Microsoft Access Tab Control
Version 1.0.004

Andrew R. Miller (andrewm@microsoft.com)

TAB.OCX is an OLE Custom Control designed to be used with Microsoft Access 2.0.   The control provides a "tabbed
" user-interface, much like the dialogs in Microsoft Word for Windows 6.0 and Microsoft Graph 5.0.   The tab control provides the following features:

* Three Different Special Effects
* Tabs can be located on top, left, right, or bottom borders
* Tab text can be rotated 0, 90, 180, or 270 degrees
* Tab body can be "Open
" or
"Closed
" (see below)
* BeforeUpdate and AfterUpdate events
* Works with Access 2.0

Installation
Upgrading from a Previous Version
Using the Control
Properties
Events
Methods
Samples
Distributing the Control
Creating a Custom Setup Containing TAB.OCX
Notes

Installation

Before installing this OLE Control, you must be running OLE 2.02 (Microsoft Access 2.0 includes OLE 2.01). You can find OLE 2.02 in Library 10 of the MSACCESS forum on Compuserve as OLE202.ZIP.

Once OLE 2.02 has been installed, you need to install and register the tab control files.   Unlike a Visual Basic Custom Controls (.VBX's), OLE Controls must be reigstered with the system before they can be used.   The following section describes how to install and register both TAB.OCX as well as OC25.DLL, the 16-bit OLE Controls support DLL.

Installing and Registering the Tab Control Files

1) Copy TAB.OCX and TAB.LIC to your \ACCESS directory.
2) Copy OC25.DLL to your \WINDOWS\SYSTEM directory.
3) From the Program Manager or File Manager, run

        REGSVR.EXE C:\WINDOWS\SYSTEM\OC25.DLL

and

        REGSVR.EXE C:\ACCESS\TAB.OCX

If you are an Access Developer's Toolkit customer, you will note that REGSVR.EXE and MSAREG.EXE apparently do the same thing (register controls).   There is a good reason for this - MSAREG.EXE is simply a slient version of REGSVR.EXE (i.e. it doesnt put up any message boxes).

Registering Controls using Microsoft Access

Once OC25.DLL has been registered, you can register controls using REGSVR, MSAREG, or Microsoft Access (the instructions above use REGSVR for simplicity).   To register OLE Controls using Microsoft Access, follow these steps:

1) Start Microsoft Access.
2) New Form.
3) Choose **Insert Object** from the **Edit** menu.
4) Click on **Insert Control**
5) Click **Add Control...**
6) Select TAB.OCX (or another control) and hit OK.

Microsoft Access will then ask the control to register itself exactly as REGSVR does.

Unregistering Controls

REGSVR.EXE and MSAREG.EXE support a /u flag which will allow you to unregister a control.   This option will ask the control to unregister itself from the system.   For example, you would run this from the Program Manager to unregister the Tab control:

        REGSVR /U C:\ACCESS\TAB.OCX

Upgrading from a Previous Version

If you are upgrading from a previous version of the Tab Control, there are several things worth noting:

1) TAB.OCX is now version 1.0.004.

2) TAB.OCX requires OLE 2.02.

3) OC1016.DLL has been replaced by OC25.DLL.   OC1016.DLL is no longer required for the Tab Control.   If you would like to unregister OC1016.DLL, use REGSVR /u as described in the Installation help topic.   If you are using other controls like MSLB.OCX, OUTL1016.OCX, or MSACAL20.OCX, you still need OC1016.DLL. OC1016.DLL and OC25.DLL can live peacefully on your system at the same time.

4) The Tab Control now supports two Raised styles.   The **Raised (Thick Bevel)** style is the same as the **Raised** style in previous versions.   The **Raised (Thin Bevel)** style will give you tabs that are more subtle, and match up with Microsoft Access raised rectangles.

5) MSAREG.EXE is no longer included in the .ZIP file.   It has been replaced by REGSVR.EXE. MSAREG is a silent version of REGSVR.EXE that comes with the Access Developer's Toolkit.   A number of Tab Control users were confused by the silent nature of MSAREG.EXE.

6) You will need to reset your Font and Back Color properties.   Sorry, there's no way for me to upgrade these properties for you.   All other properties should be OK.

7) Performance has been slightly improved in some areas.   Stay tuned for even more improvements with left, right, and bottom oriented text.

8) This OLE Control has been built with Visual C++ 1.51 and the released version of the Control Developer's Kit (CDK).   Previous versions were built with a preliminary release of the CDK.

