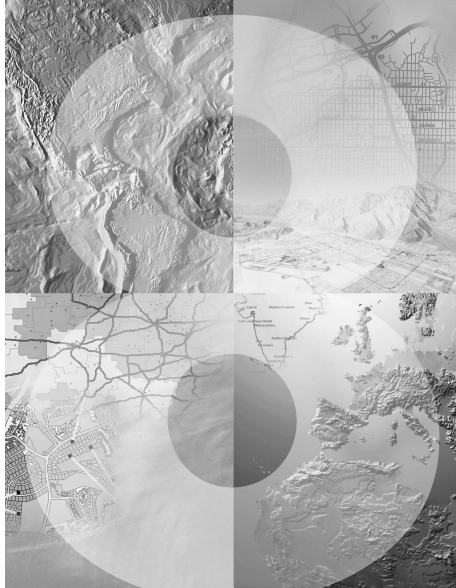


An ESRI White Paper



Customizing ArcInfo 8

June 1999



Copyright © 1999 Environmental Systems Research Institute, Inc.
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of Environmental Systems Research Institute, Inc. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by Environmental Systems Research Institute, Inc. All requests should be sent to Attention: Contracts Manager, Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is Environmental Systems Research Institute, Inc., 380 New York Street, Redlands, CA 92373-8100, USA.

ARC/INFO, ArcCAD, ArcView, *BusinessMAP*, ESRI, MapObjects, and PC ARC/INFO are trademarks of Environmental Systems Research Institute, Inc., registered in the United States and certain other countries; registration is pending in the European Community. 3D Analyst, ADF, ARC COGO, the ARC COGO logo, ARC GRID, the ARC GRID logo, AML, ARC NETWORK, the ARC NETWORK logo, *ARC News*, ARC TIN, the ARC TIN logo, the ARC/INFO logo, ArcInfo LIBRARIAN, ARC/INFO—Professional GIS, ARC/INFO—The World's GIS, ArcAtlas, the ArcAtlas logo, ArcBrowser, the ArcCAD logo, the ArcCAD WorkBench logo, ArcCatalog, ArcCensus, ArcCity, the ArcData logo, the ArcData Online logo, ArcDoc, ARCEDIT, the ARCEDIT logo, ArcExplorer, the ArcExplorer logo, ArcExpress, the ArcExpress logo, ArcFM, the ArcFM logo, the ArcFM Viewer logo, ArcIMS, the ArcIMS logo, ArcInfo, ArcLogistics, the ArcLogistics Route logo, ArcMap, ArcObjects, ARCPLOT, the ARCPLOT logo, ArcPress, the ArcPress logo, the ArcPress for ArcView logo, ArcScan, the ArcScan logo, ArcScene, the ArcScene logo, ArcSchool, ArcSDE, the ArcSDE logo, ArcSdl, ARCSHELL, ArcStorm, the ArcStorm logo, ArcToolbox, ArcTools, the ArcTools logo, ArcUSA, the ArcUSA logo, *ArcUser*, the ArcView GIS logo, the ArcView 3D Analyst logo, the ArcView Business Analyst logo, the ArcView Data Publisher logo, the ArcView Image Analysis logo, the ArcView Internet Map Server logo, the ArcView Network Analyst logo, the ArcView Spatial Analyst logo, the ArcView StreetMap logo, the ArcView StreetMap 2000 logo, the ArcView Tracking Analyst logo, ArcVoyager, ArcWorld, the ArcWorld logo, Atlas GIS, the Atlas GIS logo, AtlasWare, Avenue, the Avenue logo, the *BusinessMAP* logo, DAK, the DAK logo, DATABASE INTEGRATOR, DBI Kit, the Digital Chart of the World logo, the ESRI globe logo, the ESRI corporate logo, ESRI—Team GIS, ESRI—The GIS People, FormEdit, Geographic Design System, Geography Matters, GIS by ESRI, GIS for Everyone, GISData Server, IMAGE INTEGRATOR, *InsiteMAP*, MapBeans, MapCafé, the MapCafé logo, the MapObjects logo, the MapObjects Internet Map Server logo, NetEngine, the NetEngine logo, the PC ARC/INFO logo, PC ARCEDIT, PC ARCPLOT, PC ARCSHELL, PC DATA CONVERSION, PC NETWORK, PC OVERLAY, PC STARTER KIT, PC TABLES, the Production Line Tool Set logo, Spatial Database Engine, SDE, the SDE logo, the SDE CAD Client logo, SML, StreetMap, TABLES, The World's Leading Desktop GIS, ViewMaker, *Water Writes*, and Your Personal Geographic Information System are trademarks and ArcData, ARCMail, ArcOpen, ArcQuest, *ArcWatch*, ArcWeb, Rent-a-Tech, @esri.com, and www.esri.com are service marks of Environmental Systems Research Institute, Inc.

The names of other companies and products herein are trademarks or registered trademarks of their respective trademark owners.

