

# Macro Actions

**T**his article provides a summary of all the macro actions available in Microsoft Access 2010. The macros are organized in the following functional categories, as listed in the macro action catalog in Access 2010:

- ADP Objects
- Data Entry Operations
- Data Import/Export
- Database Objects
- Filter/Query/Search
- Macro Commands
- System Commands
- User Interface Macros
- Windows Management

Note that you won't see the ADP Objects category displayed in the Action Catalog unless you are working with an Access Data Project (.adp). The Trusted column indicates whether the macro action is a trusted action that can run even when the database is not trusted. (A macro action that is trusted has a lightning icon next to its name in the Action Catalog.) When an action is not trusted, Access won't run the action unless it is in a trusted database. (A macro action that is not trusted has an exclamation point next to its name in the Action Catalog.)

Table A6-1 ADP Objects

Macro Action	Purpose	Trusted
CopyDatabaseFile	In an Access project file (.adp) connected to a Microsoft SQL Server database, copies the currently connected database to a new file. The user must have system administrator privileges on the server to perform this action. You cannot execute this action in an Access desktop database (.accdb).	No
OpenDiagram	In an Access project file (.adp) connected to an SQL Server database, opens a table relationship diagram in the server database in Design view. You cannot execute this action in an Access desktop database (.accdb).	No
OpenFunction	In an Access project file (.adp) connected to an SQL Server database, opens a function in the server database in Datasheet, Design, PivotTable, or PivotChart view or in Print Preview. If the function is a data definition command or the equivalent of an Access action query, executes the function without returning data. If the function returns data that is editable, you can specify whether the function datasheet should be opened to add new records only; to add, edit, and delete records; or to provide a read-only view of the data. In Microsoft Visual Basic, you can also use the Open or Execute method to open the function and return any results to a recordset. You must use the OpenFunction macro action within a Visual Basic procedure if you want the function to open in the user interface. You cannot execute this action in an Access desktop database (.accdb).	No
OpenStoredProcedure	In an Access project file (.adp) connected to an SQL Server database, opens a stored procedure in the server database in Datasheet, Design, PivotTable, or PivotChart view or in Print Preview. If the stored procedure returns data that is editable, you can specify whether the stored procedure datasheet should be opened to add new records only; to add, edit, and delete records; or to provide a read-only view of the data. In Visual Basic, you can also use the Open or Execute method to open the stored procedure and return any results to a recordset. You cannot execute this action in an Access desktop database (.accdb).	No

Macro Action	Purpose	Trusted
OpenView	<p>In an Access project file (.adp) connected to an SQL Server database, opens a view in the server database in Datasheet, Design, PivotTable, or PivotChart view or in Print Preview. If the view returns data that is editable, you can specify whether the view datasheet should be opened to add new records only; to add, edit, and delete records; or to provide a read-only view of the data.</p> <p>In Visual Basic, you can also use the Open or Execute method to open the view and return the results to a recordset.</p> <p>You cannot execute this action in an Access desktop database (.accdb).</p>	No
TransferSQLDatabase	<p>In an Access project file (.adp) connected to an SQL Server database, transfers the currently connected database to another server and database name. If you want, you can transfer only the table structure. The user must have system administrator privileges on the target server to perform this action.</p> <p>You cannot execute this action in an Access desktop database (.accdb).</p>	No

Table A6-2 Data Entry Operations

Macro Action	Purpose	Trusted
DeleteRecord	Deletes the current record from the source table.	Yes
EditListItems	Opens the Edit List Items dialog box, where you can add, edit, or delete items in a combo box or list box control. The Row Source Type must be Value List, and you'll need to set the focus to the control if the control does not already have focus. You can set the focus to the combo box or list box control by using the GoToControl macro action.	Yes
SaveRecord	Saves the current record changes.	Yes