Using the Control

Inserting a Tab Control

Once OC25.DLL and TAB.OCX have been copied to your system and registered with OLE, you will need to insert the control into a form or report to use it.

1) Start Microsoft Access
2) New Form.
3) From the **Edit** menu, choose **Insert Object**.
4) Click on **Insert Control**
5) Select "Tab Control
" and hit OK.

Setting Properties

After inserting it into a form, you can set the properties of the control by right clicking on the control and selecting **Tab Control Object > Properties...** from the menu.   Optionally, you can select the control and then choose **Tab Control Object > Properties...** from the **Edit** menu.   Either of these methods will bring up the OLE Controls property sheet that will allow you to set various properties.   For more information on these properties, see Properties.

The property sheet should look like this:
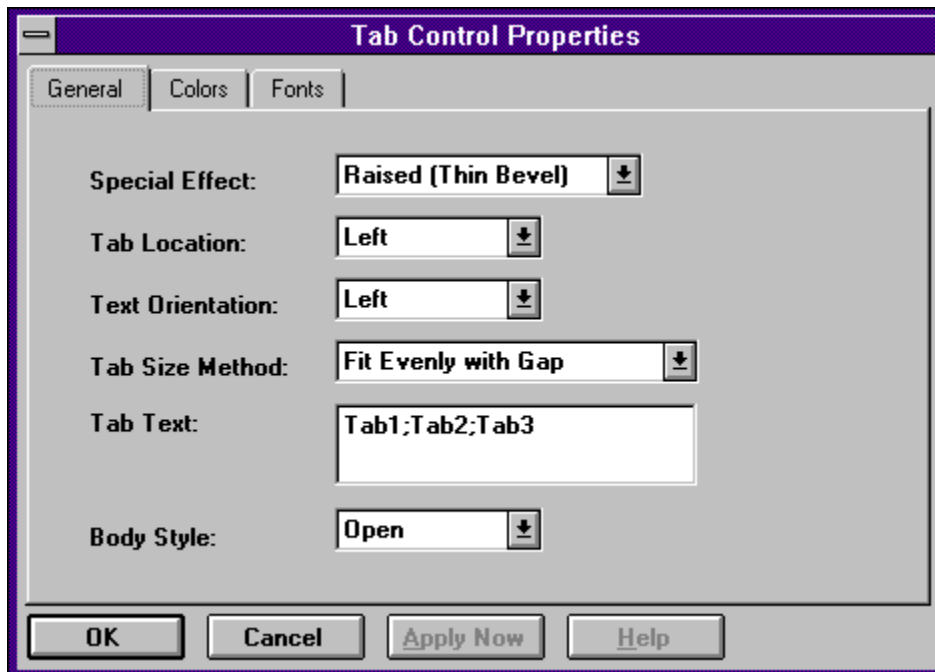
NOTE: If you don't see a Font or Color tab on your property sheet, then OC25.DLL has not been registered properly.   For more information, see Installation.

A Note About Windowed Controls

OLE Controls are windowed controls, which means they have their own window (Access controls like

command buttons typically do not).   In Access, windowed controls paint in a layer above the form, which means you Access controls will never paint on top of a windowed control.   This is a problem for a control like the tab control, since it is designed to live behind other controls.

To get around this, the tab control exposes a property called "Body Style
".   This property has two settings: Open and Closed.   A
"closed
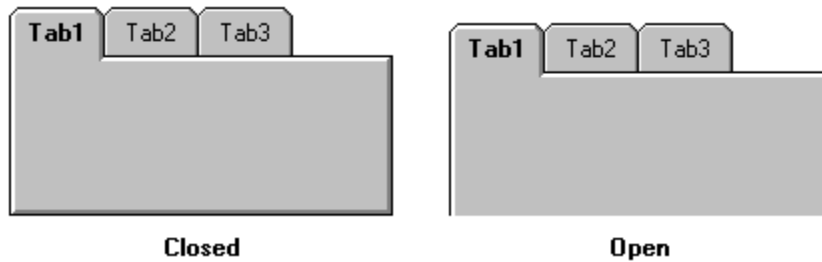" tab control consists of the tabs on top of a rectangle.   An
"open
" tab control is just like the closed control, but the
"bottom
" border is not painted.   So, if you
're tabs are on the left hand side, the right border of the tab control body will not be painted.

