QUALRUS™

The Intelligent Qualitative Analysis Program

Come visit us on the web at www.qualrus.com

Idea Works pledges to donate a minimum of 1% of income from sales of Qualrus to the World Wildlife Fund to help preserve the walrus and other endangered species.

To make your own contribution directly to the World Wildlife Fund, go to their site at http://www.worldwildlife.org/

Qualrus™

The Intelligent Qualitative Analysis Program

Quick-Start Manual: Getting Started with Qualrus™

By Edward Brent, Pawel Slusarz, and Alan Thompson
Idea Works, Inc.™

Contact Information:

Qualrus may be purchased directly from Idea Works, Inc. or from Sage Publications/Scolari

Idea Works, Inc.
100 West Briarwood, Columbia, Missouri 65203 USA
Voice: (573)445-4554 Fax: (573)446-2199
E-Mail: support@ideaworks.com
Web page: www.ideaworks.com.

Distributor Contact Information:

Sage Publications, Inc.
North and South America 2455 Teller Road, Thousand Oaks, CA 91320 (805)499-0721, Fax: (805)499-0871, E-mail: scolari@sagepub.com. Web page: www.scolari.com or www.sagepub.com.
All Other Countries 6 Bonhill Street, London EC2A 4PU, England +44(0) 171 330 1222, Fax +44(0) 171 374 8741 Web page: www.scolari.co.uk

This quick-start manual takes you through the basic steps for getting started with Qualrus. It describes how to install the program, begin work on a new project, identify sources, develop a coding scheme, assign codes, perform simple initial analyses, save your files, and generate a report. This quick-start manual alone should be sufficient to get you up and running with Qualrus and provides a brief overview of these key steps. For more details regarding each of the features of Qualrus and procedures for using them effectively, see the full Qualrus manual available in pdf, html, and MS-Word formats from our web site at http://www.ideaworks.com.

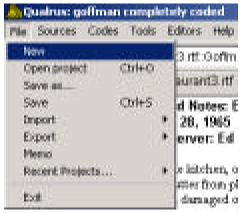
Installing Qualrus

To install Qualrus on your computer from a CD, place the CD in your CD reader. The setup program will begin automatically. Just follow the directions on your screen. If, for some reason, setup does not start automatically, then open the CD drive icon in your "my computer" folder, double click on the "setup.exe" file on the CD, then follow the program's instructions.

To install Qualrus by downloading the self-extracting file from the Internet, select "Run this program from its current location" instead of saving. The program will begin. Then follow the instructions.

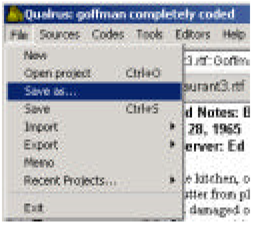
Starting the Program

To begin Qualrus, open your Qualrus file folder and double-click the left mouse button on the Qualrus icon. Qualrus automatically starts up with a "welcome" project. This project contains only the default information. You can add sources, codes, and code assignments to this project and save it as a new project with a project name. If you have run Qualrus before, then, each time the program is restarted, it automatically loads the last edited project. The name of the project will be displayed in the title bar of the main Qualrus screen just after "Qualrus:". If your program opens to an existing project, simply select the "file" option in the menu bar at the top of the main Qualrus screen, as shown here, and select the "New" option to create a new project.



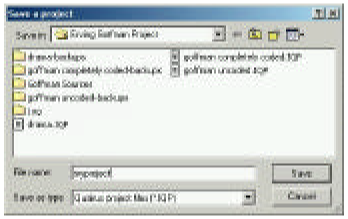
Saving the Project

At any point you can save the current project by selecting the "File" option in the menu bar at the top of the main screen, then selecting either "save" to save the project with its current name or "save as" to save it with a new name.



Let's save the project now by selecting "save as" then giving the project the name, "myproject."

Just like using a word processor, spreadsheet, or any other computer program, you should regularly save your work so that if your computer crashes you will not lose your data.



File name: myproject

Save as type: Qualrus project files (*.tQP)

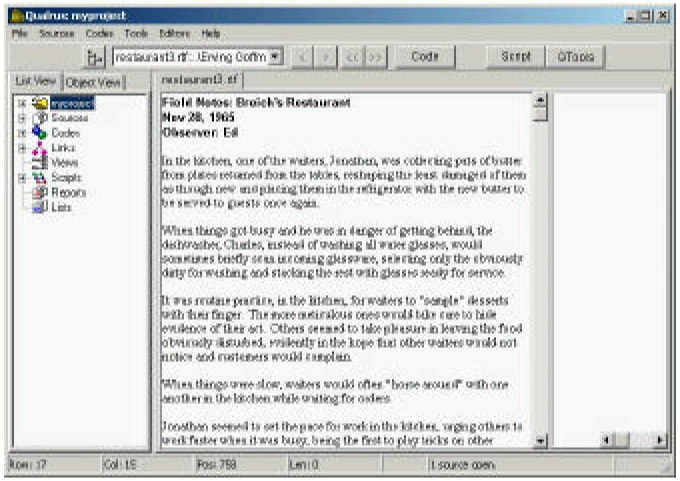
NOTE: Qualrus *automatically saves your project periodically with the same name as the current project. For additional protection against data loss, you should occasionally save new versions of the project with new names, keeping older versions as backups.*

Now we're ready to try opening a source, creating codes, and other common qualitative analysis tasks.

Opening a Source

From the "Sources" item in the menu bar select "Add new source" to add a new source. Then select the "restaurant3" file. If you followed the default installation, this file should be located in a sub-folder within the main folder where you installed the program. If your main folder is "Qualrus" then inside that folder should be the "Erving Goffman Project" directory. Within that directory should be the "Goffman sources" directory. In that folder you should find this source file ("restaurant3"). Highlight this file and click on the "Open" button to add this source to the project.

The main Qualrus window, shown here, has three large sub-windows. In the left sub-window is the project tree (an outline of the key components of the



Field Notes: Broich's Restaurant
Nov 28, 1965
Observer: Ed

In the kitchen, one of the waiters, Jonathan, was collecting pats of butter from plates returned from the tables, reshaping the least damaged of them as though new and placing them in the refrigerator with the new butter to be served to guests once again.

When things got busy and he was in danger of getting behind, the dishwasher, Charles, instead of washing all water glasses, would sometimes briefly scan incoming glassware, selecting only the obviously dirty for washing and stacking the rest with glasses ready for service.

