City of Madison
Information Technology
Recommended Internal Controls for MS Access

Many Access applications contain data that feed other applications at a higher level of the information system, which in turn feed the highest level of management information, including the organization's financial statements. Thus, Access can be, and often is, a material application in the overall MIS; a chain is only as strong as its weakest link.

Access is a file-server, not a client-server, database, and the usual front-end application and back-end database of the client-server architecture are not separated in Access, but combined and contained in the same file (with an MDB file extension). Since the Access application file contains the application process business rules, the user interface (forms, reports) to the data, the database's metadata, the physical database itself and the database engine, the controls to consider include not only the usual application control categories of input, processing and output controls, but also standard database controls. Because Access contains the application's Visual Basic (VB) code and other database and application objects in the same file, the distinction between application controls and database controls becomes blurred.

This document will discuss:

1. Database Controls
2. Input Controls
3. Processing Controls
4. Output Controls
5. Auditing Controls

1. Database Controls

1.1. User Access Controls

1.1.1. Prevent unauthorized access to data tables and queries.

1.1.1.1. There is a single password for any and all users to open the database.

A single open-the-application password can be assigned to open the application, but once users are allowed in the database, they have full access to all menus, design screens, table data, etc.; thus, this is a weak access control. The open-the-database password can also be changed or turned off by any user who knows the password, is in the database and has access to the standard menu.

1.1.1.2. There is a separate password to view and/or change the VB code in general modules or class modules.

In addition to the open-the-database password, Access has a separate programming password that can be set up to disallow a user to view and change the VB procedures and modules (code). This programming password should always be set up in any Access application; otherwise, any other controls that are in place may be compromised by a user with knowledge of VB programming methods. There are other methods to control user access to the VB code; see below.

1.1.1.3. The standard menu bar with its various administrator functions and permissions can be hidden and a customized menu bar inserted instead, thus denying access to many administrator or database owner powers.

The control weakness of having various administrator or database owner functions available to all users on the standard menu was mentioned previously. This can be controlled by replacing the standard menu with a customized menu without such functions. Creating and employing a customized menu are relatively easy to do, but experience shows that this simple, yet powerful, control is rarely found in Access applications. In fact, most Access applications are left totally unsecured, with the database window, all database and application objects, and the standard (full) menus exposed and accessible to all users. Several techniques, including the custom menu bar solution suggested previously, are available to hide and secure these windows, objects and design screens.

1.1.1.4. There is a user-level specific login and password with granular permissions down to the database and application object levels (tables, forms, etc.).

A powerful user-level security function is available in Access. Users can be set up with logins and passwords and assigned to database roles (called groups in Access) that have individualized permissions. Permissions are definable down to the database and application object level, with several choices of available permissions on an object, including read-only; open and update, but no design changes; and several others.
By default, Access user-level security is always "on" and contains a single user named "Admin" belonging to the administrator group and having a blank password. In this state, opening the application by anyone will not invoke the

login/password screen, and the user will be automatically allowed in as the Admin user with unlimited permissions due to belonging to the administrator group. Setting up Access user-level security is rather complex (which is probably why it is rarely seen) and will not be covered in detail here. Instructions can be found in the Access help file, in articles and white papers on the Microsoft Developer Network (MSDN) web site, and in Access how-to books at bookstores. Setup requires the usual database security tasks of creating users/passwords/groups, assigning granular permissions to groups and finally assigning users to groups. The security groups and their permissions must be carefully designed to be effective. (Note: These are not network file permissions that are managed by IT. These are internal application permissions that need to be managed by the application developer or administrator). Access provides a "canned" security report to document the security program's users, groups and permissions that have been set up; thus, the IT auditor can quickly determine the degree of user-level security that has been implemented.

1.2. Development Tools Controls

1.2.1. Restrict or remove source code programming language and design view of database objects, so they cannot be viewed or changed by a user.

1.2.1.1. Start-up parameters are used to hide and/or disable design views of database objects, hide standard menus, enable custom menu, and disable special key functions.

A set of startup parameters is available to hide the database window and, therefore, limit access to all Access objects, open a specific start-up screen and disable special function shortcut keys. If the database window and objects are thus hidden, a set of user interface screens (forms) must be present to allow the users to navigate around the application and carry out the tasks assigned to them, since they will no longer be able to assess database objects by simply clicking on them in the database window. Access references and literature refer to these user interface navigation screens as a "switchboard form.".

1.2.1.2. A compiled version (MDE) of the MDB file can be created to remove user access to all database and application objects, including design views and VB code.

Another powerful database access control is to "compile" the Access MDB file into an MDE file, which is easily done from the standard menus. By definition, an MDE version of an Access application will prevent the following actions by users:

- *Viewing, modifying or creating forms, reports or VB modules in design view*
- *Adding, deleting or changing references to object libraries or other databases*
- *Changing VB code (an MDE file contains no source code)*
- *Importing or exporting forms, reports or VB modules (however, tables, queries and web pages can still be imported and exported if the standard menu is available)*

Combining MDE compilation with the custom menu and start-up parameters discussed provides a strong access control environment and is recommended on all commercial Access applications. Access applications without these database-level

access controls would probably have to be classified as having materially weak controls.