Table A6-3 Data Import/Export

Macro Action	Purpose	Trusted
AddContactFromOutlook	Adds a saved contact from your Microsoft Outlook address book into your database. Running this macro action opens the Select Names To Add Outlook dialog box, where you can select the saved contact in your address book to add to your Access table. Note that you must have a table, query, or bound form open before running this action. To use the action, your table field properties must also be mapped to WSSFieldID property values. You need to use Visual Basic for Applications (VBA) code to set up these field mappings. If you use the Contacts Application Part, you can examine the field properties for the Contacts table using VBA code to see the WSSFieldID field mappings needed to use this action with a custom table.	Yes
CollectDataViaEmail	Opens the Collect Data Through E-mail Messages wizard, where you can create a Hypertext Markup Language (HTML) message or InfoPath form to collect data for your database. Note that you must have a table, query, or form open before running this action.	Yes
EmailDatabaseObject	Outputs a table, query, report, module, view, stored procedure, or function datasheet, or a form datasheet to an HTML (.htm, .html), a Microsoft Excel (.xls, .xlsb, .xlsx), a Rich Text Format (.rtf) a portable document format (.pdf), an XML Paper Specification format (.xps), or a text (.txt) file and embeds the data in an electronic mail message. You can output a module as plain text only. You can specify to whom the message is to be sent, the message subject, additional message text, and whether the message can be edited before it is sent. You must have email software installed that conforms to the Messaging Application Programming Interface (MAPI) standard.	Yes

Macro Action	Purpose	Trusted
ExportWithFormatting	Outputs the named table, query, form, report, module, view, stored procedure, or function to another file format. The formats include HTML (.htm, .html), Microsoft Excel (.xls, .xlsb, .xlsx), text files (.txt), Rich Text Format (.rtf), Portable Document Format (.pdf) or XML Paper Specification Format (.xps). Modules can be output only in text format. You can also optionally start the application to edit the file. For forms, the data output is from the form's Datasheet view. For reports, Access outputs all controls containing data (including calculated controls) except ActiveX controls.	Yes
ImportExportData	Exports data to or imports data from another Access, dBASE, Microsoft SharePoint list, or SQL database. You can also use this action to attach tables or files from other Access, dBASE, or SQL database, or from text or spreadsheet files. You can also import or export the definition of queries, views, stored procedures, functions, diagrams, forms, reports, macros, or modules to or from another Access desktop database (.accdb) or project file (.adp).	No
ImportExportSpreadsheet	Exports data to or links or imports data from Excel or Lotus 1-2-3 spreadsheet files.	No
ImportExportText	Exports data to or links or imports data from text files. You can also link or import tables embedded within HTML (.htm, .html) files.	No
ImportSharePointList	Imports data or links data from a SharePoint Services site. If you want, you can also import all display values for lookup columns.	No
RunSavedImportExport	Runs a saved import or export specification you previously saved using the Import Wizard or Export Wizard.	No
SaveAsOutlookContact	Saves the current record as a new contact in your Outlook address book. Note that you must have a table, query, or bound form open before running this action. To use the action, your table field properties must also be mapped to WSS-FieldID property values. You need to use VBA code to set up these field mappings. If you use the Contacts Application Part, you can examine the field properties for the Contacts table using VBA code to see the WSSFieldID field mappings needed to use this action with a custom table.	Yes

Macro Action	Purpose	Trusted
WordMailMerge	Opens the Microsoft Word Mail Merge Wizard, where you can select the option to link to an existing Word document or create and link to a new document. If you choose to embed the data in an existing document, the wizard displays the Select Microsoft Word Document dialog box, which asks you to specify the document location. Note that you must have a table, query, or form open before running this action.	Yes

Table A6-4 Database Objects

Macro Action	Purpose	Trusted
CopyObject	Copies any database object to the current database using a new name or copies any database object to another Access database using any specified name.	No
DeleteObject	Deletes any table, query, form, report, macro, module, view, stored procedure, function, or diagram. If you do not specify an object, the action deletes the object currently selected in the Navigation pane. In Visual Basic, you should use the Delete method of the object collection to delete an object.	No
GoToControl	Sets the focus to the specified control. In Visual Basic, you should use the SetFocus method of the control to move the focus.	Yes
GoToPage	Moves to the specified page in a form. In Visual Basic, you should use the GoToPage method of the form object.	Yes
GoToRecord	Moves to a different record and makes it current in the specified table, query, or form. You can move to the first, last, next, previous, or new record. When you specify the next or the previous record, you can also specify a parameter to move by more than one record. You can also go to a specific record number or to the new-record placeholder at the end of the set. In Visual Basic, you can also use RunCommand RecordsGoToFirst, RecordsGoToLast, RecordsGoToNew, RecordsGoToNext, or RecordsGoToPrevious. A more efficient technique is to search in the form's recordset and move the form to the desired record by setting the form's Bookmark property.	Yes

