

Microsoft Access 97 Security Manager

1.0 Introduction

No security administration tool can replace or be used effectively or safely without a fundamental knowledge of how Microsoft Access user-level security works. The following references will provide the user of the Security Manager with a good grounding in this knowledge:

- Microsoft Access 95 Security White Paper - Q148555
- Microsoft Access Security FAQ - Q165009
- Overview of How to Secure a Microsoft Access Database - Q132143
- Microsoft Access 97 Developer's Handbook. Microsoft Press. 1997
- Microsoft Jet Database Engine Programmer's Guide, 2nd Ed. Microsoft Press. 1997
- Access 97 Developer's Handbook, 3rd Ed. Sybex. 1997
- Microsoft Access security Help topics

The Security Manager enables the database-security administrator to make knowledgeable security decisions as the database and its workgroup evolve. By presenting all security settings in a single view, the administrator can set and display permission settings and assign ownership with all the information necessary to make informed decisions. In addition, complete management of user and group accounts is one click away. Finally, the ability to create logs of the database's entire security state and then to set the database's security to any of these log-runs gives the administrator a means to experiment with "what-if" scenarios or to create multiple security states for different situations.

For example, an application might normally use one set of high-level object permissions for a group and its members, but an on-site emergency may require that another group of users be assigned these permissions temporarily so that they can perform functions from which they are usually restricted. If a log of the database's security state has been created to meet this eventuality, one member of the emergency group can be permitted to apply this special log to the database, thereby granting the group all the necessary permissions.

2.0 Installing the Security Manager

The Security Manager is installed as a standard Microsoft Access add-in. To install it, on the **Tools** menu, point to **Add-Ins**, and then click **Add-In Manager**. In the **Add-in Manager** dialog box, click **Add New**. In the **Open** dialog box, browse to the location of Sm97.mde and click **Open**. Microsoft Access will copy Sm97.mde to the \Program Files\Microsoft Office\Office folder. Click **Close** to complete the Security Manager add-in installation.

The Security Manager has not been configured for multi-user, networked use. For organizations that want to install the add-in on various

workstations throughout the organization, the Sm97.mde can be placed on a network share and installed by following the steps above.

Because the Security Manager logs information about the security state of the database and members of the session's workgroup, its user must have Read and Write permissions on the file itself as well as Read, Write, Create, and Delete permissions on the folder containing the add-in.

3.0 Security Accounts that Can Run the Security Manager

The Security Manager simply checks whether the logon account is a member of the current workgroup's Admins group.

By default, the Admins group is granted all permissions on all new database objects. Even if individual Admins members do not have full, Administer permissions for database objects, their Admins membership will give them full, implicit permissions to all database objects and the database.

Ideally, the database owner should be the administrator of the database's security. The database owner has full permissions on the database and all current and newly created objects. Having the Admins group retain Administer permission on all objects provides a critical backup to the database owner account in the event that account becomes corrupted and cannot be re-created due to loss of the logon name and PID.

4.0 Security Manager Features Summary

The Security Manager add-in was designed to make the tasks of managing and troubleshooting Microsoft Access user-level security easier. By providing all the data involved with granting and revoking permissions in a single view, the Security Manager's **Permissions** tab enables the user to make completely informed permissions decisions. The **Accounts** tab similarly enables complete management of users, groups, passwords, and group membership. The **Logs** tab allows the user to record the entire permissions and ownership state of the database, experiment with various permissions settings on the **Permissions** tab, and then set the security state back to its previous state.

The following sections discuss the features of the Security Manager.

4.1 Permissions Tab

4.1.1 Users, Groups, and Memberships

Microsoft Access user-level security includes the concepts of explicit and implicit permissions. Explicit permissions are those assigned to

individual user accounts, whereas implicit permissions are those assigned to groups. Members of the groups inherit the permissions granted to the groups, thereby granting the members implicit permissions. When managing user- or group-account permissions, it is essential to know what groups the user belongs to as well as the members of any group. Upon selecting either a user or group, the account's membership is displayed in the **Membership** list box.

