Containership on Forms

by Dennis Santoro

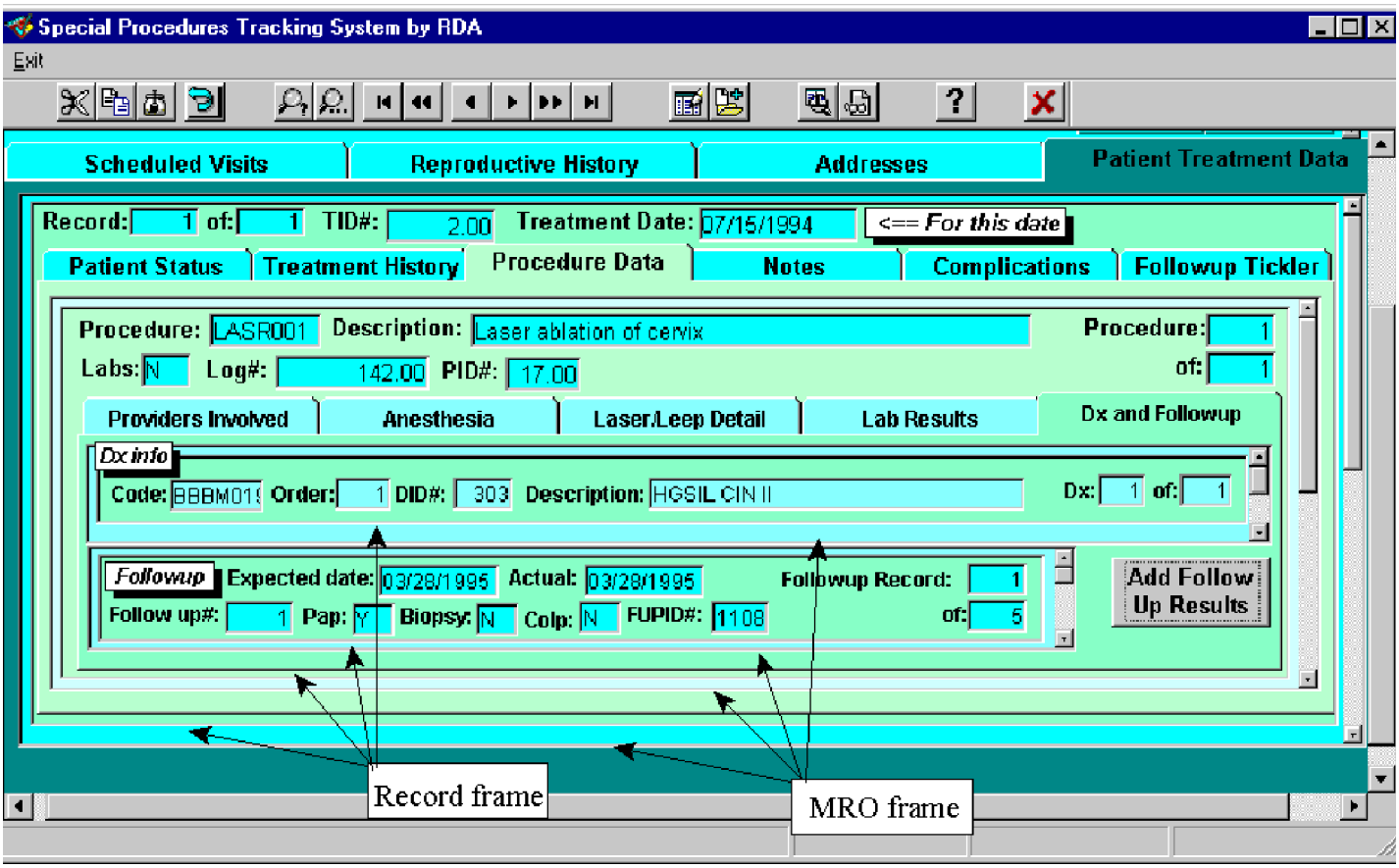
This document is designed to display some instances of containership of objects on Paradox forms.

Generally it is best to place fields on forms inside of a record object. Record objects are available in Multi Record Objects (MROs) and Table Frames (TFs). Unfortunately, the Paradox form expert will frequently place fields on a form without a record object as a container. This can cause a number of problems. The most commonly experienced problem with bare fields is the lack of implicitly posting modified records. When a user is in a record object and modifies the data, the record object automatically generates an attempted post when the user departs the record object. This behavior is not generated by field objects. So if a user creates a new record without the fields contained in a record object, and then moves to a child table, since they have not left a record object the newly created parent record will not be posted to the table. Since the lack of a post means the parent record does not actually exist yet, attempting to create a child for a nonexistent record will produce an error.