It was routine practice, in the kitchen, for waiters to "sample" desserts with their finger. The more meticulous ones would take care to hide evidence of their act. Others seemed to take pleasure in leaving the food obviously disturbed, evidently in the hope that other waiters would not notice and customers would complain.

When things were slow, waiters would often "horse around" with one another in the kitchen while waiting for orders.

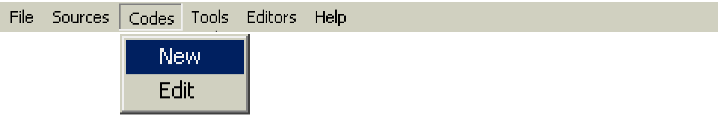
Jonathan seemed to set the pace for work in the kitchen, urging others to work faster when it was busy, being the first to play tricks on other current project). In the middle sub-window is displayed the source file (in this case, text from the restaurant3 file). On the right is a window displaying any codes assigned to segments of the source. (At this point, no codes have been assigned yet, so nothing is displayed in the right sub-window.)

Creating a Coding Scheme

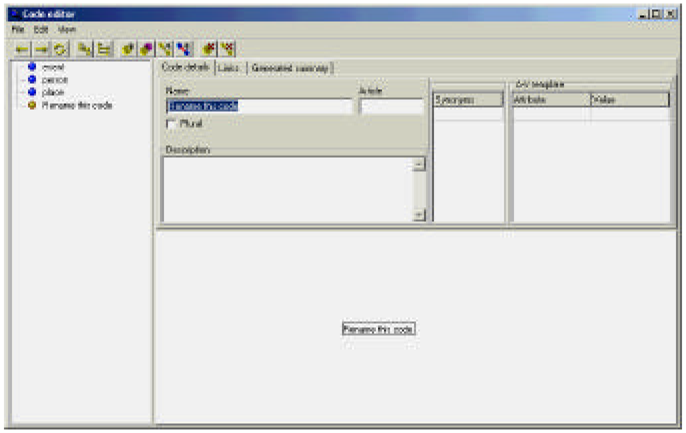
To create a coding scheme we must first create one or more codes and then specify any links we want among those codes.

Creating New Codes

To create a new code, select "Codes" in the menu bar at the top of the Qualrus screen, then select "new" in the pull-down menu that appears.



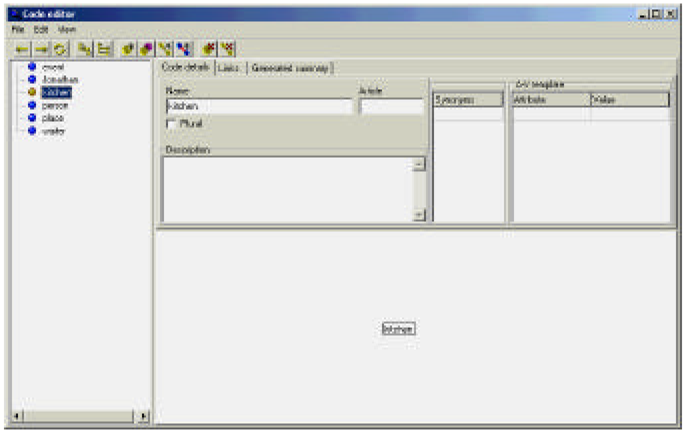
This brings up the Code Editor window shown here opened to a newly created code with the temporary name, "Rename this code." You should then highlight the code name in the "Name" field as shown here and type in the new name for this code.



To add codes from within the Code Editor window, select the new-code button, . (As you pass the mouse over each icon you will see a text description of the button; the correct one will say "New code.")

Continue in the same manner until you have added three codes: kitchen, Jonathan, and waiter. When you have done this the Code Editor should look much as it does below, with the list of codes now in the project displayed on the left. By clicking the left mouse button on any of those codes you can make that code be the one displayed in detail.

You can add as many codes as you like. However, creating codes is only one part of conceptualizing your project. You must also consider how those codes are linked to one another.



Linking Codes

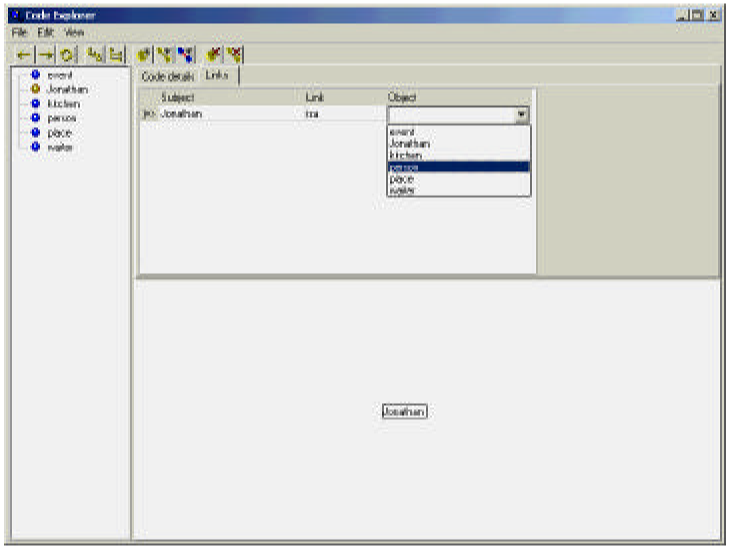
We are now ready to create links among the codes. There are many possible links among codes. Links can express logical relationships among codes such as X is a Y or X is part of Y. Links can also express empirical relationships such as X is in Y or X causes Y. By default, Qualrus includes within any new project a set of common link types such as "isa" and "part of." Users can also define their own link types using the Link Editor described in Chapter Four of the manual.

You might be wondering why we want to bother with links among codes. Indeed, if you want to "get on with" the project you may want to jump right into assigning codes to segments now and consider links among codes later, if ever.

However, links are important for both you and the program. For you, links help you conceptualize your problem and see relationships among the different elements that are coded. Recognizing hierarchical relationships in which some codes are special cases of others can suggest additional codes you may have forgotten. Thinking about other possible links among codes helps you to reason through possible causal relationships, sequential connections, associations, memberships, network links, and a host of other relationships.