To create a tabbed dialog look, the user can combine an open tab control with an Access rectangle, or with a closed tab control that has been converted to a picture.   Both the Access rectangle and the picture object can live beneath other Access controls, so we avoid the problem described above.
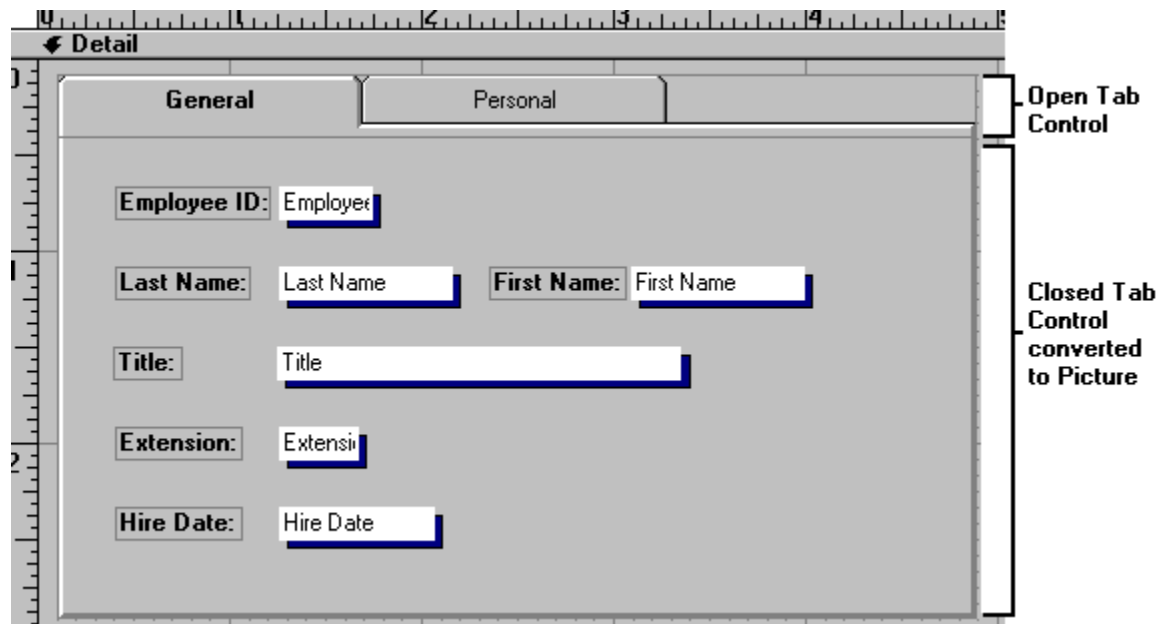
The table below shows what controls you should use together to get the desired effect.   Note that tab styles of Normal and Raised (Thin Bevel) are recommended over Raised (Thick Bevel) because they can both be combined with an Access rectangle.   An Access rectangle is lighter-weight than a Picture, so memory will be saved and speed will be increased.

Special Effect   Combine With

Normal	Access Rectangle with Special Effect = Normal
Raised (Thick Bevel)	Rasied (Thick Bevel) Tab Control converted to a Picture
Raised (Thin Bevel)	Access Rectangle with Special Effect = Raised

NOTE: Positioning an open Tab Control over a rectangle or bitmap precisely can be a chore.   Try using the Ctrl+Arrow and Shift+Arrow key combinations to move and size the controls.

Properties

BackColor - The color to use when painting the background of the control.

BodyStyle - Whether or not the tab is open.
> 0 - Closed
> 1 - Open

Font - A font object that has these attributes:
> Font.Name
> Font.Size
> Font.Weight
> Font.Italic
> Font.Strikeout
> Font.Underline

SpecialEffect - The special effect for the control.
> 0 - Normal
> 1 - Raised (Thick Bevel)
> 2 - Raised (Thin Bevel)

TabCaption() - Read-only property array that returns the caption for a particular tab.   Zero based.

TabLocation - Where the tabs live.
> 0 - Top
> 1 - Left
> 2 - Right
> 3 - Bottom

TextOrientation - Which way the text "faces
"
> 0 - Up
> 1 - Left
> 2 - Right
> 3 - Down

TabSizeMethod - How to size the tabs
> 0 - Fit Evenly with Gap
> 1 - Fit Evenly

TabText - A semi-colon delimited list of captions for the tabs.

Value - The value of the tab control (what tab is selected, zero-based).
> NOTE: This is the default property.

Events

<u>BeforeUpdate</u> - Fires before a new tab is selected.   Can be cancelled by setting Cancel=True.
<u>AfterUpdate</u> - Fires after a new tab has been selected.

Methods

The tab control currently has no methods.