Macro Action	Purpose	Trusted
OpenForm	Opens a form in Form, Design, Datasheet, PivotTable, PivotChart, or Layout view or in Print Preview. You can also apply a filter or a Where condition in Form, Datasheet, PivotTable, or PivotChart view or in Print Preview. Access ignores any filter or Where condition when you open the object in Design view. If the form is already open, the OpenForm action puts the focus on the form and applies any new Filter or Where Condition argument you specify. From a Visual Basic procedure, you normally execute the macro action to open a form. However, you can also open a form that has a module by setting a form object equal to a new instance of the form's class module. See Chapter 24, "Understanding Visual Basic Fundamentals," on the companion CD, for details. This action is not trusted when you set the View argument to Design or Layout.	Dependent on argument
OpenReport	Prints a report or opens a report in Report view (the default), Print Preview, Layout view, or Design view. For printing, Report view, and Print Preview, you can also specify a filter or a Where condition. This action is not trusted when you set the View argument to Design or Layout.	Dependent on argument
OpenTable	Opens a table in Datasheet, Design, PivotTable, or PivotChart view or in Print Preview. You can specify whether the table datasheet should be opened to add new records only; to add, edit, and delete records; or to provide a read-only view of the data. In Visual Basic, you can also use the OpenRecordset method to create a recordset from a table. This action is not trusted when you set the View argument to Design.	Dependent on argument
PrintObject	Prints the currently selected object.	Yes
PrintPreview	Opens the currently selected object in Print Preview.	Yes
RenameObject	Renames the specified object in the current database.	No
RepaintObject	Forces the repainting of the window for the specified object. Forces recalculation of any formulas in controls on that object. If you do not specify an object type and name, repaints the active window. In Visual Basic, you should use the Repaint method of a form.	Yes
SaveObject	Saves any table, query, form, report, macro, module, view, stored procedure, function, or diagram. If you do not specify an object type and object name, the definition of the currently active object is saved. If you provide only an object name, the active object is saved with the new name you specify (in a Save As operation).	No

Macro Action	Purpose	Trusted
SelectObject	Selects the specified object. Restores the object's window if it was minimized. If the object is in the process of opening (for example, a form referenced in a previous OpenForm action), SelectObject forces the object to finish opening before performing the next action. Use this action after OpenForm or BrowseTo when you need to immediately reference the form, a property of a control on the form, or data in a control on the form. In Visual Basic, you should use the SetFocus method of the form to force it to finish opening or to move the focus to the form window.	Yes
SetProperty	Changes selected properties of a control on a form or report or selected properties of a form or report. The properties that you can change with this action are Enabled, Visible, Locked, Left, Top, Width, Height, Fore Color, Back Color, Caption and Value. If you are using a web database, you cannot set the Left, Top, Width, or Height properties on web form controls. In Visual Basic, you should use a Let statement.	Yes
SetValue	Changes the value of any control or property that you can update. For example, you can use the SetValue action to calculate a new total in an unbound control or to affect the Visible property of a control (which determines whether you can see that control). In Visual Basic, you should use a Let statement.	No

Table A6-5 Filter/Query/Search

Macro Action	Purpose	Trusted
ApplyFilter	Restricts the information displayed in a table, form, or report by applying a named filter, a query, or an SQL WHERE clause to the records in the table or to the records in the underlying table or query of the form or report. ApplyFilter always operates on the currently active window and does not work for subforms. In Visual Basic, you should set the Filter or OrderBy property of the form or report and set FilterOn or OrderByOn to True.	Yes
FindNextRecord	Finds the next record that meets the criteria previously set by a FindRecord macro action or in the Find And Replace dialog box. In Visual Basic, you should provide your own custom search form and perform the search by using the Find, FindFirst, or FindNext method of the form's recordset.	Yes



