

The Twenty-Something Rules For Access Development

Compiled by Bill Mosca from the thread “Minimizing Access Grief” started by Jack Malki at http://tech.groups.yahoo.com/group/ms_access_professionals . All rules are notated with their authors.
Last Edited on 21-Sep-2008.

1. (Jack Malki) Disable Name AutoCorrect (Options>General Tab).
Name AutoCorrect only changes names in tables and queries. Any instance in Code is ignored. Therefore the changes could be disruptive. Name AutoCorrect has also been identified as a source of database corruption in Access 2000 without SP3. To disable Name AutoCorrect, click on Tools>Options>General tab. This must be done for EACH database.
2. (Jack Malki) In each table's properties, set Subdatasheet to [none].
Subdatasheets force Access to check all the relationships before loading the database. This can really slow up loading time. To set the Subdatasheet property, open the table in design view and click on View>Properties. Change [Auto] to [none]. Note: that setting will switch back to Auto again if you open the table in design view. To prevent this, implement Rule #1.
3. (Jack Malki) For all form properties, AllowDesignChanges is set to Design View Only.
It's an unprofessional slipup if the properties page should show when a form is opened by the user. Since you would never allow the user to change your design, there is no need for him/her to see the properties.
4. (Jack Malki) Never use the Lookup Wizard.
*It can create unnecessary indexes, force reports to use combo boxes instead of textboxes thereby slowing up load time, the actual field being bound is not apparent, queries will not sort properly, and much more.
See also: <http://www.mvps.org/access/lookupfields.htm>*
5. (Jack Malki) Don't use ActiveX objects.
Users will probably not have the same version of the object as you have, causing your application to blow up. They may not even have the .ocx file so you will have to include it in your installation.
6. (Beth M) Reference Database objects explicitly as either DAO or ADODB.
If/when you make the transition to ADO (or find you need to utilize something in ADO) in the future, you won't have to trace backwards and correct throughout the database to work in a mixed environment. Ex: Dim db As DAO.Database
7. (A.D.Teipal) Always ensure that dates are in a non-ambiguous format (e.g. dd-mmm-yyyy).
Relying on the system format for dates can lead to ambiguity when dealing with portability across different countries. Failing to do this can also break your code that builds and executes queries. The query engine requires short dates in mm/dd/yyyy format. When you concatenate a date value in an SQL string, VB will use the default short date format from the computer – which could be “wrong” for SQL. If you use a non-ambiguous format, the query engine will do the proper translation.
8. (A.D.Teipal) Avoid spaces and special characters (such as #) in names of objects.
Although Access allows spaces and special characters in names, brackets are needed to qualify them whenever they are used in code. The added effort to ensure proper coding is not worth it. Use proper capitalization as in CustomerOrderDetail, or use underscores to join the words as in customer_order_detail, or even a mixture to make reading easier as in PDF_FileName.

9. (A.D.Tejpai, Brett Collings) Use a naming convention, any convention - but work it out before your first key-click so as to differentiate the names of tables/queries/reports etc. and code variables, constants, routines, etc.

http://msdn.microsoft.com/library/default.asp?url=/ARCHIVE/enus/dnaraccgen/html/msdn_20naming.asp