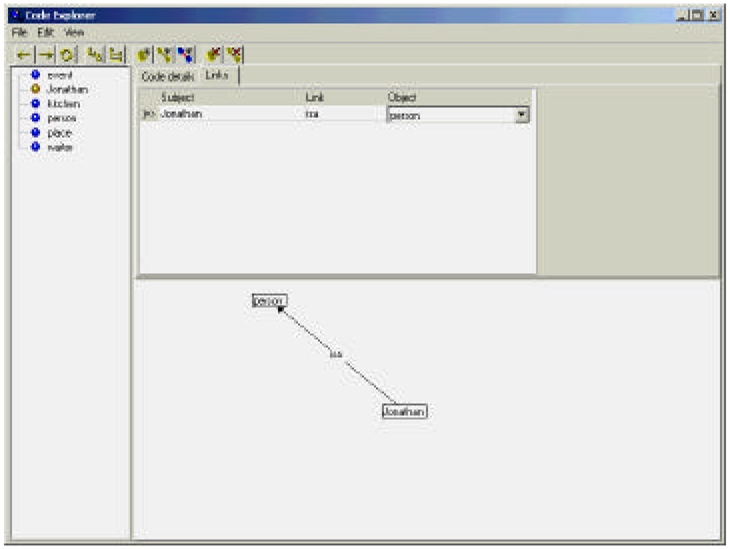
Links are also important for the program. You see, the links among codes contain within them a wealth of useful knowledge about the codes that the program can use to help suggest codes, describe codes, and generate meaningful accounts of your data. You don't have to enter these links among codes to use Qualrus. But if you don't do so, the program will be much less able to act intelligently to provide you with useful advice and assistance. Let's illustrate this point with some specific example codes.

Select the 'Jonathan" code in the list at the left in this window. Then, with the "Jonathan" code displayed in detail, click the left mouse button on the "Links" tab in the Code editor window. This shows the links grid containing three columns: subject, link, and object. "Jonathan" is in the subject column. The Link column's pull-down menu contains a list of all current link types. The Object column's pull-down menu contains a list of all currently available codes. Select the Link pull-down menu and then select the "isa" link there. Finally, in the object column, select "person" from the pull-down menu there.



Subject	Link	Object
Jonathan	isa	

Once you have entered all three components: the subject, the link, and the object, you have now successfully linked Jonathan with the "isa" link to person (representing the fact that "Jonathan is a person"). The link appears in the row, and the diagram in the bottom right of the Code editor window now displays the link graphically as well.



Subject	Link	Object
Jonathan	isa	person

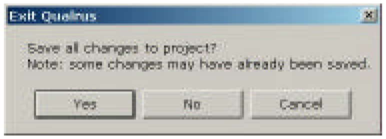
While the Links page is open for viewing, you can add a new row in the links grid by clicking on any of the fields in the bottom row and then depressing the "down arrow" key (alternatively, you can left-click on the "Link two codes" button, . This creates a new row where information for an additional link can be added. Similarly, you can delete a row and its corresponding link by first selecting any item in the row then left-clicking on the "Delete link between two codes" button, .

The graph for the currently displayed code shows that code in the center of the diagram. All of the

codes to which the code has a direct link are displayed around that code.[1] While there isn't much going on yet, you can easily imagine that for a complex coding scheme with lots of connections among codes you can end up with very complex diagrams.

To see such complex diagrams, let's switch projects now to another project in which we have continued to add codes relevant to Goffman's concepts of front-stage and back-stage behavior, and to identify links among those codes. We will use that project to illustrate how your project will look as it becomes more complete, and to show you how the program becomes better able to help you as its knowledgebase grows.

Exiting the Program



Save all changes to project?
Note: some changes may have already been saved.

When you are ready to exit a project, select the 'File" option in the main screen's menu bar again and this time select "exit." You will be asked if you want to save all changes to the project. Say "Yes." And the program will close. Let's do this now.

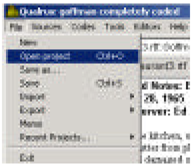
Reopening Qualrus to the same Project

To reopen a project or to open a new project, double-click the left mouse button on the Qualrus icon to begin the program. Qualrus will automatically open up to the last project you were working on. You will be taken to the main window and the source and its codes

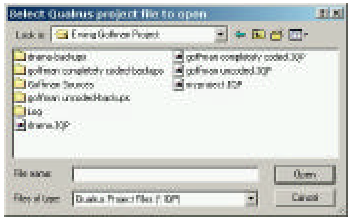
[1] Note, the code editor graph only displays links between the focus code and codes directly related to it. To see more complex links among many codes, we must use the Views Editor (described in Chapter 7 of the manual).

will be displayed there. You can quickly and easily continue where you left off.

Opening another Project



Once Qualrus is running you can switch to a different project. Let's do this now by selecting the "File" option on the main screen's menu bar then select "Open project". You will then see the window with the title, "Select Qualrus Project File to Open," shown here. Use this window to navigate to the project you want to examine next.


Select Qualrus project file to open
Look in: Erving Goffman Project
drama-backups
goffman completely coded backups
Goffman Sources
goffman uncoded-backups
Log
drama.3QP
goffman completely coded.3QP
goffman uncoded.3QP
myproject.3QP
File name:
Files of type: Qualrus Project Files (*.3QP)
Open
Cancel

For this example, let's open the "Goffman uncoded" project. You may have to move to the "Erving Goffman Project" sub-folder inside the folder where you installed Qualrus. Once you find that folder, select the "Goffman uncoded" project. This project uses the same data we were just using in "myproject" but it has many more codes and links among codes already added. In short, this project looks as "myproject" might look if you continued to elaborate your coding scheme. We want to open this more complete version of the project now so that we can show you how to assign codes to segments and how Qualrus can help you decide which codes are appropriate.

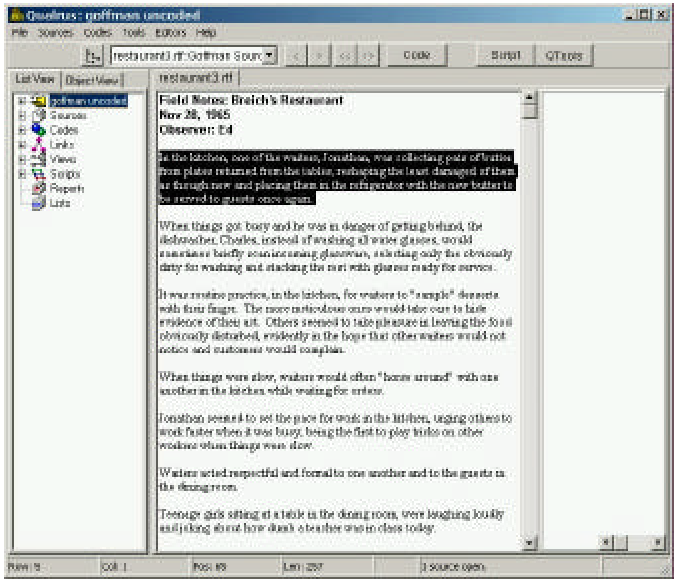
Once you are viewing the "Goffman uncoded project," select "Editors" in the main menu then select the "Codes" option to open the Code Editor for this project.