Macro Action	Purpose	Trusted
FindRecord	Finds a record that meets the search criteria. You can specify in the macro action all the parameters available in the Find And Replace dialog box. In Visual Basic, you should provide your own custom search form and perform the search by using the Find, FindFirst, or FindNext method of the form's recordset.	Yes
OpenQuery	In an Access desktop database (.accdb), opens a query in Datasheet, Design, PivotTable, or PivotChart view or in Print Preview. If you specify an action query, Access performs the updates specified by the query. If the query returns data that is editable, you can specify whether the query datasheet should be opened to add new records only; to add, edit, and delete records; or to provide a read-only view of the data. In Visual Basic, you can use the OpenRecordset method to create a recordset from a query that returns records or use the Execute method to run an action query. You cannot execute this action in an Access project file (.adp). This action is not trusted when you set the View argument to Design.	Dependent on argument
Refresh	Refreshes the data in the active object (form, report, or datasheet). In Visual Basic, you can use the Refresh method for the same functionality.	Yes
RefreshRecord	Refreshes the data in the current record only for the active object (form, report, or datasheet).	Yes
RemoveFilterSort	Removes any filter or sort applied to the active object (form, report, or datasheet).	Yes
Requery	Refreshes the data in a control that is bound to a query (such as a list box, a combo box, a subform, or a control based on an aggregate function, such as DSum). When other actions (such as inserting or deleting a record in the underlying query) might affect the contents of a control that is bound to a query, use the Requery action to update the control values. Use Requery without an argument to refresh the data in the active object (form, report, or datasheet). In Visual Basic, you should use the Requery method of the object.	Yes

Macro Action	Purpose	Trusted
RunSQL	<p>Executes the specified action query statement (INSERT INTO, DELETE, SELECT...INTO, UPDATE) or data definition query statement (CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, DROP INDEX). (Note: You can't enter more than 255 characters in the SQL Statement argument. If you need to run a more complex query, define a query object and use the OpenQuery action.)</p> <p>In Visual Basic, you should use the Execute method to run an action or data definition query contained in a string argument of any length. Also, the RunSQL command executed from Visual Basic accepts an SQL statement up to 32,767 characters.</p>	No
SearchForRecord	<p>Searches for a record using the search criteria you specify and makes it current in the specified table, query, or form. You can start your search from the previous, next, first, or last record.</p> <p>In Visual Basic, you should provide your own custom search form and perform the search by using the Find, FindFirst, or FindNext method of the form's recordset.</p>	Yes
SetFilter	<p>Restricts the information displayed in a table, form, or report by applying a named filter, a query, or an SQL WHERE clause to the records in the table or to the records in the underlying table or query of the form or report. When you are working with web forms, you cannot use the ApplyFilter action, so you need to use the SetFilter action to apply a filter to a web form.</p>	Yes
SetOrderBy	<p>Applies a sort to the records in a table, or the records from the form or report's underlying table or query. When you are working in a web database, you can use the SetOrderBy action to sort the records displayed in a web form.</p>	Yes
ShowAllRecords	<p>Removes any filters previously applied to the active table, query, or form.</p>	Yes

Table A6-6 Macro Commands

Macro Action	Purpose	Trusted
CancelEvent	<p>Cancels the event that caused this macro to run. You can't use a CancelEvent action in a macro that defines menu commands or in response to an event that cannot be canceled.</p> <p>In Visual Basic, you should set the Cancel parameter of the event procedure to True to cancel the event.</p>	Yes

Macro Action	Purpose	Trusted
ClearMacroError	Clears any information stored in the MacroError object, including the error number, error description, macro name, action name, condition, and any arguments. Access resets the error number to 0 after you run this action.	Yes
Echo	Controls the display of intermediate actions while a macro runs. In Visual Basic, you should set the Echo property of the Application object to True or False.	No
OnError	Specifies how Access should handle an error when running your macro. You can turn off error trapping, skip to the next action, or go to a specific macro in the same macro group to handle the error.	Yes
OpenVisualBasicModule	Opens a module or procedure in Design view. If you specify a module name and no procedure name, Access opens the module to the Declarations section. You can also specify only a procedure name so long as the procedure is a public procedure in a standard module. In a Visual Basic procedure, you should open a module by setting a module object equal to the name of the module in the Modules collection. To open the module of a form or report, the form or report itself must be open.	No
RemoveAllTempVars	Clears from memory all temporary variables that you create by using the SetTempVar action. Access automatically clears all temporary variables from memory when you close the database.	Yes
RemoveTempVar	Clears from memory a single temporary variable that you create by using the SetTempVar action. Access automatically clears all temporary variables from memory when you close the database.	Yes
RunCode	Executes a Visual Basic function procedure. Other actions following this action execute after the function completes. You cannot call a Visual Basic subprocedure from this action. (Note: If the function returns values, you cannot inspect them in the macro.) In Visual Basic, you should call the subprocedure or assign the function call to a return variable.	Yes

