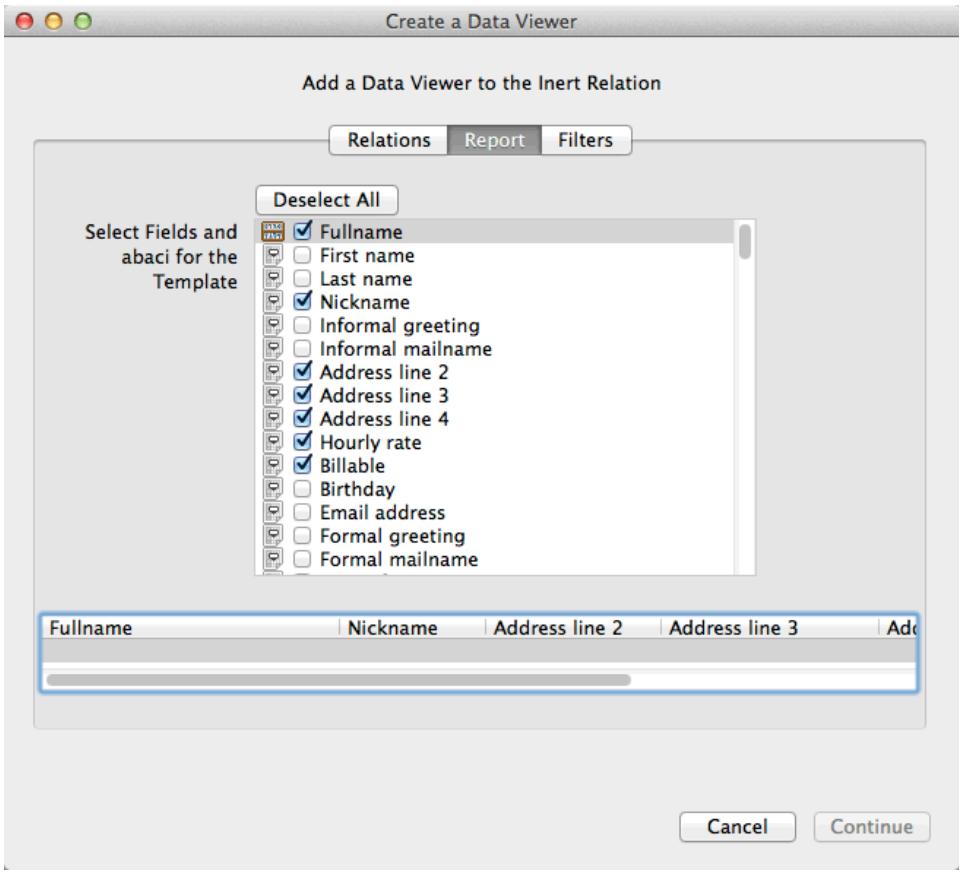
This tool provides a way to create and customize a quick report and attach live filters to it. By default, the report is created in the *Inert Forms* relation. If your collection doesn't have this relation, the Helix Workbench will offer to create it for you. Otherwise, you will be prompted which relation acts as your relation that only contains inert fields.

On the first tab, identify the relation that has the data that you want to display. Give the report a name. Also, if you want to be able to double-click a line of the report and examine the record in an entry view, then choose that view from the popup list. The report (the list view) will be created in the inert relation.



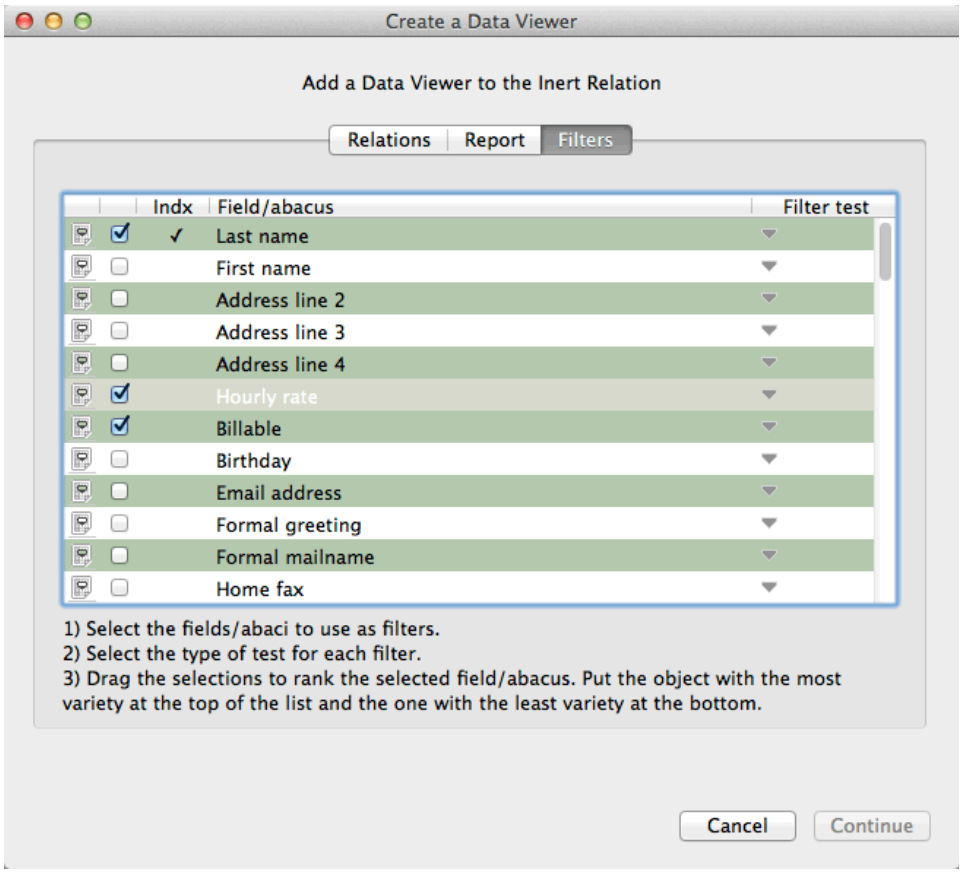
On the Report tab, select the fields and abaci that you want to include in the report. Drag the objects up and down in the list to change the column order for them to display in the report.

As you select objects, a sample report line displays at the bottom of the tab. You can resize the columns on this sample to change the size of the columns in the view that will be created.



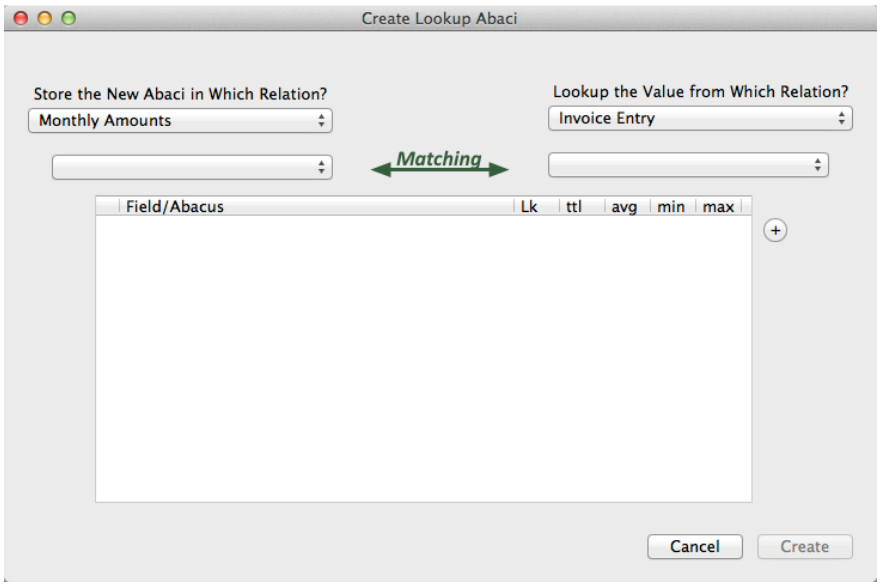
On the Filters tab, select the fields and abaci to be used as filters for the report. The table indicates whether the object is a field or an abacus and whether the object is indexed. (Try to use indexed objects as filters in order to obtain the fastest results.) For each filter that you select, choose the filter test from the popup menu. The options are: =, ≠, <, ≤, >, ≥, < • <, ≤ • ≤, and Defined/Undefined.

You can also drag the filters up and down the list to improve the speed of your search. Place the filter at the top that will have the greatest variety of values. For instance, in a list of clients, if you have fields for company name, country, state, and city, the order of your filters should probably be: Company name, City, State, Country. You would expect to have the most different country names in the relation. You can have multiple clients in a city, so the city would have the 2nd most unique values, followed by state, and by country.



Create Lookups

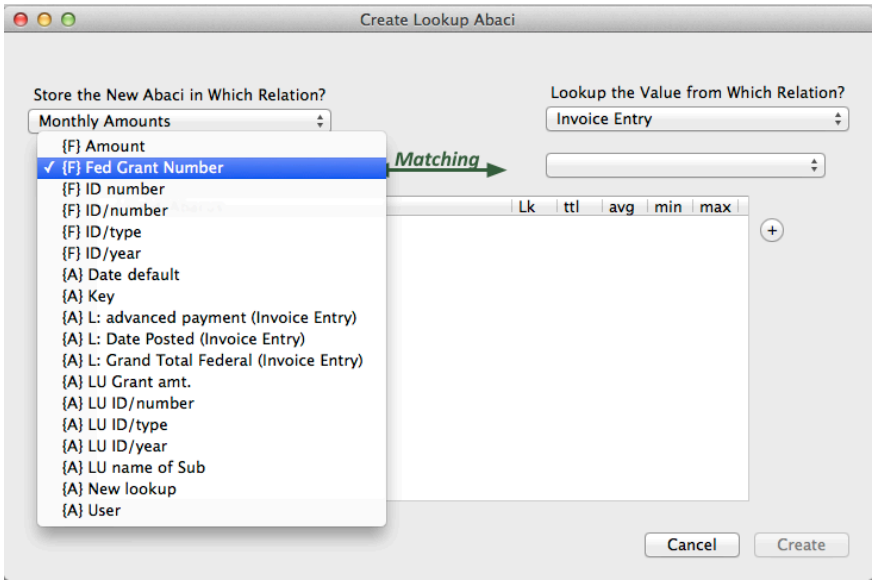
Lookups are the power behind relational databases. Normally, an abacus can only use fields and calculations from within that relation. The lookup tiles, however, let us use data from one relation in the abaci of a different relation. Examples would be to lookup the total purchases (from the *Purchases* relation) for a client (in the *Clients* relation) or the ability to print an entire mailing label (from the *Clients* relation) on an invoice (in the *Purchases* relation).



If two relations have a relationship where a Lookup makes sense, then it's often the case that you want to create multiple lookups. This tool makes it fast to create a bunch of these lookup abaci all at once.