1.3. Data Concurrency Controls

1.3.1. Protect against deadlocks when two query or update processes access the same data item at the same time.

1.3.1.1. Multiple levels of data locking are available for concurrent users and can be set by the individual Access database administrator (DBA), and not IT.

Access has multiple levels of data-locking controls, all the way from exclusive read-only access for a single user, to multiple users with several record-locking capabilities. Individual data item (field) locking is not available. The level and type of record locking are database parameters set by the owner or administrator of the database.

1.4. Cryptographic Controls

1.4.1. Convert all clear text to encrypted text, which cannot be read and interpreted with any type of text editor.

1.4.1.1. The entire database must be encrypted/decrypted; encryption by a specific data field is not available.

An entire Access database can be easily encrypted from the standard menu; however, any user with access to the database standard menu can just as easily decrypt it. Even with this weakness, encrypted data provide some security against unauthorized users viewing the file in a low-level disk editor or against interception if the entire MDB file is transmitted across the network or Internet. Microsoft reports that encrypted Access files lose approximately 15 percent in speed performance. It is important to keep in mind that encryption is useful only against outside hackers trying to view the confidential data using some type of editor. If the hacker can open the database, the Access software will automatically decrypt the data, as it must for any authorized user.

1.5. Existence Controls

1.5.1. Protect the existence of the database by establishing backup and recovery procedures.

1.5.1.1. There are no available Access functions for backup or recovery; manual IT procedures or general network controls must be used for backup and recovery of entire Access database file.

Access does not have any specific functions for backup and recovery. These important controls need to be designed into the application code itself and/or should be a part of the general IT controls surrounding the Access application. Databases must be on a network folder to be included in the general IT backup process.

1.6. Data Validation Controls

1.6.1. Ensure the accuracy, completeness and consistency of data that are input and maintained in the database.

1.6.1.1. Table attributes can be assigned properties such as data type and size, data required (y/n), default values, input masks, and validation rules (e.g., value ranges).

The Access table designer and related table attribute design properties allow a variety of validation controls at the database level. Properties available to the table designer are data type, input mask, data required (y/n), validation rule (data range) and embedded data code lists (e.g., two-character codes for states). Of course, the table designer should be hidden from ordinary users.

1.7. Audit Trail Controls

1.7.1. Provide a log (journal) of database activity of users and application events and data.

1.7.1.1. There are no available Access functions for audit trails or logs; application must provide custom programming for this.

Access does not have any specific functions for creating and maintaining audit trails. These important controls need to be designed into the application code itself and/or should be a part of the general IT controls surrounding the Access application.

2. Input Controls

2.1. Batch Controls

2.1.1. Protect integrity of data input by reconciling input number of records, hash totals and monetary amount totals input to pre-input totals.

2.1.1.1. There are no available Access functions for batch controls; the application must provide custom programming for this.

Access has no built-in capability for input batch controls. This simple, yet effective, control must be programmed into the application using VB procedures and SQL queries to compute, record and display batch record counts, hash totals and monetary totals.

2.2. Data Code Controls

2.2.1. Provide checks on the integrity of data codes, such as customer number or inventory number, by restricting input data to specifically allowed values; prevent transcription or transposition errors.

2.2.1.1. Specific required values from a drop-down list on the input form can be placed on the data item textbox.

2.2.1.2. Check digits are not specifically supported in Access; they must be programmed into the application.

Data code controls (e.g., check digits) are very limited in Access and, in general, must also be custom-programmed if needed. One exception is to use Access wizards to attach combo box droplists of allowed data to textboxes on data input forms.

2.3. Validation Controls

2.3.1. Ensure the accuracy, completeness and consistency of input data items.

2.3.1.1. Input form data fields (textboxes) can be assigned physical properties, such as input masks or range tests.

2.3.1.2. Input form data fields (textboxes) can be assigned a validation rule and error text.

2.3.1.3. The entire input form can be secured as read only, edit only, delete only, input only or any combination.

2.3.1.4. Individual input data fields (textboxes) on forms can be secured by disabling and/or lockout

Setting table attribute properties for data validation control at the database level has already been discussed. Input data validation at the application level can also be implemented by

creating validation rules and validation error text attached to the data field textboxes on the data input form.

Although the same validation rules in the tables and input form may be redundant, it is preferable to have them in both places. For instance, the data range of the attribute may be increased at the table level by the administrator, due to a business rule change, but data input for certain events (transactions) may still need the older and stricter data validation rule that can be enforced at the application data input screen level. In addition to range validation controls, non-null data requirements, input format masks and default value validation controls can also be created for data input fields on the input screen.