Navigating Among Codes

You can navigate among the codes in the coding scheme by double-clicking on any code in the diagram with the left mouse button. That code then becomes the displayed code and you see a new diagram with it as the center "focus" code. You can also navigate among codes by left-clicking once on any code in the list of codes in the left-hand panel. Do this now to examine some of the codes in this "Goffman uncoded" project. For example, look at the "front-stage setting" and the "hide secret indulgences" codes. These illustrate how codes can have complex links to other codes.

Selecting Segments

Now that we have several codes identified along with a few links among those codes we are ready to assign codes to particular segments in the source document. To mark a segment for coding, close the code editor window by clicking on the "x" box at the top right of the window or by selecting file and close from the main menu bar at the top of the window. If the source document is not displayed in the main window, then use the pull-down list of sources at the top of the main project window to select the "restaurant3" source. You should then see this source displayed in the center panel of the main Qualrus screen.



Field Notes: Breich's Restaurant
Nov 28, 1965
Observer: Ed

In the kitchen, one of the waiters, Jonathan, was collecting pats of butter from plates returned from the tables, reshaping the least damaged of them as though new and placing them in the refrigerator with the new butter to be served to guests once again.

When things got busy and he was in danger of getting behind, the dishwasher, Charles, instead of washing all water glasses, would sometimes briefly scan incoming glassware, selecting only the obviously dirty for washing and stacking the rest with glasses ready for service.

It was routine practice, in the kitchen, for waiters to "sample" desserts with their fingers. The more meticulous ones would take care to hide evidence of their act. Others seemed to take pleasure in leaving the food obviously disturbed, evidently in the hope that other waiters would not notice and customers would complain.

When things were slow, waiters would often "horse around" with one another in the kitchen while waiting for orders.

Jonathan seemed to set the pace for work in the kitchen, urging others to work faster when it was busy, being the first to play tricks on other workers when things were slow.

Waiters acted respectful and formal to one another and to the guests in the dining room.

Teenage girls sitting at a table in the dining room, were laughing loudly and joking about how dumb a teacher was in class today.

Row: 5 | Col: 1 | Pos: 69 | Len: 257 | 1 source open.

Identifying data segments and assigning codes to them takes place from within this main Qualrus window. Initially, this program looks much like other qualitative analysis programs in its coding view. To the left is the project tree. The file to be coded (in this case a rich-text file that can include various fonts, boldface and underlining) is displayed in a scrollable window in the middle. To the right is a linked window containing markers to indicate the location of different text segments and codes assigned to those segments. When a text segment is marked, a bracket will appear in the right "Code" window of the Sources Editor. The codes assigned to that text also appear to the right of that bracket in the 'Code" window. This bracket indicates the lines of text included in the segment.

Since segments are marked with the mouse, they can range in size from a single character to several pages of text. Text segments do not have to include all words on a line. Text segments can also overlap with one another or can be nested within other segments.

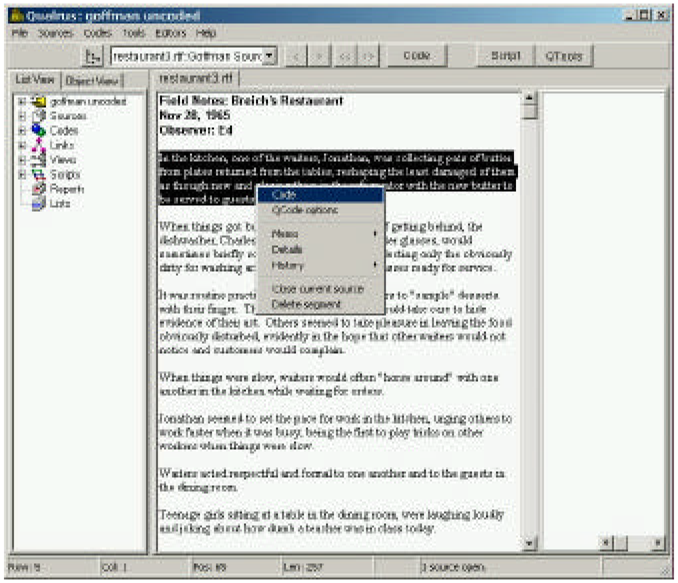
Segments can even be several layers deep with segments within segments within segments within…well, you get the picture.

When segments overlap or are nested, the brackets indicating each segment in the "Code" window are displayed in different colors. Codes assigned to a specific segment will appear in the same color as the bracket marking that segment.

A segment in the source can be marked in the same way you would mark a segment of text in most word processors. Place your mouse cursor at the beginning of the segment, press the left button down and, holding the button down, move the cursor to the end of the segment. The segment will be highlighted as you do this. When you have successfully highlighted the segment you want to mark, release the button. Note: double-clicking the left mouse button marks a complete word and triple-clicking it marks an entire paragraph. For example, let's mark the first paragraph by triple-clicking on it. You should see that paragraph become highlighted with black background and white text as shown in the screen above.

Assigning Codes to Segments

While a segment is highlighted, place the mouse over it and click the right mouse button. This will display a popup menu on top of the segment as shown below.



Field Notes: Breich's Restaurant
Nov 28, 1965
Observer: Ed

In the kitchen, one of the waiters, Jonathan, was collecting pats of butter from plates returned from the tables, reshaping the least damaged of them as though new and … with the new butter to be served to guests

When things got b… getting behind, the dishwasher, Charles … glasses, would sometimes briefly … only the obviously dirty for washing … ready for service.