Step 1. Select the target relation and the source relation from a list of relations. The source relation is where the data already is. The target relation is where you're going to save the abacus that gets the value from the source relation and lets another formula or template use it

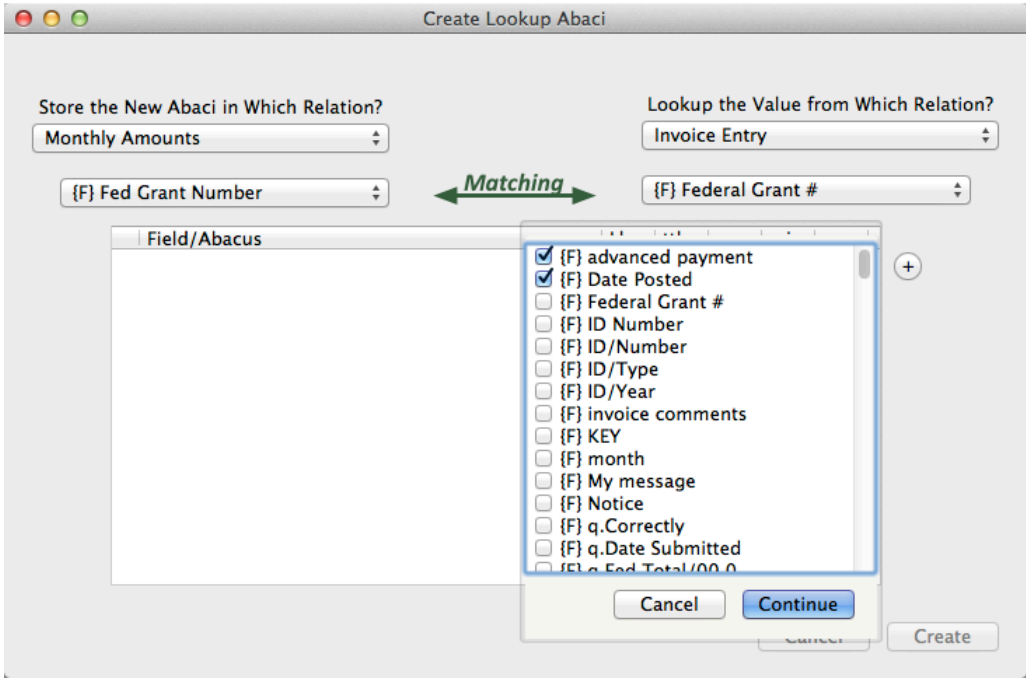
Step 2. Tell Helix how to find the record in the source relation that has the information needed by the target relation. For instance, if the source relation is the *Invoice Entry* relation, then the matching item is going to be the *Invoice#* field. In other words, identify the field or abacus will let you find just the record you're looking for. You'll also select the field or abacus in the target relation that has the value in it that will match the field/abacus in the source relation.

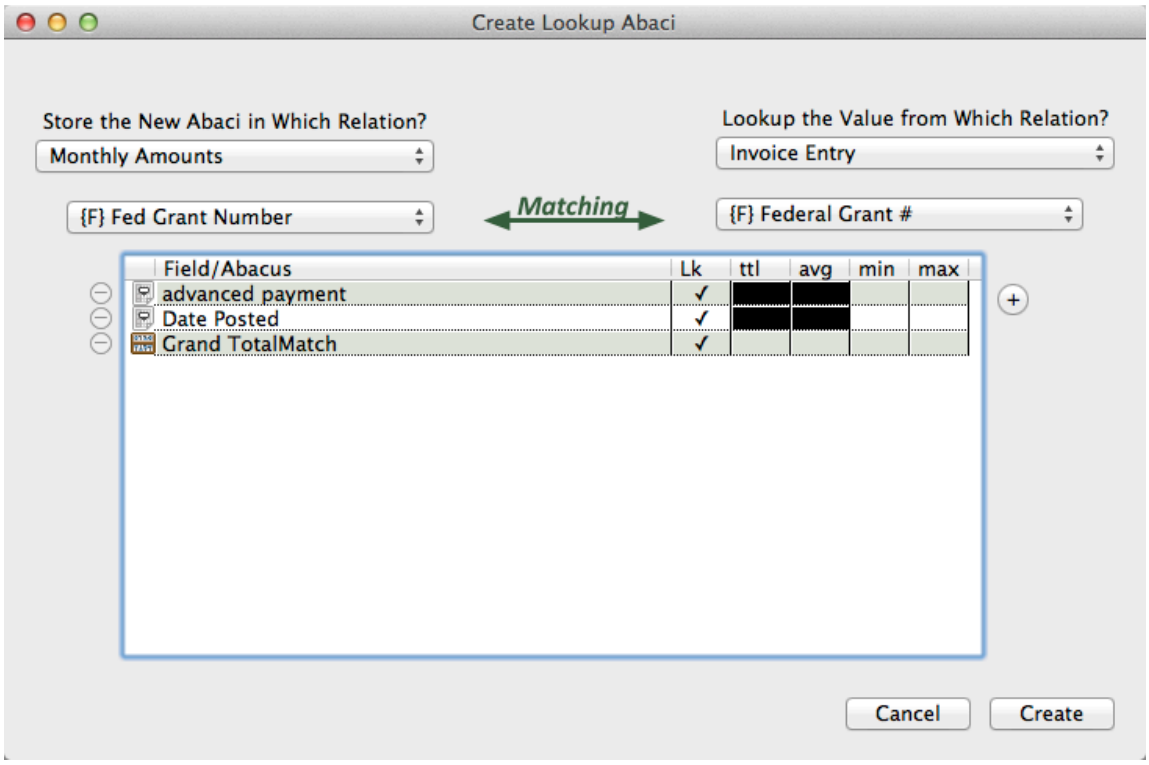


The {F} and {A} before the names indicate whether the object you can select is a field or an abacus. The source relation list will only show objects that (1) have the same datatype as the target relation's field/abacus, and (2) have an index where that field/abacus is the first item in the index. Efficient Lookups are based on indexed fields and abaci; the Helix Workbench enforces this practice.

Step 3. Press the ⊕ to show the list of fields and abaci in the source relation that can be used in the Lookup. Select all of the objects that you want to build a Lookup for and press the *Continue* button.

Step 4. You can select multiple lookup abaci to create for each of the items you select.
Lk = Lookup
ttl = Sub-total
avg = Sub-average
min = Sub-minimum
max = Sub-maximum
By default, this tool will create an abacus with a *Lookup* tile for each of the objects you select. Click in the boxes to select and deselect the abaci to create. Some of the fields are not permitted because the result is meaningless. You can press the ⊕ to add more objects, and you can press the ⊖ to remove an object.





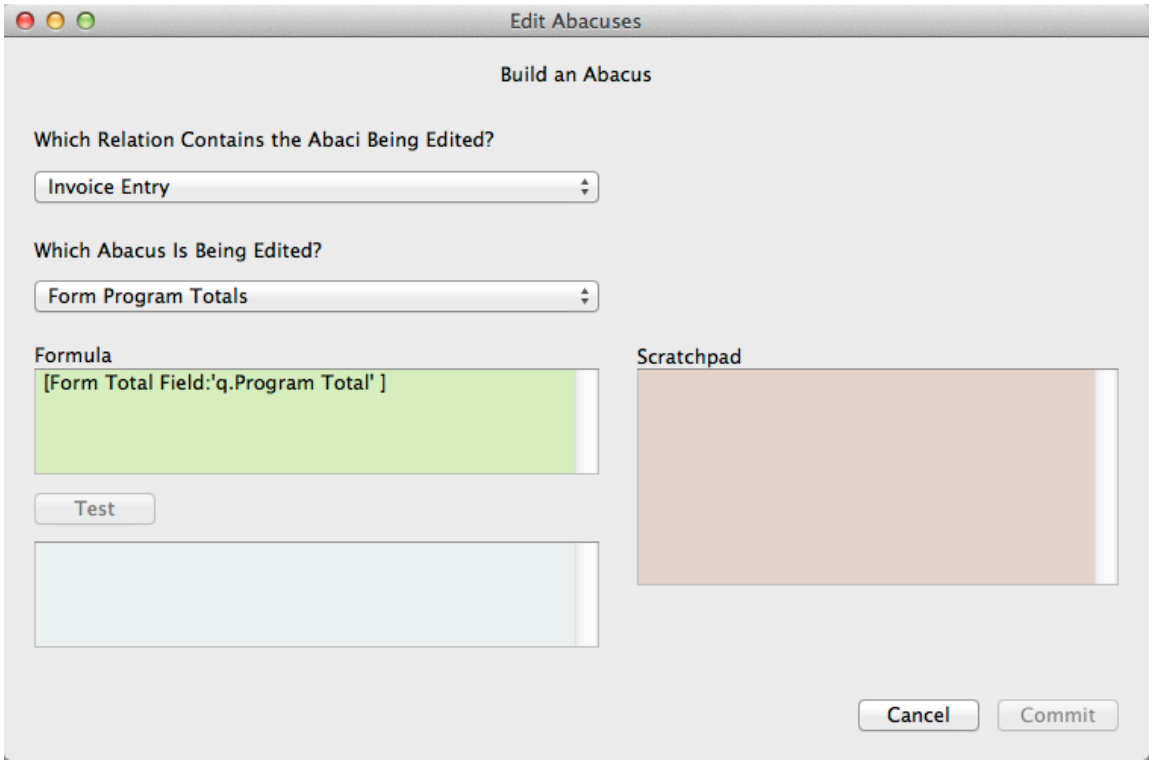
Edit Abacus

One of the coolest features of Helix RADE is the ability to enter formulas. Sure, it's fun to drag tiles around, but when you have an existing abacus and you just want to make some simple edits or use a piece of a formula in a similar abacus, then it's real convenient and fast to be able copy and paste the formula.

This tool gives you a large area to type your formula and a button that previews how the formula will be saved to the abacus. It also contains a scratchpad area where you can paste snippets of code and copy back to the formula you're working on.

