

This article first appeared in the March 2008 issue of
Novell Connection magazine.

A Better Base

*Building your carefree database in OpenOffice.org—made simple
with forms*

by BrainStorm, Inc.

In the first article in this series on OpenOffice.org Base, I hoped to tempt you into trying out this database app that's included in the OpenOffice.org suite. In the most recent article, I explained Subforms. But now it's time to take a step back and look at just plain old forms for those of you who aren't as familiar with databases.

As you know, databases can seem pretty daunting at first. If you're new to the concept, you might even find that your first attempt at creating one is less than successful. But don't let that deter you. A properly functioning database will save you a lot of time in the long run. This is especially true when it comes to the forms feature of OpenOffice.org Base.

Inputting large amounts of data would be more than just a little frustrating if it weren't for forms. Since you know how to build tables, create unique fields, link common fields from separate tables, and from the last article, create subforms, let's take a look at populating tables with data using simple forms. Obviously, this can be done by manually entering data in each table; but forms are really what you'll want to master.

The Scenario

Let's review our scenario: You're the IT director for a company with a slew of employees, and you have a lot of hardware to manage. Therefore, you're going to build an inventory database to help you manage and track the printers, laptops, desktops, scanners, fax machines and other important equipment floating around your company site. You want to include serial numbers, purchase dates, user information, vendor details and lots of other data about each item. You're also just starting out, so none of this information is really in one place, meaning it can't be easily exported from a spreadsheet to the database; we'll cover that in another article. What you need is a simpler, more straightforward way of getting the right information into your tables. (For more information on more advanced form features, check out the most recent article, [Great Form: Discovering the Power of a Subform](http://novell.com/connectionmagazine/2008/01/tech_talk_04.html), novell.com/connectionmagazine/2008/01/tech_talk_04.html)

Making Refinements

In the first article, I recommended that you carefully plan each table for your database. Then I showed you how to create those tables. And now it's confession time—and advice-giving time—after setting my database tables aside for a few days and spending some time boning up on forms, I decided that my tables needed a little tweaking.

Why? Well, sometimes a table contained information it didn't need, and other times two tables contained too much redundant information. I simply wasn't being specific enough. But that's okay. It's also exactly why I suggest that you create your tables, set them aside for a few hours—or even days—then come back to them for some fine tuning.

For example, I made these tweaks to the Status table: I added a few fields, namely the maintenance and comments fields, where IT employees could make notes about a certain product or provide details and dates about the maintenance history. I also added a description field, where I could enter a simple term to identify each product (laptop, desktop, etc.).

I changed the Manufacturer table to a Vendors table because what I really want to know is *where* we purchased the product. The manufacturer is secondary to this information—and I added a field called make/model for it to the Status table.

At this point, I realized that the Inventory Details table was, at best, redundant, so I scrapped that too and moved vital fields from it to the Status table.

For the most part, this is what I had left, and what we'll use to move forward:

- **Product Status table**, with fields for each product's description (laptop, desktop, etc.), the product's serial number, the product's make/model, the vendor's ID, the purchase date and price, the user's employee ID number, the estimated product life, the current condition (new, fair, poor) and fields for comments and maintenance issues. (See Figure 1—Product Status Table.)
- **Employee/User table**, with fields for the employee's ID number, first and last names, department (marketing, sales, accounting, IT, product development, etc.) and phone extension, as well as the serial number of each product that employee is using. (See Figure 2—Employee/User Table.)
- **Vendors table**, with fields for each vendor's ID, name, address, contact name, phone number, e-mail address, etc., as well as the field for notes that replaced the old field for discounts. (See Figure 3—Vendors Table.)
- **List tables** for the department and condition fields in the first two tables

Description	SerialNumber	Make	VendorID	DateAcquired	PurchasePrice	EmployeeID	EstimatedProductLife	Comments	Maintenance	Condition

Figure 1: This is the main table in our IT Inventory Database. Several of the fields in the table, such as those for the Serial Number, Vendor ID, Employee ID, and Condition, are linked to other tables.

EmployeeID	FirstName	LastName	Extension	DepartmentID	ProductSerialNumber(s)

Figure 2: This table is a secondary table in the IT Inventory Database. It is linked to the main table through the Employee ID field.