Customizing ArcInfo 8

An ESRI White Paper

Contents	Page
Introduction	1
What Does It Mean to Customize Your Application?	1
How Can I Customize ArcInfo 8?	2
ArcMap and ArcCatalog	2
Running Macros and the VBA Editor	4
ArcToolbox	5
Building Your Own Controls	7
Customizing with VBA	8
Customizing Outside VBA	9
Conclusion	10
Appendix A	11
Appendix B	13

Customizing ArcInfo 8

Introduction

This paper introduces the customization environment present within the new applications in ArcInfo™ 8—ArcToolbox™, ArcCatalog™, and ArcMap™—and the vast array of possibilities it holds. As a system, ArcInfo 8 contains the same set of core modules it always has: ARC, ARCEDIT™, ARCPLOT™, ARC GRID™, and ARC TIN™, among others. The customization of the core modules is not discussed in this paper. For information on that topic, please refer to the online documentation.

What Does It Mean to Customize Your Application?

Customization, by definition, means the tailoring or alteration of some *object*, according to a set of predetermined requirements or specifications. Application customization, therefore, means the modification of the user interface or functionality to fit the specific needs of a user or market niche. Users of the new ArcInfo 8 applications have the opportunity to customize in an environment that is highly flexible and extensible. The new applications, ArcCatalog and ArcMap, are based on Microsoft's Component Object Model (COM), where the basic building block for application programming and problem solving is the object. In the case of ArcInfo 8, as in any object-oriented application, there is a set of requirements that any object needs to meet if it is to work properly in the schema of the host application or with other objects present in an object model. In the paradigm that is ArcInfo 8, once these conditions are met, a custom object, tool, toolbar, or script can be plugged into the application framework of ArcCatalog or ArcMap. The ArcToolbox application is written using the Open Development Environment (ODE) technology that was introduced to ARC/INFO® software users at Version 7.1.1. It is written in Visual Basic 6 (VB6) and comes with reusable components to provide application developers a suite of powerful geoprocessing tools.

What does this mean for ArcInfo users? It means customizing the user interface or extending the functionality of these new applications is easier to do and more robust than ever before. There are three major classes of customization available in ArcInfo 8: simple interface manipulation, the use of Visual Basic for Applications (VBA) to create custom functions or macros, and the writing of COM objects with an appropriate development environment to create custom feature objects or distributable tools or even a stand-alone application (based on the objects that comprise ArcInfo 8).

Each version of ArcCatalog and ArcMap is shipped with a licensed version of Microsoft VBA. Microsoft designed VBA to be embedded in applications to provide a seamless, integrated programming environment throughout the application. VBA is built into the MS Office line of products whereas Visual Basic (VB), one of Microsoft's stand-alone programming environments, is not. Both VB and VBA provide many useful tools, and each offers a complete Interactive Development Environment (IDE) helpful for easy application development. VB and VBA hide much of COM's intricate nature, allowing users to focus on their work rather than spend valuable time learning a new programming concept. VB has proven to be the most popular choice among developers due to its

relatively short learning curve and its expansive capabilities. VB and VBA are *not* the only available choices for developers. Since ArcCatalog and ArcMap are built on COM, any programming language that supports COM, such as Visual C++ (VC++), Java, or Delphi, can be used for application development or customization.

How Can I Customize ArcInfo 8?

There are several ways in which ArcCatalog and ArcMap can be customized, from simply changing the look of the user interface by adding or removing user controls (menus, tools, or toolbars) to adding custom Dynamic Link Libraries (DLLs) as new user controls. A custom tool can consist of a single function, or it can contain form menus that perform user-defined tasks.

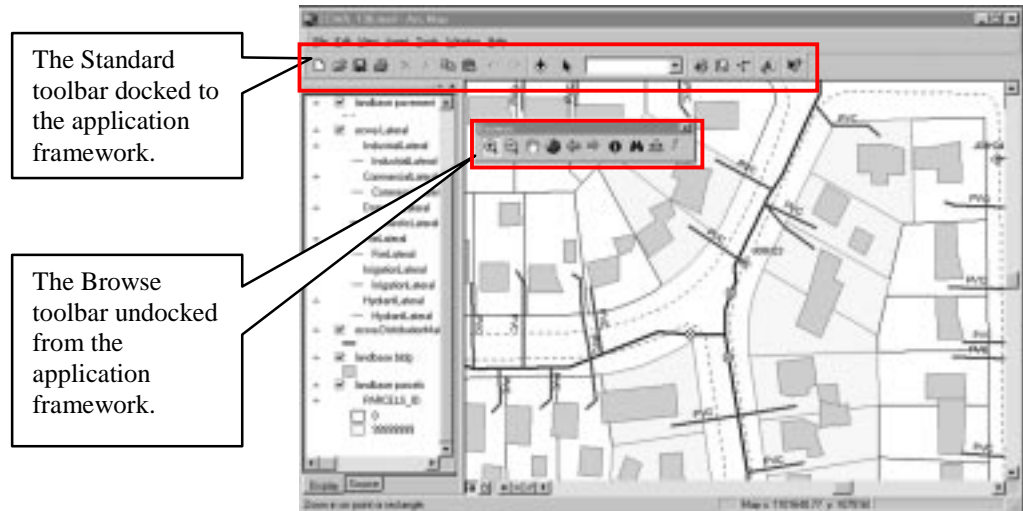
ArcMap and ArcCatalog

The simplest method of customizing the user interface is by adding or removing toolbars on the user interface. This procedure is done the same way in both ArcMap and ArcCatalog. The following diagram is the Standard toolbar found on the main user interface of ArcMap.