Macro Action	Purpose	Trusted
RunDataMacro	Runs a named data macro attached to a table. If the named data macro has any parameters, Access displays text boxes on the macro design surface for each parameter value. Actions following this action run after the named data macro completes if you set the Wait For Post Processing form property to Yes.	Yes
RunMacro	Runs a macro. Actions following this RunMacro action run after the called macro completes. You can specify a number of times for the macro to execute or a condition that, when true, halts the macro execution. Caution: A macro can run itself, but you should provide conditional testing that exits the macro so that you don't create an unending loop. In Visual Basic, use the Do and For statements to create iterative or conditional code loops. Although it is possible to execute RunMacro in a Visual Basic procedure, you should write the equivalent Visual Basic statements within your procedure instead.	Yes
RunMenuCommand	Executes an Access built-in command. The list of available commands includes all commands that you can execute from any of the ribbons. If you use macros to define a custom menu, you can use a RunMenuCommand action to make selected Access menu commands available on your custom menu. Any command that modifies a design element (such as applying an AutoFormat) is not trusted.	Dependent on argument
SetLocalVar	Creates a temporary local variable and lets you set it to a value that you can reference while the macro executes. The value of the variable stays in memory so long as the macro is running. Once the macro finishes, Access clears the local variable from memory. You'll find this action useful when you need to perform calculations inside the macro actions, such as calculating a running sum.	Yes
SetTempVar	Creates a temporary variable and lets you set it to a value that you can reference in other areas of your database. The value of the variable stays in memory as long as the database remains open or until you clear the variable using either the RemoveTempVar or RemoveAllTempVars action.	Yes

Macro Action	Purpose	Trusted
SingleStep	Places the currently running macro in Single Step mode and opens the Single Step dialog box. Click Continue to run the rest of the macro. If you single-step through an entire macro, Single Step mode will still be in effect when the next macro runs. When executed from Visual Basic, the next macro to execute will run in Single Step mode.	Yes
StartNewWorkflow	Opens the Workflows dialog box, which lists all defined workflows for a linked SharePoint list. You can click the Start button next to a specific workflow to start it. If no workflows are defined for the linked SharePoint list, Access displays no information in the Workflows dialog box. Note that you can use this action only with linked SharePoint lists.	Yes
StopAllMacros	Stops all macros, including any macros that called this macro. You cannot execute this action from Visual Basic.	Yes
StopMacro	Stops the current macro. You cannot execute this action from Visual Basic.	Yes
WorkflowTasks	Opens the Workflows Tasks dialog box, where you can view a list of any workflow tasks associated with a linked SharePoint list. If no workflows are defined for the linked SharePoint list, Access displays no information in the Workflow Tasks dialog box. Note that you can use this action only with linked SharePoint lists.	Yes

Table A6-7 System Commands

Macro Action	Purpose	Trusted
Beep	Produces a sound.	Yes
CloseDatabase	Closes all Access windows and closes the database. If you have any unsaved objects open, Access prompts you to save them.	Yes
DisplayHourglassPointer	Changes the mouse pointer to an hourglass icon while a macro runs. In Visual Basic, you can also set the Mouse-Pointer property of the Screen object.	Yes
OpenSharePointList	Browses to the selected linked SharePoint list in the Navigation pane using your web browser. Note that you can use this action only if you have one or more linked SharePoint lists in your database.	No

Macro Action	Purpose	Trusted
OpenSharePointRecycleBin	Browses to the Recycle Bin page on your SharePoint site. Note that you can use this action only if you have one or more linked SharePoint lists in your database.	No
PrintOut	Prints the active datasheet, form, module, or report. You can specify a range of pages, the print quality, the number of copies, and collation. Use an Open action first if you want to apply a filter or a Where condition.	No
QuitAccess	Closes all Access windows and exits Access. You can set options to save all changes (the default), prompt the user to save, or exit without saving. In Visual Basic, you should use the Quit method of the Application object, which has the same options. If you specify Prompt, this macro is trusted. If you specify Save All or Exit, the macro is not trusted.	Depends on argument
RunApplication	Starts another Windows-based or MS-DOS-based application. In Visual Basic, use the Shell function or ActiveX application automation to open and control another application.	No
SendKeys	Stores keystrokes in the keyboard buffer. If you intend to send keystrokes to a modal form or a dialog box, you must execute the SendKeys action before opening the modal form or the dialog box. In Visual Basic, you should use the SendKeys statement.	No
SetWarnings	When enabled, causes an automatic Enter key response to all system warning or informational messages while a macro runs. For warning messages displayed in a dialog box, pressing Enter selects the default button (usually OK or Yes). Run this action when your code is about to execute action queries and you do not want the user to see the update warnings. Setting SetWarnings to No does not halt the display of error messages. Use the Echo macro action with the Echo On argument set to No to avoid displaying the error messages. In Visual Basic, you should set the SetWarnings property of the Application object to True or False.	No

