



Create Powerful Bound Forms in Access

With this technique, you can use bound forms in Access and avoid the performance hit that can occur from having a form bound to a large recordset.

By Mark Liquorman, Access Developer, Liquorman Inc

I prefer to use bound forms in Access. However, I'm aware of the performance penalties that arise from having a form bound to a large recordset, especially over a network. Therefore, I try to have only a single record at a time in my recordset. The following technique seems to handle this well.

I start with a form (frmInvoices) for handling data from my Invoices table. I place an unbound control (txtPrimaryKey) on the form. The form's RecordSource property is:

```
SELECT * FROM Invoices
WHERE InvoiceNo = Forms!frmInvoices.txtPrimaryKey
```

When the form opens, the recordset contains no records, because txtPrimaryKey contains no text. To display a specific invoice, enter the invoice number into txtPrimaryKey, whose AfterUpdate event contains this single line of code:

Me.Requery

If the selected invoice number is in the table, the invoice displays; otherwise, a new record displays. If you use another control or form to search for an invoice (say, by date or customer), you merely have to place the selected invoice number in txtPrimaryKey and tell the form to requery; in fact, txtPrimaryKey doesn't even have to be visible.

You can also easily put in command buttons to move about the table. Moving to a new record is just a matter of using code like this:

```
txtPrimaryKey = Null
Me.Requery
```

Moving to the last invoice in the table is just as simple:

```
txtPrimarykey = Dmax("InvoiceNo", "Invoices")
Me.Requery
```

So, I get all the benefits of using bound forms, but few of the penalties.