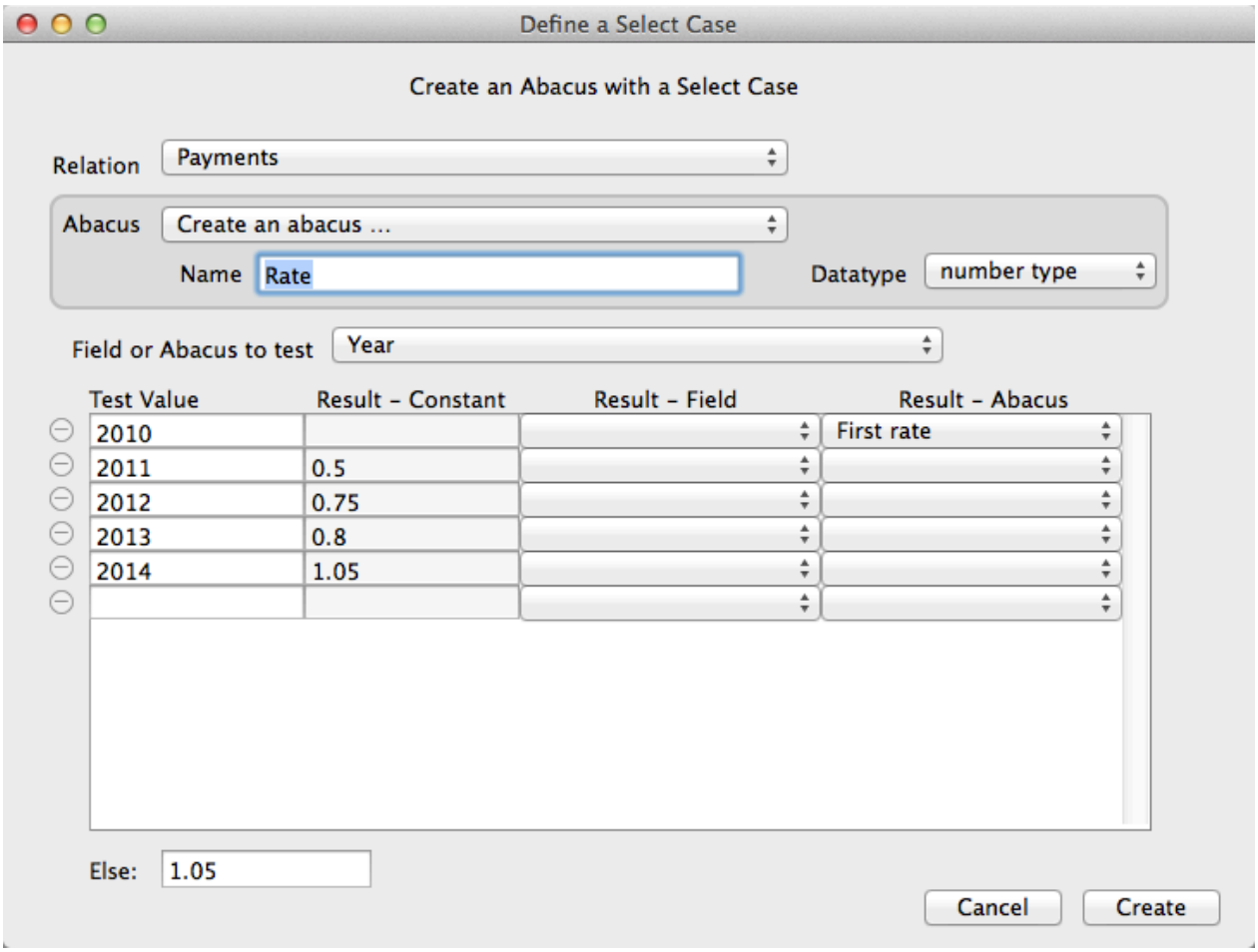
When you select a relation from the first popup, the next popup will list all of the abaci in the relation. When you select an abacus from that list, its formula is displayed in the formula area. When you're done making your edits to this formula, press the *Test* button. The way that Helix sees your formula appears in the box below the formula area. The *Save* button only becomes enabled after you test your formula. If you're satisfied with the formula, you can press *Commit*, and the tool will replace the formula in the abacus.

You can also select an option to create a new abacus from the list of abaci. In that case, a field appears where you can name the abacus, and you can enter its formula, test it, and save it.



Generate Abacus from a Select Case Statement

What originally attracted me to Helix was its use of tiles to build abaci. Sometimes, it can get tedious dragging IF tiles to create an abacus that has a series of tests to determine the value to use. This tool creates an abacus from the values you enter into a table.



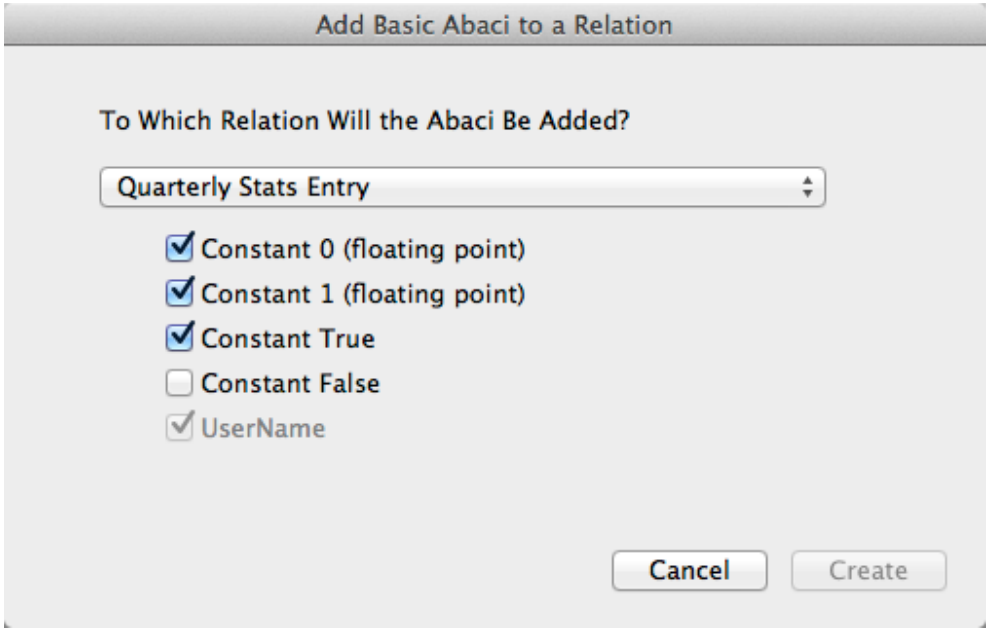
This case creates an abacus named Rate as a number.

If the year = 2010, then Rate = the abacus First rate.
If the year = 2011, then Rate = .5
If the year = 2012, then Rate = .75
If the year = 2013, then Rate = .8
If the year = 2014, then Rate = 1.05
If year = anything else, then Rate = 1.05

Add Basic Abaci to a relation

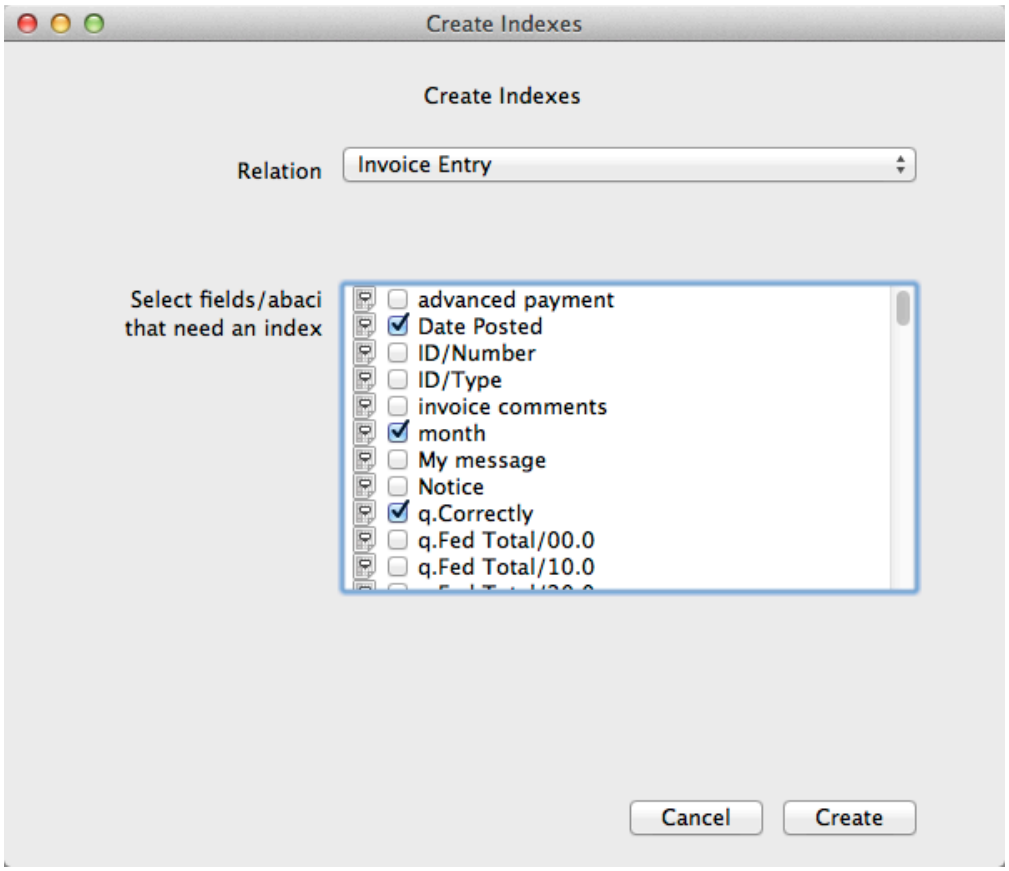
There are certain abaci that I recommend be included in every relation because they are used so often. This tool can add a True constant, a 0 constant, a 1 constant, and a UserName tile to the chosen relation.

You can turn off which abaci will be created. If an abacus already exists, the tool will not let you create a second copy of it.



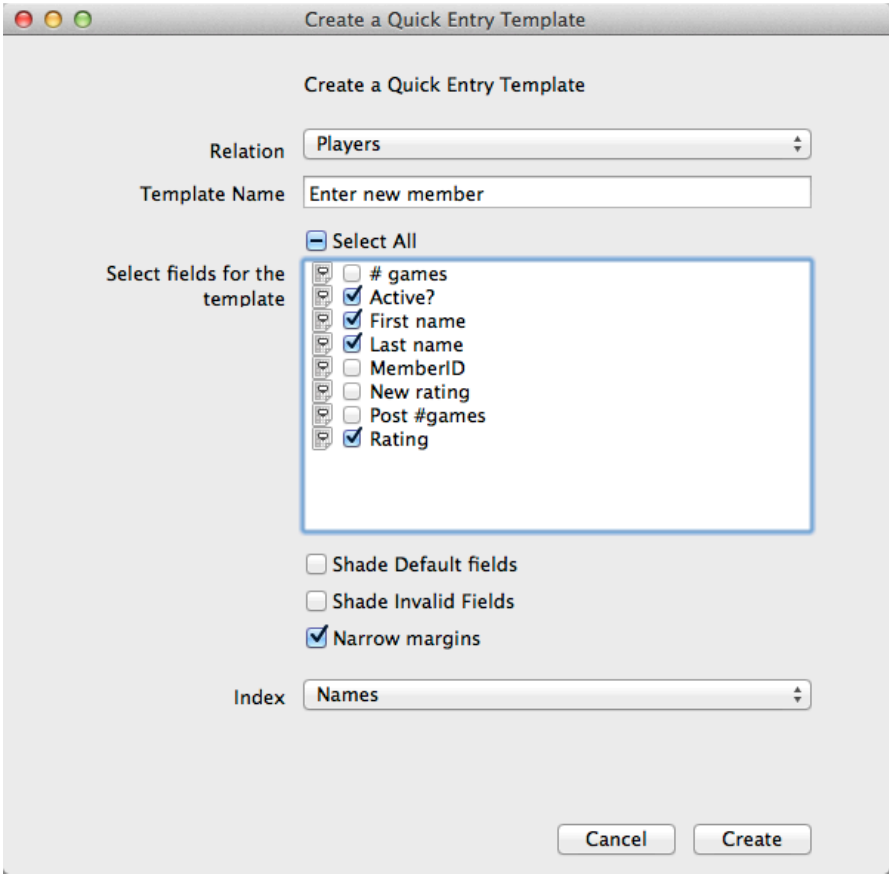
Add Indexes

The workbench provides a tool to quickly generate new indexes. After you select a relation, every field and abacus that is not yet indexed appears in the list. Select the items that you want to index, and press *Create* to create these indexes.