## INSIDE OUT

### Avoid SendKeys If Possible

Although you can set an optional parameter to wait until Windows processes the keystrokes, you have only limited control over where Windows actually delivers the keys. For example, if the user clicks another window at the moment you issue the SendKeys action, Windows delivers the keystrokes to that window. If you want to deliver the keystrokes to a dialog box, you must issue the SendKeys with no wait, immediately open the dialog box, and hope that Windows will send the keystrokes there. For example, if you want to open the Find dialog box to perform a search on the LastName control and set the default Match to Any Part Of Field, you need to execute the following commands:

```
GoToControl Control Name: LastName
```

```
SendKeys Keystrokes: %H A %N
```

```
Wait: No
```

```
RunCommand Command: Find
```

The SendKeys command queues up Alt+H A Alt+N, which is the key combination you could use to move to the Match option (H is the hotkey), press A to select the entry beginning with that letter (Any Part Of Field), and then use Alt+N to get back to the Find What (N is the hotkey) box. This works a vast majority of the time, but if some other application pops forward a dialog box right after the SendKeys and before the RunMenuCommand, the keystrokes won't go where you intended.



The bottom line is always to work hard to figure out an alternative way to accomplish a task that you might be tempted to do with SendKeys. In the previous example, a custom search form would work much better. See Chapter 25, "Automating Your Application with Visual Basic," on the companion CD, for details about how to build a custom search form (also known as custom query by form).

Table A6-8 User Interface Macros

Macro Action	Purpose	Trusted
AddMenu	Adds a drop-down menu to a custom menu bar or to a custom shortcut menu for a form or a report. This is the only action allowed in a macro referenced by a Menu Bar or Shortcut Menu Bar property. The arguments to AddMenu specify the name of this menu bar and the name of another macro that contains all the named commands for the menu and the actions that correspond to those commands. An AddMenu action can also build submenus by referring to another macro that uses an AddMenu action. Any custom menu that you define with macros appears on the Add-Ins tab on the ribbon when you open a form or report that references the menu macro. This feature is retained for compatibility with versions of Access before Access 2007.	Yes
BrowseTo	Browses to a different top-level form object or report object or browse to a different form or report displayed inside a subform control, such as a navigation control. You can also apply a filter or a Where condition to the form or report you browse to. If you are browsing to a web continuous form in a web database, you can also specify a specific page that you want to browse to when you view the form in a web browser. If the form returns data that is editable, you can specify whether the form should be opened to add new records only; to add, edit, and delete records; or to provide a read-only view of the data. If the underlying query for the form or report has any parameters, Access displays text boxes on the macro design surface for each parameter value.	Yes
LockNavigationPane	Prevents users from deleting objects or deleting object shortcuts in the Navigation pane.	Yes
MessageBox	Displays a warning or an informational message in a dialog box, and also produces a sound if you want. The user must click OK to close the dialog box and proceed. In Visual Basic, you should use the MsgBox function. When you use the function, your code can also specify optional buttons and test the return value to determine which button the user clicked.	Yes



