



AutoKey 3.0 Demo Application

by
Resource Development Associates

© Copyright 2000,2001 & 2002 by Dennis Santoro
and Resource Development Associates.
All rights reserved.

Installation and Use

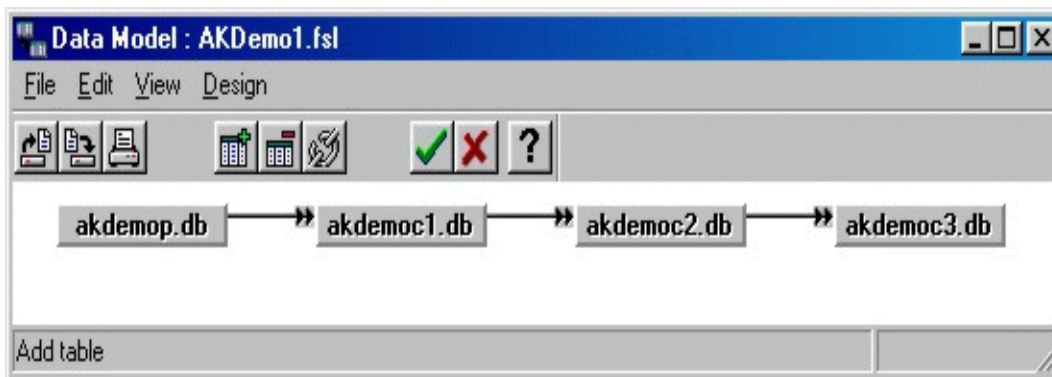
AutoKey's Demonstration application is designed to show AutoKey users a variety of ways that AutoKey's library methods can be implemented. This demo application has calls to all of the library methods in various forms and includes a number of demonstrations of complex containership and multilevel data models which users should also find helpful both in implementing AutoKey and in their database and form design in general. The code in the included forms is all open and available for your use and extensively commented. This should help even relatively inexperienced users to more fully understand the code and may teach you something about actions, library calls and custom types as well.

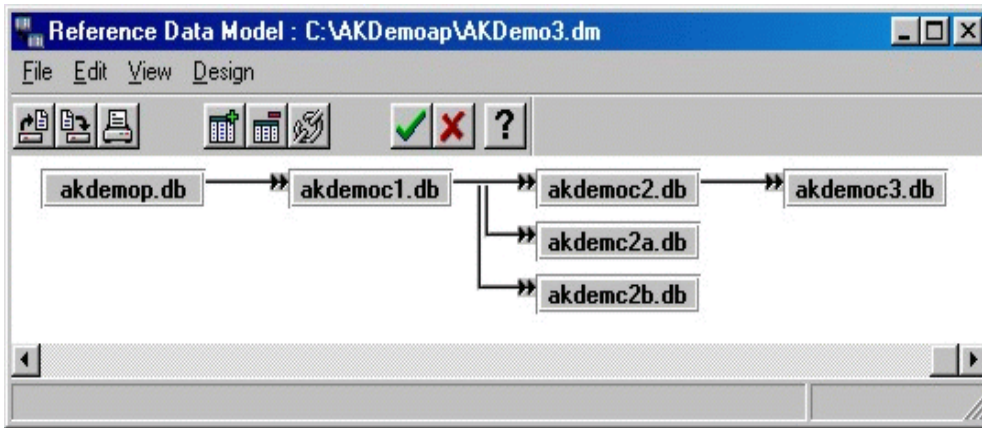
This demo application will not run without AutoKey. It can be opened and looked at without AutoKey however and so, may be valuable as a teaching tool anyway. The demo will run with any version of AutoKey so you will only need a single copy of the demo no matter how many version of AutoKey you have (for the various versions of Paradox for which AutoKey is available) unless you plan to distribute the demo application as well.

The self extracting file contains 4 forms, AKDemo1.fsl, AKDemo2.fsl, AKDemo3.fsl and AKDemo4.fsl. These forms use the tables also contained in the self extracting file. The tables include: Akdemop.db (parent table), AKdemoc1.db, AKdemoc2.db, AKdemoc3.db, AKdemoc2a.db and AKdemoc2b.db (various child tables), Akdemop1.db (parent table for the fourth form), AKdemc1a.db (child table for the fourth form), and Idtable.db which is the AutoKey control table for deployment of this demo application.