4.1.2 Permissions

The **Permissions** check boxes instantly assign and revoke their associated permissions; there is no need to confirm their action. The Security Manager follows the Microsoft Access permission conventions of granting or revoking associated permissions whenever certain permissions are granted and revoked. For example, checking the **Update Data** permission for a table will also check the **Read Data** and **Read Design** permissions. Similarly, revoking **Read Design** for a form also revokes **Modify Design**.

The Security Manager provides great flexibility when modifying permissions settings. The user can first specify a user or group, and then process permissions on various objects and types of objects, or the user can first select a particular object and then process permissions for different users and groups. The user can even switch to the **Accounts** tab in the middle of a permissions session, administer user or group accounts, and then switch back to the **Permissions** tab, which will still be pointing at the same account and object.

In addition to the permissions in the Microsoft Access **Permissions** dialog box, the Security Manager includes **Create New Tables/Queries**. If the option button for **Permissions on New Objects** is selected and either **Tables** or **Queries** is selected in the **Types** combo box, the **Create New Tables/Queries** check box enables the security administrator to assign or revoke a user or group's ability to create new tables and queries within the database. This permission cannot be checked for tables and unchecked for queries. The setting will apply to both types of objects.

Note The security administrator will find Microsoft Access user-level security much easier to manage if permissions are only assigned to group accounts and are revoked from all user accounts. Providing permissions to individual users then becomes simply a matter of making them members of the appropriate groups.

4.1.3 Explicit or Group Permissions

When the **Explicit or group permissions** option is selected, the user can assign or revoke an object's permissions for the selected user or group account. As with the Microsoft Access **Permissions** dialog box, only permissions applicable to the **Types** will be enabled.

4.1.4 Implicit Permissions

Selecting the **Implicit permissions** option displays the least restrictive permissions on the selected object or object type that are inherited (implicit) from all the groups to which the account belongs. For example, if a user does not have Read Data permission for a particular query, but belongs to at least one group that has Read Data permission on the query, the user will be able to exercise this permission as if it were assigned explicitly.

4.1.5 Cumulative Permissions

Cumulative permissions are the total of all the least restrictive permissions granted to a user account plus all the groups to which the account belongs. Whereas explicit permissions apply to individual users and implicit permissions apply to groups (and are inherited by members), cumulative permissions sum the two sets of permissions. For example, a user might only have the permission to Open/Run a form. The Users group, to which the account belongs, only has Read Design and Modify Design permissions. In this example, the account's cumulative permissions are Open/Run, Read Design, and Modify Design.

Note The **Cumulative permissions** value is stored in a container property called **AllPermissions**. **AllPermissions** is a read-only property and applies only to user accounts. Although its value is stored in each log-run, it is not written back to the container objects when setting database security from a log-run. As security is set from a log-run, Jet recalculates the **AllPermissions** value.

4.1.6 Types

When the **Permissions on new objects** option button is cleared (its default setting), selecting from **Types** populates the **Current object** list box (described below) with the appropriate objects. If **Databases** is selected, the list box is empty and its label changes to **<Current Database>**.

4.1.7 Current Object

The **Current object** list box enables selection of any database object for permissions processing. Selecting an item from the **Types** combo box (described above) repopulates it. All system tables and temporary queries are excluded from **Objects**. However, the list includes any table beginning with "Usys", such as a UsysRegInfo table, because these are user-created objects.

4.1.8 Run with Owner's Permissions

The Security Manager indicates the status of the Run with Owner's Permissions for all queries. The **Run with Owner's Permissions** check box is read-only; the user cannot change a query's Run Permissions in the Security Manager.

The **RunPermissions** property maps to the presence or absence of the WITH OWNERACCESS OPTION declaration in a query's SQL statement. When **RunPermissions** is set to **Owner's**, Microsoft Access includes the declaration in the SQL; if the property is set to **User's**, the declaration is absent from the SQL.

