

Excel to Access with AccessLinks

Peter Vogel



Excel has an add-in that lets you link an Access database to an Excel spreadsheet with a few mouse clicks. With a basic connection established, you can build as complex a front end as you want. Peter Vogel shows you how.

DIFFICULT as I find it to believe, there are people who don't use Access for everything. Many of those people use Excel not only as a spreadsheet but also as a database. Unlike Access, however, Excel doesn't offer the reporting or data entry features that Access includes as a matter of course.

With Excel 2000, users can integrate their favorite tool with the power of Access through a simple add-in. The underlying technology, called AccessLinks, allows Excel users to create Access databases, forms, and reports from a menu choice. What's relevant for Access developers is that AccessLinks provides a quick and easy way to integrate Access forms and reports with Excel data.

Linking

Once you add the AccessLinks add-in to Excel 2000 or 2002, users have access to three new options on the Excel Data menu (see the sidebar, "Getting AccessLinks"). The Convert to MS Access option takes a selected area of an Excel spreadsheet and exports it to either a new or existing Access database (see Figure 1). This is a one-way transfer and is intended to allow

users to move an application from Excel to Access permanently. One warning: When exporting to an existing database, the wizard requires you to enter the full pathname even if the MDB file is in the same directory as the Excel spreadsheet.

Exporting Excel data to Access isn't for the faint of heart. Selecting the Convert to MS Access menu choice actually loads Access into memory and calls Access's Import command. The user is then walked through Access's Import Wizard. This requires the user to answer questions about field size and type—not decisions that you'll want your users to make in a production system. Once you've finished exporting your data, however, the resulting database is bound to the spreadsheet. If the user now works with either of

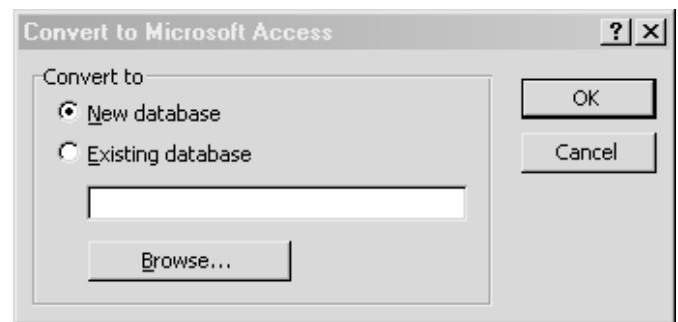


Figure 1. Exporting a spreadsheet to an Access database.

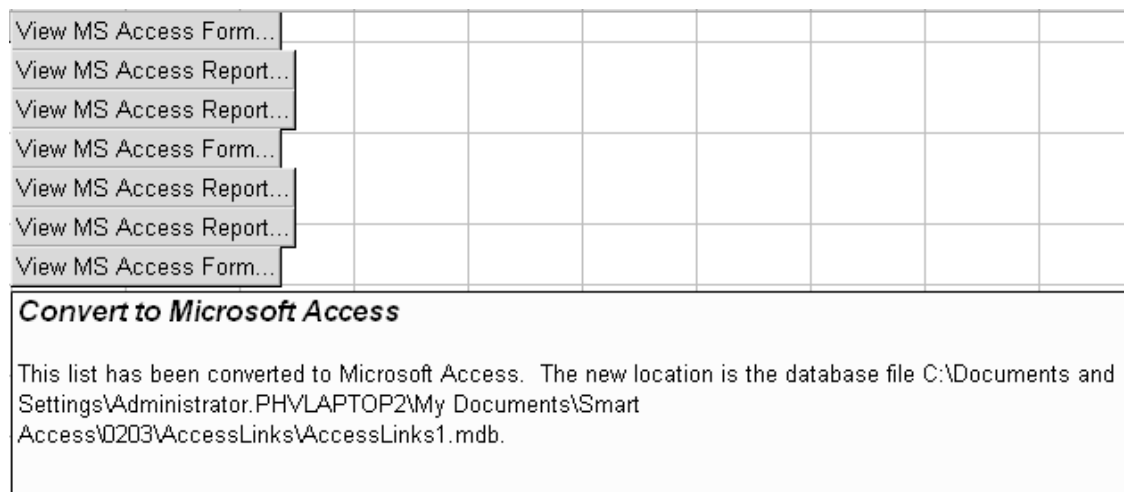


Figure 2. Access report and form buttons added to an Excel spreadsheet.

the other two commands (MS Access Form or MS Access Report), the resulting forms or reports will automatically be directed to the same database.