Make Quick Entry Form

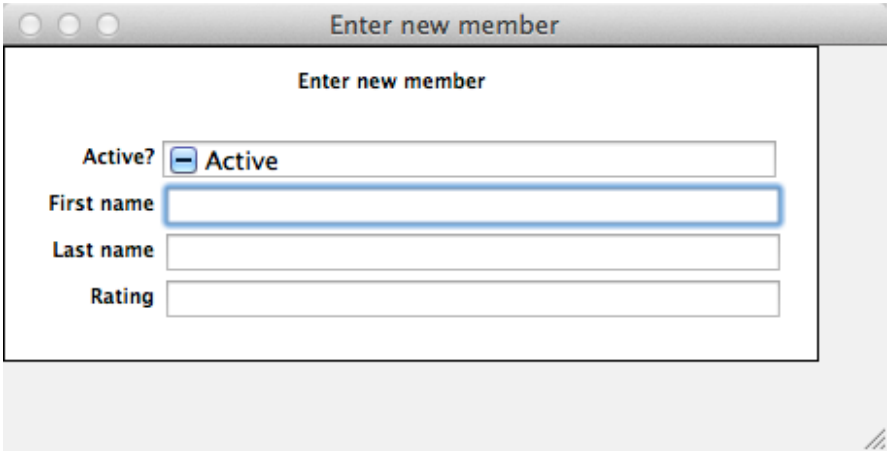
Sometimes, you want to quickly create an entry template that can be used to enter some test data. Sometimes, you want to be able to get a head start on a more formal entry form. In both cases, this tool to make a quick entry form helps. There are some differences between the Helix Workbench tool and the traditional Helix function that creates a quick entry. The workbench lets you select a subset of the fields to include in the template and what order to display them in.



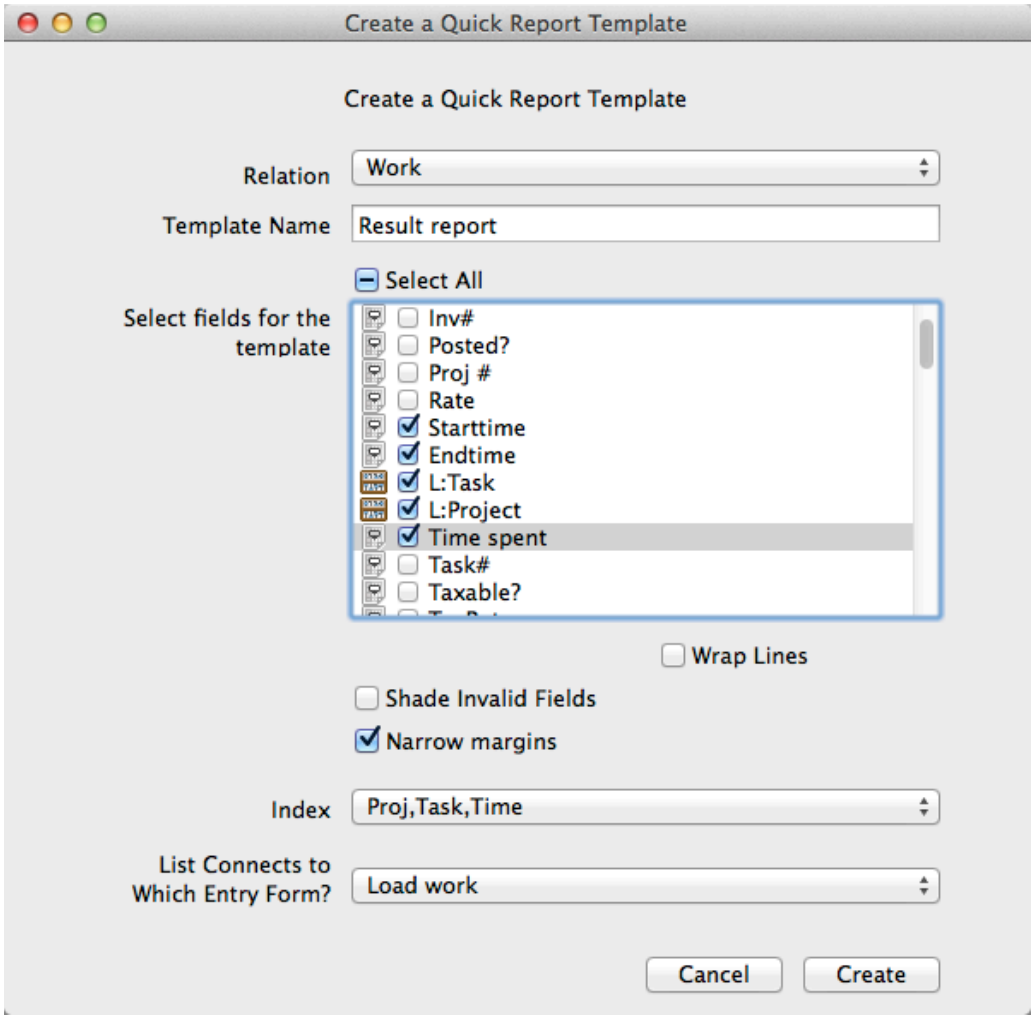
The tool begins by asking you to select the relation for the quick entry template, what to name the template, and what index (if any) to use for the view. When you select a relation, the fields in the relation will appear. You can select the fields to add to the template and rearrange the order they'll appear on the template by dragging fields up and down in the list.

The options allow you to shade defaulted fields (typically, this is unset), shade invalid fields (also, this is typically unset), and narrow margins (reduces the size of the template so the window doesn't take up so much room) when possible.

When you press the *Continue* button, this tool will create a template with the selected fields and a view with the same name that uses the new template and the specified index. Also, if this relation already has a post named *clhAssign* (that assigns the next counter# to the counter field), then that post is attached to this new view for *post on edit*.



Make Quick Report

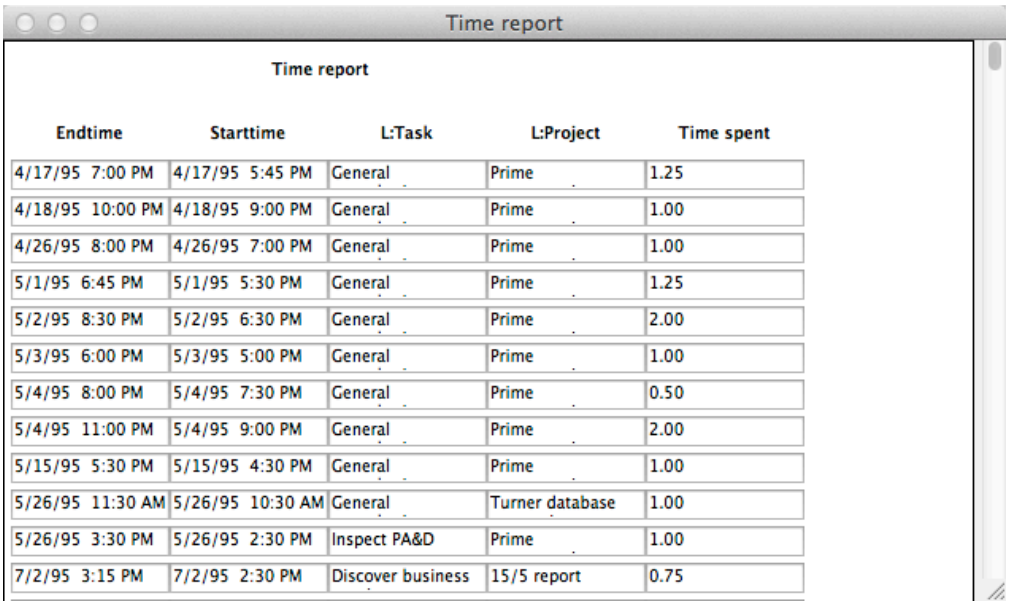


Sometimes, you want to quickly create a quick view of what's in a relation. Most of the time, you're only interested in seeing some of the fields or abaci. There are some differences between the Helix workbench tool and the traditional Helix function that creates a quick report. The workbench lets you select a subset of the fields to include in the template and what order to display them in. It also lets you include abaci in the report.

The tool asks you to select the relation for the quick report template, what to name the template, and what index (if any) to use for the view. When you select a relation, the fields and abaci in the relation will appear. You can select the items to add to the template and rearrange the order they'll appear on the template by dragging items up and down in the list.

The quick report has other options that you won't find in the default Helix quick report (or in the quick entry). The tool lets you choose a entry form to use for list processing. If you double-click on a record in your report, the selected record(s) will be displayed in this entry form. You can also tell the tool whether the items on the report should wrap at the end of the page or whether they should just continue to the right. If you're planning to just review this report online and not print it, you may find it preferable to not wrap the lines. You can specify that fields are shaded if they contain invalid values. And you can have your report use narrow margins (reduces the size of the template so the window doesn't take up so much room) when possible.

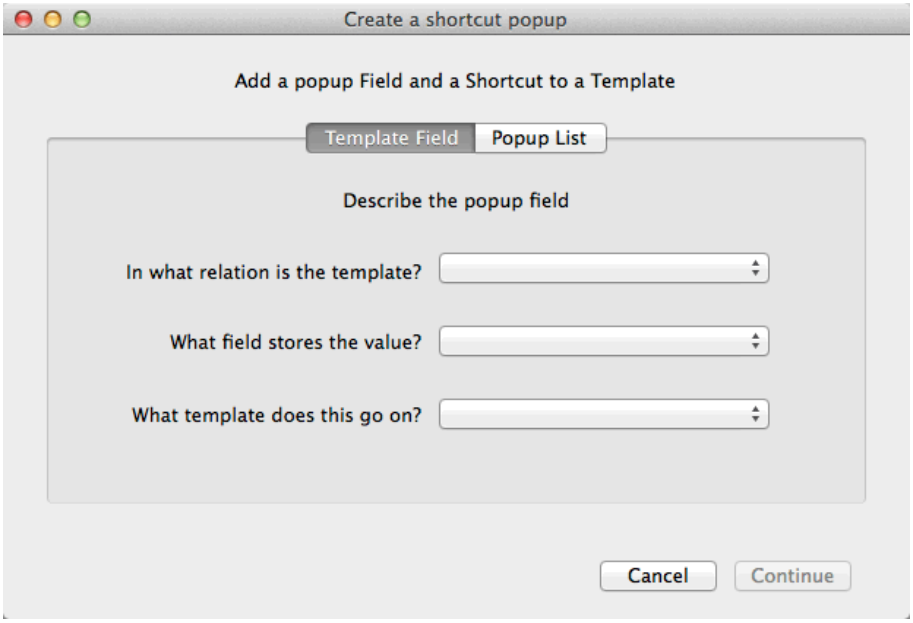
When you press the *Create* button, this tool will create a template with the selected fields and a view with the same name that uses the new template and the specified index.