The concept of query Run Permissions is central to maintaining a high level of security over the design and contents of database tables, while still allowing users to view and manipulate their data. In short, Run Permissions enables the query's user to manipulate data with the table permissions granted to the query's owner. When a query's **RunPermissions** property is set to **Owner's**, the owner referred to is the owner of the query. Because the user in a secured database normally has no permissions on any of the tables, a query with **RunPermissions** set to **Owner's** allows the developer to provide selective access to table data. By creating queries that include only specific table fields and selected rows, and then setting the query's **RunPermissions** to **Owner's**, the database administrator or an administrative group can ensure that tables remain secure, yet users can still process their data. This key security concept is discussed at length in the references listed at the beginning of this document.

4.1.9 Permissions on New Objects

If the **Permissions on new objects** option button is selected, the user can assign default permissions for all new objects of the type selected in the **Types** combo box. These permissions do not affect the permissions on current objects, but only those created from that point on. **Permissions on new objects**, in conjunction with **Types**, enables the same functionality as selecting <New Tables, queries, and so on> from the Microsoft Access permissions **Object Name** combo box.

4.1.10 Change Owner

Change Owner assigns ownership of the **Current object** selection to the user or group selected in the **Users or Groups** list box. The change is immediately reflected in the **Owner** display. Objects can be owned by either user or group accounts. Only queries with their **RunPermissions** set to **Owner's** (see section 4.1.8) may not have their ownership changed, either in Microsoft Access or with the Security Manager. See article Q120885 - "Can't Change Ownership When RunPermissions Set to Owner's" for further information about this issue.

4.1.11 Displays

4.1.11.1 Permission Values

Permissions are stored as long integers within the document for each database object and within each container representing a type of

database object. Each user and group account will have its particular permissions value stored in each document and in each container. Microsoft Access and the Security Manager convert these values to familiar permissions like Insert Data. This display will be particularly valuable to users who are, or want to become, familiar with the manipulation of the actual permissions values.

4.1.11.2 Object Owner

Owner displays the current object's owner and indicates whether the account is a user or a group account. If the object's owner has been deleted from the workgroup information file, **Owner** will display **<Unknown>**.

4.1.11.3 Logon User

Logon User displays the name of the currently logged on user.

4.1.11.4 Db Owner

Db Owner displays the owner of the current database.

4.1.11.5 Workgroup

Workgroup displays the name of the current workgroup information file.

4.2 Accounts Tab

The Security Manager duplicates all of the functionality available in the Microsoft Access security interface for managing users, groups, passwords, and group memberships.

4.2.1 Users, Groups, and Passwords

The functionality and user interface elements are very similar to those found in the Microsoft Access security interface, including addition and deletion of users or groups, as well as password management.

4.2.2 Memberships

Assign/Remove Members enables the security administrator to assign groups to members or assign members to groups. In the default state, the **USER-Groups** option button is selected. This enables Available Groups to be selected from the **Available Groups** list box for the current user in the **Users** list box. If the **GROUP-Users** option is

selected, the list boxes switch positions so that the administrator can now assign Available Users to the group selected on the left.

In addition, the Security Manager emphasizes the relationship of availability and current Memberships by removing the selected account from the **Available Groups** or **Users** list. Thus, as groups are assigned to users (or users are assigned to groups), the **Available Groups** list shrinks.

Note No accounts are deleted during this process; they are simply deleted from the membership availability list. Assignments and removals from the Memberships list can be performed with the **Assign** and **Remove** buttons, or by double-clicking the account in either the **Available Groups/Users** list or the **Memberships** list.

4.3 Logs Tab

The ability to log and write back the permissions state of the database is a feature not found in the Microsoft Access user-level security menus. Security administrators should find many uses for this feature, including experimenting with "what-if" scenarios, backing up the database security state, or creating logs of various security states for possible fine-tuning of a Microsoft Access application at a customer site. Logs, referred to as log-runs in the Security Manager, are stored in the Sm97.mde. See Section 6.0 for information about periodically compacting Sm97.mde.