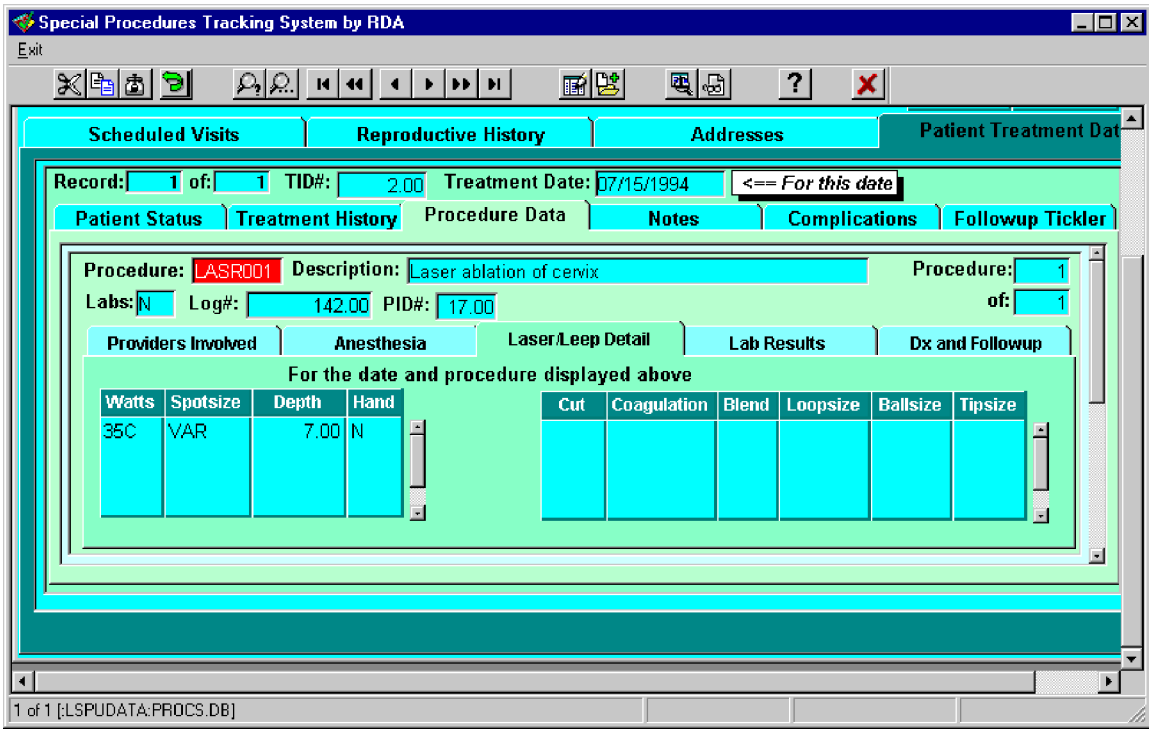
Another advantage here is the ability to call the container object to act on the table, the record or the restricted view. For example, and empty() call to a child MRO which is properly linked in your data model will empty only the records that meet the restricted view, not the whole child table to which it is connected. This is extremely useful behavior for editing and for designing cascading deletes. In many cases you will find that having a record object or container for a record object simplifies your programming since you have more built in functions to take advantage of in your form. These principles apply to reports as well although that will not be dealt with explicitly here.

In addition, when records are contained in MROs and then have child tables embedded in them, not only will the view be restricted on the child tables but the relationships among records become obvious. The graphics in this document show some of the uses and advantages of this approach to form design.