SupplierID	SupplierName	Address	City	StateOrProvince	PostalCode	CountryOrRegion	ContactName	ContactTitle	EmailAddress	FaxNumber	PhoneNumber	Notes

Figure 3: Like the User/Employee Table, the Vendors table is secondary to the main table. It is linked to the main table through the Supplier ID field, which shares common information with the Vendor ID field in the main table.

Gathering the Data

As my database grows, I may find that I'd like to add additional tables; but my top priority now is to populate the tables I do have. One way to do this is to simply open the table and begin entering the data. You can certainly choose that option. To do so:

1. Open the database that contains the desired tables.
2. Select Tables from the Database column at the left of the screen.
3. Double-click the main table, or, alternately, right click the table and select Open.
4. Begin entering your data, pressing Tab to move to each new field and Enter to move to the next record. (See Figure 4—Manually Populating Tables.)
5. Click the Save button in the toolbar after entering each record.

Doing it this way works, but it can also be very tedious, not to mention fairly inefficient.

Description	SerialNumber	Make	VendorID	DateAcquired	PurchasePrice	EmployeeID	EstimatedProductLife	Comments	Maintenance	Condition
Laptop	1234567	MS-122	121	12/07/07	1900.00	12345	3 years	system restore necessary		New

Figure 4: If you have only a few tables in a database, with relatively little information needed in each table, you can easily populate your tables by opening them and entering the data by hand. Simply tab over from field to field. Remember to save your data before closing the table.

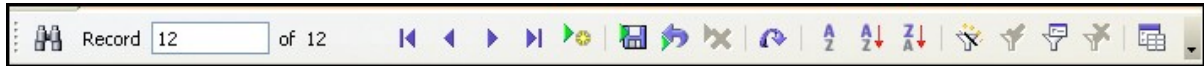


Figure 5: The Form Navigation Toolbar employs VCR-like buttons to help you scroll through each record. It is also the place to go to save, delete, undo, or further control a record while entering data.

The most efficient means of filling those tables with data is to create data entry forms to help you along. At their most basic, forms are straightforward and can be quickly created with the Form Wizard. I recommend you start there.

Creating a Basic Data Entry Form

In our scenario, one of the more basic tables is the Employee/User table. If you use the Wizard to create a form, you could delegate the data gathering and have an intern quickly gather some information from Human Resources and enter it in the form, thus populating the table. To create the form:

1. Open the database that contains the Employee/User table.
2. Select Forms from the Database column at the left of the screen.
3. Click Use Wizard to Create Form.
4. Select the table you want to use (in this case the Employee/User table) from the *Tables or queries* field.
5. Select the fields you want to appear in the form and move them to the *Fields in the form* pane. It's a good idea here to put the fields in the order you want them to appear in the form. Moving them around later can cause more trouble than you'd expect.
6. Click Next to move to number 5 in the *Steps* pane at the left of the Wizard. (We'll cover Subforms in our next article.)
7. Make selections for steps 5 through 8 as desired. These steps will allow you to design how the form looks—including how each record displays and the colors and visual effects used—and what it is named.
8. Click Finish.

Entering Data in a Form

The next step is to enter your data, which is really quite straightforward: Enter the data in the first field, press Tab to move to the next field, enter more data, and keep tabbing and entering data until you've finished entering data for that record. One of my favorite features is that each new record is saved automatically when moving to the next record. Of course, you can also click the Save button on the Form Navigation toolbar at any time to make sure each record is saved.

Speaking of the Navigation toolbar, it's imperative that it be displayed whenever you're entering new data or even just looking at a form. Use the VCR-like buttons on the bar to move to the next record, find an existing record, delete an entire record, and so on. The toolbar is accessible by clicking View | Toolbars | Form Navigation. (See Figure 5—Form Navigation Toolbar.) You can also use the toolbar to sort data in a large form, filter data or to view the form as a table while entering the data.