4.3.1 New

Each time that a new log-run is created, a set of records is added to the Security Manager's log table. Each log-run is identified by its name and the date and time it was created. Before clicking **New**, the user should provide a brief **Description** to further identify the new log-run.

After clicking **New**, the Security Manager will verify that the user wants to create the log-run. As the database's security settings are being logged, the status bar in the lower-left corner indicates the log-run's progress. Following completion of the log-run, a message will be displayed to indicate its completion.

4.3.2 Set Security

Setting the database's security from a log-run re-assigns the permissions for all user and group accounts on all objects (documents) and types of objects (containers), as well as the owner of each object. As with creating a new log-run, the user is prompted to continue, the status bar indicates progress, and the user is advised when the process has completed.

Four types of errors can occur when setting database security:

- 1) An object contained in the log-run has either been deleted or renamed since the log-run was created.
- 2) A user or group account has been deleted from the workgroup since the log-run was created.
- 3) An object's Owner has been deleted from the workgroup since the log-run was created.
- 4) The log owner of a query with **RunPermissions** set to **Owner's** is different from the query's current owner. The ownership of this type of query cannot be changed while setting security from a log-run, due to the Jet rules governing the ownership of a query whose **RunPermissions** is set to **Owner's**.

If any of these errors (or variances) occur, messages are displayed indicating the cause as well the name of the affected object, object type, user/group, or owner. The process then continues until completed. After setting the database security state, the user can view a variance log created during the process if errors occurred. Each time security is set from a log-run, the previous variance log is deleted. The user may view any variance log by selecting the **Errors** option.

The Security Manager detects if the current workgroup information file is different from the one joined when the log-run was created and will prevent the log-run from being applied.

4.3.3 Delete

To delete a log-run, select the log-run from the list and click **Delete**. The entire security log cannot be deleted in a single operation. Instead, individual log-runs are deleted one at a time.

4.3.4 Log-Run Names

The default name of a new security log is "Scratch_nn" where nn is a number between 01 and 99. The default name can be overridden by any name that the user wants, but the auto numbering only applies to the default name. Duplicate log names are detected and prevented.

4.3.5 Security Last Set From

This prompt displays the log-run name last used to set the database's security state.

5.0 Deleting and Renaming Objects While Using the Security Manager

In general, the Security Manager should be closed and reopened if the user is creating, deleting, or renaming new objects. However, if a new database object is created while the Security Manager is opened, the user can simply refresh the objects list by selecting another **Types** and

then reselecting the desired **Types**, or pressing F9 after clicking on the **Current object** list.

The user should avoid deleting database objects or security accounts between creating a log run and setting the security state from that log. However, the Security Manager will detect and log any errors that occur because of missing/renamed objects or deleted accounts.

The Security Manager was designed to allow the developer to work with database objects while the Security Manager is loaded. It can be minimized after changing security settings to allow the developer or security administrator to test the changes in the database itself, and then return to the Security Manager with all its pointers unchanged.

6.0 Compacting the Add-In

Because the Sm97.mde contains a dynamic table for logging security settings and log-runs will be created and deleted frequently in normal usage, the user should compact the database periodically. The security of the Sm97.mde allows the administrative user to compact the database. To do so, the user should open the Sm97.mde, and then on the **Tools** menu, point to **Database Utilities**, and then click **Compact Database**.

Note The Security Manager should not be in use when compacting Sm97.mde.

7.0 Restrictions of the Security Manager

The Security Manager does not provide the capability to change the ownership of Types (containers). Effective user-level security does not require that the database owner or any other account take ownership of the container objects.

The Security Manager enables setting permissions on the current database, but not on the Databases container (Types).

Although changing the ownership of the database itself could have been easily included within the Security Manager's functionality, this means of ownership does not grant the new database owner the security permissions and capabilities normally associated with ownership of the database. The only real way to achieve these administrative capabilities is to either create the database or to create the database and import all its objects into a new database. Because of the critical importance of ownership to Microsoft Access security, users should familiarize themselves with this concept by referring to the sources listed at the beginning of this document.