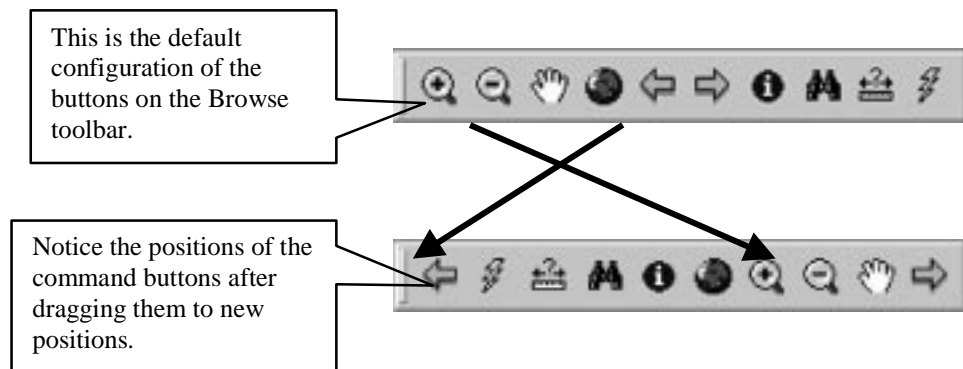
Just as with any toolbar in ArcMap or ArcCatalog, it has certain properties that allow users to manipulate them in order to produce the user interface they want. Any toolbar shown, by default, can be checked on or off, consequently displaying it on or hiding it from the user interface. Once a toolbar is added to the user interface it need not remain in the position it was added. It can be dragged to and from the application framework; this is known as docking or undocking. Undocking a toolbar enables it to float about the desktop but function as a part of the application, whereas docking it "glues" the toolbar someplace on the application framework. This allows the user to set the appearance of the application as desired. The following diagram shows two toolbars, one docked and one undocked.

J-8219



Not only can toolbars be repositioned, added, or removed from the user interface of either ArcMap or ArcCatalog, but command buttons can be manipulated in much the same way. They can be moved anywhere along their parent toolbar or dragged to different toolbars altogether. For example, a user can group those tools used most often close together on a single toolbar, increasing work efficiency for users by minimizing their search for the tools necessary to complete a task. New toolbars can be created and command buttons or pulldown menus can be added or dragged to these toolbars.

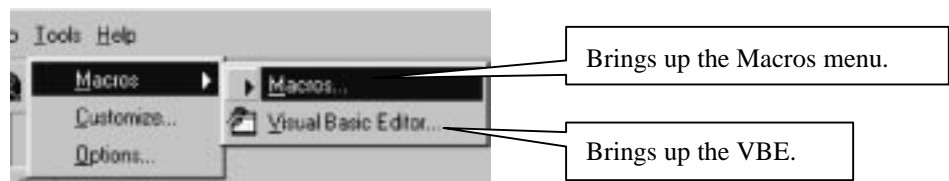
The ability also exists to remove a command tool from the interface by dragging it off the application framework and dropping it onto the desktop. See the figure below to see an example of how command buttons can be reordered along the Browse toolbar.



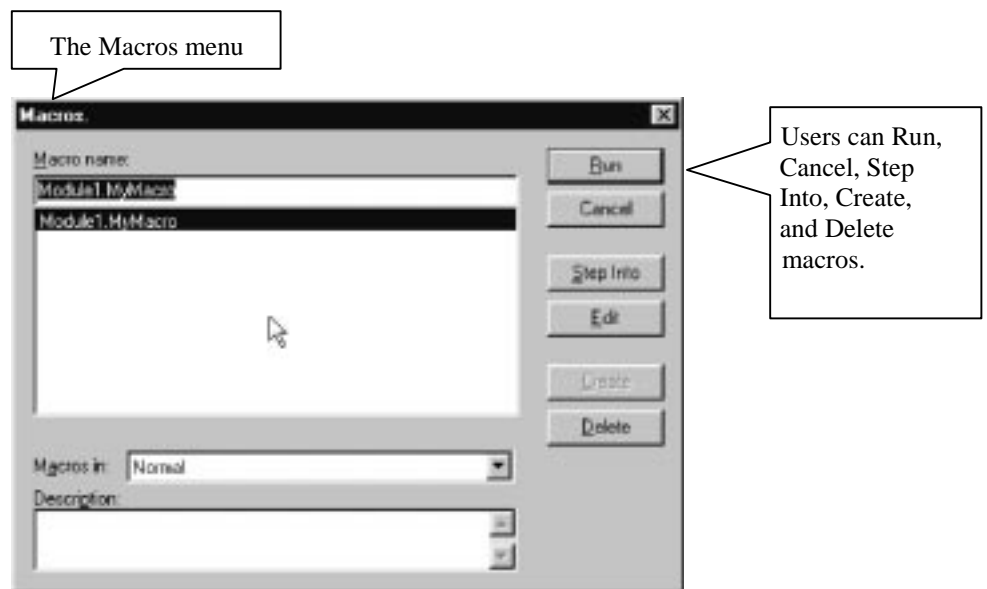
To perform any of the customizations described thus far, the Customize dialog window must be open. The Customize dialog is the first step in adding new or available command tools to the user interface. This dialog is brought up by right clicking on any open area at the top of the application framework or by choosing the Customize option under the Tools pulldown menu. See **Appendix A** for more details on the Customize dialog. Also present in the Customize dialog is a button enabling users to map their keyboards to perform specified tasks. This allows the user to set "hot keys" to perform the action typically done by clicking a button or running a script.