The AKdemo1 and AKdemo2 forms use the data model below. As you can see it contains a 4 level data model where each level has a 1 to many (1:M) connection to the level below. The connections are from the primary key in the parent table to the foreign key in the child table. The

primary key of each table is, of course, a meaningless integer which is set by AutoKey. The foreign key is also a meaningless integer, on which there is a secondary index which is used for





the link. The foreign key contains the value of the parent record's key. This primary key - foreign key relationship is a very stable relationship and is highly recommended as a way to design tables and relationships. The AKdemo3 form uses a data model where the 3rd

level contains 3 tables, only one of which has a child. Level 2 having 3 child tables complicates the model and the containership to some extent. These tables are also keyed and linked in the same way as the previous data model. The AKDemo4 form uses a model similar to the top two tables in the other two data models. The exception here is that since this form was specifically designed to show how to use the 2 AutoKey methods that use the dynarray return, and since one of these demonstrates a padded key value and the other illustrates using a long integer, concatenated with a string to create a key, the keys have to be different. The AKDemoP1.db uses an A10 field to hold a padded key that is 10 characters long in all circumstances although the incrementing digit keeps each value unique(e.g.0000001859). The AKDemC1a.db uses an A10 field to hold a key which is concatenated from the first 3 letters of a value in the parent table's RecVal field plus the returned incremented integer (e.g. RDA123). In this case the foreign key link also has to be A10. These functions can be useful for certain keying systems. They could also be used to generate such things as non key unique customer numbers or non key invoice numbers which always have the same number of characters or a specific pattern (such as the first n letters of the customer name followed by a unique integer).

We encourage you to explore these data models, the linking and ways to manipulate the data and forms by manipulating them, modifying the forms, modifying the implementation code, adding additional tables (you will need to create your own additional IDTable.db for that but multiple Idtable.db files can be used in a single application if you are careful about the alias calls or you can add the data from the current Idtable.db to your own after creating appropriate keys for the records in your own IDTable.db. (**Note:** please keep a copy of the demo in it's original form in a safe place in case you need to recover from any changes you make.)

Installing the AutoKey Demo Application

You must have a copy of AutoKey for the demo to work. Install the tables and forms in a folder (preferably one of its own) by double clicking the self extracting zip file. Any folder name will do. Below this folder create a folder called DemoApID. Move the files Idtable.db, Idtable.PX and IDTable.val to this new DemoApId folder from the folder you expanded the files in. Copy your AutoKey.ldl library file to this DemoApId folder as well. The demo is now installed.

Running the Demo Application

To run the application, open paradox. The :Work: alias can point anywhere. Open any of the demo application forms. This will create a project alias AKDemo to the folder the form is in and another, AKDemoLib to the DemoApID folder. If you do not save them these aliases will only exist for the duration of the session. You may wish to check out this alias creation code in the forms (as well as several other coding techniques you will find in them). Once a form is open you can add and edit records at will. You can start or end edit mode from the normal menus, from the toolbar that is included in the AutoKey library, by pressing the insert key or from the edit toggle button on the forms. Note the toggle button changes to the appropriate state no matter how you toggle the edit mode. This is a useful feature from a usability point of view. Records may be added with the add record button or with the insert key. In some places autoappend is also available.

You can toggle the forms into design mode at any point to review the code. You should open one form in run mode first however to create the aliases.

That's it. Explore. Play with it. Have fun. We hope you learn some interesting stuff.

Licensing

RDA believes in clear and simple licensing. You must have a Full license for RDA's AutoKey and adhere to it to run this application. A copy of the AutoKey Demo Application allows the licensee to use the AutoKey Demo Application at any one site at a time. It should be treated like a book, for the most part. If you install it on a server it is permissible for all users on that server to access the application. Installations on individual machines should be done only so as the copies will not be run simultaneously. For example, you can install and use the application on your office and home machine as you will not be in both places simultaneously. You can not, however, freely distribute this to your clients or friends in such a way that it can be used simultaneously on a single license.

Remember that in installing and using this application you must also meet the AutoKey licence agreement. See the AutoKey documentation for that information.

On the other hand, if you build 2 applications which include RDA's AutoKey library and install them both at the same site, only your full license is required. For each additional site you install either or both applications in you would need an additional distribution license.

Contact us to discuss volume terms or other licensing issues.