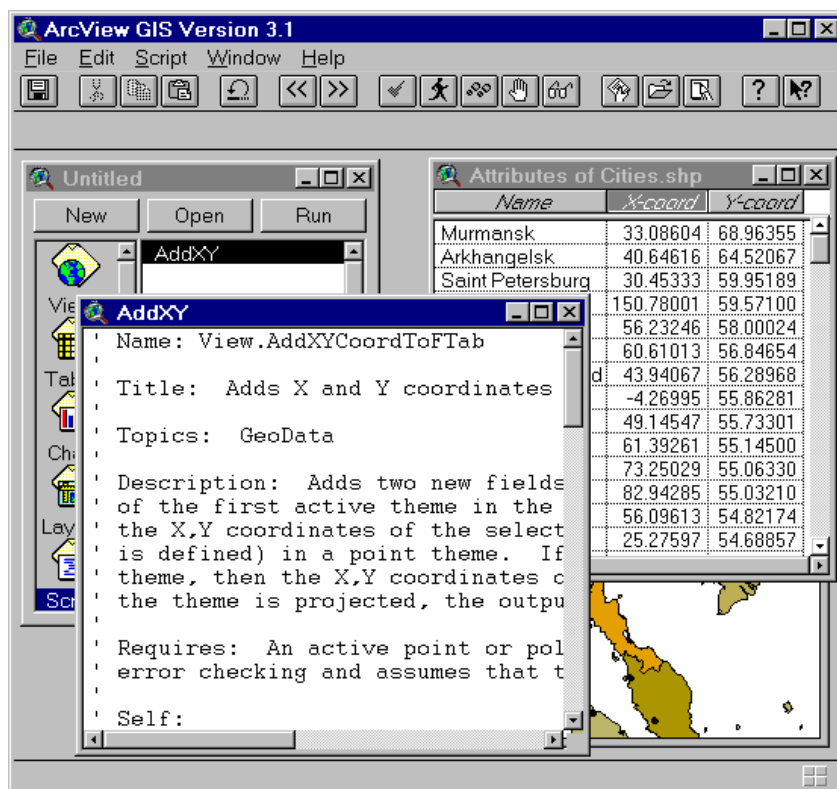


Programming in ArcView 3.x GIS using Avenue

Avenue is ArcView's object-oriented programming language used for customizing and developing ArcView applications. This Guide introduces customizing the ArcView interface, utilising existing Avenue scripts, and developing new scripts. It assumes prior working knowledge of ArcView 3.x.



Document code: **Guide 94**
Title: **Programming in ArcView 3.x GIS using Avenue**
Version: **1.4**
Date: **April 2007**
Produced by: **University of Durham Information Technology Service**

Copyright © 2007 University of Durham Information Technology Service

Conventions:

In this document, the following conventions are used:

- A typewriter font is used for what you see on the screen.
- A **bold typewriter font** is used to represent the actual characters you type at the keyboard.
- A *slanted typewriter font* is used for items such as filenames which you should replace with particular instances.
- A **bold font** is used to indicate named keys on the keyboard, for example, **Esc** and **Enter**, represent the keys marked Esc and Enter, respectively.
- A **bold font** is also used where a technical term or command name is used in the text.
- Where two keys are separated by a forward slash (as in **Ctrl/B**, for example), press and hold down the first key (**Ctrl**), tap the second (**B**), and then release the first key.

Contents

| | | |
|-----------|--|-----------|
| 1 | Introduction | 1 |
| 2 | Customizing ArcView | 1 |
| 2.1 | Saving customized settings | 3 |
| 3 | Using existing Avenue scripts..... | 3 |
| 3.1 | System scripts..... | 3 |
| 3.2 | Sample scripts | 3 |
| 4 | Avenue basics..... | 4 |
| 4.1 | Statements..... | 4 |
| 4.1.1 | Assignment statement and variables..... | 5 |
| 4.1.2 | For Each statement | 5 |
| 4.1.3 | If...Then...Else statement..... | 5 |
| 4.1.4 | While statement | 5 |
| 5 | ArcView classes..... | 5 |
| 6 | Writing new scripts..... | 6 |
| 6.1 | Using the Script Editor | 6 |
| 6.2 | Script conventions..... | 7 |
| 6.3 | Portability | 7 |
| 7 | Examples | 7 |
| 7.1 | User input..... | 7 |
| 7.2 | Working with views | 8 |
| 7.3 | Working with tables..... | 8 |
| 7.4 | Accessing files | 8 |
| 7.5 | Calling a script from another script | 8 |
| 8 | Creating menus..... | 9 |
| 9 | Interapplication communication..... | 9 |
| 10 | Avenue scripts — worked examples..... | 9 |
| 10.1 | Run a system script | 9 |
| 10.2 | Associate script with a new button..... | 10 |
| 10.3 | Add a script from a text file | 10 |
| 10.4 | Open a user-chosen text file | 11 |
| 10.5 | Create a user-defined loop | 11 |
| 10.6 | Select and zoom to features in a theme | 11 |
| 11 | Further reading — books and manuals | 12 |
| 12 | Other sources of information | 12 |