Running Macros and the VBA Editor

Yet another way for users to customize the functionality of ArcMap or ArcCatalog is to execute macros directly without adding them to a command tool (assigning code to a command button or tool and physically attaching them to the interface). Under the Tools menu, the Macros submenu contains two additional options, Macros and Visual Basic Editor (VBE).



Selecting the Macros option brings up the Macros menu, where users can create VBA macros.



Macros written in this mode can then be executed directly from this window by clicking on the Run button. Executing macros in this way enables users to write and use custom tools without having to physically add them to the user interface. These macros can optionally be exported from the VBE and saved as .BAS modules and stored on disk. These macros are also saved in the specified template file so any users of that template have access to any of these exported modules.

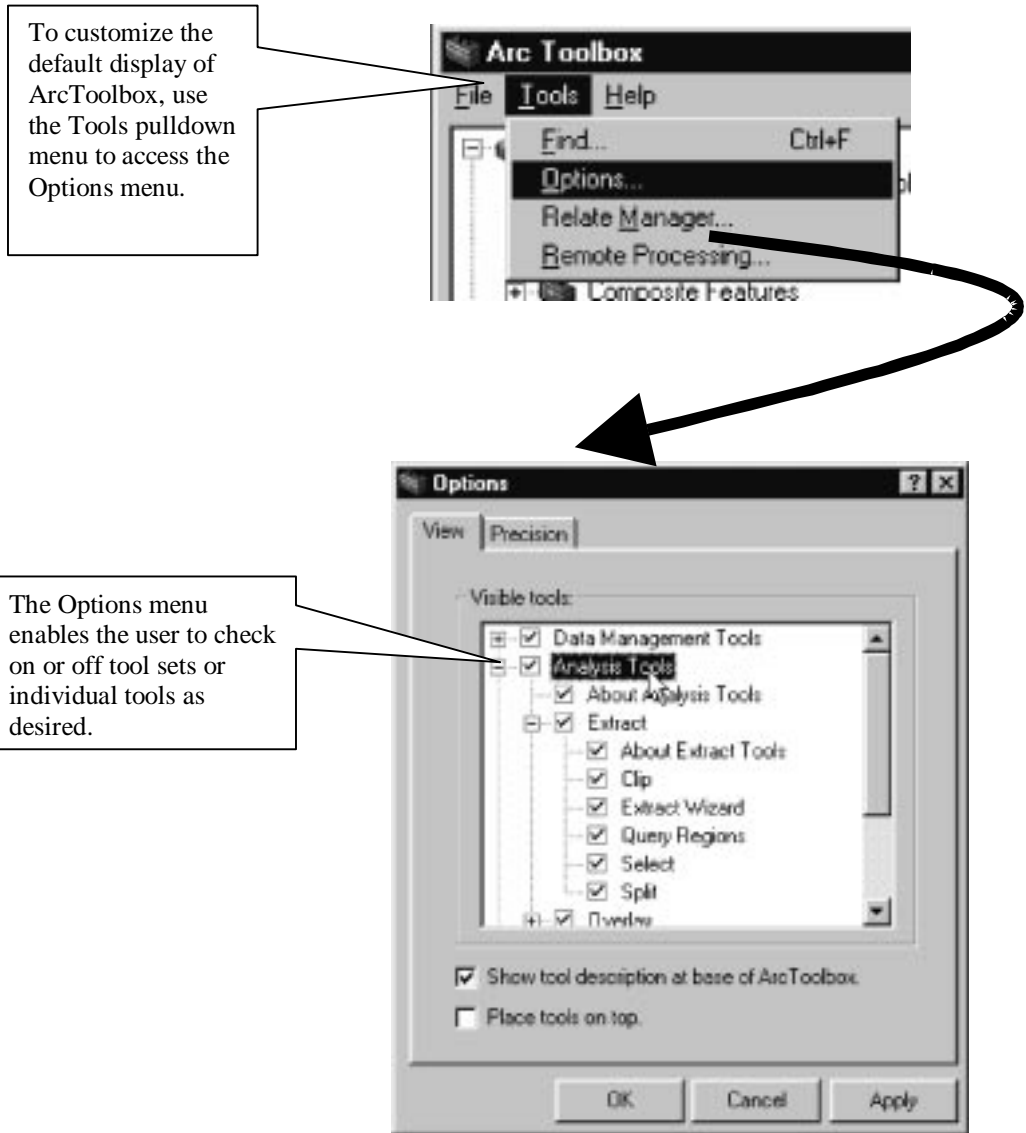
Bringing up the VBE enables users to create macros, menus, toolbars, or other user interface controls. This is due to the fact that VBA 6 can compile a DLL to be run from the user interface of either ArcMap or ArcCatalog. User controls consist of button controls, combo box controls, edit controls, or tool controls. Button controls execute a function that has no interaction with the map canvas. Tool controls execute some function that does interact with the map canvas. Edit controls hold user-defined edit tasks (i.e., create new fire hydrant), and combo box controls contain a list of user-defined tasks. For a picture of the Visual Basic Editor, see **Appendix B**. The VBE choice is yet another method for users to enter the development environment of ArcCatalog or ArcMap.