3. Processing Controls

3.1. Error Messages and Logs

3.1.1. Display and record appropriate information for any errors encountered during processing procedures.

3.1.1.1. There are no available Access functions for error logs; the application must provide programming for this.

3.2. Run-to-run Batch Control Totals

3.2.1. Protect integrity of data processed by reconciling number of records, hash totals and monetary amount totals before, during and after processing steps, and compare to previously saved batch input totals.

3.2.1.1. There are no available Access functions for batch controls; the application must provide programming for this.

3.3. Validation and Data Code Controls

3.3.1. Ensure the accuracy, completeness and consistency of data items processed.

3.3.1.1. There are no available Access functions at the processing step; the application must provide programming for this.

Access has no built-in capability for processing controls. Error messages and logs, run-to-run batch control totals, data code controls, validation controls, and the audit trail of processing steps must be custom-programmed into the application using VB code operating on database objects (tables, queries, forms, reports, etc.). A well-controlled Access application would have all of these controls, and the absence of these controls would certainly indicate material weakness in internal control for the Access application and, therefore, perhaps other processes linked to the Access application.

4. Output Controls

4.1. Report Output Controls

4.1.1. Provide limitation over the production and distribution of reports.

4.1.1.1. Report open permissions can be assigned to individual qualified users through user-level permissions.

4.1.1.2. There are no available Access functions for report distribution controls.

4.2. Display Output Controls

4.2.1. Provide limitation over the production and distribution to data in forms (i.e., screens and windows).

4.2.1.1. Input form open permissions can be assigned to individual qualified users through user-level permissions.

Access can assign user-level permissions to open and, therefore, view or print the data contents of data display in forms and hard-copy reports. This user-level access control is enforced via user login names and passwords and was described previously. Access has no built-in capacity to control output distribution once it can be viewed or printed by a qualified user.

5. Auditing Controls

Based on the previous discussion of the database and input, processing and output controls found in and around Access applications, the following are suggested internal control questions to be asked concerning any material Access application. These questions do not by themselves make a professionally designed Access audit program, although they could lead to a good start on one. The more "yes" answers to the questions about the application, the stronger the controls could be rated and the smaller the control risk could be judged, assuming the evidence collected in the tests of controls corroborated the "yes" answers.

Access internal control questions to be asked include:

1. Is the standard menu hidden and replaced by a custom menu that is absent all of the database application developer functions?
   a. If the answer is "no," the application has serious control problems, since most of the other controls applied to the application can be compromised by the user's access to the developer tools.

2. Is the installed application compiled to an MDE file?
   a. If the answers to numbers 1 and 2 are both "yes," internal control is definitely achievable in this Access application, although additional controls may be needed (see numbers 3 through 12).
   b. If the answer to number 1 is "yes" but the answer to number 2 is "no," internal control is achievable, but many controls automatically available through a "yes" answer to number 2 will have to be achieved by a combination of several other controls listed below.
   c. If the answer to number 1 is "no" and number 2 is "yes," it is doubtful that control can be achieved due to the powerful development tools available to the ordinary user.
   d. If the answers to number 1 and number 2 are both "no," then control is definitely not achievable.

3. Has a password been set up to restrict access to the VB code found in general and class modules and procedures?
   a. This control is not needed if a compiled MDE file is created (i.e., if a "yes" answer was given to number 2).

4. Has a single password needed to open the Access file been set up for all users?
   a. It is not needed if user-level controls are implemented (see number 8).

5. Is the Access file encrypted?
   a. If database performance degradation (approximately 15 percent) due to encryption is a problem, a strong network access control system may be effective and used in lieu of encryption. See 1.4 Cryptographic Controls.

6. Are data concurrency controls set up by the database administrator or owner? See 1.3 Data Concurrency Controls.

7. Have data validation controls been set up at the table level for appropriate data fields?
   a. Non-null data requirement?
   b. Default values?
   c. Data code drop-down lists for textboxes (if applicable)?
   d. Input masks (if applicable)?
   e. Data validation ranges and error text messages and error logs?

8. Are user-level security controls with individual permissions implemented?
   a. This is easy to determine by opening the Access database and seeing if user login and password are prompted.

9. Are run-to-run batch processing controls programmed into the application? See 3.2 Run-to-run Batch Control Totals.

10. Are audit trail controls programmed into the application? See 1.7 Audit Trail Controls.

11. Are error log controls programmed into the application? See 3.1 Error Messages and Logs.

12. Are existence controls (backup and recovery) in place (not really an application control)? See 1.5 Existence Controls.

Conclusions

Access is used worldwide to house business database applications. Most Access applications are totally unsecured or, at best, only partially secured and, therefore, are easy prey for even an unsophisticated hacker or malevolent employee. Access has a variety of security tools and methods available to the designer, and they should be investigated and tested (if necessary) by IT. The IT staff must be familiar with the tools to perform a competent audit of material Access applications that should be controlled and secured. For a well-controlled Access application, the following should be performed:

- The standard menu should be replaced with a custom menu
- Controls must be present for:
  - Data Validation
  - Input
  - Processing
- User-level control permissions must be implemented
- The MDB file should be compiled to an MDE file