

Chapter 1 : Visual Basic 6 (VB6) - ADO and the ADO Object Model

What we are going to discuss in this article is Recordset object but you might think why we have chosen to name the article as ActiveX Data Objects (ADO), this is because you must have an idea of ADO technology in order to understand the bits and bytes of Recordset object.

If a barcode scanner is not available for testing, consider these hand-held barcode scanner kits. Install and Register the Barcode Control in Windows Install the ActiveX Barcode Control by running the executable file included in the package; this will register the control in Windows and install several implementation examples. The barcode control must be installed and registered in Windows before an application will be able to identify it. Several installation tutorials are included in this document. Adjust the Barcode Control Properties The ActiveX Barcode Control properties determine the data that will be encoded in the barcode as well as other options such as barcode type and size. There should now be a toolbox visible. On the toolbox, choose the "more controls button. Size the control as necessary. To adjust the barcode control properties, right-click on the control and choose properties. When finished, click on the "design mode button. If problems are encountered editing the properties of the control, press the design mode button to enable it. The data encoded in the barcode may be static or modified with the use of VBA code. The following describes this process: Install the barcode object on the merge document and determine the name of the object by right-clicking on it. In this case, it is "BarCode1". If unable to access the properties of the object, choose the Developer and select Design Mode. Determine the name of the merge field used to encode in barcode. In this example, it is "Merge1". After the file is imported, it should show up in the modules section. Click on this module to reveal the code. Choose Edit - Replace. Change the name of the object from Barcode1 to the object name determined in step 1 and choose Replace All. Change the name of the merge field from Merge1 to the actual merge field determined in Step2 and choose Replace All. A print preview macro is included that will print one page from a selected row number. If an error message is received such as "macros in this project are disabled" that will not allow the macro to run, change the security level of the macros as outlined in this Microsoft KB. The normal method of performing a mail merge will create multiple pages with a barcode that is not linked to a mail merge field. The only method of linking the barcode to a mail merge field is by executing the macro. When used in reports and forms, these controls may be dragged into the design area and connected to database fields. Open the report in design mode. Move and size the control necessary. To change the properties of the control, such as barcode height and symbology type, right click on the control and choose properties. In the Property Sheet drop down menu, select Report and change the Record Source to the table where the barcode field of the ActiveX control resides: To bind the barcode to a data source, select the table in the control source field in the same manner as for a text control. Verify the correct data is encoded in the barcode. If it does not, consult Access Help to determine how to get the correct data in the Control Source field. The result of the Control Source is what will be encoded in the barcode. When working with multiple tables, it may be necessary to specify both the table and field in the control source; for example: For a complete barcode generator object that stays embedded in the database, consider the Native Barcode Generator for Access. On the tool box, choose the "more controls button. Click and drag the control for desired placement in the spreadsheet and size it as necessary. To modify the properties of the control, the user must be in design mode, then right-click on the control and choose properties. When finished, click on the "design mode button" to exit. To link a cell to the data to be encoded in the barcode, enter the cell in the "LinkedCell" property of the control. ActiveX Controls cannot be used to populate a column in Excel and must be placed individually. If this is needed, consider barcode fonts which can be used to generate a column of barcodes. If the control does not appear in the list, choose Customize to add it to the list of controls and select Finish. The control will appear in the web page. Right-click on the control to adjust properties as necessary. The control must be installed on every PC that accesses the web page displaying the ActiveX control. To enable the control to automatically install, if it is not already installed, enter the file name and path of the signed CAB file, which is provided in the download package. ActiveX Controls only work with web

browsers that support ActiveX.

Chapter 2 : ADO: ActiveX Data Objects [Book]

Introduction to ActiveX Data Object This tutorial provides you a quick introduction of ActiveX Data Object (ADO) Statistics. Total Hits - Total Votes - votes Vote Up - 45 votes.

Contains the information used to establish a connection to a data store. ConnectionTimeout Indicates how long to wait while establishing a connection before terminating the attempt and generating an error. CommandTimeout Indicates how long to wait while executing a command before terminating the attempt and generating an error. State Indicates whether a connection is currently open, closed, or connecting. Provider Indicates the name of the provider used by the connection. Version Indicates the ADO version number. CursorLocation Sets or returns a value determining who provides cursor functionality. To use a Connection object, simply specify a connection string, which identifies the data store you want to work with, and then call the Open method to connect. The easiest way to open a connection is to pass the connection string information to the Open method. To determine whether the Connection object worked, you can use the State property of the Connection object. Close End Sub If you need to connect to only one data store, the procedure followed in the above code is the easiest way. Alternatively, you can create a Connection object and set the ConnectionString property before calling the Open method. This approach allows you to connect to one data store and then reuse the Connection object to connect to another data store. Close End Sub This method also gives you the opportunity to set other properties of the Connection object before connecting. For instance, you might want to set the connection time-out: It is becoming increasingly popular to not have to rely on existing ODBC data stores. This eases the setup burden. You can use a different provider by setting the Provider property of the Connection object. However, this shows you how you would change the provider when you want to use other OLE DB providers. The Execute method is used to send a command an SQL statement or some other text to the data store. If the SQL statement returns rows, a Recordset object is created. Close End Sub Remember that the returned Recordset object from connection. In the following example, the command passed to the data source is a Delete statement. Because no rows are returned, you do not need to explicitly use a Recordset object. How many rows were deleted? You can use the recordsAffected parameter to find out. Close End Sub In the next example, the command passed to the data store specifies the name of a stored procedure to run. Because rows are returned, you do need to use a Recordset object. Do While Not rs. Close End Sub Show:

Chapter 3 : My Careers: ADO - Activex Data Object

ADO is a programming interface to access data in a database Accessing a Database from an ASP Page The common way to access a database from inside an ASP page is to.

It can be used without writing any code at all! Or, it can be a central part of a complex database management system. This icon may not appear in your Visual Basic toolbox. The Components window will appear. The control will be added to your toolbox. That control is still included with Visual Basic 6. Make sure you are not using this data control for the work in this class. This control is suitable for small databases. You might like to study it on your own. Connect to a database. Open a specified database table. Create a virtual table based on a database query. Pass database fields to other Visual Basic tools, for display or editing. Such tools are bound tools controls , or data aware. Add new records or update a database. Trap any errors that may occur while accessing data. One row of a table is accessible to each data control at any one time. This is referred to as the current record. The arrows are used to navigate through the table rows records. As indicated, the buttons can be used to move to the beginning of the table, the end of the table, or from record to record.

Chapter 4 : Database Programming using VC++ and ADO | VC++ In Easy Steps.

ADO Programmer's Guide for using ADO Objects. 01/19/; 2 minutes to read Contributors. In this article. This documentation provides an overview of how to use ADO objects to work with data from various data sources.

For those who have experience with previous data-access object models in VB, the main points of comparison between ADO and the earlier data access models are as follows: ADO provides fewer object classes than earlier models, and most objects can be instantiated directly by the programmer instead of having to be instantiated through other objects. The ADO Recordset object may be instantiated independently of any other object, and then attached to a Connection or Command object—or you can open the Recordset by calling methods of the Connection and Command objects. Making objects less dependent on each other has interesting and useful consequences besides just conceptual clarity: As a consequence of the example just given, you can create an ADO Recordset object that never has to be connected to an existing data source—and you can therefore use such a disconnected Recordset to track and maintain virtual data created entirely within your application or to maintain data from a database offline. ADO provides the programmer more opportunities to finetune the data cursor of a Recordset. A data cursor or just a "cursor" represents a set of resources initiated by a data connection that provides a connection to a specific row in a set of data. The data cursor can change position to point to a different row of data. ADO enables you to directly specify where the cursor is implemented client or server side and several different types of cursors. ADO provides better overall performance than earlier object models. ADO is more resource-efficient than earlier object models. Following are the objects exposed in the ADO object model: Specifies information about the physical connection with a data source. Stores information about actions performed on the data, such as data modification and retrieval. You can use a Command object to execute actions on the data or to return data from the server in a Recordset object. Provides a rich selection of properties, events, and methods to expose data in a field-row format, and thus allows you to programmatically traverse, examine, and manipulate specific fields in specific rows of data. Parameters collection of the Command object made up of Parameter objects. Contains information about parameter values that are passed by a Command object. Fields collection of the Recordset object made up of Field objects. Contains information about field structure and content of the data in a Recordset object. Properties collection made up of Property objects. Contains information about provider-specific properties of Command and Parameter objects. Errors collection of the Connection object made up of Error objects. Contains information about the most recent error that occurred when attempting an ADO operation. The rest of this chapter discusses how to use ADO at various levels, beginning at a very automated level and ending with lower-level manipulation in code of the object model and of SQL data providers.

Chapter 5 : Download calendrierdelascience.com from Official Microsoft Download Center

ActiveX Data Objects (ADO) is a collection of software components providing a programmatic interface to access the data sources from client applications. ADO acts as a layer to access any data store in a generic way from the application code.