1 Introduction

Avenue is ArcView's object-oriented programming language used for customizing and developing ArcView applications.

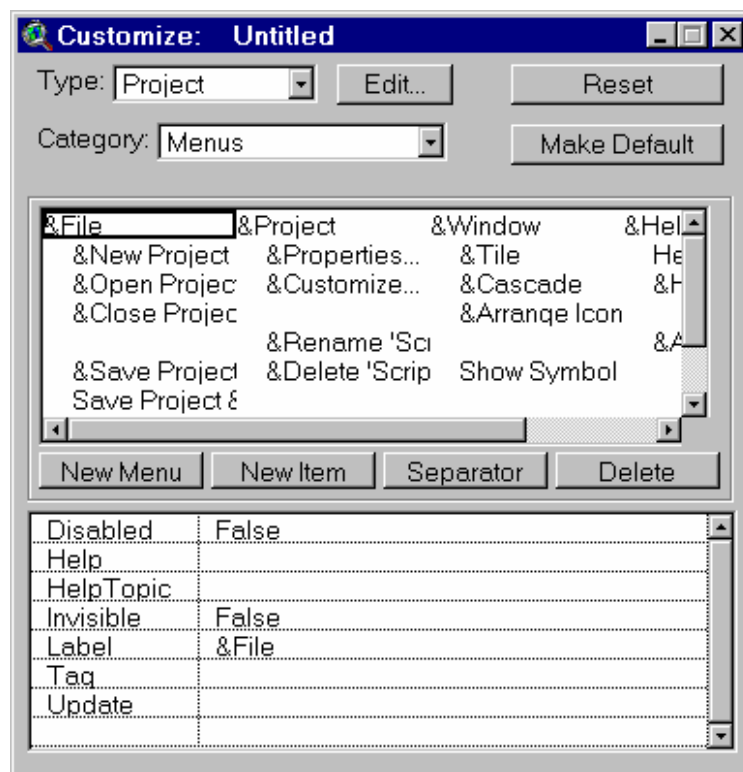
ArcView is composed of many Avenue scripts. These are all available to the user, making it very easy to customize. There are also a large number of sample scripts that can be used to access functionality not available through the normal ArcView interface.

This Guide covers customizing the ArcView interface, accessing and using the Avenue system and sample scripts, and a basic introduction to developing scripts, including worked examples.

2 Customizing ArcView

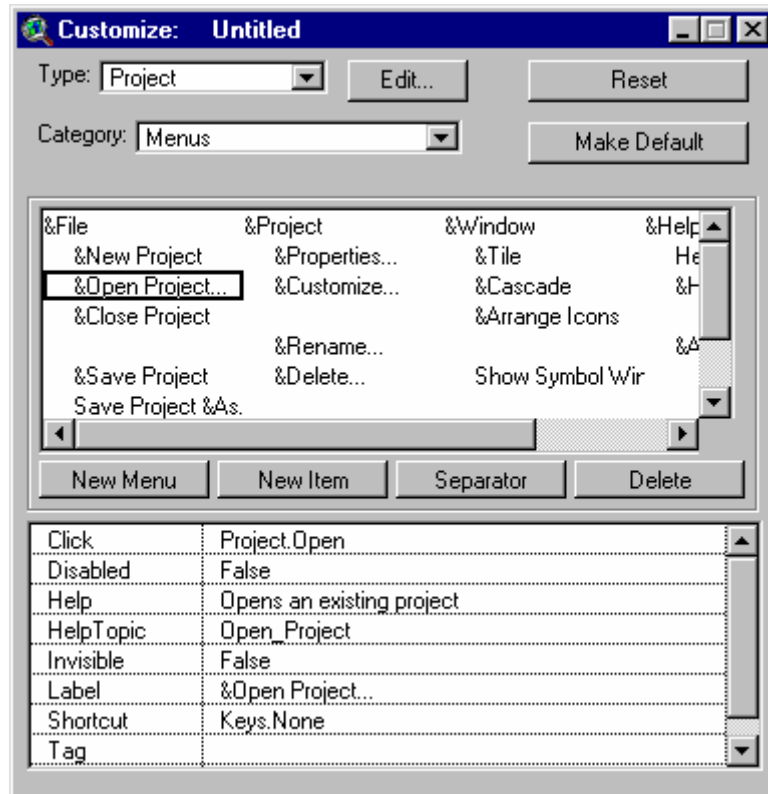
It is very simple to customize the ArcView interface to tailor it for a users needs. The customizing process also provides a good introduction to how ArcView works and how all it's functionality (provided by Avenue scripts) can easily be utilised to develop applications.