Macro Action	Purpose	Trusted
NavigateTo	Selects the category to display from the list available under Navigate To Category on the Navigation Pane menu. Use the optional Group argument to filter the selected category to display a specific group of objects.	Yes
Redo	Redoes the last action. This action performs the same functionality as clicking the Redo button on the Quick Access Toolbar.	Yes
SetDisplayedCategories	Shows or hides the selected available category listed under Navigate To Category on the Navigation Pane menu. You can also show or hide all categories.	Yes
SetMenuItem	Sets the enabled or checked status of a menu item on a custom menu bar or a custom shortcut menu. Menu items can be enabled or disabled, checked or unchecked. In Visual Basic, you should set the Enabled property of a custom CommandBar control to enable or disable it. Set the State property of a custom CommandBar control to check or uncheck it. Any custom menu that you define with macros or in Visual Basic code appears on the Add-Ins tab on the ribbon when you open a form or report that references the menu. This feature is retained for compatibility with versions of Access before Access 2007.	Yes
ShowToolbar	Shows or hides any custom toolbars. Although you cannot design custom toolbars in the user interface, you can define a custom toolbar with macros or in Visual Basic. In Visual Basic, you can also set the Visible property of a CommandBar object. Any custom toolbar that you define with macros or in Visual Basic code appears on the Add-Ins tab on the ribbon when you open a form or report that references the menu or when you execute a ShowToolbar command. This feature is retained for compatibility with versions of Access before Access 2007.	No
UndoRecord	Undo all changes to the current record. This action performs the same functionality as clicking the Undo button on the Quick Access Toolbar.	Yes

Table A6-9 Windows Management

Macro Action	Purpose	Trusted
CloseWindow	Closes either the specified window or, if no window is specified, the active window for a table, query, form, or report. If the Navigation pane has the focus when you execute a CloseWindow action with no window specified, Access closes the database. You can also indicate whether to save the object when Access closes it. This action is not trusted unless you set the Save argument to Prompt.	Depends on argument
MaximizeWindow	Maximizes the active window.	Yes
MinimizeWindow	Minimizes the active window.	Yes
MoveAndSizeWindow	Moves and sizes the active window.	Yes
RestoreWindow	Restores a maximized or minimized window to its previous size.	Yes

The following table lists the 26 web-supported macro actions and their arguments. Web macro actions, in general, support fewer arguments than client macro actions. You can refer to the tables above for descriptions of these macros. Note that all web macros are trusted.

Table A6-10 Web-Supported Macro Actions

Category	Macro Action	Arguments
Data Entry Operations	DeleteRecord	None
	SaveRecord	None
Database Objects	GoToControl	Control Name
	GoToRecord	Record
	OpenForm	Form Name, Where Condition, Data Mode, Window Mode
	OpenReport	Report Name, Where Condition, Window Mode
	SetProperty	Control Name, Property, Value
Filter/Query/Search	RefreshRecord	None
	Requery	Control Name
	SetFilter	Where Condition, Control Name
	SetOrderBy	Order By, Control Name
Macro Commands	ClearMacroError	None
	OnError	Go to, Macro Name

Category	Macro Action	Arguments
	RemoveAllTempVars	None
	RemoveTempVar	Name
	RunDataMacro	Macro Name
	RunMacro	Macro Name
	RunMenuCommand	Command
	SetLocalVar	Name, Expression
	SetTempVar	Name, Expression
	StopAllMacros	None
	StopMacro	None
User Interface Macros	BrowseTo	Object Type, Object Name, Path to Subform Control, Where
	MessageBox	Message
	UndoRecord	None
Windows Management	CloseWindow	None

Microsoft renamed some of the macro action names in Access 2010. The following table lists the older macro names and their new equivalents. Note that you can still type the older macro name in the Add New Action combo box and then press Enter, and Access displays the new macro action name on the macro design surface. You can also type the older name in the Action Catalog search box, and Access filters the search to any similar macro names or descriptions.

**Table A6-11** Renamed Macro Actions

Older Macro Action Name	Access 2010 Macro Action Name
AddFromOutlook	AddContactFromOutlook
BrowseSharePointList	OpenSharePointList
Close	CloseWindow
Delete	DeleteRecord
FindNextRecord	FindNext
Hourglass	DisplayHourglassPointer
Maximize	MaximizeWindow
Minimize	MinimizeWindow
MoveSize	MoveAndSizeWindow
MsgBox	MessageBox
OpenModule	OpenVisualBasicModule

Older Macro Action Name	Access 2010 Macro Action Name
OutputTo	ExportWithFormatting
Print	PrintObject
Quit	QuitAccess
Rename	RenameObject
Restore	RestoreWindow
RunApp	RunApplication
RunCommand	RunMenuCommand
Save	SaveObject
SharePointSiteRecycleBin	OpenSharePointRecycleBin
TransferDatabase	ImportExportData
TransferSharePointList	ImportSharePointList
TransferSpreadsheet	ImportExportSpreadsheet
TransferText	ImportExportText
Undo	UndoRecord