Contents

Section 1 Learning model basics
Changes for chapter 1, Introducing model building.................................3
Changes for chapter 2, Setting up interactive models.............................11

Section 2 Programming with model tools
Changes for chapter 3, Establishing flow of control..............................24
Changes for chapter 4, Working within the modeling environment..........47

Section 3 Setting up advanced modeling techniques
Changes for chapter 5, Using multiple inputs........................................50
Changes for chapter 6, Using model iterations....................................51
Changes for chapter 7, Building model documentation..........................63
Introduction

This supplement has updates, changes, additions, and corrections for *Getting to Know ArcGIS ModelBuilder* (2011). The developers of ArcGIS for Desktop provide quarterly software releases, and these changes often dictate corresponding changes in workbooks. This supplement provides information so that you can use the book with ArcGIS 10.3 for Desktop. Also, there remain some errors or unclear workbook instructions found by users, such as you, that simply need to be fixed. Please visit [http://esripress.esri.com/contact/index.cfm](http://esripress.esri.com/contact/index.cfm) to report any errors or problems that you encounter.

Here’s how to use this supplement: when working through a chapter of the workbook, first scan this supplement’s corresponding section, identified by chapter number and title, and look for changes, provided by page number and section or step of the workbook. A good practice, so as not to miss any changes, is to make a mark in the workbook for each change or addition in this booklet, and then when coming to a mark in the workbook, read and use the material in this booklet. Alternately, just
keep both the workbook and this supplement open, checking for changes as you work. All of the exercises can be worked in ArcGIS Pro with the exception of exercise 2c and exercise 2d.
Section 1 Learning model basics

Changes for chapter 1, Introducing model building

Exercise 1a

Page 6, step 3 replacement

3. Find the Select a Subdivision model in the Samples toolbox. Right-click it, and then click Edit. An input variable, a tool, and an output variable are displayed. Pause the mouse over the Select tool and note that the parameters for the tool are displayed. Close the model window.
Exercise 1b

Page 12, step 6 figure replacement

Page 13, step 9

Note: If you have pinned the Catalog window in place, the Search window will appear as a tab at the bottom of the window.
Page 13, step 10 figure replacement
Exercise 1c

Page 20, step 6 browse button replacement

Page 21, step 6 figure replacement
Exercise 1d

Page 26, step 2 replacement

2. Validate that the model has done what you asked by navigating to the C:\ESRIPress\GTKModelbuilder\MyAnswers folder and examining the Results geodatabase. Open the properties of the PD_1234 feature class, go to the fields tab, and note that the field structure of the template file was used when the feature class was created. Then check the XY Coordinate System tab and note that the spatial reference system was also set.
Exercise 1e

Page 32, step 3 replacement

3. Navigate to the Data folder and find the file Car.gif.

Highlight the file and click Open.
Page 34, step 7 figure replacement
Page 34, step 8 figure replacement

Exercise 1f

Page 51, step 30 replacement

30. Since both the new buffer and the output table of the Copy Rows tool will contain the desired results, right-click the PD_Buffer and PD_Notify ovals and select Add to Display. This will ensure that both the buffer feature class and the output table are added to the map document.
Changes for chapter 2, Setting up interactive models

Page 66, added text and figure after caption “A tool parameter is set to accept user input”:

In addition to making a variable a model parameter, the user can apply certain filters to it to help control what is allowed to be entered. In the Model Properties box, the parameters for the model will be shown in a matrix with a column for filters. Depending on the input being requested, the filter can control the file, field type, workspace, or feature class type, provide a validation range, or even present a pick list of values. These can help enforce data integrity rules on your model and prevent the user from making mistakes in selecting appropriate input.
Did the parameters for your model appear in the same order as the example in the book? These can be rearranged into any order you like.

Simply open the model properties, go to the Parameters tab, and arrange them to suit your needs.
Before you begin the exercise, examine the steps needed to complete the task:

The police department may also need to use this on other point feature classes with different accident data, or even other types of incidents. You can make the input feature class a model parameter and, since this tool is only designed for point feature classes, you will add a filter to prevent using other types.

Before you begin the exercise, examine the steps needed to complete the task:

- Create a stand-alone variable and rename it.
- Make the new variable a model parameter.
- Use variable substitution to make the new variable part of the output file name.
- Set up a parameter filter.
Exercise 2b

Page 76, steps 8 to 13

8. Right-click the Accident Schema input variable and make it a model parameter. Then open the Model Properties and go to the Parameters tab.