If you have used VB or any other language that supports COM to create a DLL that contains custom tools or toolbars, these items can easily be added to the ArcMap or ArcCatalog user interface. Use the "Add from file..." button on the Customize dialog to register these custom tools or toolbars with the application. A file dialog will pop up to allow you to browse for your DLL. Then the Added Objects dialog appears, reporting which new objects have been added to the application. Once your custom tools and toolbars have been added to the application, you can add them to the user interface in the same way you add built-in tools and toolbars to the user interface.

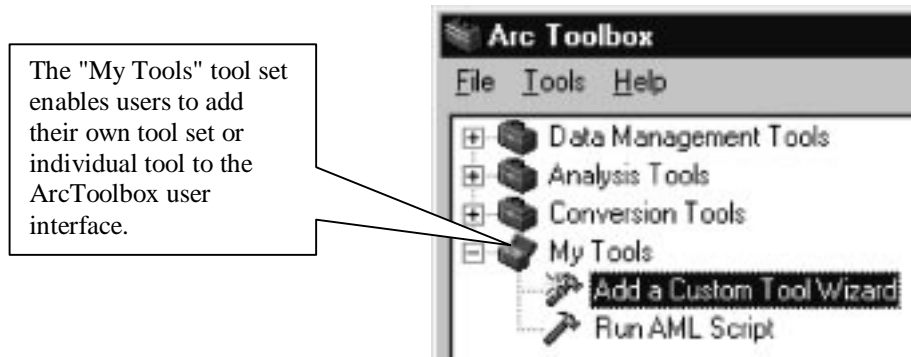
Any and all of these changes can be made persistent, or saved, to the application by saving it to the *Normal* or user-specified template. ArcMap and ArcCatalog operate with a concept of templates similar to MS Word. The default template in ArcMap is known as the normal template file. When ArcMap is started, the application creates a default document file. Users can add data and save a map to a document file. Customizations are also stored within the document file loaded when the application was started. These files can be reopened in subsequent ArcMap sessions as customizations made to those templates are restored and reactivated.

ArcToolbox

ArcToolbox also can be customized in several different ways. One way is to simply show only those tool sets the user chooses. Selecting the Tools pulldown menu and choosing the "Options..." choice will open the Options menu. This menu allows the user to check entire tool sets or individual tools on or off, thereby hiding (or showing) the tools the user wants or needs.



Another way to customize the ArcToolbox is through the "My Tools" tool set. This tool set has options enabling the user to create their own tool set category in addition to their own custom tool. The first option in this tool set is named "Create Your Own Tool." This tool pops up a wizard that walks the user through the process of pointing at any registered DLL or EXE on the host machine and implements it as a tool on the ArcToolbox user interface. Below is a picture of where this tool set exists in the ArcToolbox interface.

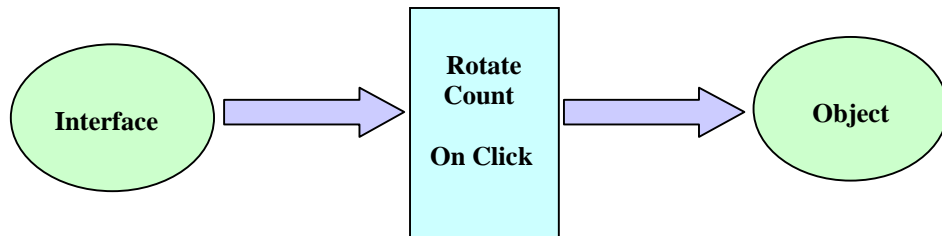


Also present in this tool set is the "Run AML Script" tool. As the name states, it starts a tool that allows the user to select any ARC Macro Language (AML™) or several AML files to run in batch mode on the system and run them. They are not added as a tool; rather, they are executed the moment the user clicks the OK button or when they are scheduled to run.