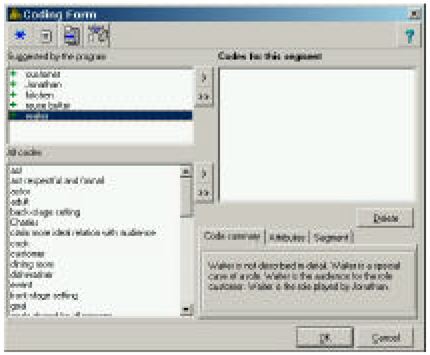
It was routine practi… to "sample" desserts with their finger. … take care to hide evidence of their act. Others seemed to take pleasure in leaving the food obviously disturbed, evidently in the hope that other waiters would not notice and customers would complain.

When things were slow, waiters would often "horse around" with one another in the kitchen while waiting for orders.

Jonathan seemed to set the pace for work in the kitchen, urging others to work faster when it was busy, being the first to play tricks on other waiters when things were slow.

Waiters acted respectful and formal to one another and to the guests in the dining room.

Teenage girls sitting at a table in the dining room, were laughing loudly and joking about how dumb a teacher was in class today.



Waiter is not described in detail. Waiter is a special case of a role. Waiter is the audience for the role customer. Waiter is the role played by Jonathan.

Select "code" from the popup menu and you will see the Coding Form window. (You can also open the Coding Form window once a segment is selected by left-clicking on the "Code" button on the bar at the top center of the main window or by selecting "Sources" from the menu bar and then selecting the "Code selected text" option.)

NOTE: If you try to open the Coding Form window without first selecting a segment for coding you will see a message in a popup window reminding you to first select a segment, then code it.

This Coding Form window contains three lists of codes. On the left at the bottom is a list of all codes available in the project currently. On the right is a list of all codes currently assigned to this segment (currently none are assigned and this list is empty). However, it is the list at the top left of the Coding Form window that is most interesting. This is the list of codes suggested by the program.

Qualrus employs a number of intelligent coding strategies to suggest codes. These include natural language understanding strategies to recommend codes based on linguistic patterns in the text of segments, embedded expert system scripts that infer one code from other codes, case-based reasoning to suggest codes based on how similar segments were coded in the past, and a series of machine learning strategies that look for recurring patterns in data then recommend codes based on those patterns. These intelligent strategies are discussed in greater detail in Chapter Six of the main manual. Here, let's simply say that between the built-in procedures for suggesting codes provided with the program, and additional procedures you can add which are specific to your own project, Qualrus can often provide surprisingly helpful advice.

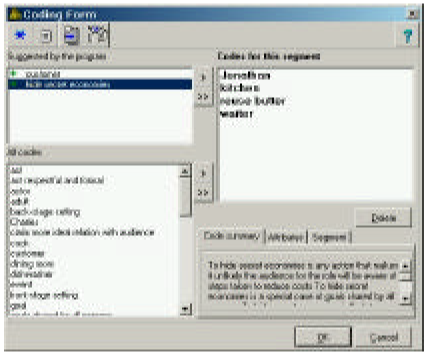
In this example, the program recommends five codes for the first segment: "customer", "Jonathan", "kitchen", "reuse butter", and "waiter." You can then select any of these "suggested" codes that you want to assign to the segment by highlighting them with the mouse and then pressing the ">" key to move the codes to the "Codes for this Segment" list on the right.

Note: You do not have to do what the program suggests. Those are only suggestions. You still make the final determination of what codes should *be assigned. You can add codes not suggested by the program. You can also choose not to add codes suggested by the program.*

Four of these codes (kitchen, Jonathan, reuse butter, and waiter) make sense. The fifth code, "customer," is understandable since the text mentions "guests." But no guests are actually present in this case, so let's not include the customer code. To accept codes recommended by the program we need to highlight them then left-click the mouse on the top ">" button (the one by the list of suggested codes) to add those codes to the list on the right of codes for this segment. Our Coding Form now looks like the following screen.

New Recommendations Based on Assigned Codes

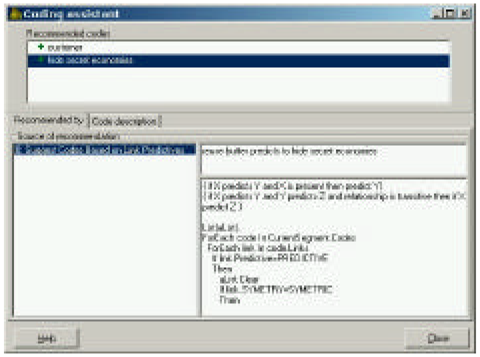
Notice that, after we selected four of the five recommended codes, those codes are now in the "Codes for the segment" list on the right. But now there are two codes in the list of codes "suggested by the program." The program has *added* another code!



This occurs because when you added codes to a segment the program now "understands" more about that segment and uses the new information to suggest additional codes. You can then decide whether to add those additional codes or not.

Explaining Recommended Codes

Qualrus not only recommends codes but also provides tools to help the researcher make a reasoned decision about whether to accept the program's advice. For example, to see why the program is recommending "hide secret economies" click the right mouse button over "hide secret economies" to open the Coding Assistant window shown here.



reuse butter predicts to hide secret economies

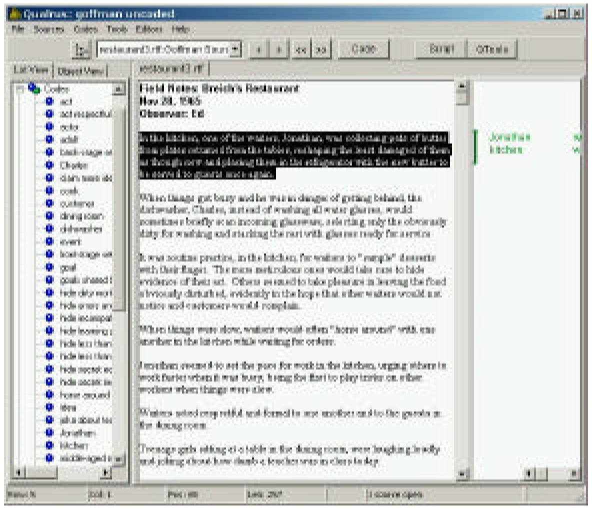
```
{ If X predicts Y and X is present then predict Y}
{ If X predicts Y and Y predicts Z and relationship is transitive then if X
predict Z }

ListaList
ForEach code In CurrentSegment.Codes
  ForEach link In code.Links
    If link.Prediction=PREDICTIVE
    Then
      aList.Clear
      If link.SYMETRY=SYMETRIC
      Then
```

This window has a list of the currently recommended codes in the top field. Left-click the mouse on any of those codes to highlight it. In the bottom field is a field containing a list of sources of the recommendation in the left column. In this case, "hide secret economies" is recommended by a script named "select codes based on link predictives." When you click the left mouse button on the source of the recommendation, details available for that source are displayed on the right. Selecting that source we see the details on the right. In the top of the right-hand column is an automatically generated text explanation of that script and in the bottom right is the complete script. The Coding Assistant form is described in more detail in Chapter Six of the Qualrus manual.