![Image of Accident Buffer Analysis Properties window]

9. Click the word None in the filter column next to Accident Schema, and in the pull-down box select Feature Class.
10. Clear all the boxes next to the feature class types except Point. Click OK, and then OK again. Save and close the model.

11. Make sure there is an accident location selected, and double-click the Accident Buffer Analysis tool in the
Catalog window. Accept the default feature class entry, enter a buffer distance of 200 feet, and an incident number of 2233. Click OK.

12. Add the PD_2233_Buffer feature class and the PD_2233_Notify table to your Table of Contents. Examine the contents of the table and the buffer that was created.

13. Try running the tool again, and this time instead of using the default feature class, navigate to:

    “F:\ESRIPress\GTKModelbuilderInstrutor\Data\OleanderFireDept.mdb\Base Data.”

**Note:** Only point feature classes are shown, even though there are other types in this location. Select FireStations, enter an incident number of 3899, and set a buffer distance of 500 feet. Click OK. Examine the results and then close ArcMap.
Page 77, new text after “What you’ve learned so far” box:

You were able to place a restriction on the types of feature classes the user could select. This type of restriction can also be placed on other entry types with a filter, and you can even create a pick list for some types.

Page 80, new text above “Exercise 2c”

Note: If you loaded the sample data in a location other than the default on the C: drive, you may experience broken data links in the layer files. To fix the broken data link for the Generic Line FC.lyr file, in ArcMap, right-click the .LYR file and select Properties. Click the Source tab and you'll see that the second value ("Location:"') is set to:

C:\ESRIPress\GTKModelBuilder\Data\CityOfOleander.mdb

Click the Set Data Source button and navigate to the drive and location where you installed the exercise data. For example, if you installed on the E: drive, your location would be (including the feature dataset name and the feature class name):
You can move the dialog box to the side and see the correct feature dataset name and feature class name while you are doing this. Once you have reset the data source location, click OK to close the layer properties.

There are two other .LYR files in the \Data folder which will also need to be fixed: PublicWorks Line FC.lyr and ZoningDistricts.lyr.

**Exercise 2c**

**Page 85, step 12 replacement**

12. **Double-click the Public Works Notify model in the Catalog window. In the Public Works Notify dialog box, click the drawing template Feature_Set.**
13. In the map area, draw a line down the middle of Kimble Dr, double-clicking the last point to save the line. Notice that the feature set is temporarily added to the Table of Contents and is drawn with a preset symbol that was extracted from the symbology layer file template.
Exercise 2d

Page 92, step 9 replacement

9. Double-click the Public Works Double Notify model to run it. Notice that the Feature Set now has two templates to choose from; one to perform a 40-foot buffer and one to perform a 400-foot buffer. These are created from the values that were set in the domain of the Num_Integer field in the template file you used for the feature set variable.
Select the 400 Feet template and draw a line along the center of Erica Ln, double-clicking to end the feature.
10. Next, click the 40 Feet template and draw a line along the center of Carol Way.

11. Finally, use the 400 Feet template again to draw a line along the center of Amy Way. Change the name of the output table to **PW_Tuesday** in your results geodatabase and click OK to finish running the model.
Section 2 Programming with model tools

Changes for chapter 3, Establishing flow of control

Exercise 3a

Pages 104–105, step 11 replacement

11. Finally, use the coordinate system Browse button to set this value. The Favorites should include the correct value (NAD_83_StatePlane_Texas_North_Central_FIPS_4202_Feet).

If not, import the coordinate system from the Accident Information feature dataset in the City of Oleander geodatabase.
Spatial Reference Properties

XY Coordinate System  Z Coordinate System

- Favorites
  - NAD_1983_StatePlane_Texas_North_Central_FIPS_4202_Feet
- Geographic Coordinate Systems
- Projected Coordinate Systems
- Layers

Current coordinate system:

NAD_1983_StatePlane_Texas_North_Central_FIPS_4202_Feet
WKID: 2276 Authority: EPSG

Projection: Lambert_Conformal_Conic
False_Easting: 1968500.0
False_Northing: 6561666.66666666
Central_Meridian: -98.5
Standard_Parallel_1: 32.13333333333333
Standard_Parallel_2: 33.96666666666667
Latitude_Of_Origin: 31.66666666666667
Linear Unit: Foot_US (0.3048006096012192)
Exercise 3b

Page 113, step 2 figure replacement

Page 114, step 3 figure replacement
Page 118, step 13 figure replacement

Page 119, step 14 replacement

14. Run the Public Works Notify tool again, and this time use the selection template to draw a line over Peterstow Dr.

Set the output table to Peterstow in the results geodatabase. Click OK.
Exercise 3c