Each ArcView document (view, table, chart, layout) has it's own graphical user interface (GUI) and associated menu, button and tool bars. These can be customized by adding/removing menus, buttons or tools. With the Project window active in ArcView select **Project | Customize** from the menu bar to display the **Customize** dialog box:



The **Type:** box displays which document is selected — in this case Project. Other options include View, Table, Chart or Layout. The **Category:** box displays whether settings for the Menu, Button or Tool bar are shown.

An existing item can be used to illustrate how the menu system works. By clicking on the **&Open Project...** entry we can see what happens when a user chooses **File | Open Project** from the menu bar:



The **Control Properties** for **&Open Project...** are shown in the lower part of the dialog box. The description for each of these is:

- **Click:** name of script executed when menu item chosen by user
- **Disabled:** sets whether menu item can be chosen
- **Help:** help text shown in status bar
- **Help Topic:** name of help topic shown when asking for item help
- **Invisible:** determines whether item is visible on menu bar
- **Label:** text that appears on menu bar for item
- **Shortcut:** keystroke to access menu item
- **Tag:** can be used to store information on item. Not used by ArcView
- **Update:** script to run if change to state of active window occurs

The Customize dialog box will be used later in the worked examples to associate a script to a new item on the button bar.

2.1 Saving customized settings

To save a customized GUI click on **Make Default**. This will put a copy of **default.apr** in your home directory. Everytime you start up ArcView it will use this personal copy of **default.apr** containing your customized settings. **Default.apr** is in addition to projects containing your Views, Themes, Tables etc. These should still be saved as projects (e.g. **mywork.apr**). To return to the system **default.apr** delete or rename the personal copy of **default.apr** in your home directory.

3 Using existing Avenue scripts

There is a wealth of existing Avenue scripts available to ArcView users, including system and sample scripts.

3.1 System scripts

System scripts allow access to the Avenue code used in the ArcView interface. These scripts can be used to form the base for developing applications and new functionality. The **Script Editor** is used to write and run scripts:

- 1 To create a new Script Editor window double-click on the script icon in the Project window:



- 2 To open a system script into the script editor window choose **Script | Load System Script** from the menu that will display the **Script Manager**.
- 3 Select **Project.Open** and click **OK** and the script to open a project will be pasted into the script editor.
- 4 To compile the script press the **compile** button.
- 5 If no errors are returned, press the **run** button to execute the script.

compile 

run 

The option to save the existing project will be given and then the **Open Script** dialog box will be shown. **Project.Open** is the script that is run when choosing **File | Open Project** from the menu.

3.2 Sample scripts

ArcView's online help contains many sample scripts under the heading **Sample scripts and extensions | Sample scripts**. These scripts are classified into sections and include a brief description and purpose of the script. For example, under the heading **Views | Data conversion/alteration** the script Adds X and Y co-ordinates of features to Attribute Table is shown. By choosing this entry and clicking on the **source code** link at the top of the page entry we can access the full script. To use this code in the Script Editor select all the script, choose **Options | Copy**

and then **Edit | Paste** in the Script Editor. Finally, compile and run the script.

Note: A Theme must be highlighted in a View immediately before running the script.

These sample scripts can also be accessed directly from the ArcView installation directory at \$AVHOME/samples/scripts. To bring one of these into the Script Editor choose **Script | Load Text File** from the menu and select the appropriate file.

4 Avenue basics

This Guide does not introduce the theory of object-oriented programming, but details briefly how objects are used in ArcView and Avenue.

Arcview and Avenue are composed of objects - a view is an object, a theme is an object, a table is an object.

Avenue programming involves sending requests to objects:

theView.GetDisplay

The request (GetDisplay) to the object (theView) is separated from the object by a dot (.). GetDisplay returns the screen area of the view (theView).