Close the Coding Assistant form, then add the "hide secret economies" code to our list of codes assigned to this segment and close the Coding Form by selecting the "OK" button. This applies the codes to this segment and returns you to the main Qualrus window as displayed here.

Note: Codes in the "Codes for this segment" list are not actually assigned to a segment until you click the "OK" button at the bottom right of the Coding Form.

Now we see a green marker in the right-hand window marking the segment and the codes assigned to the segment appear in green there as well. When you have finished assigning codes to a segment you can go on and highlight another segment for coding.



Field Notes: Breich's Restaurant
Nov 28, 1965
Observer: Ed

In the kitchen, one of the waiters, Jonathan, was collecting pats of butter from plates returned from the tables, reshaping the least damaged of them as though new and placing them in the refrigerator with the new butter to be served to guests once again.

When things got busy and he was in danger of getting behind, the dishwasher, Charles, instead of washing all water glasses, would sometimes briefly scan incoming glassware, selecting only the obviously dirty for washing and stacking the rest with glasses ready for service.

It was routine practice, in the kitchen, for waiters to "sample" desserts with their fingers. The more meticulous ones would take care to hide evidence of their act. Others seemed to take pleasure in leaving the food obviously disturbed, evidently in the hope that other waiters would not notice and customers would complain.

When things were slow, waiters would often "horse around" with one another in the kitchen while waiting for orders.

Jonathan seemed to set the pace for work in the kitchen, urging others to work faster when it was busy, being the first to play tricks on other workers when things were slow.

Waiters acted respectful and formal to one another and to the guests in the dining room.

Teenage girls sitting at a table in the dining room, were laughing loudly and joking about how dumb a teacher was in class today.

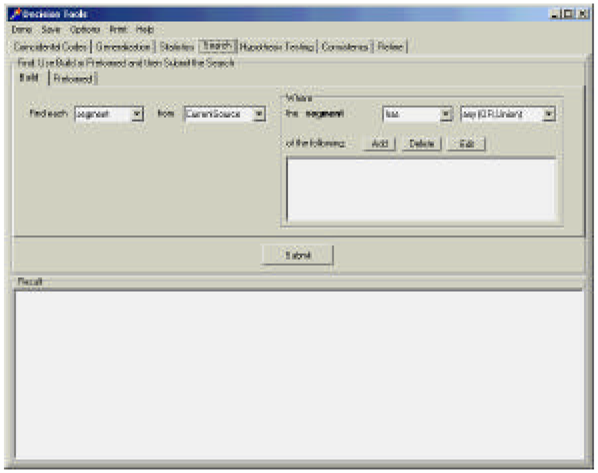
We can continue coding segments in the same manner. Go ahead and play around with the program, marking other segments and looking at the codes recommended by the program. Practice accepting suggested codes. Try adding codes from the list of all codes in the bottom as well. You may also want to familiarize yourself with the procedure for deleting codes from a segment. For more details on any of these procedures, see Chapters Five and Six in the Qualrus manual.

Analyzing the Data

Qualrus is fundamentally different from other qualitative analysis programs by using intelligent strategies to recommend appropriate codes. It is also fundamentally different from those other programs in terms of what it can do to help you analyze and make sense of the data both as you code and in the final analysis stages of the project.

To best examine the analysis tools in Qualrus, we want to open a new project. Use the file option in the menu bar at the top of the main window and select "Open project" then select the "Goffman completely coded" project. (If you are unsure of how to do this, review the procedure for opening a project above.)

Once you are viewing the "Goffman completely coded project," open the QTools window by selecting the "Tools" option on the main menu bar and then selecting "QTools" from the pull-down options. You will then see the QTools window as shown below. You can select the different tabs across the top of this window to view the various tools.


Decision Tools
Done Save Options Print Help
Coincidental Codes | Generalization | Statistics | Search | Hypothesis Testing | Consistency | Refine
First Use Build or Preloaded and then Submit the Search
Build | Preloaded
Find each segment from CurrentSource
Where
the segment has any (O R Union)
of the following: Add Delete Edit
Submit
Result

Qualrus has a comprehensive range of intelligent tools to assist researchers in analyzing the data from conducting searches, to identifying patterns in the data, to testing hypotheses, to generalizing or refining concepts.

The **search tool** permits researchers to specify complex searches, looking for segments, codes, sources, paragraphs, or sentences having specific characteristics based on Boolean combinations of codes and text.

The **hypothesis testing tool** allows the researcher to examine all segments to test specific hypotheses. Researchers can see a summary of how well the hypothesis accounts for the data and can also examine the segments that support the hypothesis and those for which it is not true.

The **categorizing tool** helps researchers identify evolving categories by creating different stacks, sorting segments into those stacks, then assigning codes to each stack.

The **coincidental codes tool** allows researchers to follow up, looking at the co-occurrences of two specific codes in segments to determine whether codes might be related to one another. Using this tool researchers can quickly see how often two codes co-occur and compare and contrast segments where both codes occur, only one occurs, or neither occur to help assess any possible relationship.

The **refine tool** helps researchers examine either codes or relationships among codes to determine whether those need to be further refined, such as when two codes may be related but only for a certain subclass of segments.

The **generalization tool** helps researchers identify codes that may be good candidates for generalization, such as when two different codes may be special instances of the same more general code or one code may be a special case of another.

