ADO Data Bound Dialog Wizard: Overview

The ADO Data Bound Dialog Wizard is a Visual C++ component that guides you through the process of creating a Microsoft Foundation Class Library (MFC) data bound dialog box with ActiveX Data Objects (ADO).

A data bound dialog box provides record level visual access to your data. You can manipulate data in a recordset either programmatically, through the ADO object model, or visually. The ADO Data Bound Dialog Wizard provides a visual interface that enables you to create a dialog box whose controls are bound to the fields of a **Recordset**. Using the Wizard, you can automatically generate all of the resources, classes, and Component Object Model (COM) initialization code necessary to add a data bound dialog box to your project.

Use the ADO Data Bound Dialog Wizard to get a jump-start in writing ADO code in your Visual C++ applications and to learn about ADO 2.0 native C/C++ data type binding support, which allows C/C++ clients to fetch data into native C/C++ types such as `CHAR` and `INT` as well as `VARIANT`s.

NOTE   You must first create or open an MFC project before launching the ADO Data Bound Dialog Wizard.

See Also

ADO Data Bound Dialog - Specifics

ADO Data Bound Dialog - Results

ADO Data Bound Dialog Wizard: Specifics

The ADO Data Bound Dialog Wizard guides you through the process of adding a bound dialog box to your application. You provide the information about your data and the ADO Data Bound Dialog Wizard automatically adds the dialog box to your project.

For more information on each wizard step, see:

- Step 1 - Enter a Connection String
- Step 2 - Specify the Data to Bind
- Step 3 - Specify Cursor Options
- Step 4 - Review Wizard Options

For more information on the exact changes made to your project, see ADO Data Bound Dialog - Results.

See Also

ADO Data Bound Dialog - Overview

ADO Data Bound Dialog Wizard: Results

The ADO Data Bound Dialog component makes the following changes to your project:

- **Adds a source (.cpp) and an include (.h) file to your project.**
  The default name is `RsCgDlg.cpp` and `RsCgDlg.h`. If a project contains multiple data bound dialog boxes, these file names are appended with a number starting with 1. For example, the second data bound dialog box in your project would be named `RsCgDlg1.cpp` and `RsCgDlg1.h`.
- **Adds a dialog box in the project's resource file.**
  The default name is `CG_IDD_RECORDSET`.
- **Adds code to the project's application InitInstance to initialize COM.**
  If you have previously included COM initialization code in your application, you may need to remove the code that the ADO wizard inserts.

You will need to integrate the generated dialog box (`CG_IDD_RECORDSET`) into your application by creating an instance of the generated dialog class and calling its `DoModal()` member function.

See Also

ADO Data Bound Dialog - Overview
ADO Data Bound Dialog - Specifics

ADO Data Bound Dialog Wizard: Step 1 - Enter a Connection String

Complete this step to specify how to connect to your data.

Step Options

Connection String Field
Enter a valid ODBC or OLE DB connection string such as a Data Source Name (DSN).

Build
Click to create an OLE DB connection string.

On the **Provider** tab of the **Data Link Properties** dialog box, select an OLE DB provider, then specify connection properties on the **Connection** tab for the selected provider. Depending on your provider, other properties may need to be specified to establish a connection.

NOTE For more information about the **Data Link Properties** dialog box, press F1 from within the dialog box.

See Also

Step 2 - Specify the Data to Bind

Step 3 - Specify Cursor Options

Step 4 - Review Wizard Options

ADO Data Bound Dialog Wizard: Step 2 - Specify the Data to Bind

Complete this step to specify whether the **Recordset** is based on a command or a table name.

Step Options

Command Text
Enter a valid command whose result set you want to return to a dialog box. You can enter any command text such as a SQL command or a stored procedure as long as it is row returning.

Table
Select the table whose fields you want to return to a dialog box.

See Also

Step 1 - Enter a Connection String

Step 3 - Specify Cursor Options

Step 4 - Review Wizard Options

ADO Data Bound Dialog Wizard: Step 3 - Specify Cursor Options

Complete this step to specify cursor options for the ADO **Recordset** object. The CursorLocation and CursorType properties specify whether to get a client cursor, a server cursor or no cursor at all.

NOTE For more information about ADO cursor properties, see the *ADO Programmer's Reference*. This documentation is available in the Microsoft Data Access SDK.

Step Options

Select the cursor location that you want to use.
This property allows you to choose between various cursor libraries accessible to the provider. Usually, you can choose between using a client-side cursor library or one that is located on the server.

Select a cursor location:

- **adUseClient** - Uses client-side cursors supplied by a local cursor library. Local cursor engines will often allow many features that driver-supplied cursors may not, so using this setting may provide an advantage with respect to features that will be enabled. For backward-compatibility, the synonym **adUseClientBatch is also supported**.
- **adUseClientBatch** - Same as adUseClient.
- **adUseServer** - Default. Uses data provider- or driver-supplied cursors. These cursors are sometimes very flexible and allow for additional sensitivity to changes others make to the data source. However, some features of the Microsoft Client Cursor Provider (such as disassociated recordsets) cannot be simulated with server-side cursors and these features will be unavailable with this setting.
- **adUseNone** - No cursor services used. (This constant is obsolete and appears solely for the sake of backwards compatibility.)

Select the cursor type that you want to use.
Use the CursorType property to specify the type of cursor that should be used when opening the **Recordset** object. The CursorType property is read/write when the **Recordset** is closed and read-only when it is open.

Select a cursor type:

- **adOpenForwardOnly** - Forward-only cursor. Default. Identical to a static cursor except that you can only scroll forward through records. This improves performance in situations when you only need to make a single pass through a recordset.
- **adOpenKeyset** - Keyset cursor. Like a dynamic cursor, except that you can't see records that other users add, although records that other users delete are inaccessible from your recordset. Data changes by other users are still visible.
- **adOpenDynamic** - Dynamic cursor. Additions, changes, and deletions by other users are visible, and all types of movement through the recordset are allowed, except for bookmarks if the provider doesn't support them.
- **adOpenStatic** - Static cursor. A static copy of a set of records that you can use to find data or generate reports. Additions, changes, or deletions by other users are not visible.

See Also

Step 1 - Enter a Connection String

Step 2 - Specify the Data to Bind

Step 4 - Review Wizard Options

ADO Data Bound Dialog Wizard: Step 4 - Review Wizard Options

Complete this step to review the changes that the ADO Data Bound Dialog Wizard will make to your project. Changes include the following:

Files that Will be Created
Lists the name and location of the source (.cpp) and include (.h) files that the ADO Data Bound Dialog Wizard generates.

Other Modifications
Lists all modifications that the ADO Data Bound Dialog Wizard makes to your project files.

Warnings
Lists any ADO Data Bound Dialog Wizard issues that may cause unpredictable behavior in your project.

For more information on the exact changes made to your project, see ADO Data Bound Dialog - Results

See Also

Step 1 - Enter a Connection String

Step 2 - Specify the Data to Bind

Step 3 - Specify Cursor Options