10. (A.D.Tejpai) In a db that has freshly created/copied/imported objects, better carry out recompilation - before attempting to use the db.
When a VB project is not compiled, Access must compile it "on the fly" every time you run the application. To compile a project, open any module in VBE and choose Compile from the Debug menu. Click Save if the compiler reports no errors to permanently save the result.
11. (A.D.Tejpai) As far as possible, avoid use of timer
If timer event of a form is used, and the situation permits, incorporate setting back the timer interval to zero at a suitable point in the code module. Leaving a timer running can disrupt such things as data entry flow.
12. (Beth M) Close and set to Nothing all objects via code that are opened before exiting app.
Closing the application or exiting a procedure without doing so may leave memory leaks.
13. (Beth M) Never use reserved names (i.e., Name, Date) within table structure.
This leads to ambiguity between implicit functions/methods and the names.
14. (John Viescas) Avoid storing calculated values.
If records should be edited in the future, existing calculations will be wrong. Be sure to write the appropriate transaction-protected code to do the calculation if you need to do so to resolve a performance problem.
15. (Jack Malki) Never Allow Users Direct Access To Tables.
There is just too much potential damage. You cannot control what the user is doing as you can with forms. If your user insists on Datasheet view, use a Form in Datasheet View instead.
16. (Beth M) When creating variables for numbers, use Long instead of Integer if there is a possibility of the number growing VERY large.
17. (Brett Collings) Always base forms and reports on queries not tables.
*The least number of records loaded in the recordset, the better the performance. Also, when you use a saved query, Access does not have to compile the query every time the object opens. If you use a table name, Access must build a SELECT * query and compile it.*
18. (Brett Collings) Before you ever touch the form and screen design pages, design your data structure and read up about normalization.
This may seem to be a waste of time because Access is so easily edited, but that ease can lead you into painting yourself into a corner and having to redo hours/days/weeks worth of work.
19. (Brett Collings) As soon as you start on your first form design, read up and try subforms/subreports.
Easy rule-of-thumb is if the data is linked to a foreign key, it probably needs to be on a subform/subreport.
20. (John Viescas) Always use Option Explicit in every module.
Using Option Explicit makes debugging much easier and will force code integrity. Not using it can lead to undeclared variables acting in ways you would never suspect since they all become Variants by default. You can set that in Tools>Options in the VBE window. Check the box for Require Variable Declaration.
21. (John Viescas) Include error handling code in every procedure.
Your users will find ways of using your application that you never expected. Errors happen. Being able to control them is essential.

22. (Zameer Abdulla) The 10 commandments of Access at [MVPs.org/access](http://www.mvps.org/access/tencommandments.htm) is a good list in itself.
23. (Just about everyone) Never use macros.
They do not support error handling. If you currently have any macros, convert to VBA. The only useful task for macros is AutoKeys.
24. (John Viescas) Set the form's KeyPreview = Yes
The most common example: preventing the user from using Page Down or Page Up to move to other records when custom navigation buttons are provided. These can be trapped in the form's KeyDown event and set to 0 to "swallow" the keystroke.
25. (John Viescas) Never define a Validation Rule without Validation Text. Also consider using a Validation Rule in preference to setting Required=Yes.
When you do not provide Validation Text, Access displays an ugly message: "One or more values are prohibited by the rule: <The Code of Your Rule Here>" A user isn't going to understand what you're trying to enforce with your comparison predicate. Tell them in plain language. If you use a Validation Rule and Text instead of Required=Yes, you can tell the user exactly what must be supplied. Be sure to leave Required=No because Access tests that rule before your Validation Rule.
26. (John Viescas) Avoid allowing both Zero Length and Null values in a field.
Users can't see any difference between zero length and Null values in the user interface. Pick one or the other, but not both.
27. (John Viescas) Create meaningful captions for fields in queries if they are displayed to the user.
See #8 above. For example, if the field name is CustLastName, the caption should probably be Last Name. Conversely, do not put captions in table fields as this can lead to difficult to debug SQL.
28. (A.D. Tejpal) Always explicitly declare query parameters.
Whenever a query involves inputs through parameters (whether user input or reference to form's control), it should be a standard practice to incorporate a Parameters clause at the beginning, explicitly specifying the data-type. This would be somewhat akin to the practice of declaring Options Explicit in VBA.

To add a parameter, open the query in design view and click on Query>Parameters...

In the first example, the parameter is used for a field in the query. In the second example, the parameter is in the criteria to limit the records returned. Example 1 will have unexpected results if no parameters are declared. The CboDate value is a variant as all control values are. So a date might be interpreted as a number. Specifying it as a date insures it will be interpreted properly.

Example 1:

```
PARAMETERS [Forms]![F_Sale]![CboDate] DateTime;  
SELECT [Forms]![F_Sale]![CboDate] AS RefDate_1  
FROM T_Sale;
```

Example 2:

```
PARAMETERS [Forms]![F_Sale]![CboDate] DateTime;  
SELECT *  
FROM T_Sale  
WHERE RefDate = [Forms]![F_Sale]![CboDate];
```