Page 129, new box and figure between steps 1 and 2

Python scripts can be edited within ArcGIS using any available Python code editor. This is set in the Geoprocessing Options. On the main menu, click Geoprocessing > Geoprocessing Options and in the Script Tool Editor/Debugger section type in the path to the Python editor you wish to use. By default this book uses IDLE that ArcGIS installs with Python. The path for the IDLE editor is:

<Install Drive Letter>:\Python27\ArcGIS10.1\Lib\idlelib\idle.pyw
Follow the accompanying code for this script. You can see at the top that it imports the ArcPy module. Then it creates an input variable to hold the user’s suggested feature dataset name. It then sets the workspace environment using input from the model. This tells the script where to search for the existing feature dataset. The last part runs the arcpy.Exists method, which tests for the existence of the specified data object and
checks the input feature dataset name. If the item exists, the script
returns a value to the model that will cause it to skip the creation process
along with the name of the existing feature dataset. Otherwise, it will
return a value that will signal the model to create the specified feature
dataset.

Page 130, step 3 figure replacement
4. Double-click the Check Exists script to open the parameters input window. Set Input FDS Name to New FDS Name and Input Workspace to Workspace. Click OK and note that the script moves to the Ready to Run State. Click Auto Layout to organize the model elements.
Change “Output False” to “Output Boolean.”

Page 131, step 7 figure replacement

Page 132, step 8 replacement

8. On the ModelBuilder menu bar, click Insert > Model Only Tools > Merge Branch and drag the tool to an open area of the model.
9. Double-click the Merge Branch tool to open its dialog box.

In the In Values text box, add the variables Output FDS and Existing FDS. Make sure the variables are in this order. If not, reverse them. Click OK.
Page 133, step 10 figure replacement
Exercise 3d

Page 143, step 1

Update the Python version from “Python 2.6” to “Python 2.7.”

Page 143, step 3

Change index values (removing index 2) to:

0 - Input Feature Class Name (required)

1 - Output Boolean Variable for True/False value
Page 144, step 3 figure replacement

```python
# FC_Check.py
#
# This script will test for the existence of a user specified Feature Class.
# The output will be either True or False.
#
# Arguments:
#
# 0 - Input Feature Class name (required)
# 1 - Output Boolean Variable for True/False value
#
# Your Name
# Month Year
# ArcGIS 10.1
#
```

Page 144, step 4 figure replacement

```python
# Load ArcPy module
# This makes all the ArcGIS geoprocessing tools available in Python.
import sys, os, arcpy
```

Page 144, step 5 figure replacement

```python
# Get the input from the model
InputFC = arcpy.GetParameterAsText(0)
```
Replace the second set of code to type with this:

```python
arcpy.SetParameterAsText(1,"False")
raise Exception("Feature Class Already Exists!")
```

Replace the next set of code to type with this:

```python
arcpy.SetParameterAsText(1,"True")
arcpy.AddMessage("Feature Class Does Not Exist!")
```

Page 145, step 6 figure replacement

```python
# Statement to check the existence of the feature class.
if arcpy.Exists(InputFC):
arcpy.SetParameterAsText(1,"False")
raise Exception "Feature Class Already Exists!"
else:
arcpy.SetParameterAsText(1,"True")
arcpy.AddMessage("Feature Class Does Not Exist!")
```

Page 145, step 6 new text after figure

It may seem counterintuitive to set the “Does not exist” case to have an output value of True, but when used in a model the output will be set as
a precondition to the “Create Feature Class” tool. Only when the precondition is true will the tool run.

Page 146, step 7 replacement figure
Page 147, step 10 replacement figure

Add Script

Name:
FCExists

Label:
Check Feature Class Existence

Description:
Check to see if the user-specified feature class exists.

Stylesheet:

- Store relative path names (instead of absolute paths)
- Always run in foreground
Page 147, step 11 replacement figure

Page 148, step 12

Change “any value” to “Any value.”
Page 148, step 12 replacement figure

Page 148, step 13

Change “Output True” to “Output FC Boolean.”
14. When you have the two variables configured correctly, click Finish. The script is now a Script Tool in the Chapter 3d toolbox.
16. Perform the following steps, similar to exercise 3c, to incorporate the script into the model's processes:

   a. Connect the NewFC variable to the Check Feature Class script, making it the input feature class.

   b. Connect the Output FC Boolean variable to the Create Feature Class tool as a precondition.

   c. Save and close the model.
Exercise 3e

Page 154, step 1 replacement figure
Page 154, step 2 replacement figure

```python
# Load the ArcPy module
import sys, os, arcpy
```

Page 154, step 3 replacement figure

```python
# Get the input from the model
InputSHP = arcpy.GetParameterAsText(0)
```

Page 155, step 4 replacement figure

```python
# Create a describe object
#
descSHP = arcpy.Describe(InputSHP)
#
# Print shapefile properties
#
arcpy.AddWarning(descSHP.ShapeType)
arcpy.AddWarning(descSHP.CatalogPath)
```

Page 157

In the text after step 11, remove “an if-then-else or.”
Changes for chapter 4, Working within the modeling environment

Page 163, added text after caption “When only the file name is entered...”