Indong Glenn Mark A. A part of MDAC , it provides a middleware layer between programming languages and OLE DB a means of accessing data stores, whether they be databases or otherwise, in a uniform manner. ADO allows a developer to write programs that access data without knowing how the database is implemented; developers must be aware of the database for connection only. ADO was introduced by Microsoft in October ADO is made up of four collections and twelve objects. Fields This collection contains a set of Field objects. The Collection can be used in either a Recordset object or in a Record object. In a Recordset object, each of the Field objects that make up the Fields collection corresponds to a column in that Record set object. Errors All provider-created errors are passed to a collection of Error objects, while the Errors collection itself is contained in a Connection object. The connection object stores information about the session and provides methods of connecting to the data store PowerPoint Presentation: Command After the connection object establishes a session to the data source, instructions are sent to the data provider via the command object. The command object can send SQL queries directly to the provider through the use of the CommandText property, send a parameterised query or stored procedure through the use of a Parameter object or Parameters collection or run a query and return the results to a dataset object via the Execute method. Recordset A recordset is a group of records, and can either come from a base table or as the result of a query to the table. The RecordSet object contains a Fields collection and a Properties collection. The Fields collection is a set of Field objects, which are the corresponding columns in the table. The data are updated at the data source after the record is changed and the Update method is called. The recordset is locked using adLockBatchOptimistic and each time Update is called the data are updated in a temporary buffer. Finally, when UpdateBatch is called the data are completely updated back at the data source. To start the transaction, the programmer invokes the BeginTrans method and does the required updates. When they are all done, the programmer invokes the CommitTrans method. RollbackTrans can be invoked to cancel any changes made inside the transaction and rollback the database to the state before the transaction began Transaction PowerPoint Presentation: Record This object represents one record in the database and contains a fields collection. A RecordSet consists of a collection of Record objects. Stream A stream, mainly used in a RecordSet object, is a means of reading and writing a stream of bytes. It is mostly used to save a recordset in an XML format, to send commands to an OLE DB provider as an alternative to the CommandText object and to contain the contents of a binary or text file. Parameter A parameter is a means of altering the behaviour of a common piece of functionality, for instance a stored procedure might have different parameters passed to it depending on what needs to be done; these are called parameterised commands. A property object can be either a built-in property " it is a well-defined property implemented by ADO already and thus cannot be altered " or can be a dynamic property " defined by the underlying data provider and can be changed PowerPoint Presentation:

Chapter 6 : ADO Introduction

ActiveX Data Objects (ADO) is a collection of software components providing While using this site, you agree to have read Microsoft Activex Data Objects Library from operational data and other sources that is designed to serve a particular.

How to Use the Sample Extract the contents of the. To use the Visual Basic project: This sample code works with both ADO 2. Press the F5 key to run the program. A form for the demonstration appears. This sample creates a copy of OrdersTemplate. It then uses ADO to connect to the workbook and opens a Recordset on a table that is a defined range in the workbook. Follow these steps to do this: On the Insert menu in Excel, select Names, and then select Define. Note that the defined name has grown to include the newly added records. Quit Microsoft Excel and return to the Visual Basic application. This sample creates a copy of EmpDataTemplate. Data is added at defined ranges or tables in the workbook. When the data is transferred, the connection is closed and the workbook that results is displayed in Excel. After you examine the workbook, quit Microsoft Excel, and then return to the Visual Basic application. This sample creates a copy of ProductsTemplate. When the row additions are complete, the ADO Connection is closed and the workbook is displayed in Excel. The macro runs when the workbook opens; if the new "Products" worksheet exists in the workbook, the macro code formats the worksheet and then the macro code is deleted. This technique presents a way for the Web developer to move formatting code away from the Web server and onto the client. A Web application could stream a formatted workbook that contains data to the client and allow macro code that would perform any "conditional" formatting that might not be possible in a template alone to run at the client. This sample produces the same results as Sample 1, but the technique that is used to transfer the data is slightly different. In Sample 1, records or rows are added to the worksheet one at a time. Once the transfer is complete, the Excel table is detached from the Access database and the workbook that results is displayed in Excel. Quit Excel, and return to the Visual Basic application. The last sample illustrates how you can read data from an Excel workbook. Select a table in the drop-down list, and then click Sample 5. The Immediate window displays the contents of the table that you selected. Note that the used range does not necessarily begin on row 1, column 1 of the worksheet. The used range starts at the upper left-most cell in the worksheet that contains data. If you select a specific range address or a defined range, the Immediate window displays the contents of only that range on the worksheet. Note that the default path for the home directory is C: Copy the following files to the folder you created in the previous step:

Chapter 7 : What are ActiveX Data Objects (ADO)? - Definition from Techopedia

â€¢ The ADO (ActiveX Data Object) data control is the primary interface between a Visual Basic application and a database. It can be used without writing any code at all! Or, it can be a central part of a complex database management system.

Chapter 8 : How Do I Use the Connection Object in ADO?

calendrierdelascience.com - Connecting to databases in VBA sounds like it should be complicated, but using the ActiveX Data Objects, or ADO, Object Library makes life an awful lot easier! This.

Chapter 9 : Introduction to ADO - ADO: ActiveX Data Objects [Book]

Select Microsoft ActiveX Data Objects x.x Library from the list. Verify that at least the following libraries are also selected: Verify that at least the following libraries are also selected: Visual Basic for Applications.