One last way to customize ArcToolbox is to write a stand-alone tool (for example, in Visual Basic), compile it as a DLL or EXE, and add it to ArcToolbox as a custom tool. In order for the toolbox to successfully execute your tool, it needs to know what type of tool to expect. In VB, you must add "ESRI ArcToolbox Tool Executer" as a reference in your new ActiveX DLL project to get access to the IToolboxTool interface. Your tool class must also implement the IToolboxTool interface. You need not fill out code for each of the IToolboxTool methods, but you must at least stub in the code in order to be able to successfully compile your project. The IToolboxTool methods allow you to set the tool's modality and optionally pass on a reference to a preexisting Arc Automation Server. In most cases, your custom tool will consist of a form containing buttons, text boxes, combo boxes, and so forth, to allow user interaction with the tool. In the "ToolboxTool_ExecArcTool" method, you can simply use the ".show" property on the form you have created in the class project. Once the code is written and compiled correctly, you have a DLL. The "Add Custom Tool" option in the "My Tools" tool set will walk you through browsing for this DLL and registering it with the application. Once this is done, the custom tool will execute from the ArcToolbox and perform those functions you have written.

Building Your Own Controls

The Component Object Model, best known simply as COM, consists of objects and interfaces to those objects. In COM, the interfaces provide the mechanism for object manipulation. *Interfaces* are made up of one or more related methods, properties, and/or events. *Methods* represent some action the object is to take such as Rotate or Draw. The *Properties* describe an object; that is, they represent the attributes or characteristics of an object. Examples of properties of an object could be Count or Name. *Events* are methods that notify the rest of the system when something has happened relating to the object they are associated with. Interfaces then determine how clients of an object interact with and manipulate the object, as well as define what can and cannot be done with the object.



Anyone interested in customizing ArcCatalog or ArcMap has a choice in the development environment to use. Should the built-in programming language VBA be used, or should some other language that supports COM such as VB6 be selected? There are definite advantages and disadvantages to either selection.

Customizing with VBA

Opting to use VBA as the means for customizing ArcMap or ArcCatalog is a good choice due to the fact that it is fully integrated within the applications. With VBA, there is no overwhelming need to compile a DLL, register it onto the host computer, and load it into the application in order to execute it as a custom tool. This is possible with VBA. VBA 6 has the ability to compile DLLs. VBA provides a run-time development environment for the user to create custom processes, enhancing any version of the application. For example, using VBA, ArcMap (or ArcCatalog) can be currently running when the need to create a new macro arises. There is no need to shut down the application, open the development environment, create the macro, and finally restart ArcMap in order to test the macro. With VBA you have the ability to debug your macro in the application it is running while you are writing it. Variables and results can be checked and performance can be monitored without having to exit the current ArcMap session. VBA is also the simplest choice to make, as it is relatively easy to learn in comparison to other programming environments. Nevertheless, VBA does have its drawbacks. Since the scripts you write are local to the map document file, source code can be exposed to a user. If you need your code compiled, encrypted, or inaccessible, you should be writing DLLs in a programming environment that will facilitate your requirements, not VBA. The following code is a simple VBA example showing how one could code the object diagram displayed at the end of the next section of this paper (**Customizing Outside VBA**).

```

Dim pParcelUpdate as IUpdate ' declaration
Set pParcelUpdate = New Parcel' assignment/creation
pParcelUpdate.Split' call Split on IUpdate

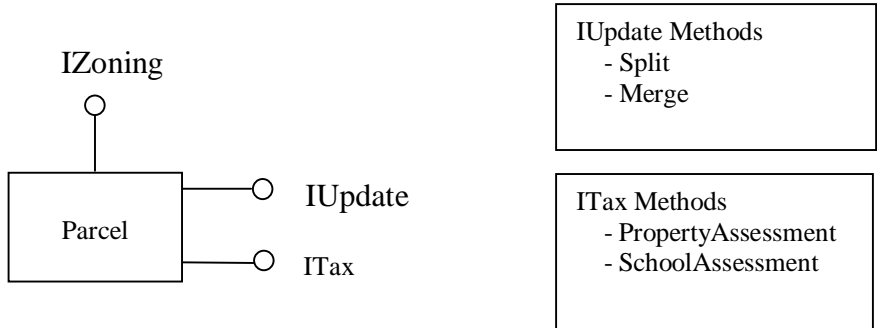
Dim pParcelPay As ITax' declaration
Set pParcelPay = pParcelUpdate ' QI (Query Interface) for ITax
pParcelPay.PropertyAssesment' call PropertyAssesment on ITax
Set pParcelPay = Nothing' Release pParcelPay
  
```

For example, in ArcMap a user can program a simple message box that tells the current scale of the map on the screen by adding a subroutine and in it writing the following code. It does nothing more than load the current map document and return the scale of the active view.



Customizing Outside VBA

Why choose a programming language other than VBA? If you already know and are proficient with another language that supports COM, such as Visual C++, you can continue to use it. If you are a developer who is looking to create and distribute a command tool (toolbar or menu) and do not want to be restricted to the scope of a single document file, or just want to maintain proprietary code, then creating custom COM components is what you want to do. Creating custom COM commands requires a different approach. You must manage the interfaces and the implementation of the object programmatically. The diagram below displays a simple COM object. The "Parcel" is an object and IUpdate and ITax are interfaces to it. IUpdate and ITax have methods such as Split and PropertyAssessment. The class Parcel takes on any properties that IZoning, another interface, may already have of its own.