An equal sign is used to assign an object to a variable:

theProject = av.GetProject

In this case the variable theProject is assigned the current project obtained by the GetProject request on the object av. Note: The object **av** references the whole ArcView application.

Some requests accept arguments, and are contained in brackets following the request:

theView = theProject.FindDoc("UK map")

The object returned by the FindDoc request on theProject is passed to theView. The FindDoc request takes an argument ("UK map").

Chapter 14 'ArcView Customization' in the book Inside ArcView GIS by S. Hutchinson & L. Davies gives a good overview of these concepts. ESRI's 'Using Avenue' provides more detail. See the Further reading section for information on these books.

4.1 Statements

Statements in Avenue organise the way in which you can send requests to objects. The following statements are included in Avenue:

4.1.1 Assignment statement and variables

The assignment statement makes a reference to an object:

variable = object

e.g.:

```
theView = GetActiveDocument
theSum = 5 + 12
```

Global variables are prefixed by an underscore (_)

e.g.

```
_totalSum
```

Strings must be quoted.

e.g.:

```
theMessage = "Hello"
```

4.1.2 For Each statement

For Each iterates through a loop:

```
for each f in 1..10
  MsgBox.Info( f.AsString, "" )
end
```

4.1.3 If...Then...Else statement

If...Then...Else is used for decision making:

```
If (theNumber = 10) then
  MsgBox.Info("Number is 10", "")
else
  MsgBox.Info("Number is not 10", "")
end
```

4.1.4 While statement

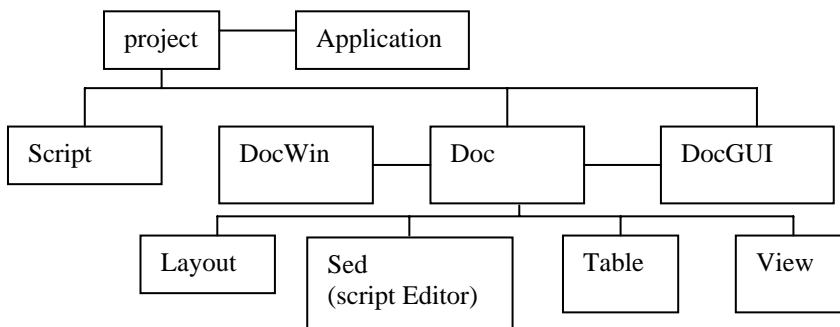
While repeats as long as the expression is true:

```
while (theCount < 20)
  theCount = theCount + 1
  MsgBox.Info(theCount.AsString, "")
end
```

5 ArcView classes

Avenue classes define the common properties for a group of objects. For example, the Doc class defines properties for the Layout, SEd, Table and View objects. As well as these common properties, objects have their own additional properties.

Application Framework:



Comprehensive Help on the Class hierarchy and a functional overview is contained in the ArcView online help system accessible from the menu bar using **Help | Help Topics**.



Context sensitive Help on Objects and requests can be accessed by highlighting the word in the Script Editor and pressing the Script Help button.

A full list of ArcView requests can be seen in the online Help under **Sample scripts and extensions | Other samples**.

6 Writing new scripts

6.1 Using the Script Editor

The Script Editor is used to write new scripts:

compile

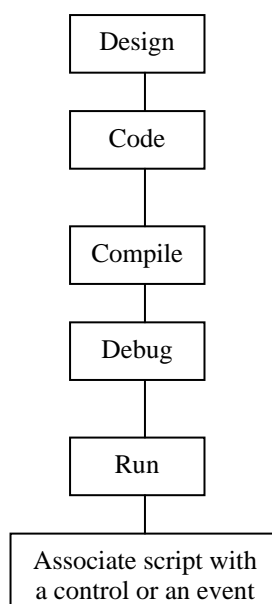


Once complete, the compile button is used to check for bugs. Once these have been eliminated the run button executes the script. Scripts are saved as part of an ArcView project. They can be given names through the **Script | Properties** menu option. Scripts can also be written out to a text file with the menu option **Script | Write Text File** using the extension **.ave**.

run



Stages in developing a script:



6.2 Script conventions

Script names normally start with the document to which they belong e.g. **View.Add**.



Avenue is not case sensitive but variable names normally begin with a lowercase letter e.g. **theNumber**. Objects and requests begin with an uppercase letter e.g. **MsgBox**, **AsString**.