Time report
Time report
Endtime
Starttime
L:Task
L:Project
Time spent
4/17/95 7:00 PM
4/17/95 5:45 PM
General
Prime
1.25
4/18/95 10:00 PM
4/18/95 9:00 PM
General
Prime
1.00
4/26/95 8:00 PM
4/26/95 7:00 PM
General
Prime
1.00
5/1/95 6:45 PM
5/1/95 5:30 PM
General
Prime
1.25
5/2/95 8:30 PM
5/2/95 6:30 PM
General
Prime
2.00
5/3/95 6:00 PM
5/3/95 5:00 PM
General
Prime
1.00
5/4/95 8:00 PM
5/4/95 7:30 PM
General
Prime
0.50
5/4/95 11:00 PM
5/4/95 9:00 PM
General
Prime
2.00
5/15/95 5:30 PM
5/15/95 4:30 PM
General
Prime
1.00
5/26/95 11:30 AM
5/26/95 10:30 AM
General
Turner database
1.00
5/26/95 3:30 PM
5/26/95 2:30 PM
Inspect PA&D
Prime
1.00
7/2/95 3:15 PM
7/2/95 2:30 PM
Discover business
15/5 report
0.75

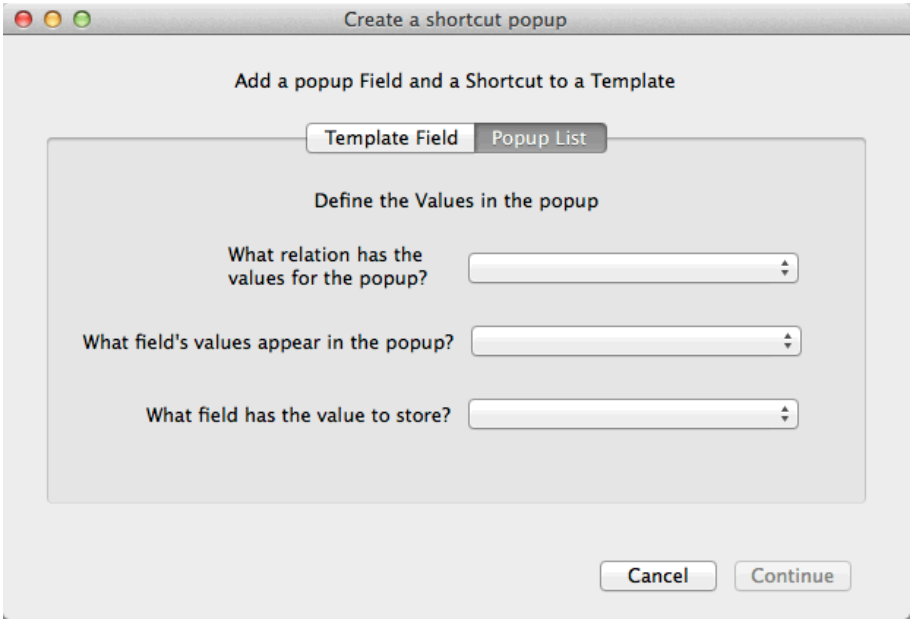
Create Shortcut Popup

A shortcut popup is a useful item to add to an entry form. The command adds a data rectangle to a template and formats the rectangle as a popup rectangle. If a data rectangle already exists on the template with the field, then that rectangle is converted into a popup. The popup is a *dynamic* popup, which means that the choices in the popup come from the records and a field in another relation. For example, on a sales entry form, there may be a popup for *color*. Your collection has a separate relation that store the available colors for the product, and those values can be selected in the popup.

A second data rectangle is also added to the template, and it contains the shortcut for the popup. The popup will only display values that begin with whatever is typed into the shortcut. What this means, is let's say that the popup displays the cityname from another relation. If there are 200 cities, you wouldn't want someone to have to scroll through all 200 of them. If the shortcut has no value, then no cities are shown in the popup. As soon as the user types an *A* into the shortcut, then the popup lists all cities that begin with the letter A. If the user types *AB* into the shortcut, then the popup only lists cities that begin with the letters Ab. There's one other special feature to note. If the relation that supplies the values for the list has duplicate values, only one of them will be displayed in the popup.



To define the shortcut popup, define the template field and then the popup list. On the *Template Field* tab, first select the relation that contains the template and the field that stores the selected value. Once the relation is selected, then select the field that will store the selected value from the popup and the entry template that the data rectangle will be stored on.



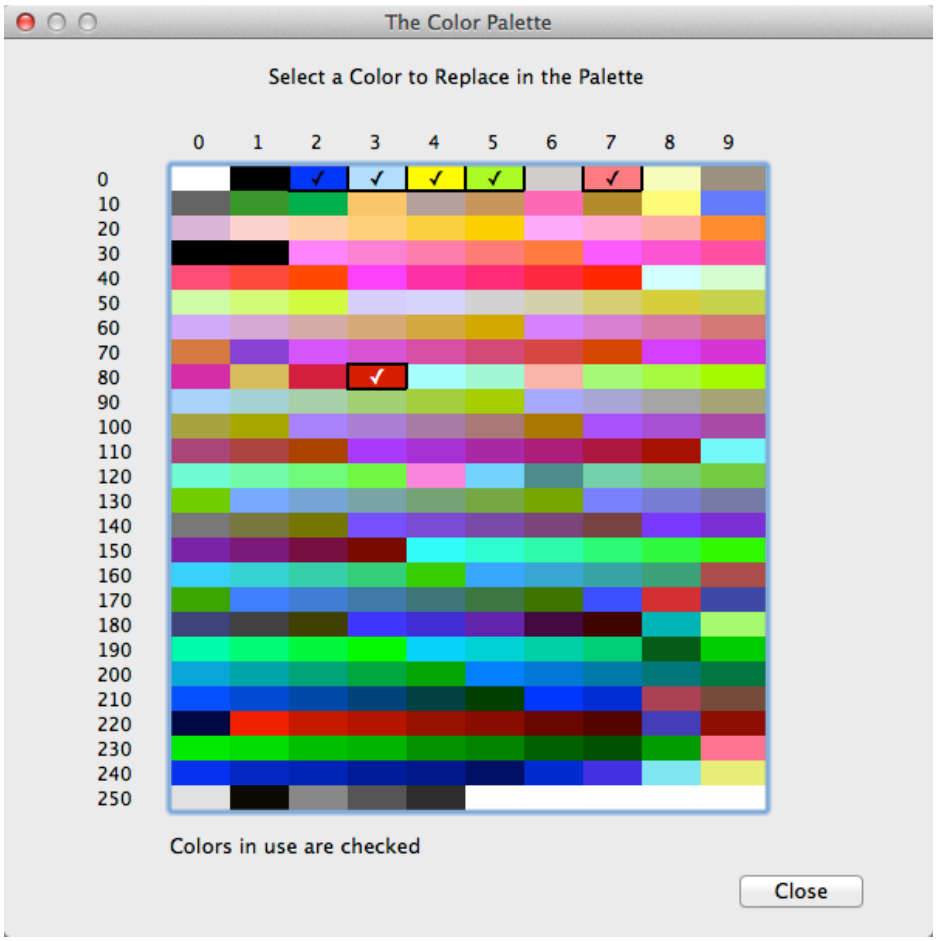
On the *Popup List* tab, there are three values to supply. (1) What relation contains the records with the values for the popup? (2) What field's (or abacus') values will populate the popup? Only fields and abaci that are indexed can be selected here. That's guarantees that the popup list will be sorted and also allows duplicate values to be suppressed. (3) What field has the value to store? Note that only fields and abaci that share the same

datatype as the field that stores the value can be selected here.

It's common that the field's value to store is different from the field's value to display. If you're selecting a product, you probably want the product description in the popup, but you want to store the product code. Or if you are entering an employee's time, you want the popup to show the employee's name, and the value stored is the employee's ID.

Edit Color Palette

Color is an important element in well-designed entry forms and reports. You can specify the color to be used as background areas in rectangles and as conditional colors by assigning a Color ID from the color palette in Helix RADE Template Inspector.



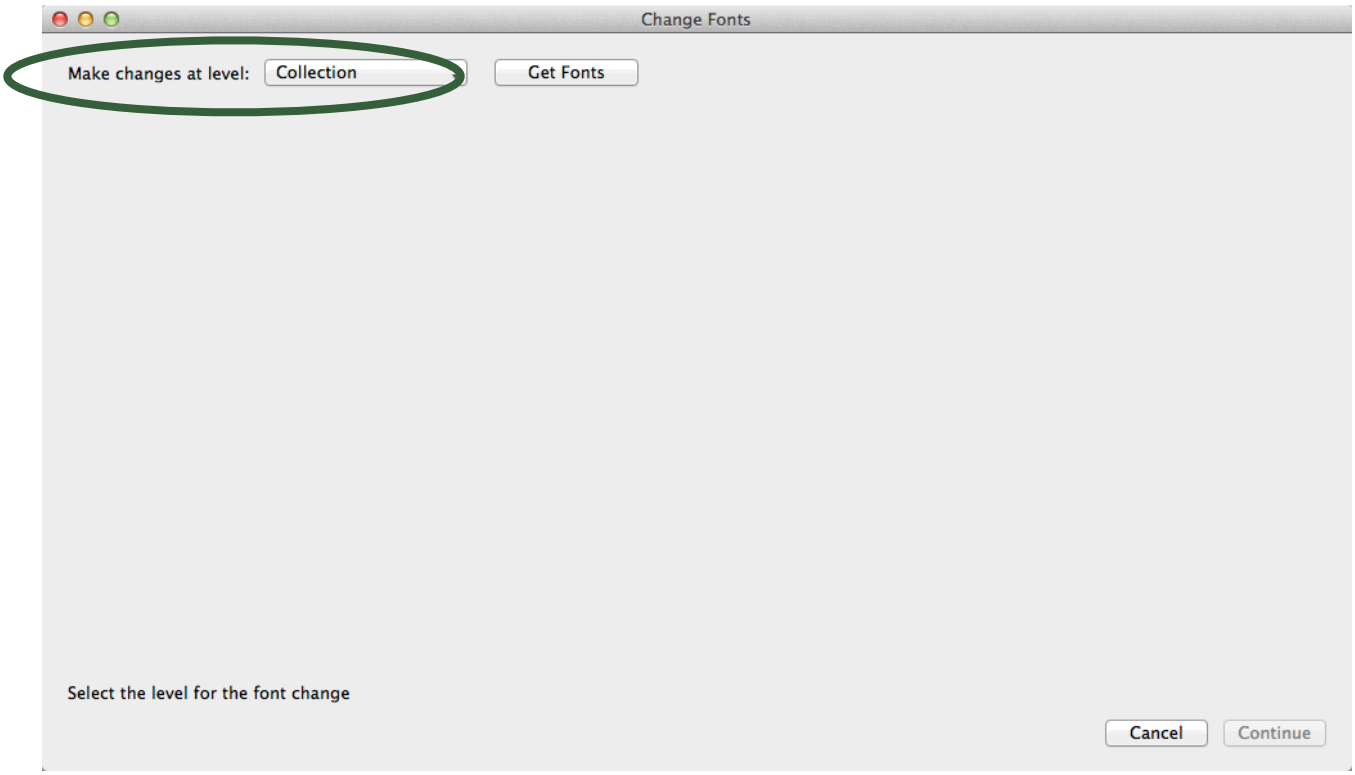
What if you want to add or change a color though? This tool makes this process more visible. The tool begins by examining the collections and then displaying a palette that shows the color for every color number. The only way to "add" a color is really to replace a color that's in the palette. Click on the color that you don't want to use. Apple's standard color chooser appears and lets you select a replacement color.

