[#1] [K2] [S3] **DAOTable--General Information**

The DAOTable sample demonstrates using the MFC DAO classes to create common database objects: Databases, Tables, Queries, Fields, and Indexes. This dialog-based application maps the properties of these objects to controls the user can set and/or view. The program source code is organized so that most database interaction is isolated from the user interface code to ease finding examples of how to use MFC DAO.

In addition to demonstrating the use of the MFC DAO classes, this sample can be a useful tool for creating simple Access databases. You can create MDB files from scratch, create and delete tables and queries, add and delete fields and indexes in the tables, and modify existing queries.

MFC DAO and Its Use of Exceptions

One thing you will notice if you run this sample as a debug build with tracing enabled is that the sample throws numerous first chance exceptions. In reality, what you are seeing is MFC DAO's way of handling error conditions. For many MFC DAO methods, as documented in the VC++ 4.0 on-line documentation, no return value is provided to indicate success or failure, rather an exception is thrown if an error occurs. DAOTable handles those exceptions either by simply displaying the error message provided by the exception mechanism or by using the exception to gather information (e.g. exception processing is used to determine if an object already exists in a given collection). For more information, please see the documentation listed below.

More Information

For more information about the MFC DAO classes, refer to the on-line help for Visual C++ 4.0. For additional documentation on DAO, refer to the help files in the DAO SDK which is provided on the VC++ 4.0 CD-ROM.

Files in project

Except for those files that are common to all MFC projects, all files that make up DAOTable are listed below with the type of components they contain and a brief description of their contents.

Project Files:

DATABASE.CPP	- MFC DAO - database specific functions
DATABASE.H	(also demonstrates C++ exception handling)
FIELD.CPP	- MFC DAO - field specific functions
FIELD.H	
INDEX.CPP	- MFC DAO - index specific functions
INDEX.H	
QUERYDEF.CPP	- MFC DAO - querydef specific functions
QUERYDEF.H	
TABLEDEF.CPP	- MFC DAO - tabledef specific functions
TABLEDEF.H	
ADDQYDLG.CPP	- USER INTERFACE - query definition dialog
ADDQYDLG.H	
DAOTABLE.CPP	- APPLICATION - main application objects
DAOTABLE.H	
DAOTDLG.CPP	- USER INTERFACE - main dialog
DAOTDLG.H	
ADDIXDLG.CPP	- USER INTERFACE - add indexes dialog
ADDIXDLG.H	
ADDTBDLG.CPP	- USER INTERFACE - table definition dialog

[1] HID_GENERAL
[2] General Information
[3] General Information

ADDTBDLG.H
LISTCTRL.CPP - USER INTERFACE - derived CListCtrl class
LISTCTRL.H
ADDDBDLG.CPP - USER INTERFACE - database definition dialog
ADDDBDLG.H

#4 K5 $6**DAOTable Main Dialog**

Database

Edit Box: Enter the full or relative path of an existing Access database file to which you want to connect. Leave blank to browse for file. Specify a new name to create a database file.

Connect Button: If you specify a path in the edit box, then pressing this button opens the specified file or displays a creation dialog it if it does not exist. If you leave the edit box empty, then a dialog is displayed that lets you browse for an MDB file. Once a successful connection is made, the Table and Query portions of the main dialog are enabled.

Table

Combo Box: Either enter the name of a new or existing table by typing or select an existing table from the drop-down list.

Fields Button: Press this button to view existing fields in a table and/or add new fields

Indexes Button: For existing tables, press this button to view existing indexes and/or add new indexes to the table. Not available for new tables until fields have been added.

Delete Button: For existing tables, press this button to delete the table. You are prompted for acceptance via a message box before the table is deleted.

Query

Combo Box: Either enter the name of a new or existing query by typing or select an existing query from the drop-down list.

Definition Button: Press this button to view or modify the definition of an existing query or define a new query

Buttons

Done: Press this button to exit the application.

Help: Press this button to see this help page.

[4] HIDD_DAOTABLE_DIALOG
[5] Main Dialog
[6] Main Dialog

[#7] [K8] [$9] DAOTable Database Definition Dialog

Controls

Database Name Edit Box: (Read-only) Reflects the name chosen in the main dialog.
Encrypt Check Box: Select whether to create the database with encryption or not.
Access Version Radio Button Group: Specify the version of database to create. For peak efficiency, use Version 3.0.

Buttons

Done: Press this button to create the database. To exit the dialog without creating the database, simply close the dialog using the windows system menu.
Help: Press this button to see this help page.

[7] HIDD_ADD_DATABASE_DLG
[8] Database Definition Dialog
[9] Database Definition Dialog

#10 K11 $12 **DAOTable Table Definition Dialog**

NOTE: For existing fields, the majority of controls are read-only. This need not indicate that the actual property is read-only, it merely indicates that no support for updating properties in existing tabledef objects is provided in this application. Refer to the MFC DAO class and DAO SDK documentation for details on property updatability.

Table Name Edit Box: (Read-Only) Edit box that displays the table name specified in DAOTable main dialog

Field Information

Position Edit Box/Spin Control: Enter the desired ordinal position of the current field. While this value automatically increments for each new field, it is acceptable to have fields share the same ordinal position. Note: When using DAOTable's Previous and Next buttons to move through the collection, you will notice that the field collection does not appear to be ordered by ordinal position. However, the ordinal position does have an effect that is consistent with the documented behavior of this property--for instance, the ordinal position determines the order of columns returned by SQL statements such as "Select * from table_name."