While the convert functionality isn't particularly useful to developers, the other two commands can be. The MS Access Report and MS Access Form menu choices both do roughly the same thing: start Access, insert a link to the data into the database, and start either the Form or Report Wizard. Again, you probably don't want your users generating forms in a production system. However, giving users the ability to use the Access Report Wizard from within Excel might be a good idea. Once either process is complete, a new button is added to the spreadsheet for each form or report that's created (see Figure 2, on page 21). As Figure 2 also shows, the Convert to MS Database option also adds a text box to the spreadsheet documenting what database the spreadsheet is working with.

If a user clicks on one of the buttons that's been added to the spreadsheet, Access is loaded into memory and the report or form is then run. If it's a form that's at the other end of the button, the user can update or enter data from the form into the spreadsheet. If it's a report that's loaded, the user gets a print preview of the report.

Once the link is established, you can use it from either Access or Excel. For instance, a user can open the Access database and generate the reports from within Access using the Excel data; alternatively, the user can open a form and update the Excel spreadsheet. At this point, an Access developer is in a position to let users continue to keep their data in Excel but get the reporting and data integrity benefits of Access.

The reports and forms created by AccessLinks are, of course, limited to what you can create with Access's Form and Report wizards. However, once the forms and reports are created and linked to the Excel

Getting AccessLinks

To download the Excel add-in, go to <http://office.microsoft.com/downloads/2002/aclnk.aspx>. Once you've run the download, you'll have a file called aclnk.exe. Running the executable will install the add-in's file (AccLink.xla) to any directory that you want. The default is the add-in directory for Excel, probably the best place to put it. This doesn't make it possible for you to use the add-in, however.

To actually use the add-in, you'll need to add AccessLinks to Excel through the Add-In Manager from the Tools | Add-Ins dialog. When it appears, you should find the Microsoft AccessLinks add-in in the list (see Figure 3). Checking off the AccessLinks add-in on the list and then clicking on the OK button

will integrate the add-in with Excel. To determine whether the integration was successful, check the Excel Data menu. It should include three new items as shown Figure 4:

- MS Access Form...
- MS Access Report...
- Convert to MS Access...

To remove the add-in, just bring up the Add-Ins Manager and clear the AccessLinks check box.

Figure 3. The Excel Add-In Manager showing the AccessLinks add-in.

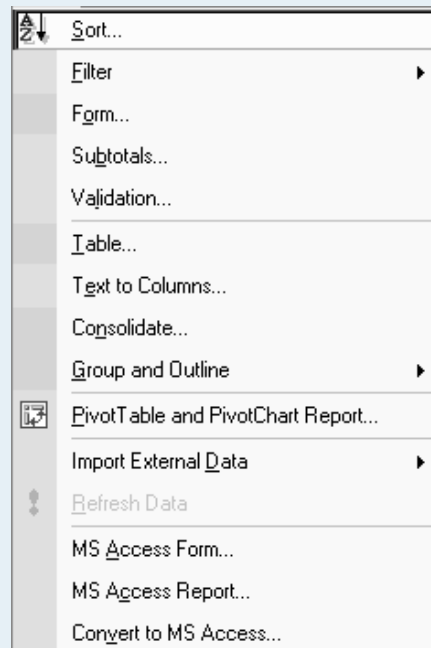
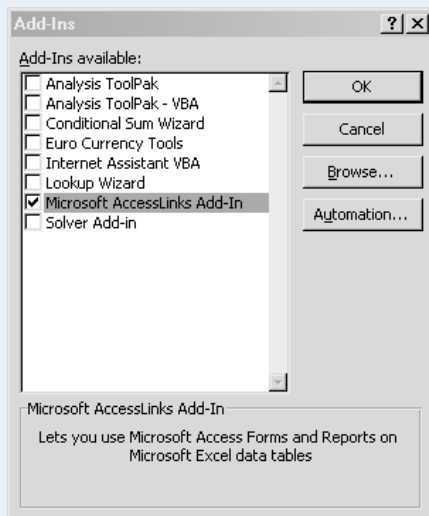