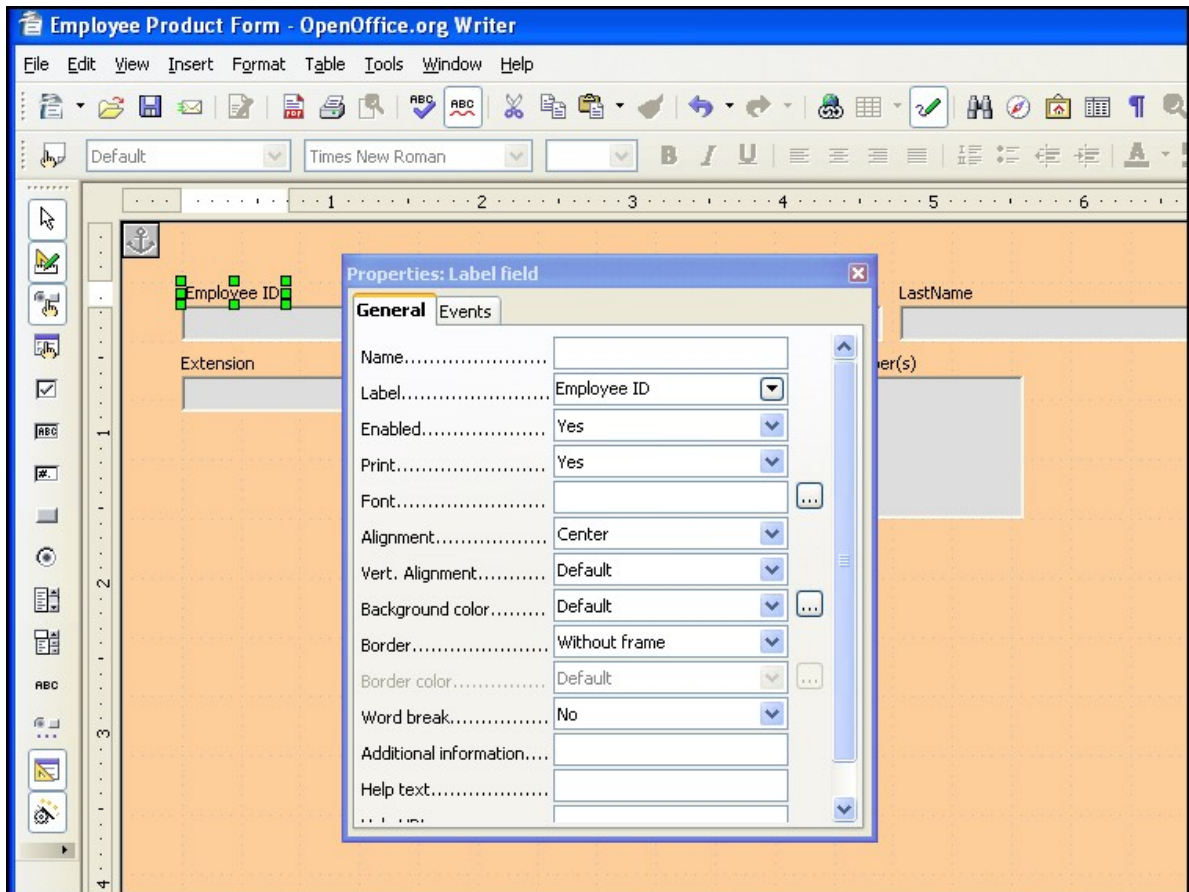


Figure 6: You'll likely want to change the name of many of the labels on your forms. This is because the label names come directly from your tables, which require that a field be named with only one word.

Editing a Form

If, like me, you open a form and find that it's really not designed the way you'd hoped, like the field for a serial number is too short or you decide to change the name of a field, you can edit it quite easily, even if you've already begun entering data. This is nice for a variety of reasons. For example, Base dictates that table fields be one—and only one—word; but that's no reason to make the person doing the data entry look at a field that reads *SerialNumber* instead of *Serial Number* all day long.

To edit a form:

1. Open the database that contains the form you want to edit.
2. Select *Forms* from the pane on the left.
3. Right click the desired form and select *Edit*.
4. The form will open, displaying only the fields and labels; all data will be hidden.
5. Edit away.

Here are a few important edits you'll likely have to make at some point:

Changing a Field Label: Move your cursor over the

label you want to change and Ctrl-click the left mouse button.

The label alone will be selected. Right click the label and select *Control*. The properties box for the label field will open, in which you can enter a new name in the *Label* field, change the font or

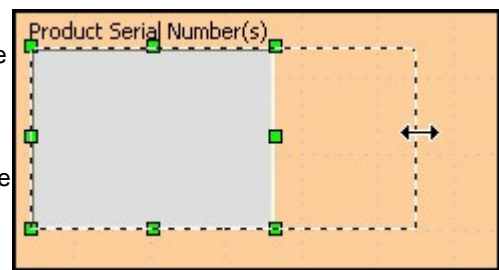


Figure 7: After creating a relationship, a line will display between the common fields, indicating the type of relationship. This is a one to one relationship.

alignment of the label, and more. (See Figure 6—Labelfield Properties.) Click the Exit X in the top right when you're finished. Save your changes.