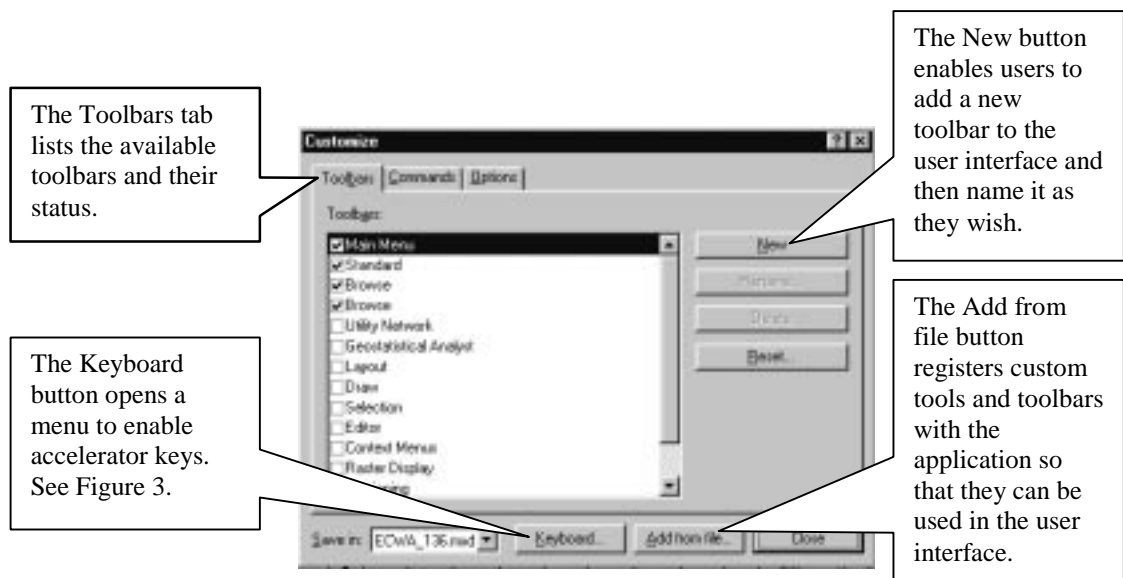


Conclusion The customization environment in ArcInfo 8 is more robust and flexible than ever before. The opportunities COM has brought forth has enabled ESRI to build the most extensible set of applications in the geographic information system community. ArcInfo 8 is a developer's dream, with every component that can be seen in the software being accessible and available for use in any developer's application or custom tool.

Appendix A

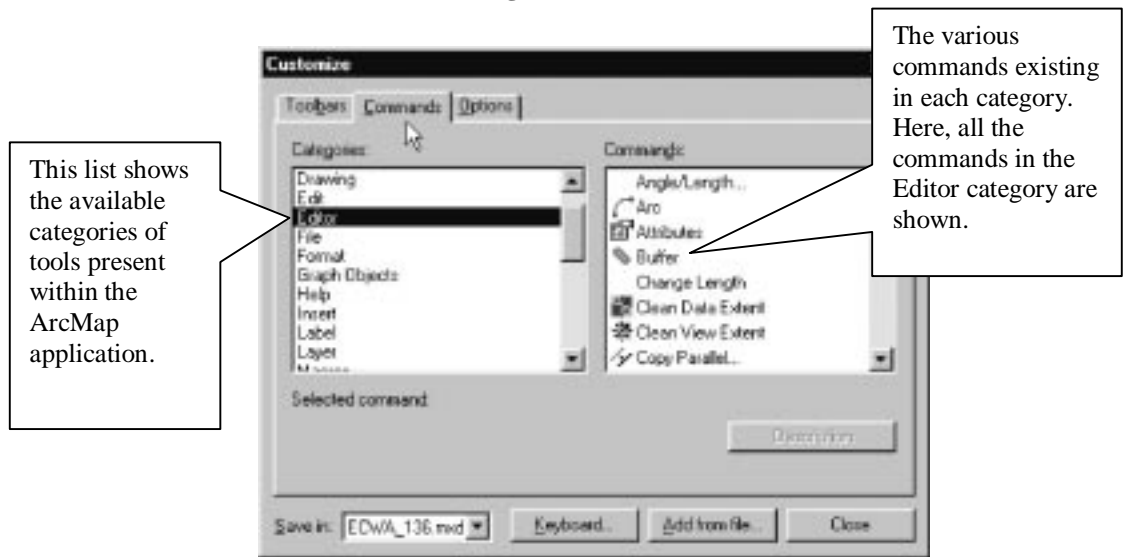
This section details the Customize menu user interface. There are three tabs to this menu. The first two deal directly with customizing via the addition of tools or toolbars to the user interface. The third tab, the Options tab, not shown in this document, gives the user the option to display the application using large icons and more. The first tab, the Toolbars tab, displays a list of the toolbars already active on the user interface, those that are checked, and the other toolbars that can be added to the application. On this dialog, users have the ability to add a New toolbar, one they can name and add to the user interface. The Reset button gives the user the chance to reconfigure the toolbar back to how it was in the Normal template format.