Comments can be inserted into an Avenue script by prefixing them with a single quote ('):

```
' simple.ave  
' simple avenue script  
' created 10 March
```

Blank lines in Avenue are ignored and can be used to break up and improve readability of the script.

The line continuation character ~ is used to continue long lines onto more than one line.

Indenting statements can be made easier using the format buttons   icon:

```
for each t in 1..10  
  MsgBox.Info( t.AsString, "Count=")  
end
```

6.3 Portability

Avenue scripts are portable across platforms, including Unix, Windows and Macintosh. Some differences do however exist. For example the methods in which directory and file names are accessed:

UNIX:

```
theFile = "$home/workarea/coords.txt".AsFileName
```

Windows:

```
theFile = "c:\workarea\coords.txt".AsFileName
```

7 Examples

7.1 User input

The Class MsgBox provides a mechanism to interact with the user:

```
yourName = MsgBox.Input ("Enter your name", "Name", "")  
MsgBox.Info(yourName,"The name entered is")
```

7.2 Working with views

The following creates a new view in a project:

```
thisProject = av.GetProject
newView = View.Make
if (nil = newView) then
  MsgBox.Error ("Unable to create", "")
  exit
end
```

7.3 Working with tables

To query the data in a table:

(the following uses the sample data:

UNIX: \$AVDATA/esridata/world/cities.shp

NPCS: z:\licence\esri\av_gis30\avtutor\dbaccess\cities.shp)

```
theTable = av.GetActiveDoc
theVtab = theTable.GetVtab
theBitMap = theVTab.GetSelection
'set up the query string (vancouver)
select = "vancouver"
'set up the field to query (name)
str = "[cityname] = " +select.Quote
theVTab.Query(str, theBitMap, #VTAB_SELTYPE_NEW)
theVTab.UpdateSelection
```

7.4 Accessing files

To open a file and read and display the contents:

```
theFN = "$home/inputfile.txt".AsFileName (use this line on UNIX)
theFN = "j:\avenuecourse\inputfile.txt".AsFileName (use this line on NPCS)

If = LineFile.Make(theFN, #FILE_PERM_READ)

While (true)
  str = If.ReadElt
  if (str = nil) then
    break
  else
    msgbox.info(str, "Reading...")
  end
end
```

7.5 Calling a script from another script

The Run request is used to call a script from within a script:

```
av.Run ("Name query", "")
```

This will run the script **Name query**.

8 Creating menus

If you need to create Dialogs of your own ArcView provides the Dialog Designer extension. This is loaded from the **File | Extensions** menu option. A copy of the book *Using the ArcView Dialog Designer* may be borrowed from the IT Service Desk.

9 Interapplication communication

ArcView can be integrated into other applications using a variety of methods including Remote Procedure Calls (RPC) on UNIX systems and Dynamic Data Exchange (DDE) on Windows systems.

In addition, the operating system can be accessed using the System class.

For example, on UNIX at Durham the following will start up a new window with the ArcInfo environment set:

```
System.Execute("usr/local/share/bin/arc.init")
```

and the following will create a new directory called **temp**:

```
System.Execute("mkdir temp")
```

The following can be used on a Windows system to create a new directory **test**:

```
System.Execute("c:\winnt\system32\command.com /c mkdir j:\test")
```

Further information on interapplication communication can be found in ArcView's online Help under **Customizing and programming ArcView with Avenue | Creating an ArcView application | Integrating ArcView with other applications**.

10 Avenue scripts — worked examples

This section goes through the stages of writing Avenue scripts. The first part of each sub-section includes a brief description of the particular task required of Avenue at that stage. Use this to write the relevant Avenue script alone using this Guide and the online help, or alternatively follow the notes that work through each step of writing the script.

10.1 Run a system script

Select the System Script **Project.Save** and run.

- 1 Create a new Script Editor window.
- 2 Select the menu **Script | Load System Script**.
- 3 Select the **Project.Save** script and press **OK**.
- 4 Compile the script.
- 5 Run the script.

10.2 Associate script with a new button

Using the field properties script from the online Help sample scripts create a new button item to display the properties of a field selected in a table.

- 1 Create a new Script Editor window.
- 2 Paste the script **Displays field properties** from online Help under **Sample scripts and extensions | Sample scripts | Tables** into the script window.
- 3 Select **Script | Properties** from the menu and name the script **Table.FieldProperties**.
- 4 Compile the script.
- 5 With the Project window active select **Project | Customize** from the menu.
- 6 Set **Type:** to **Table** and **Category:** to **Buttons**.
- 7 Click on the **New** button.
- 8 Double-click on the **Click control property** and select **Table.FieldProperties** from the Script Manager. Click **OK**.
- 9 Double-click on the **Icon control property** and select a suitable icon. Click **OK**.
- 10 Click on **Make Default** and close the Customize dialog box.
- 11 Finally, create a new view, add a theme and display the themes table. The new icon should now be displayed on the button bar. Highlight a field on the table and click the new icon to display the **Field Properties**.