Samples

If you're looking for some ideas on how to use the control, check out TAB.MDB, which should have been distributed with this document and the tab control.   This database contains several forms based on NWIND data that use the Tab control.

Distributing the Control

The README.TXT file that accompanies this control describes two ways that you can distribute the control.   The README file is a bit formal, so I thought a little legal-ease to English translation was in order.   Basically, you have two options:

1) Distribute everything in TAB.ZIP, but don't distribute TAB.ZIP nor any component inside of it for money. This means you can give TAB.ZIP to your friend, your mom, your dog, upload it to a bulletin board system, copy it to ftp.foo.bar.bletch on the Infobahn, etc.

If you would like to distribute TAB.ZIP as part of a commercial product (e.g. an Access book or a samples disk), you need contact me first, as the license agreement does not permit you to distribute TAB.ZIP in this manner.

2) Distrbute TAB.OCX and OC25.DLL as part of a Microsoft Access application.   You can charge money for an application that includes TAB.OCX.   You cannot distribute TAB.LIC, which comes in TAB.ZIP, in your Microsoft Access application.

If any of this is unclear, please feel free to email me.

Creating a Custom Setup Containing TAB.OCX

Your license permits you to distribute TAB.OCX with your custom Access applications as long as TAB.LIC is not distributed.   If you are an Access Developer's Toolkit customer, you are most likely creating a custom Setup with the Setup Wizard to ship your application.   After a few modifications to the Setup Wizard files, including the Tab OLE Control in this Setup can be quick and easy.

Step 1 - Make a Change to the SetupWizard

There is a minor problem with the Setup Wizard that needs to be fixed before the TAB.OCX can be easily included in your application.   This problem only occurs on some systems, but it doesn't hurt to fix it on all systems.   Here
's the fix:

1) Open the SetupWizard with the shift key down.   This will disable the Autoexec macro.
2) Open the **zws_Setup** module.
3) Go to the **zws_FindFile** routine.
4) Replace all occurrences (there should be only two) of :

```
        "  \
"
```

with

```
        " \
"
```

(be sure to include the quotes!).   In case you can't see the difference between these two strings: there is an extra space before the slash in the string you are replacing.
5) Save the module and close the database.

Step 2 - Update SWDEPEND.INI

The first screen of the Setup Wizard allows you to select what files you will ship with your application. You could simply select TAB.OCX and OC25.DLL along with your other files, but there is an easier way.

When you add a file to the Microsoft Access Setup Wizard, it looks in a file called SWDEPEND.INI to see if there are any dependencies (i.e. are there any files required to use the file you just added?).   Since TAB.OCX requires OC25.DLL, you will want to add this entry to SWDEPEND.INI:

```
[TAB.OCX]
Uses1=OC25.DLL
Dest=$(AppPath)
```

With this completed, whenever you add TAB.OCX to your custom Setup using the Setup Wizard, OC25.DLL will be automatically included.

Step 3 - Add TAB.OCX to your Custom Setup

With steps 1 and 2 complete, all that's left is for you to add TAB.OCX to your custom Setup.   Once you add the file, the Setup Wizard should detect that it depends on OC25.DLL and tell you that it has automatically selected the dependent file.   The Setup Wizard will notice that both of these files are self-

registering, and so it will automatically include MSAREG.EXE from the \ADT directory.   It will also modify the Setup script so that OC25.DLL is registered as is TAB.OCX, making installation for your users quick and painless.

Notes

This control is not yet complete.   I've still got some performance work to do, as well as some bug fixes. If you come across problems, I
'd like to know about them.

One big feature that will be ready ˝real soon now
˝ is different tab styles.   In addition to the tab style currently shown, the control will allow the diagonal tab style (a la Excel workbooks), as well as a rectangular style (not used anywhere, but still kind of cool).

New in version 1.0.002:

    +   Text that is oriented Up (i.e. the ˝normal
˝ way) should paint faster than it did before.   For any other type of text, there is a slight performance penalty, because we need to rotate the text, and there
's no fast, easy way to do this.

+   I fixed a bug where the control would not paint after you double-click it in Form Design (which opens the ˝mini-activation
˝ window).

New in version 1.0.003:

    +   Fixed bug in version 1.0.002.   I was not freeing some resources, leading to ˝Not enough
         resources to update display.
˝

New in Version 1.0.004:

    +   Added **Raised (Thin Bevel)** Special Effect for that ˝Windows95
˝ look.

+   Minor performance improvements

+   Control has been built with recently released version of the CDK (based on Visual C++ 1.51).