There are two other system variables which are read-only and can be used specifically for storing data in a geodatabase, or for storing file-based data such as text files (.txt) or layer files (.lyr).

The first of these is the scratch geodatabase variable, or %scratchGDB%. This is set by ArcMap automatically according to the situation in which it’s used. If you have set the scratch workspace and included a geodatabase in it, then scratchGDB will be the same as the scratch workspace. If no geodatabase is set in the scratch workspace, then it will create a file geodatabase there called scratch.gdb.

If the scratch workspace has not been set, the scratchGDB will default to the user’s temporary file folder. These locations are

C:\Users\<user>\AppData\Local\Temp on Windows 7 or C:\Documents and Settings\<user>\Local\Temp on Windows XP. A new file geodatabase called scratch.gdb will be created. Note that in neither
circumstance will your scratch data be automatically deleted. It is up to the user to perform file maintenance on this geodatabase.

**Exercise 4b**

**Page 175, step 1**

Change “ArcCatalog” to “ArcMap.”

**Exercise 4d**

**Page 186, step 2 replacement figure**
Page 187, step 4 replacement figure

```python
# CreateNewFC.py
# Author: David V. Allen, GISP
# Created on: 2011-12-29 15:36:54.00000
# (generated by ArcGIS/ModelBuilder)
# Usage: CreateNewFC <Workspace> <New_FDS_Name> <NewFC>
# Description:
# Create a new Feature Class with the City of Olean drill parameters and
# spatial reference pre-set
#
# Import arcpy module
import arcpy

# Load required toolboxes
arcpy.ImportToolbox(r"C:\ESRI\Press\GTKModelbuilder\MyAnswers\Chapter 3c.tbx")
arcpy.ImportToolbox(r"C:\ESRI\Press\GTKModelbuilder\MyAnswers\Chapter 3d.tbx")

# Script arguments
Workspace = arcpy.GetParameterAsText(0)
if Workspace == ('#') or not Workspace:
    Workspace = r"C:\\ESRI\Press\\GTKModelbuilder\\MyAnswers\\Results.gdb"
    # provide a default value if unspecified

New_FDS_Name = arcpy.GetParameterAsText(1)
if New_FDS_Name == ('#') or not New_FDS_Name:
    New_FDS_Name = "New FDS Name" # provide a default value if unspecified

NewFC = arcpy.GetParameterAsText(2)
if NewFC == ('#') or not NewFC:
    NewFC = "New Feature Class Name" # provide a default value if unspecified

# Local variables:
Output_FDS = New_FDS_Name
Merge_FDS = Output_FDS
Output_FC = Merge_FDS
Existing_FDS = New_FDS_Name
Output_FC_Boolean = NewFC
Empty_FC_to_store_schema = "Empty FC to store schema"
```
**Section 3 Setting up advanced modeling techniques**

Changes for chapter 5, Using multiple inputs

**Exercise 5b**

**Page 203, step 5**

Before “Click Add,” add this instruction:

**Note:** The text file has the paths set to the default location of the sample data. If you installed the data in a different location, edit the text file and correct the paths before running the model.

**Page 204, step 8 replacement figure**
Changes for chapter 6, Using model iterations

Page 210
Change all the distances of “500 feet” to ”800 feet.”

Page 211
Change distance of “500 feet” to “800 feet.” Change the last sentence in the first paragraph to: “You’ll need to make the model repeat this process 7 times, for a total distance of 5,600 feet (a little more than a mile).” Expand bullet list to include the following:

- Create a second model
- Use a filter to make a value list
- Call a model from another model
Exercise 6a

Page 212, step 4 replacement figure
The model performs several iterations of selection to provide back data that can be used for further analysis. But the fire chief has read more of the article and has decided that this analysis should be run with the fire station as the starting point, and select the calls for service radiating out from there. The progression of selections will be the basis for the analysis he’s doing.

You will need to make a second model to allow the fire chief to select a fire station. Once the initial selection is done, you will call the first model
to complete the selections. First you need to modify the first model to make it available to the second model.