You can tell which colors are already in use in this collection, because the table displays a dark border around those colors. If you change one of the colors already in use, then it will change wherever it's being used in your reports and entry forms.

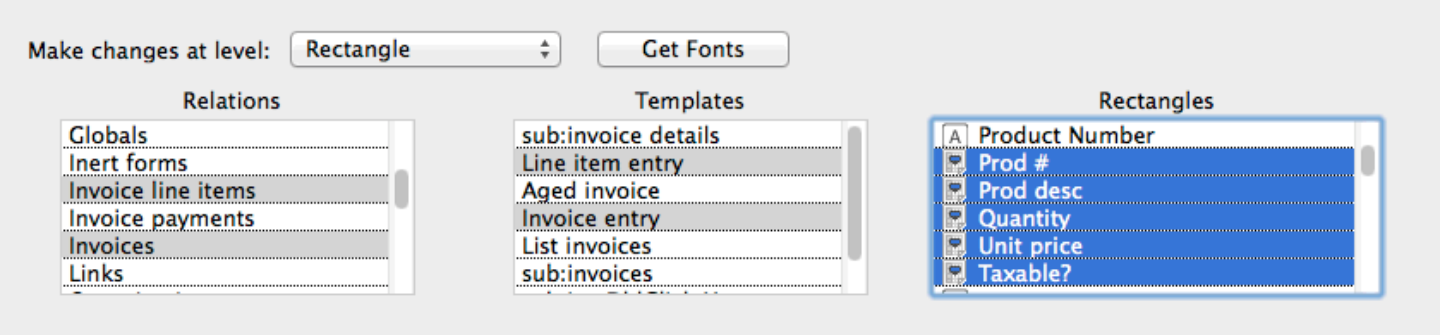
The Helix Workbench only examines colors used in rectangle backgrounds. It does not test for conditional styles.

Mass Font Changes

With the Helix Workbench, you're able to change the fonts and colors for individual rectangles on a template, all of the fonts on a template or in a relation, or even all of the fonts in the entire collection. Begin by specifying whether you want to change fonts for the collection, for relations, for templates, or for rectangles.



After you select the level that you want, you can one or more objects to drill down to the items you want to examine. In this example, we've selected the relations *Invoice line items* and *Invoices* and then selected the templates *Line item entry* and *Invoice entry*. All of the rectangles in those two templates appear in the third list. We select the ones to review and select *Get Fonts*.



The left-hand portion of the font list shows the rectangles in the templates and what fonts they currently are using. Specify the font that you want to change these rectangles to and check the ones to change.

Fonts in Use Show colors

Relation	Template	Rectangle	Current Font	Bold	Italic	Size	Mark Fonts to Update	New Font	New Typeface	New Size
Invoice line items	Line item entry	Invoice#	Times	✓	✓	12	☐	Times		12
Invoice line items	Line item entry	Line#	Times	✓	✓	12	☐	Times		12
Invoice line items	Line item entry	Prod #	Times			14	☐	Times		14
Invoice line items	Line item entry	Prod desc	Times			14	☐	Times		14
Invoice line items	Line item entry	Quantity	Times			14	☐	Times		14
Invoice line items	Line item entry	Unit price	Times			14	☐	Times		14
Invoice line items	Line item entry	Taxable?	Times			14	☐	Times		14
Invoices	Invoice entry	Invoice date	Verdana	✓		12	☐	Verdana		12

Set Default Font

In this example, all of the 14-point Times rectangles are being changed to 12-point Arial.

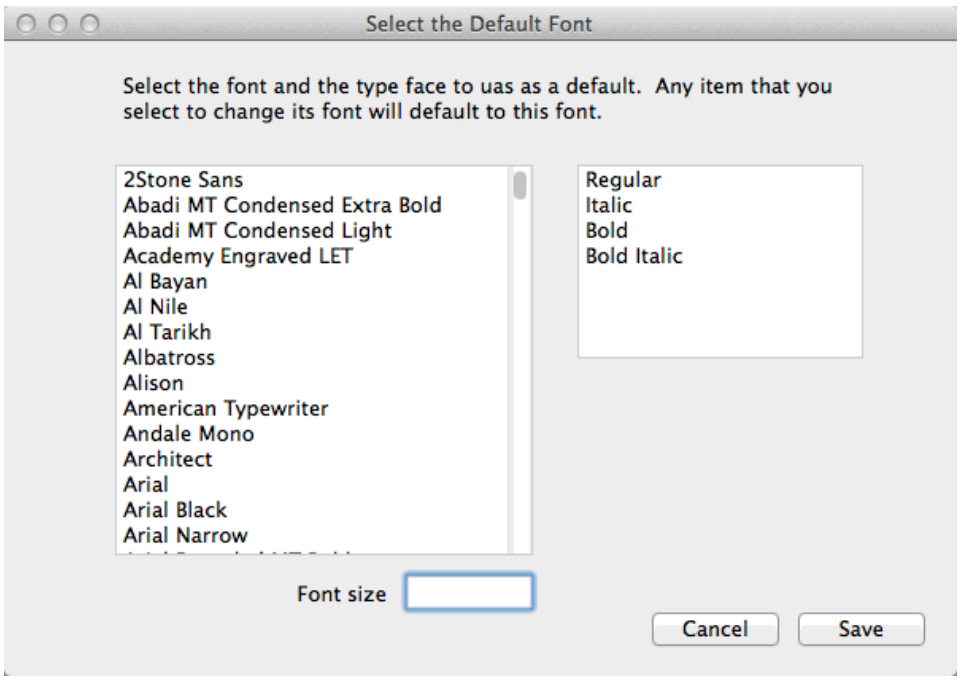
Fonts in Use Show colors

Relation	Template	Rectangle	Current Font	Bold	Italic	Size	Mark Fonts to Update	New Font	New Typeface	New Size
Invoice line items	Line item entry	Invoice#	Times	✓	✓	12	☐	Times		12
Invoice line items	Line item entry	Line#	Times	✓	✓	12	☐	Times		12
Invoice line items	Line item entry	Prod #	Times			14	☑	Arial	Regular	12
Invoice line items	Line item entry	Prod desc	Times			14	☑	Arial	Regular	12
Invoice line items	Line item entry	Quantity	Times			14	☑	Arial	Regular	12
Invoice line items	Line item entry	Unit price	Times			14	☑	Arial	Regular	12
Invoice line items	Line item entry	Taxable?	Times			14	☑	Arial	Regular	12
Invoices	Invoice entry	Invoice date	Verdana	✓		12	☐	Verdana		12

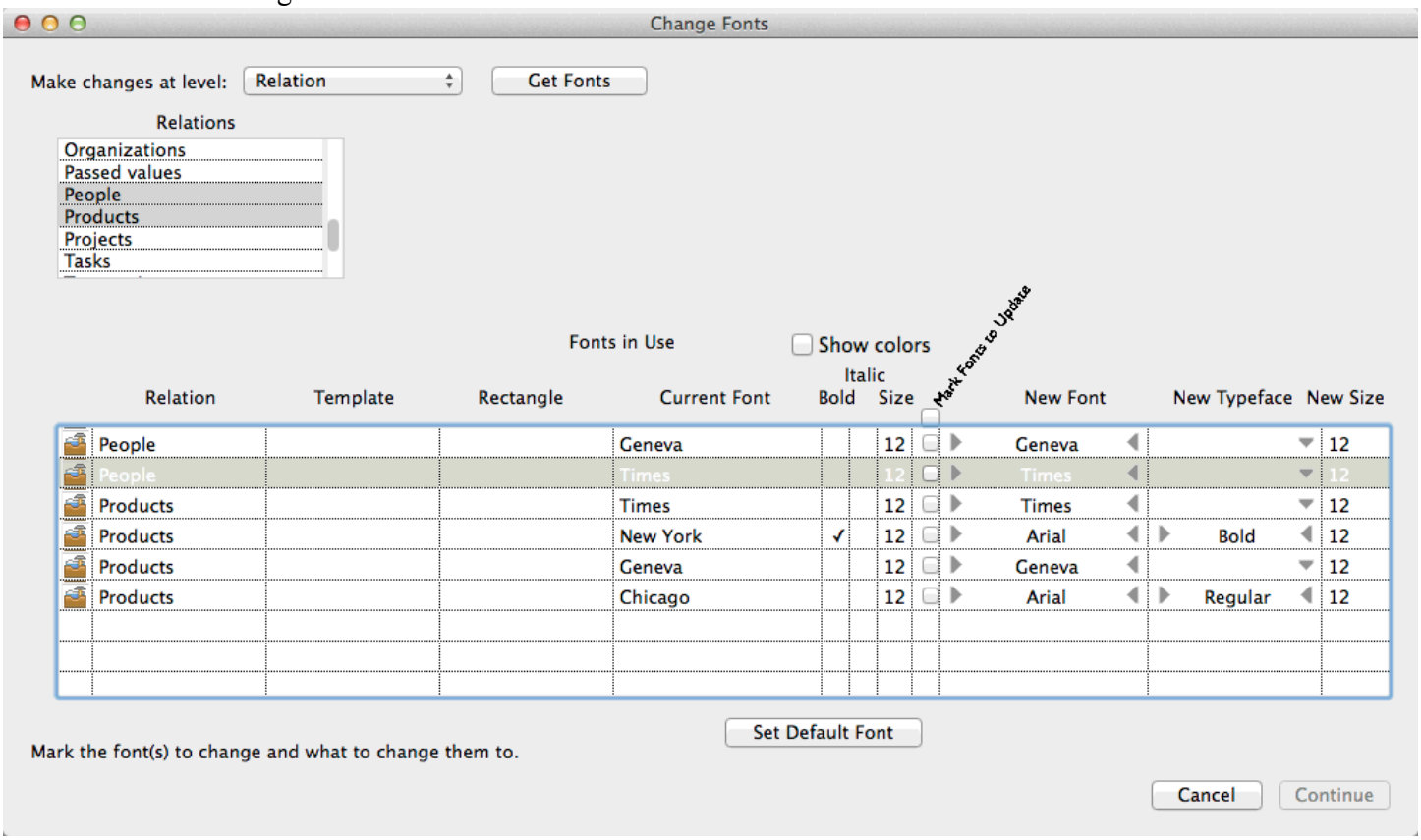
Set Default Font

The select the *Continue* button to complete the update.

