



CUSTOM TABLES

WHAT ARE CUSTOM TABLES, AND WHY SHOULD YOU CARE?

As a technology, custom tables have been supported in Act! since version 2008, Service Pack #3, however, despite their relative longevity, only a minority of Act! users are aware of them – with fewer still using them. So what are custom tables, and why should we care about them?

To appreciate custom tables properly, we need to understand what Act! is without them.

Out of the box, Act! is comprised of tables for Contacts, Companies, Groups, and Opportunities, and for the most part that is where Act! users store their field data. For the majority, this type of data structure is sufficient, with people storing things like dates of birth, anniversaries, and general contact information quite adequately within these tables.

For others, however, this type of rigid structure is quite limiting, as Contacts, Groups, and Companies do not support what I call “transactional data”. To illustrate the distinction, imagine having a contact field in your database for “Last Sale”. The data in this field would continuously be overwritten by the “latest” sales data for this contact, making any sort of history recording for this field difficult, if not impossible.

This is a simple example, but to illustrate a more complex contrast between the use of rigid vs transactional data, I’m going to use the example a mortgage broker. We could just as easily use insurance brokers managing policies, or a manufacturer tracking inventory, but the mortgage broker scenario is easiest to relate to.



[SUBSCRIBE NOW](#)

[LEARN MORE](#)

Now within the contact table, a mortgage broker organizes their customer's mortgage details in the Contact Table similar to the first illustration below, and even creates similar type fields to manage up to four mortgages per contact.

Seems straight forward enough, that is until one of those mortgages matures and this broker needs to record the details of the replacement mortgage. Where does that data go? Do they overwrite the details of the matured first mortgage, and lose that history, or do they add more fields to their database? Neither hopefully.

Unfortunately, the limitations of storing transaction data within the contact table are far more serious than the example I just gave. To illustrate my point, consider the two screen shots displayed below.

Both have a Tab dedicated to mortgages, and both both allow the user to store details for multiple mortgages per customer. The differences are this. Each of the data lines in the second illustration make up individual transactions, and look like the following form when you drill down. All data points of each mortgage are contained within the form for each mortgage, and reporting is a snap.

Now contrast this with the first illustration, where a mortgage broker chose to create fields to record each mortgage's "lender", "Funding Date", "Interest Rate", "Loan Amount", and "Maturity Date", and positioned them within Contact his layout so they made sense to them visually, and seemed both organized, connected, and allow them to record details for up to four mortgages. Why would it matter if none of the fields had any logical connection with each other, if the data was intuitive to the user?

Well, as intuitive as it may seem, this broker may be chagrined to learn that after inputting all this mortgage data into Act! for hundreds of customers, they'll be unable to run any useful

reporting for their business. Simple reports, like how many mortgages they had funded, or what the quantity or value of the mortgages they're managing, would all be unavailable to them. In fact, they couldn't even accurately report on how many mortgages they've placed with a particular lender because the user was more concerned what what appealed to they're eyes, than properly recording the data?

Disaster, right? In bookkeeping terms, this would be like setting up an accounting software that limited one invoice per customer.

Contact fields were never meant to support transactional data, which is why custom tables are a much more efficient, flexible, and robust way to manage them.

So what are custom tables? Imagine a custom table as a kind of spreadsheet within one tab of your database that you can store an infinite amount of transactional information. Each time you needed to add details for a new type of data – be it a mortgage, insurance policy, vehicle, or piece of equipment – you simply clicked on "Add", and up would pop a form where you could enter all the pertinent information for that custom entity or transaction. When you clicked save and close, it would neatly add another line to your Tab's spreadsheet. Simply repeat this process as needed.

Another advantage of custom tables is, unlike Contact fields, you can view these transactions at the Contact, Company, Group, or Opportunity level, as well as viewing all of them globally.

In the case of our mortgage broker example, custom tables would allow us to click on a button to display all mortgage transactions from past or present, and from a list view easily start filtering them based on whatever criterions your reporting needs required. As the illustration below indicates, you can also perform calculations, summaries, and export them all to Excel in a way contact fields could not support.