Resizing a Field: Move your cursor over the field you want to resize and Ctrl-click the left mouse button. The field alone will be selected. Move your cursor to one of the green handles, press the shift key, and use the mouse to resize the field as desired. (See Figure 7—Resizing a Field.) Save your changes.

Moving a Field: Move your cursor over the field you want to move. Click and drag the field with its label to the desired position on the form. Save your changes.

Beware, though! Moving fields can cause problems on occasion. Let's say, for example, that your original form contained these fields in this order: title, first name, last name, address, state, city, zip code. You didn't realize when you set up the form that you had put the field for state before the field for city, so you edit the form by moving the fields around. But when you start entering data in the form and press Tab after each field entry, the cursor still moves from title to first name to last name to address to state to city, etc. Your problem wasn't really solved. That's because moving a field in this fashion also requires resetting the tab order.

Resetting the Tab Order: Move your cursor over the field in question and Ctrl+click the left mouse button. The field alone will be selected. Right click and select *Control*. Change the number in the *Tab order* field under the *General* tab to reflect the correct location. If the state and city were reversed, as described above, you'd change the 6 in the City field to a 5 and the 5 in the state field to a 6. Close the window and save your changes.

Changing the Field Type: Move your cursor over the field in question and Ctrl+click the left mouse button. The field alone will be selected. Right click and select *Replace with*. Make your selection from the context menu that appears. (See Figure 8—Field Types.) If you change the field type, it's likely that you'll also want to edit the field's appearance to ease data entry. For example, you can select the field alone, right click, select *Control*, and then make edits under the *General* tab that would add things such as calendars, formatting parameters for currency and dates, and so on. (See Figure 9—Creating a Date Field.) Close the window and save your changes.

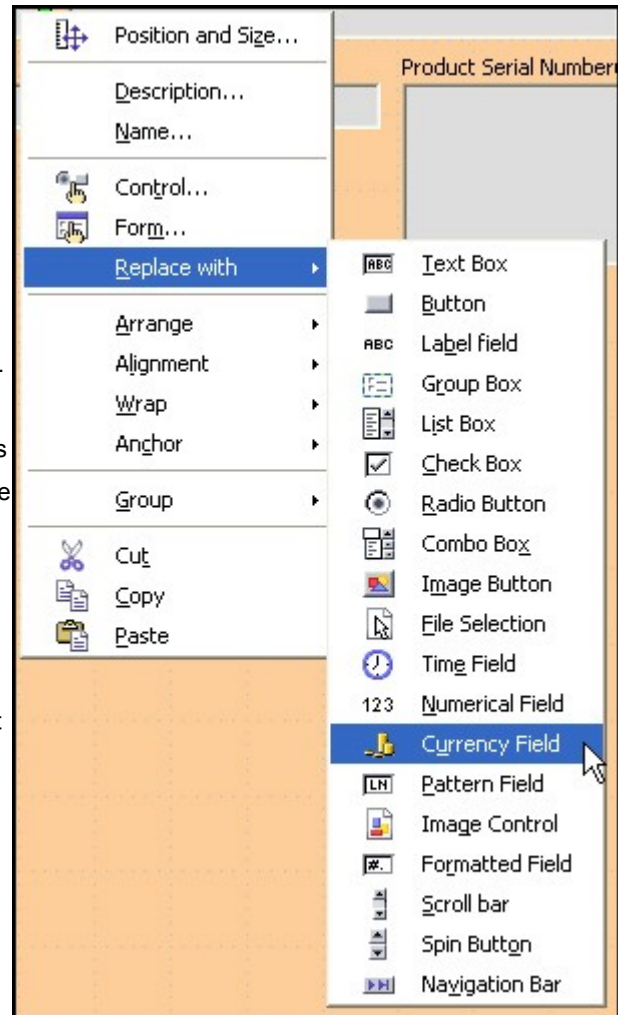


Figure 8: Selecting a field type for each field in your table will help you create tables that provide the specific information you want to relay,

Where Did It Go . . . and What's Next?