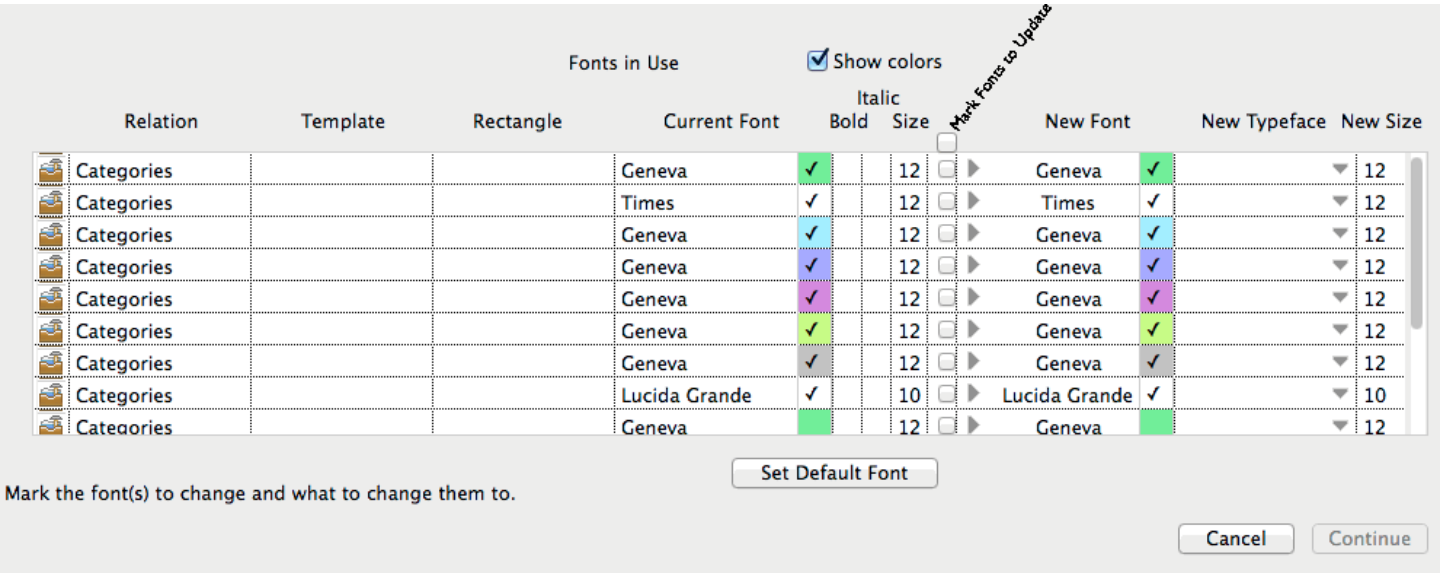
You have a second way to specify the font that you're changing to. If you have several rectangles that you're changing to the same font, you can select the *Set Default Font*. After you save the font, the typeface, and the font size, you will return to the list of all of the fonts. As you check off rectangles to convert, the default font, face, and size will automatically be selected as the target for the conversion. (You can override the defaults on a rectangle-by-rectangle basis if you want.)



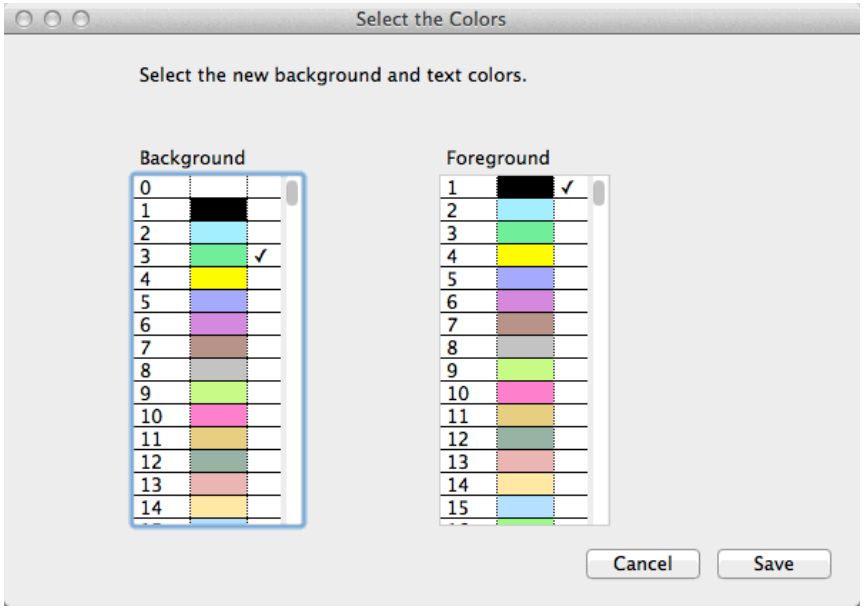
Finally, you don't have to make changes at such a low level. In this example, we've said that we want to make changes a relation level. We've retrieved the list of fonts used in the *People* and *Products* relations. *People* has rectangles that are 12-point Geneva and 12-point Times. *Products* has rectangles 12-point Times, 12-point New York, 12-point bold Geneva, and 12-point Chicago. We're about to change all of the bold New York to Bold Arial and Chicago to Arial.



You can also use mark the *Show colors* checkbox to see the colors for the rectangles and change them. When you check *Show colors*, two columns of boxes appear. The first row shows the background color for that rectangle, and the checkmark shows the color of the text. The second column shows the new colors.



To change a rectangle's color, click in the second column for the rectangle that you want to change. When the dialog box appears, the current settings are checked.The first column shows the background color in use, and the second column shows the color of the text. The columns show all of the colors that have been set up for this collection. Each collection has its own color palette; you can change the colors used by a collection with the Helix Workbench's *Edit Color Palette* command. After selecting the new colors for the rectangle, you can save the changes with *Save* or discard them with *Cancel*.



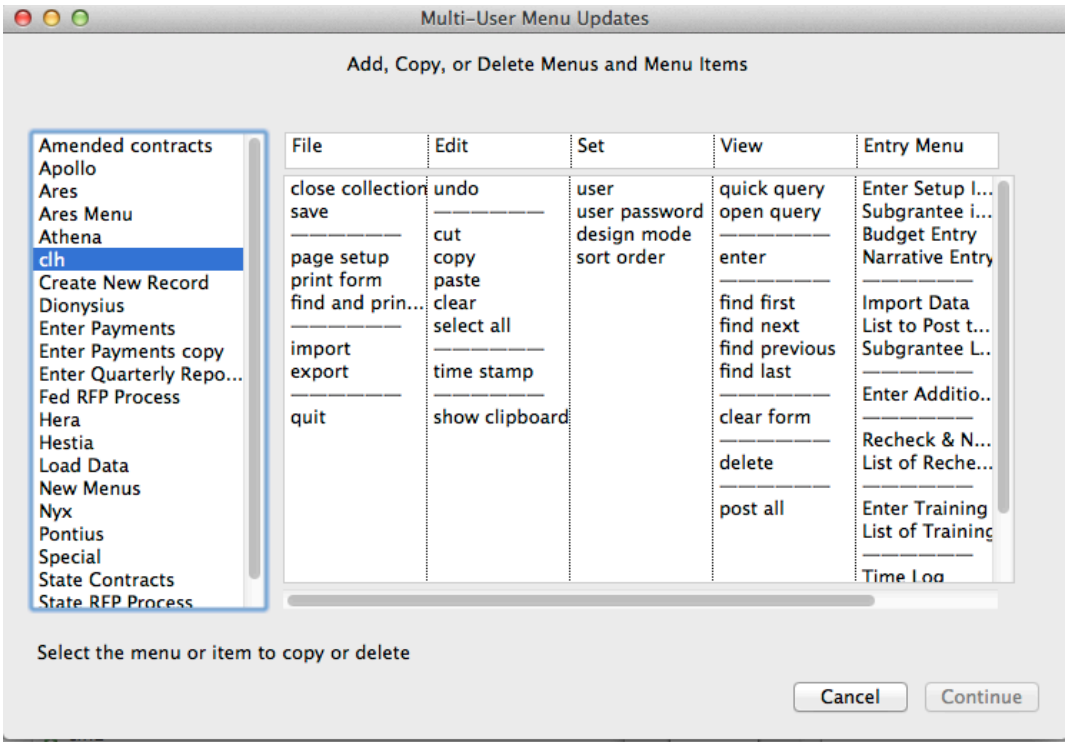
Two special notes:

1) The Helix Workbench is only examining fonts in text and label rectangles. It is not examining styled text.

2) In some cases, Mac fonts have more than one name, depending on how the font is referenced. You may find a font name in the font list, but that's not the font name that the Mac refers to it internally. When Helix 6 is told to display a font that it can't identify, it will display it as Lucida Grande. And it can be confusing, because Helix will tell the Workbench that it's using the font that you selected, but it's still displaying it as Lucida Grande.

Multi-user Menu Updates

New reports or menu options often must be applied to multiple user profiles. This feature saves you from having to repeat the changes for every profile.



First, make the changes to a single profile that you want to copy to other profiles. Then, in the Helix Workbench, select that profile as your source profile. When you select the source profile, then its menus will appear. Designate the menu or the item to copy to or delete from other profiles. You can hover over a menu item to see the full menu name.

If you select a menu, then the entire menu will be deleted or copied (see figure 2).

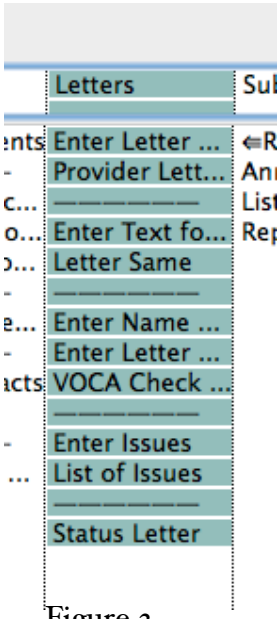


Figure 2

Figure 3 shows that submenus have an arrow following the name that will go down to the next level if you select that item. Figure 4 shows that all of the items of the submenu are selected when you select a submenu. You can continue with them all selected, or you can select a single item. You can also go return to the previous level by selecting the first item in the list (the one that has a left-pointing arrow.