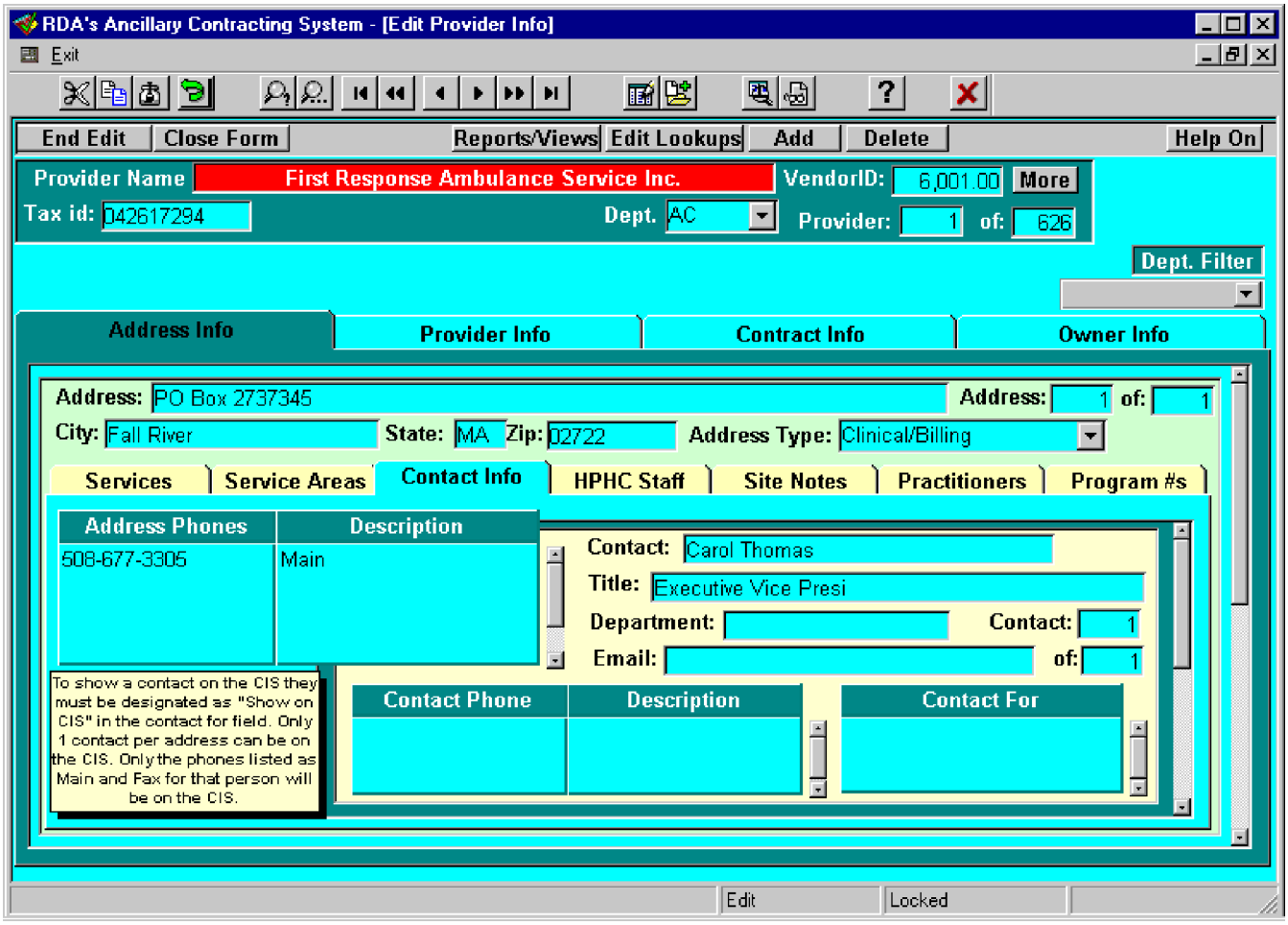
The first picture shows a Paradox form with multiple table containership. The top level is not displayed for privacy reasons but is an 1x1 MRO which contains the master record of the data model. Each page of the notebook (scheduled visits, etc. contains tables working down one branch of the data model. Displayed on the Patient Treatment Data tab is a 1x1 MRO containing treatment dates for the current patent. Contained in the record object for that MRO is another embedded notebook. Each page follows another branch of the data model. The Procedure Data page is displayed. On that page is another 1x1 MRO displaying Procedure. Embedded in this MRO is another notebook. Again the various pages follow the data model down different paths within that relationship. Displayed is the Dx and Follow page which contains two MROs that are at the same level of the data model. Note that the MRO frames are at the outside of the blue and the MRO record object frames are at the insides of the blue area. This will make it easier for you



to clearly see the containership. Clicking any of the other notebook tabs at any level here will display tables in that branch of the data model. Each notebook tab holds MROs or TFs for tables that are at the same level of the data model as the tables on the other tabs of the same notebook. Table frames can be used instead of MROs as shown in the second image.