Figure 4. The results of adding AccessLinks to Excel.

spreadsheet, there's nothing stopping you from modifying the initial forms. You can add code or modify the properties of the controls on the form. You can also extend any reports generated by the AccessLinks add-in to give you a more sophisticated output. Finally, with the link made, you can add new forms and reports to the database. If the users don't need to get to this functionality from within Excel, you can delete the buttons from within Excel. If you want to leave the buttons in place, it would be a good idea to replace their default caption ("View Access Report") with the name of the report or form to run.

What you can't do is add more buttons to the Excel spreadsheet to link to any new forms and reports. To link an Access form or report into the spreadsheet, you'll need to use the Excel menu choices to generate the initial form or report. The AccLink.xla file that holds the code for the AccessLinks menu choices is password protected, so there's no way to modify the code in the add-in.

Regardless of the benefits to users of the AccessLinks add-in, there can be real benefits to developers in linking Access to Excel data. By

running the add-in, you can quickly link an Excel spreadsheet to an Access database. You can also generate a basic form or report from which to begin your development using the add-in. For Access developers who need to get to Excel data, AccessLinks is a quick and easy way to begin. ▲

 [ACCLNK.ZIP at www.smartaccessnewsletter.com](http://www.smartaccessnewsletter.com)

Peter Vogel (MBA, MCSD) is the editor of *Smart Access* and a principal in PH&V Information Services. PH&V specializes in system design and development for systems that use Microsoft technologies. Peter has designed, built, and installed intranet and component-based systems for Bayer AG, Exxon, Christie Digital, and the Canadian Imperial Bank of Commerce. He's also the editor of Pinnacle's *XML Developer* newsletter and wrote *The Visual Basic Object and Component Handbook* (Prentice Hall, currently being revised for .NET). In addition to teaching for Learning Tree International, Peter wrote its Web application development, ASP.NET, and technical writing courses, along with being technical editor of its COM+ course. His articles have appeared in every major magazine devoted to VB-based development, can be found in the Microsoft Developer Network libraries, and will be included in Visual Studio .NET. Peter also presents at conferences around the world. peter.vogel@phvis.com.

Tip: Complex Report Selection

Peter Lintzeris

Many of my clients request a lot of flexibility in the criteria they can use to print a report. I give them a selection screen with combo boxes to allow them to choose the criteria that will control the data in their reports. I used to check whether a selection had been made and then string together a Where clause for the SQL statement that contained each criterion they'd entered. This can become very tedious when a form has multiple selection criteria. I'm now using a much simpler method: For each criterion I just create a global variable and function.

A simple selection form, for instance, might contain a "client code" combo box and a "state code" combo box. The users may select by client alone, by state alone, or by state and client. To support the client code combo box, I dimension a global variable called gvClientCode as a variant. I then create a function called GetgvClientCode that returns either an asterisk (*) if no value has been selected in the combo box, or the value in the combo box if it's been set:

```
Public Function GetgvClientCode()  
  
If IsNull(gvClientCode) Or _  
    gvClientCode = "" Then  
    GetgvClientCode = ""  
Else  
    GetgvClientCode = gvClientCode  
End If
```

End Function

I call the appropriate function in the AfterUpdate of each combo box on the selection screen. I also set all global variables to Null when loading the selection screen and frequently provide buttons on the selection screen to clear all combo boxes and set all of the global variables to Null.

Each report's Row Source property is based on a stored query. The criteria row for each field in the query calls my functions. In the query, the ClientCode field's criteria row contains the test:

Like GetgvClientCode()

If client code has been selected on the selection screen, the query is filtered by client code. Otherwise, the query will return all clients because the value returned from the function is the asterisk, giving:

Like *

I can have as many of these global variables/functions/query criteria as I want or need. Users can select any combination of values that they want. The result is simple code and tremendous flexibility. ▲