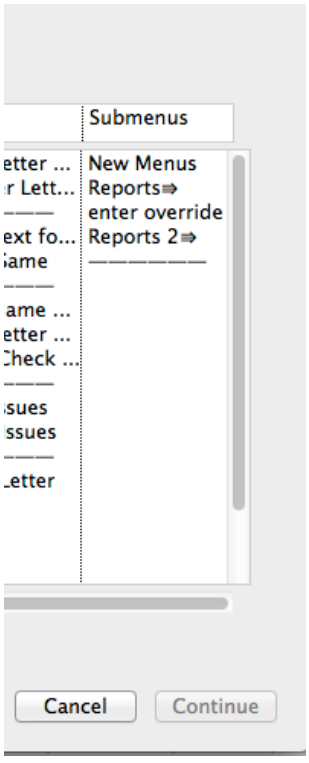


Figure 3

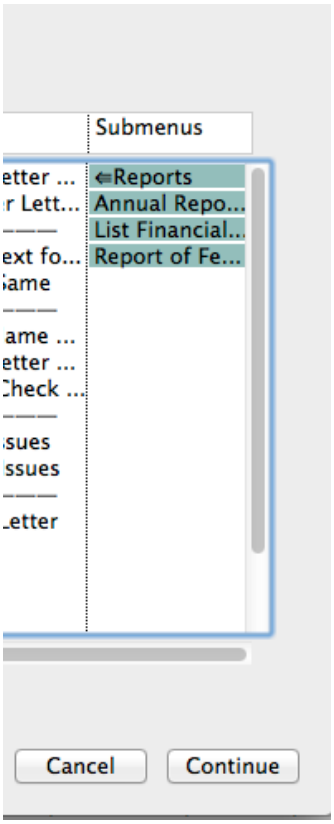
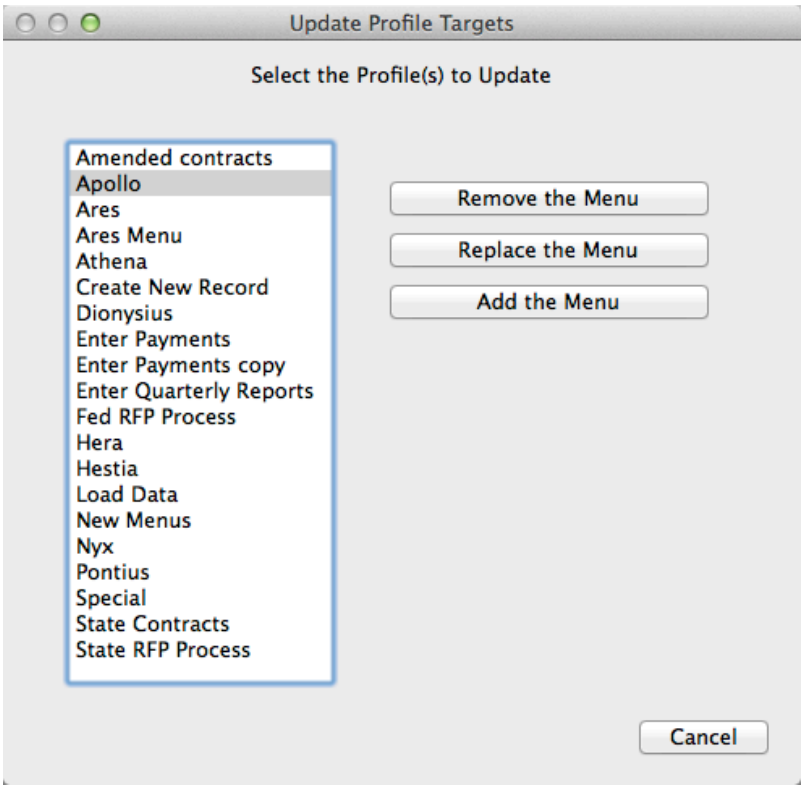


Figure 4

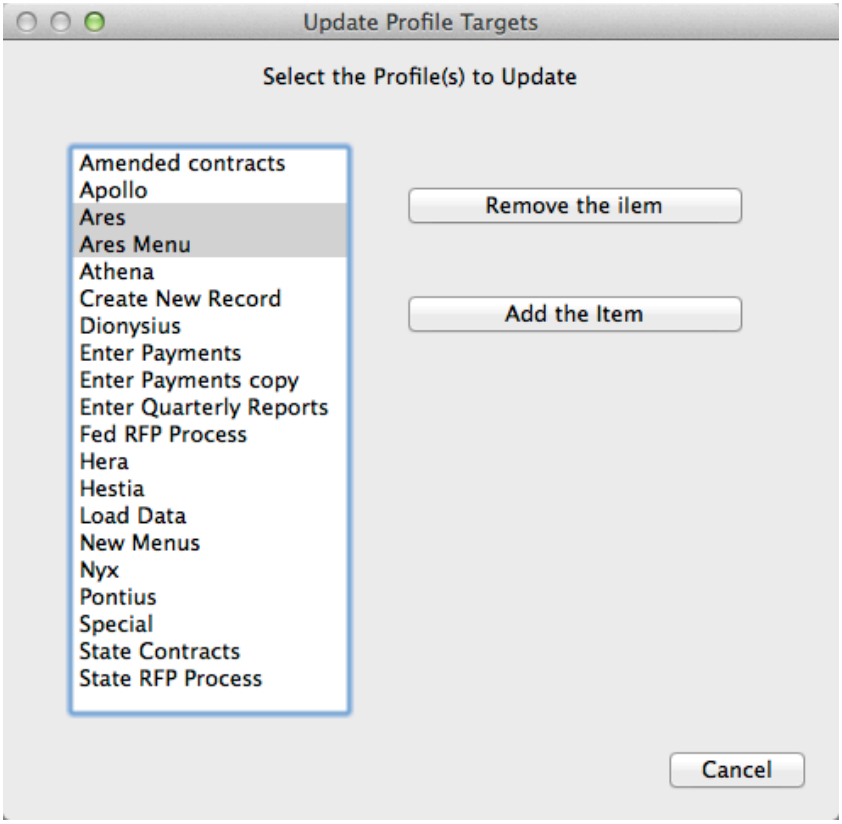
After the item to delete or copy has been selected, select the Continue button.

If you've selected a menu, then you can select one or move target profiles and either:

1. remove the menu from the source profile and all of the targets (*note that this feature is temporarily broken until Helix 7 is released)*;
2. copy the menu from the source profile to all of the target profiles and replace the version of the menu that's already there; or,
3. add the [new] menu to all of the selected target profiles.



If you select a single item, then you can remove the item from all of the selected targets. (If you remove the item, it will remove it from any menu that it appears in.) Or, you can add the item to the target profile. The Helix Workbench will do its best to find the same location in the target profile to add the menu item. If it can't find a matching menu, then a new menu will be created with the selected menu item in it.



License Agreement

This is a legal agreement between you (either an individual or an entity), the end-user, and CLH WAREs. By installing the computer software in this package ("SOFTWARE"), by loading or running the SOFTWARE, or by placing or copying the SOFTWARE onto your computer hard drive, you are agreeing to be bound by the terms of this Agreement.

CLH WAREs LICENSE

1. GRAND OF LICENSE. CLH WAREs grants to you the right to use one (1) copy of the enclosed SOFTWARE on a single computer. For purposes of this section, "use" means loading the SOFTWARE into RAM, as well as installation on a hard drive or other storage device. You may not: rent, lease, modify, translate, disassemble, decompile, reverse engineer, or create derivative works based upon the SOFTWARE and distribute to others. The SOFTWARE, together with any archive copy thereof, shall be destroyed when no longer used in accordance with this Agreement, or when the right to use the SOFTWARE is terminated.

2. COPYRIGHT. The SOFTWARE is owned by CLH WAREs and/or its licensors and is protected by United States copyright laws and international treaty provisions. You must treat the SOFTWARE like any other copyrighted material, except that you may either (a) make copies of the SOFTWARE solely for back-up or archival purposes. You may not otherwise reproduce, copy or disclose to others, in whole or in any part, the SOFTWARE.

3. OTHER RESTRICTIONS. This License Agreement is your proof of license to exercise the rights granted herein and must be retained by you. You may not rent or lease the SOFTWARE, but you may transfer your rights under this License Agreement on a permanent basis provided you transfer this License Agreement, the SOFTWARE, and all accompanying materials and retain no copies, and the recipient agrees to the terms of this Agreement. Any transfer of the SOFTWARE must include the most recent update and all prior versions.

4. GENERAL PROVISIONS. Should any provision of this Agreement be held to be void, invalid, unenforceable, or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby. If any provision is determined to be unenforceable, you agree to a modification of such provision to provide for enforcement of the provision's intent, to the extent permitted by applicable law Failure of a party to enforce any provision of this Agreement shall not constitute or be construed as a waiver of such provision or of the right to enforce such provision. If you fail to comply with any terms of this Agreement, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

LIMITED WARRANTY

LIMITED WARRANTY. CLH WAREs and its licensors warrants that if properly installed and operated on a computer for which it is designed, the SOFTWARE will perform substantially in accordance with the accompanying documentation for the current Mac OS and Helix suite version at the time of sale.

CUSTOMER REMEDIES. CLH WAREs and its licensors' entire liability and your exclusive remedy shall be, at CLH WAREs' option, either (a) return of the price paid or (b) repair or replacement of the

SOFTWARE that does not meet CLH WAREs' Limited Warranty. To make a warranty claim, return the SOFTWARE to the point of purchase, accompanied by proof of purchase, your name, your address, and a statement of defect, or return the SOFTWARE with the above information to CLH WAREs. This Limited Warranty is void if failure of the SOFTWARE has resulted in whole or in part from accident, abuse, misapplication, or violation of this Agreement. Any replacement SOFTWARE will be warranted for the remainder of the original warranty period or thirty (30) days, whichever is longer. This warranty allocates risks of product failure between Licensee and CLH WAREs. CLH WAREs' product pricing reflects this allocation of risk and the limitations of liability contained in this warranty.

NO OTHER WARRANTIES. CLH WAREs AND ITS LICENSORES DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE SOFTWARE AND THE ACCOMPANYING WRITTEN MATERIALS. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS THAT VARY FROM JURISDICTION TO JURISDICTION. CLH WAREs AND ITS LICENSORS DO NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED, ERROR FREE OR MEEET LICENSEE'S SPECIFIC REQUIREMENTS. THE WARRANTY SET FORTH ABOVE IS IN LIEU OF ALL OTHER EXPRESS WARRANTIES ON BEHALF OF CLH WAREs. ADDITIONAL STATEMENTS SUCH AS DEALER ADVERTISING OR PRESENTATIONS, WHETHER ORAL OR WRITTEN, DO NOT CONSTITUTE WARRANTIES BY CLH WAREs OR ITS LICENSORS AND SHOULD NOT BE RELIED UPON.

EXCLUSIVE REMEDIES. You agree that your exclusive remedy against CLH WAREs, its affiliates, contractors, suppliers, and agents for loss or damage caused by any defect or failure in the SOFTWARE regardless of the form of action, whether in contract, tort, including negligence, strict liability or otherwise, shall be the return of the purchase price paid or replacement of the SOFTWARE. This Agreement shall be construed in accordance with and governed by the laws of the State of Texas. Copyright and other proprietary matters will be governed by the United States laws and international treaties. IN ANY CASE, CLH WAREs AND ITS LICENSORS SHALL NO BE LIABLE FOR LOSS OF DATA, LOSS OF PROFITS, LOST SAVINGS, SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT OR OTHER SIMILAR DAMANGES ARISING FROM BREACH OF WARRANTY, BREACH OF CONTRACT, NEGLIGENCE, OR OTHER LEGAL THEORY EVEN IF CLH WAREs OR ITS AGENT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

U.S. GOVERNMENT RESTRICTED RIGHTS

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph ©(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs ©(1) and (2) of the Commercial Computer Software—Restricted Rights at 48 CFR 52.227-19, as applicable. Contract/manufacturer is CLH WAREs, Box 720367, Houston, TX 77272, USA.