This new button will now appear in ArcView everytime you run the application. To remove these changes, delete or rename the file default.apr from your home directory.

10.3 Add a script from a text file

This script imports an **e00** interchange file:

- 1 Create a new Script Editor window.
- 2 Select **Script | Load Text File** from the menu.
- 3 Open the avenue script **impe00.ave** from:
UNIX: **/usr/local/courses/gis/**
NPCS: **t:\its\gis**
- 4 Compile and run the script.
- 5 Select an **e00** file to import e.g. **country.e00** in:
UNIX: **/usr/local/courses/gis/**

NPCS: t:\lits\gis\

Once imported, add this new theme to a view to check the process worked correctly.

10.4 Open a user-chosen text file

Open a text file chosen by the user, display the file name and display the file contents in a message box.

- 1 Create a text file called textfile.txt in your home directory and enter five lines of text into the file.
- 2 Create a new Script Editor window.
- 3 Add the following Avenue statements:

```
existingFilename = FileWin.Show (*.txt,"Text files",
"Select a text file")
theFile = existingFilename.AsString
MsgBox.Info(theFile, "Chosen file is")
If = LineFile.Make(theFile.AsFileName, #FILE_PERM_READ)

While (true)
  str = If.ReadElt
  if (str = nil) then
    break
  else
    msgbox.info(str, "Reading...")
  end
end
```

10.5 Create a user-defined loop

Use the class MsgBox to ask the user to the number of times to repeat a loop. Display the current iteration of the loop in a message box.

- 1 Create a new Script Editor window.
- 2 Add the following Avenue statements:

```
n = MsgBox.Input("Enter the number of times to repeat loop","", "5")
for each t in 1..n.AsNumber
  MsgBox.Info( t.AsString, "Count=")
end
```

10.6 Select and zoom to features in a theme

Query a theme to select features and zoom to this selected set in the View.

(the following uses the sample data:

UNIX: \$AVDATA/esridata/world/cities.shp

NPCS: z:\licence\esri\av_gis30\avtutor\dbaccess\cities.shp)

- 1 Create a new Script Editor window.
- 2 Enter the following Avenue statements:

```

theView = av.GetActiveDoc
theTheme = theView.GetThemes.Get(0)
theFTab = theTheme.GetFTab
theBitMap = theFTab.GetSelection
'enter a value that is present in the theme you intend to use for testing
select = "montana"
'enter a field name that is present in the theme you intend to use for testing
expr = "[state_name] = " +select.Quote
theFTab.Query(expr, theBitMap, #VTAB_SELTYPE_NEW)
theFTab.UpdateSelection

```

- 3 Add the system script **View.ZoomToSelected** below the above statements. This should produce the following:

```

av.GetProject.SetModified(true)
theView = av.GetActiveDoc
r = Rect.MakeEmpty
for each t in theView.GetActiveThemes
  if (t.CanSelect) then
    r = r.UnionWith(t.GetSelectedExtent)
  end
end
if (r.IsEmpty) then
  return nil
elseif ( r.ReturnSize = (0@0) ) then
  theView.GetDisplay.PanTo(r.ReturnOrigin)
else
  theView.GetDisplay.SetExtent(r.Scale(1.1))
end

```

- 4 Compile the script.
- 5 Make sure you have a test theme active before running the script.

11 Further reading — books and manuals

The following manuals are available for loan from the IT Service Desk:

Introducing Avenue
Using Avenue
Using the ArcView Dialog Designer

Other books:

Inside ArcView GIS (1997). Scott Hutchinson & Larry Davies.
ArcView GIS/Avenue Developers Guide (1997). Amir H Razavi.
ArcView GIS/Avenue Programmers Reference (1997). Amir H Razavi & Valerie Warwick.

12 Other sources of information

A list of WWW links to GIS resources, and information on GIS discussion lists and newsgroups can be seen on the ITS GIS WWW pages:

<http://www.durham.ac.uk/its/software/gis>. This includes links to collections of Avenue scripts, in particular ArcScripts, ESRI's comprehensive collection of downloadable scripts.