Name Edit Box: The name for the field

Type Combo Box: Select one of the predefined data types for the field

Size Edit Box: For fixed size fields, this edit box is read-only. For variable size fields, you can specify a field size using this edit box.

Default Value: Enter a default value for the field

Required Check Box: Check if you want NULL to be unacceptable as a value for this field

Attributes

- **dbFixedField Radio Button**: Mutually exclusive with dbVariableField Radio Button—only user settable for text fields, otherwise reflects the nature of the type you select. For text fields, select this button if you want to store the field in a fixed-sized data block
- **dbVariableField Radio Button**: Mutually exclusive with dbFixedField Radio Button—only user settable for text fields, otherwise reflects the nature of the type you select. For text fields, select this button if you want the storage allocation of the text to vary with content.
- **dbAutoIncrField Check Box**: For type Long fields only—check if you want the field to be an autoincrementing counter

Validation

Validation Rule: Enter a rule by which values are checked for validity when they are specified for this field. (e.g. "between 10 and 100" for a numeric field type)

Validation Text: Enter the message that will be displayed if a value is specified for this field which breaks the validation rule.

Buttons

Done: This button is disabled until a field has been added to the table. To exit the dialog if the Done button is disabled, simply close the dialog using the windows system menu. If you have entered a field name for a new field and have not added the field and you press Done, you will be warned that new field information will be lost if you continue.

Help: Press this button to see this help page.

Previous, Next: Use these buttons to move through the field collection

Add: Press this button to add the field whose properties and attributes you have just specified to the tabledef.

Delete: Press this button to delete the current field--only valid if the field exists in the collection (i.e. deletes nothing if the field has not yet been added). You are prompted for acceptance via a message box before the field is deleted.

[10] HIDD_ADD_TABLE_DLG
[11] Table Definition Dialog
[12] Table Definition Dialog

Implementation Note:

Since a tabledef can not be appended to the tabledef collection of a database object without at least one field having been created in the tabledef, the following steps are performed when the user adds the first field:
1) Create the tabledef
2) Create the field with the specified properties (this automatically appends it to the tabledef's fields collection
3) Append the tabledef to the tabledef collection of the database
As a result, if an invalid name is specified for a table, the exception which indicates this situation is not thrown until the first field is added.

#13K14 $15 **DAOTable Add Indexes Dialog**

NOTE: For existing indexes, the majority of controls are read-only. This need not indicate that the actual property is read-only, it merely indicates that no support for updating properties in existing tabledef objects is provided in this application. Refer to the MFC DAO class and DAO SDK documentation for details on property updatability.

Table Name Edit Box: (Read-Only) Edit box that displays the table name specified in DAOTable main dialog

Index

Index Name Edit Box: Enter the name of the index you want to create

Check Boxes: Check the boxes that set the desired properties of this index—be aware that not all combinations of options are supported. (e.g. you can not specify a non-unique primary index). Also, with certain selections made, other selections can become don't-cares and need not reflect the appropriate values. Please refer to the DAO documentation on index properties for more information.

Fields in Index

List Control: (Multiple Selection) Select one or more field from the list. To select an item, click the left mouse button with the mouse positioned over the name of the field. Fields whose type does not support indexing are not displayed. You can press the left mouse button on the name of an item in the list control to select the field with ascending sort, press again to select the field with descending sort, and press again to deselect the field.

Buttons

Done: If you have entered an index name for a new index and have not added it and you press Done, you will be warned that new index information will be lost if you continue.

Help: Press this button to see this help page.

Previous, Next: Use these buttons to move through the index collection

Add: Press this button to add the index whose properties you have just specified to the tabledef.

Delete: Press this button to delete the current index--only valid if the index exists in the collection (i.e. deletes nothing if the index has not yet been added). You are prompted for acceptance via a message box before the index is deleted.

[13] HIDD_ADD_INDEX_DLG
[14] Add Indexes Dialog
[15] Add Indexes Dialog

#16 K17 $18 **DAOTable Query Definition Dialog**

Controls

Query Name Edit Box: Modify the name of the query by making changes in this edit box
Updatable Check Box: (Read-Only) Indicates whether the view created by the query is updatable or read-only
SQL Statement Edit Box: (Multiline) Specify the SQL statement that the query will execute in this edit control.

Buttons

Done: If you have entered or modified the SQL statement for the current query and have not selected Add or Modify and you press Done, you will be warned that the new or modified information will be lost if you continue.
Help: Press this button to see this help page.
Modify: Use this button to transfer modifications of the SQL statement or query name to the existing query object. If the current query has not been added, pressing Modify performs no operation and displays a warning dialog.
Add: Press this button to add the query whose properties you have just specified to the tabledef.
Delete: Press this button to delete the current query--only valid if the query exists in the collection (i.e. deletes nothing if the query has not yet been added). You are prompted for acceptance via a message box before the query is deleted.

[16] HIDD_ADD_QUERYDEF_DLG
[17] Query Definition Dialog
[18] Query Definition Dialog