Figure 1



The Commands tab (shown in **Figure 2**) contains two list views. On the left, all the available categories of tools are displayed such as Editor, Drawing, Graph Object, and more. In the right-hand list view are the corresponding commands for each category listed on the left. Any of these commands can be added to the user interface by dragging and dropping them from the Customize menu onto a toolbar of choice. If a user wishes to remove a command button from the user interface, it must be done while this menu is open. This is achieved by clicking and dragging any tool from the interface and dropping it somewhere outside the application framework, for example, onto the desktop.

Figure 2



When the user clicks on the Keyboard button, it brings up the Customize Keyboard menu (see **Figure 3**), allowing users to assign commands to accelerator keys; for example, Ctrl-C is the same as the Copy command.

Figure 3



Appendix B

The Visual Basic Editor is started in several ways. The user can right click on a command button or toolbar while the Customization menu is on screen and select the View Source option. Two other ways to start the Visual Basic Editor come from the Tools pulldown menu. From here a pull-right menu entitled "Macros" exists. Selecting this pull-right menu enables another "Macros..." option and a "Visual Basic Editor" option to become available for selection. The "Macros..." option brings up a dialog that allows you to create a new macro or select one that may already exist in the template file you are using. The "Visual Basic Editor" option starts up VBA directly where you can use its tools to locate an existing project or to create a new one.





For more than 30 years ESRI has been helping people manage and analyze geographic information. ESRI offers a framework for implementing GIS in any organization with a seamless link from personal GIS on the desktop to enterprisewide GIS client/server and data management systems. ESRI GIS solutions are flexible and can be customized to meet the needs of our users. ESRI is a full-service GIS company, ready to help you begin, grow, and build success with GIS.

Corporate

ESRI
380 New York Street
Redlands, California
92373-8100, USA
Telephone: 909-793-2853
Fax: 909-793-5953

For more information call
ESRI or your local reseller at

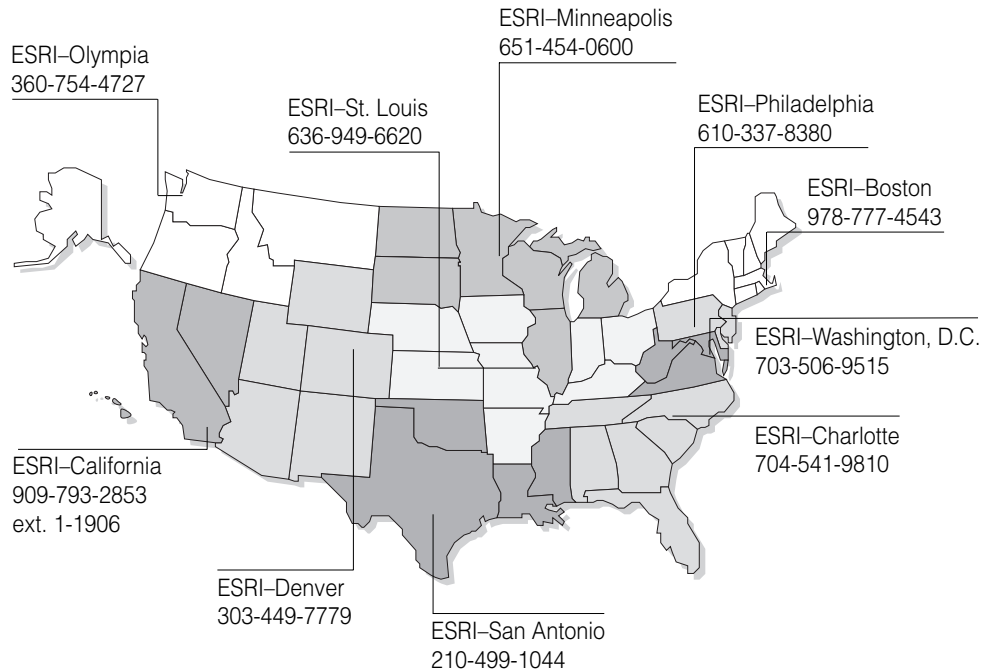
1-800-447-9778

(1-800-GIS-XPRT)

Send e-mail inquiries to
info@esri.com

Visit ESRI's Web page at
www.esri.com

Regional Offices



International Offices

Outside the United States,
contact your local ESRI distributor.

For the number of your distributor,
call ESRI at 909-793-2853,

ext. 1-1235,

or visit our Web site at

www.esri.com/international

Australia 61-89-242-1005	Hong Kong 852-2730-6883	Netherlands 31-10-217-0700	Sweden 46-23-84090
Belgium/Luxembourg 32-2-460-7480	Hungary 361-428-8040	Poland 48-22-825-9836	Thailand 66-2-678-0707
Canada 416-441-6035	India 91-11-620-3802	Romania 40-1-231-13-81	United Kingdom 44-1296-745-500
France 33-1-46-23-6060	Italy 3906-406-96-1	Singapore/Malaysia/Indonesia 65-742-8622	Venezuela 58-2-285-1134
Germany and Switzerland 49-8166-677-0 41-1-360-2460	Korea 82-2-571-3161	Spain 34-91-559-4375	

Place ESRI business partner or distributor address here.