Page 215, steps 8–16 added

8. Edit the Calls for service spread Analysis model. In the model canvas, select the variable for the input features “Response 2006(2)” and make it a model parameter. Save and close the model.

9. In the Chapter 6a toolbox, create a new model called SelectStation with a label of “Select Station for Analysis” and a description of “The user selects a station from a pick list, then the surrounding features are selected for analysis.” Click OK.
10. Create a new variable in the model canvas and set its type to String. Rename it Station and make it a model parameter.
11. Double-click the Station variable and give it a value of `Station_1`. Click OK.
12. Add the Select layer By Location tool to the Model. Set the Input Features to Response 2006 and the Selecting Features to the following, including a path and a variable substitution:

C:\EsriPress\GTKModelbuilder\Data\OleanderFireDept.mdb\Base Data\%Station%

Set the relationship to **WITHIN_A_DISTANCE** and the distance to **800 feet**. Leave the Selection Type as **NEW_SELECTION**. Click OK.
13. Open the Model Properties and go to the Parameters Tab.

   Click the box under Filter and select Value List.

![Select Station for Analysis Properties](image1)

14. In the Value List dialog, add the values Station_1, Station_2, and Station_3 on the first three lines. Click OK, and OK again.

![Value List](image2)
15. Drag the “Calls for service spread Analysis” model onto the model canvas. Delete the input for the model and use the connect tool to connect the Response2006(3) to the model. Save and close the model.

16. Double click the “Select Station for Analysis” model in the Catalog window to run it. Use the pull-down list to select
Station_1, and then click OK. Note the results in the Results window.

![](image)

The fire chief can now run this for each station and record the results for his analysis.

**Page 215**

Add to “What you’ve learned so far” box:

- How to set up a filters pick list on a variable
- How to call a model from another model
Page 216

In third paragraph, change the first sentence to:

The Calculate Value tool should be made a precondition of the final process so that the process will not run unless the precondition evaluates to True.

Page 216 replacement figure

Exercise 6b

Page 219, step 6

Change “SqMiles” to “TOTAL_POP.”
Page 219, step 6 figure replacement

Exercise 6c

Page 226, step 2

Change the last sentence to: “Be sure the relative paths check box is selected.”

Page 232

In the first line, change “model parameters” to “model properties.”
Changes for chapter 7, Building model documentation

Exercise 7a

Page 271, step 14

Change “2010” to “2011.”

Page 272, step 15 replacement figure

Select by Incident Code

Copyright 2011 - David W. Allen, GISP
Page 275 replacement figure

**Item Description**

**Title**  
My Custom Model for Running Multiple Processes

**Thumbnail**

Page 277 replacement figures

**Summary (Abstract)**

- A valid sewer line database
- A set of street centerlines
- A symbology template layer

Running this model will produce a sewer line summary. Note that all lines in excess of 48” in diameter are not maintained by the City of Oleander and should not be included in the maintenance budget.

**Usage**

1. If any edits have been made to the data since the last time this model was run, you will need to calculate the length in feet and meters into the LengthFT and LengthM attribute fields.

   Right click the field and select Calculate Geometry

2. Verify that the latest as-built drawings have been scanned and indexed to the sewer line data. The file names will be included in the output database.
Page 279

In the second paragraph, replace “And finally” with “Next.”

After the caption “The credit information is important...” add the following text:

And finally, any use limitations or the appropriate scale range can be recorded. There may be legal restrictions on the use of the scripts, how and where the model is distributed, or any information another user may need to know before using the model in their workflow. A suggested scale range can be set by moving the sliders to represent the upper and lower scale limits.
Page 279 replacement figures

Code Samples

This custom model will perform two processes that I designed and produce a results file that may be used for future analysis.

Code

```python
# Import arcpy module
import arcpy

# Script arguments
Input_Parameter = arcpy.GetParameterAsText(0)

if Input_Parameter == '#' or not Input_Parameter:
    Input_Parameter = 100 # provide a default parameter if not specified

Input_Feature_Class = arcpy.GetParameterAsText(1)

if Input_Feature_Class == '#' or not Input_Feature_Class:
```

Use Limitation

This model is placed in the public domain by the City of Oleander and may be freely distributed.

Appropriate Scale Range

City

1:50000

Buildings

1:5000

Credits

Author: David W. Allen, GISP
Date: Jan 2012
City of Oleander, Texas
Page 279 replacement caption

Information on the model’s use and scale limitations is stored here.

Exercise 7b

Page 285, step 12 replacement figure
16. Finally, type your name as the model’s author and add the 
use limitations and appropriate scale range as shown in 
the accompanying graphic.