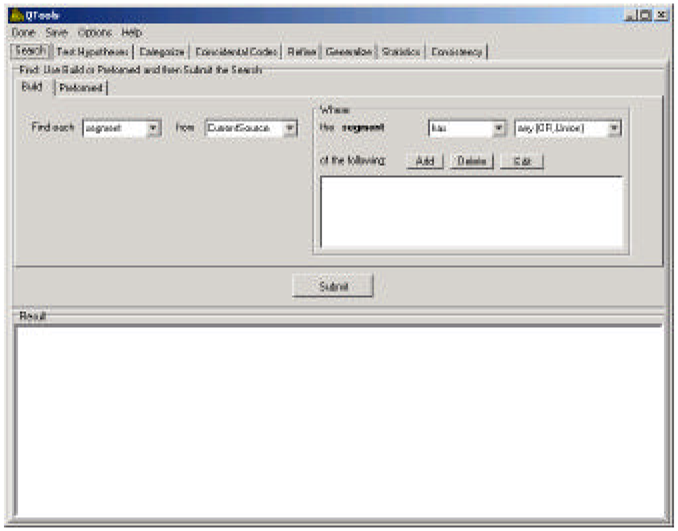
The **statistics tool** examines all codes singly and in all possible pairs to determine how frequently the code or code pair occurs in the data. Those are then rank-ordered so researchers can easily see codes that may be related to one another.

The **consistency tool** provides statistical data showing how consistent each coder is with the recommendations of the program. This information can be very helpful for assessing the training level of coders or identifying problem recommendations where the program needs to be adjusted to be more helpful.

All of these analysis tools are examined in depth in Chapter Seven of the Qualrus Manual. Here, since we are providing a quick overview of features, we will examine only two of these analysis tools: the search tool and the categorizing tool.

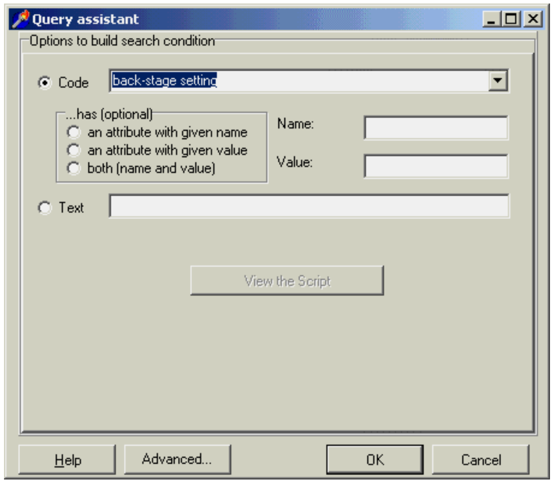
Searching for Codes and Text Combinations

To perform a search of the database we use the "search" tool. Select "Tools" and then "QTools" in the pull-down list from the menu bar at the top of the main Qualrus window. Select the "Search" tab to display the search tool form as shown here.


QTools
Done Save Options Help
Search | Test Hypotheses | Categorize | Coincidental Codes | Refine | Generalize | Statistics | Consistency
Find: Use Build or Preformed and then Submit the Search
Build | Preformed
Where
of the following Add Delete Edit
Submit
Result

Find each: segment ▾ from: CurrentSource ▾
the segment: has ▾ any (OR, Union) ▾

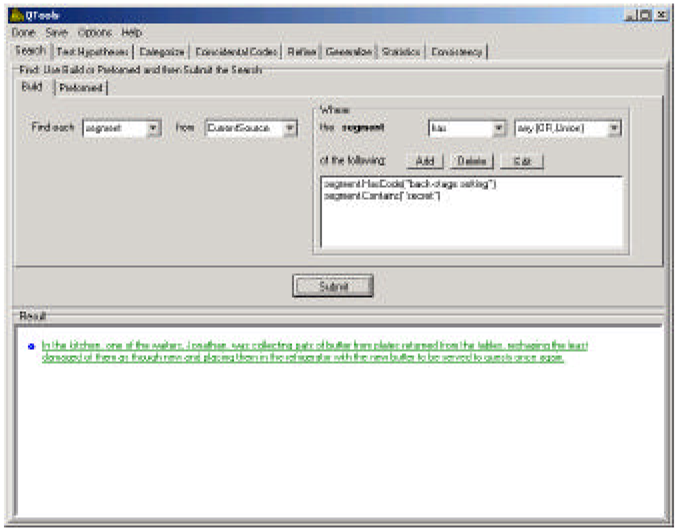
This tool can be used to perform a wide range of searches. You can search for segments, sources, codes, paragraphs, or sentences. The search can be restricted to the current source, selected sources, or all sources in the project. Let's search for segments in the current source as shown here. Here we will illustrate a fairly simple search looking for all segments having a specific code, "back-stage setting" or specific text, "secret." Click the left mouse button on the "add" button to add search criteria. This brings up the query assistant screen.



● Code: back-stage setting ▾
○ Text:

First, let's select "code" then select the code "back-stage setting" from the dropdown list of all codes. Then left-click on the "OK" button to add this condition to the search. When that is done, select the "add" button again to specify the second criterion. Now we want to add specific text, so select "text" and type in "secret" then press the "OK" button when you are done.

You should then see the search screen with the search specified as shown in this screen. You can keep adding conditions to the search criteria until you have specified the search to your satisfaction. For this example, this is enough. Next, we must left-click on the "Submit" button on this screen to actually have Qualrus perform the search. Do this now and you will then see this same screen (shown below) with the results of the search in the "Result" sub-window.


QTools
Done Save Options Help
Search | Test Hypotheses | Categorize | Coincidental Codes | Refine | Generalize | Statistics | Consistency
Find: Use Build or Preformed and then Submit the Search
Build | Preformed
Add
Delete
Edit
Submit
Result

Find each: segment from: CurrentSource
Where the segment has any (OR, Union) of the following:

```
segment.HasCode("back-stage selling")
segment.Contains("secret")
```

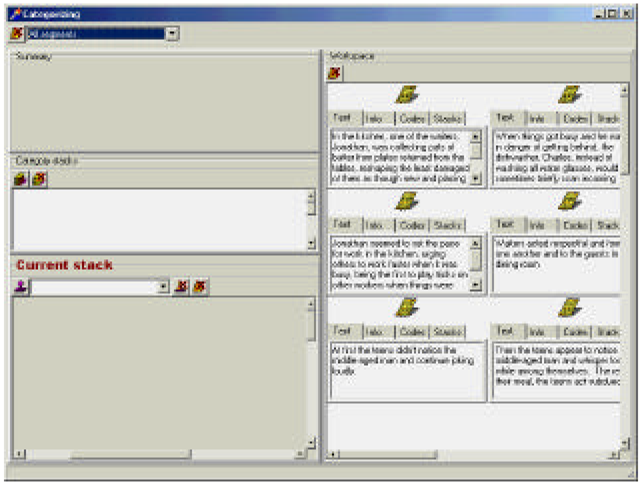
Result:

- In the kitchen, one of the waiters, Jonathan, was collecting pats of butter from plates returned from the tables, reshaping the least damaged of them as though new and placing them in the refrigerator with the new butter to be served to guests once again.