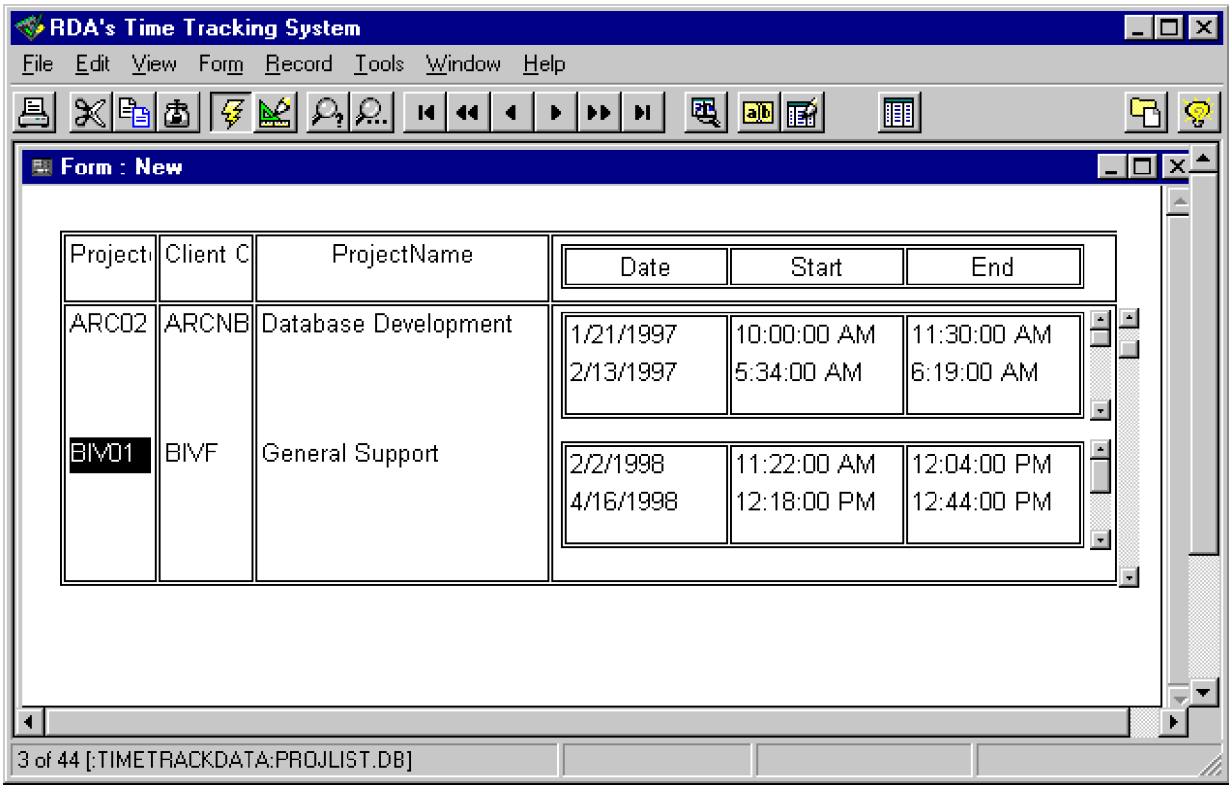


This approach can be extended in numerous ways. For example, in the form below, Address specific phone numbers are displayed in a TF that is contained by the address MRO and the Contact Info notebook tab but not by the contact MRO. This is because they are address specific and contact info related but are not contact specific. Contact specific phones are contained in the Contact phones TF which is contained in the Contacts MRO.



Additional capabilities include the ability to place a notebook in an MRO and organize the fields on various pages of the notebook by subject, editable and not editable etc. This allows greater display flexibility while still maintaining fields in a record object. You can also place boxes or other graphic organizing objects within an MRO.

Table frames are a bit more limited but a table frame can be embedded in another table frame. The next graphic demonstrates the use of one table frame embedded in another. Note that the Header of the child TF has been detached and embedded in the header space of the parent frame to keep it from repeating in display. The process of embedding table frames requires that the space you wish to embed something in be large enough to contain it. It is easiest if you create a blank column in the TF to place your child TF in. You can delete the label object and the field object from the blank column if you wish. That is recommended.



Conditions of Use and Reproduction:

This content in this paper is provided as is, with no warranties, guarantees, or claims regarding its accuracy, completeness, or usefulness. While all efforts have been made to ensure its quality and accuracy, you are solely responsible for your own use of this information.

Any statements of fact contained in this article should be interpreted as the opinions of the author, which may or may not reflect the opinions of any other entity involved in the transactions that led you to this article.

You may not distribute this information unless you meet the following conditions:

1. You obtain the permission of the author prior to such use including the specifics of what use is requested, any compensation you expect relating to use of this material. Any permissions will be deemed to be granted only for the use specifically agreed to in the document granting permission and only for the time period designated in said grant of permission. (Contact can be made via e-mail at RDAPermissions@RDAWorldWide.com. Please allow sufficient lead time).

2. All content (including this statement of conditions of use and reproduction) is provided completely unchanged.

3. Any additional conditions contained in any grant of permission are met by the user prior to any such use.

Commercial distribution can be arranged by contacting the author.

Feedback is strongly encouraged, especially constructive criticism and/or typographical/syntactical corrections. Response or action is left to the discretion of the author. Flames, abusive language, unsolicited commercial email messages (aka SPAM), and other forms of rudeness will be cheerfully ignored. Professional responses will receive priority attention. Unprofessional contact will be ignored.