Once you spend a few hours learning just what a form can do, save your form again. Saving is key in Base. After saving a form and returning to the main database screen, always click the Save button on the toolbar;

otherwise, you risk returning to the program and finding that your form no longer exists. The form is accessible only through the database with which it is associated.

For now, especially if you're new to database creation, I recommend stopping here and practicing for a while. It may not seem like you've learned much, but you're actually well on the way to having a working database with simple forms for data entry and simple tables populated with basic data. I use the terms *simple* and *basic* because databases with multiple and many-layered relationships can be tricky. (I suppose it's really true that no relationship is easy.) So, practice away and if you're one of those just beginning to work with databases, when you've mastered these skills, check out the [article on Subforms](http://novell.com/connectionmagazine/2008/01/tech_talk_04.html) (novell.com/connectionmagazine/2008/01/tech_talk_04.html) and other deeper relationships in the previous article. Those will take you to an entirely new level. **N**

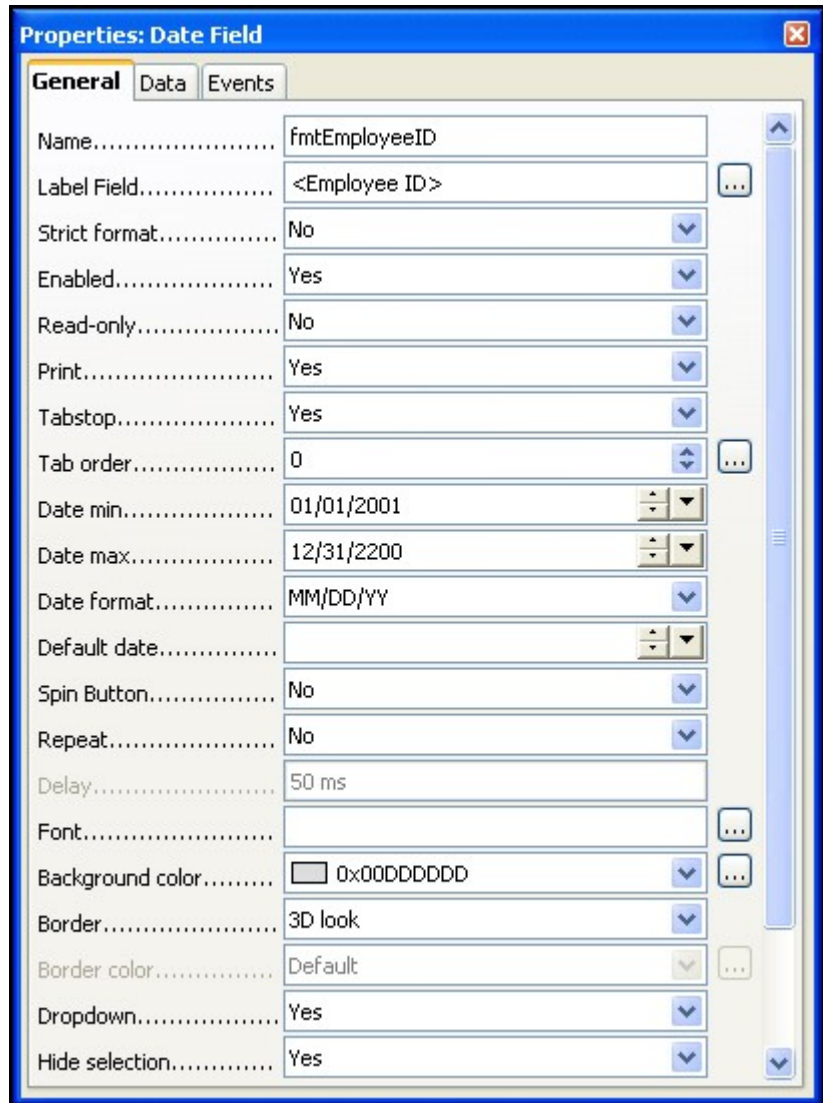


Figure 9: By selecting a field, right clicking, and choosing Control, the Properties box displays. Here you can make changes to improve data entry. This image shows that the user has made changes that will place a drop-down arrow in the field, add parameters for minimum and maximum date entries, and set the date format to MM/DD/YY.