Each segment in the database that meets these criteria is displayed in the "Result" sub-window. Note that those segments are active objects that can be selected. When you click on one of the segments (in this case there is only one), the main screen of Qualrus displays that segment highlighted in the source file. This permits you to examine the segments in the context of the other information in the source.

Categorizing Segments

The Categorizing tool is opened by selecting "Tools" and then "QTools" on the menu bar at the top of the main Qualrus window, then selecting the "Categorize" tab at the top of the Categorizing window . This opens the following screen. This screen was designed on the recommendations of well-known qualitative sociologist, Howard Becker. He pointed out that in many of his studies such as *Boys in White*, and *Making the Grade* he often wanted to categorize text segments or other information into a few broad categories. He would lay out note cards on the floor with each card having a segment of text to be coded, then sort them into stacks.[2]



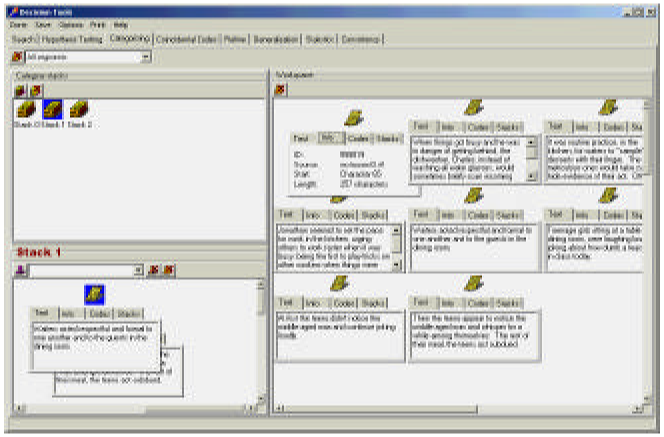
This tool lets you create as many stacks as you need, then place "sheets" representing each segment into those stacks, and apply one or more codes to all segments represented by the sheets in each stack. To use this tool, first select a list of segments for coding from the pull-down list box at the top left of this page. For this example, let's select "all segments."

Then you will see each of the segments represented as a separate "sheet" in the workspace area to the right of the window. The next step is to create stacks into which the "sheets" can be sorted by left-clicking on the "New stack" button, 📚, once for each

[2] For more information on how Howard Becker would use Qualrus for projects like *Boys in White* and *Making the Grade*, see the *Making the Grade* demo project he created on our web page.

stack you want to create. In this case, let's create three stacks by clicking on this button three times. You should then see three stacks displayed in the Category stacks sub-window. Left-click on each stack label in turn and rename them "front-stage", "back-stage", and "other." Left-click on the front-stage stack and anything in that stack is then displayed in the "Current Stack" window at the bottom left. We can drag any segment into the stack by clicking the left mouse button down on top of the icon at the top of the segment then while holding the button down, dragging the segment into the "current stack" sub-window.

By default, dragging and dropping the segment onto a stack with the left mouse button copies the segment onto the stack. If you hold down the control key while uing the left mouse button, the segment is moved to the stack and disappears from the workspace. If you want the stacks to represent mutually exclusive categories, then moving items there prevents you from inadvertently placing the same segment in multiple stacks. Let's drag two segments, copying them into the current stack. Your window should now look similar to the one below:



For example, let's sort segments into "front-stage" and "back-stage" behaviors according to Erving Goffman's definitions of those terms. Any action taking place in a setting where the audience for the role the person is playing is not present would be back-stage behavior, and if the audience is present, it would be front-stage behavior. For example, when a customer is present it would be a front-stage setting for the waiter since customers form their audience. If no customers are present (such as in the kitchen), it would be back-stage behavior. Let's put all the "back-stage" pages in the "back-stage" stack and all the "front-stage" pages in the "front-stage" stack. Any remaining pages would be placed in the "other" stack.

This tool has many advantages. Sorting pages into the stacks is very flexible. A page can be moved into a stack. Then the researcher can reconsider and move it back to the workspace and from there into another stack. The researcher can view all the pages in any one stack at a time and look for broad similarities or dissimilarities. Once the researcher has decided the stack represents a meaningful grouping of segments, they can automatically assign one or more codes to every segment represented by pages in that stack. This gives the researcher flexibility to try out categories represented by stacks before they become more rigid as assigned codes.[3]

To assign a code to every segment represented by pages in a stack all we have to do is select a code

[3] Code assignments for a specific segment can, of course, be dropped. However, for many segments this would be a tedious and error-prone task. It is far easier to drop or add a code to all segments at the same time with this Categorizing tool.

from the pull-down list of codes at the top of the current stack window in the bottom left of this screen, then, with the code to be assigned showing in the listbox, click on the "rubber stamp," , at the left of the listbox to apply that code to every segment represented by the pages in the current stack. Similarly, to remove a specific code from every segment in that stack, select the code then select the "delete selected code from all segments" button (the one with the rubber stamp crossed out on the right hand side of the code listbox).

Ready for Liftoff

If you have followed this brief introduction successfully, then you are now "checked out" on Qualrus. You should be able to do the basics involved in any project: creating a new project, adding one or more sources, creating and linking codes, assigning codes to segments, saving and reopening the project, and conducting analyses using Qualrus's tools. There is, of course, much more to Qualrus than we have been able to show you in this brief overview. You should check out the full manual. The manual is organized by tasks, so you can quickly find that part of the manual that takes you through examples using the program to accomplish specific tasks. Included with your program are all the project files used in the tutorials and other chapters in the manual so that you can follow along with your own computer as you read each chapter. Additional demonstration projects are available on our website. Those projects have been designed to illustrate a wide range of capabilities of Qualrus. If you work through each of those and follow along in the accompanying documents, you should quickly become familiar with most of the capabilities of Qualrus.

Once again, thank you for having the confidence in us to purchase Qualrus. We will do everything we can to live up to your trust by providing you with excellent service and support. We are very excited about this program and we are eager to hear how it works for you. You can contact us at our web site via email or by using the feedback forms we have in place there. We encourage you to do this. We would love to put some of your success stories in our next manual or post them on the web site and we have a system of rewards in place to thank you for those contributions. We also welcome bug reports from you so that we can continue to improve the program. Please visit our web page for program updates, news, and to download additional example projects showing how you can get the most out of Qualrus.