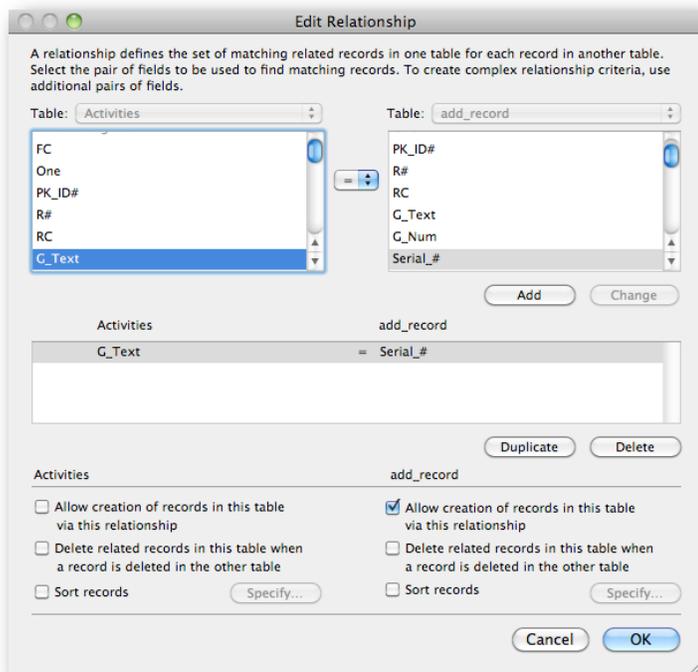


## About that portal, Captain!

There are three ways you can add records to a portal; two of which are scripted and one of which allows records to be added via the relationship which is the method we're going to discuss first. Ok, we're not going to discuss it; don't do it - it's a bad idea. "What?", I hear you say, "Why wouldn't I take advantage of something that FileMaker allows?" The simple answer to that question is that just because it's allowed doesn't mean it's a good idea. In truth, the reason why it isn't a good idea is that you end up with a ghost row at the bottom of your portal which causes a lot of confusion to lots of users, especially if you have a delete button on every row. (Some of your users will have so much fun pressing that delete button on the last row that they won't get any work done!)

So, having put that to rest, let's talk about the two scripted methods of adding related records. Which you use depends on whether you need the portal records sorted by a specific field or fields. If you do, then you create the related record by setting a variable (or global) to each the key fields of the parent record, opening a new window, going to the related table (not necessarily, the specific TO), creating a new record, setting those fields, allowing the user to enter other values in that new window, committing the record and finally closing the window. Record added, portal correct, done deal!

The second way, and which is the one I tend to use most of the time, is via a relationship, created for this specific purpose).



Here is our **portal** relationship:

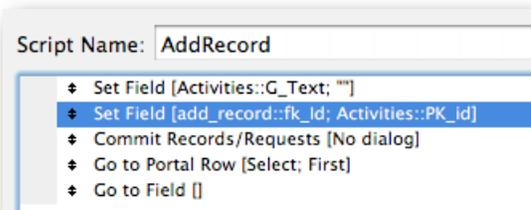
**Important:** Note that this relationship is sorted - you can have the sort on the portal itself instead of in the relationship if you want, but it must be sorted by the **serial\_#** field **descending**:



**G\_Text** and on the other side, I have a field called **Serial\_#** which is a number field that auto-enters a serial # into each new record; this one is vitally important - it must enter a serial #. The relationship, which we'll call **Add\_Record** goes from **G\_Text** to **Serial\_#** and allows for creation of related records.

Remember that the only purpose of this relationship is to create the related record.

In our script, we first set **G\_Text** to "", then we set the first of our key fields in the relationship. Let's say, for example, that the portal relationship uses **pk\_id = fk\_id** (Primary key = foreign key). Thus in the **Add\_record** relationship, we are going to set the FK\_ID field with the PK\_ID value from our parent table. Here is the script.



So, what just happened? Well, the related record has been created. For some reason, and I have no idea why this happens, because the **G\_Text** field is set to "", the relationship sees that it needs a value and sets it to the value of the **serial #** field in the child table, thus creating a valid relationship. With the record created, we commit the record (which is **vital**) and go back to our portal where the newly created record is at the top of the portal, providing that you've **remembered** to sort that portal by the **serial\_#** field descending.

