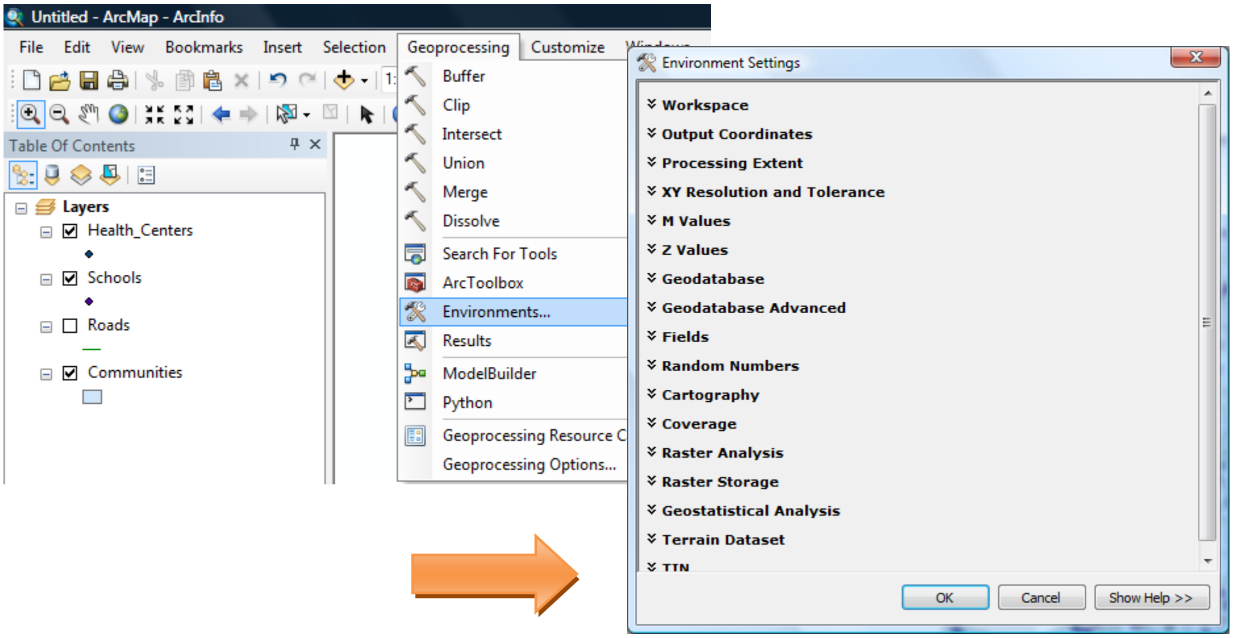
Chapter 5: Environment settings

5.0 Introduction

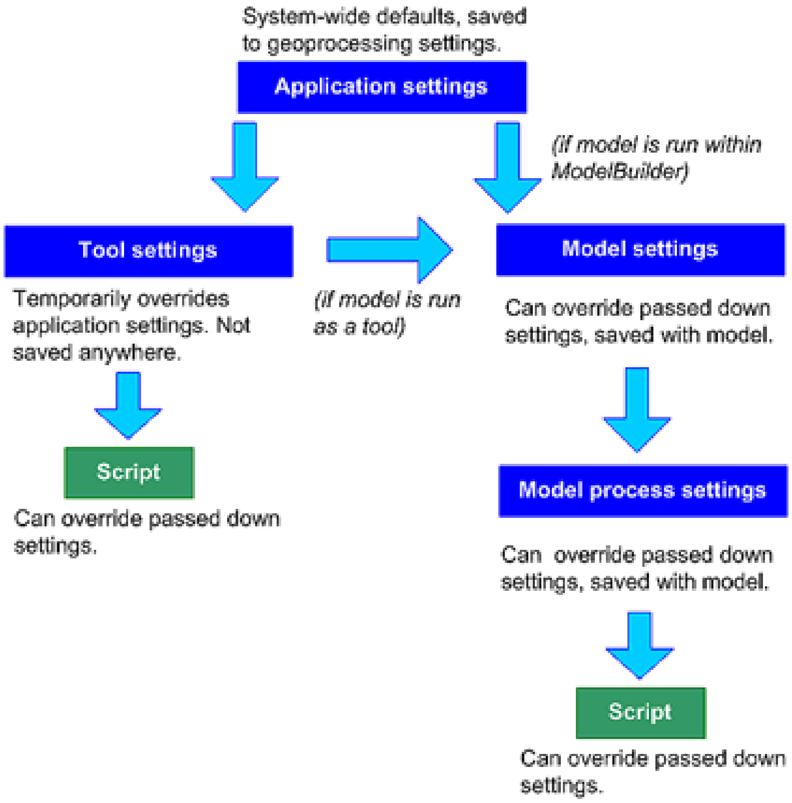
- Geoprocessing environment settings can be thought of as additional parameters that affect a tool's results.
- They differ from normal tool parameters in that they are set separately from the tool and are interrogated and used by tools when they are run.
- Don't appear in the Tool's dialog
- Environment settings such as an area of interest, the coordinate system of the output dataset, and the cell size of a new raster dataset can all be specified and honored by the tools.
- Organized by Category



- Can be set in Python
- When a script executes it utilizes various input parameters in addition to related environment settings.
- Tools only use the environment settings that they need
- Can use the tool help in the Tools Dialog to get more information on each of the various options.

5.1 Levels of Hierarchy

- There are four levels of environment settings: application, tool, model, and model process.

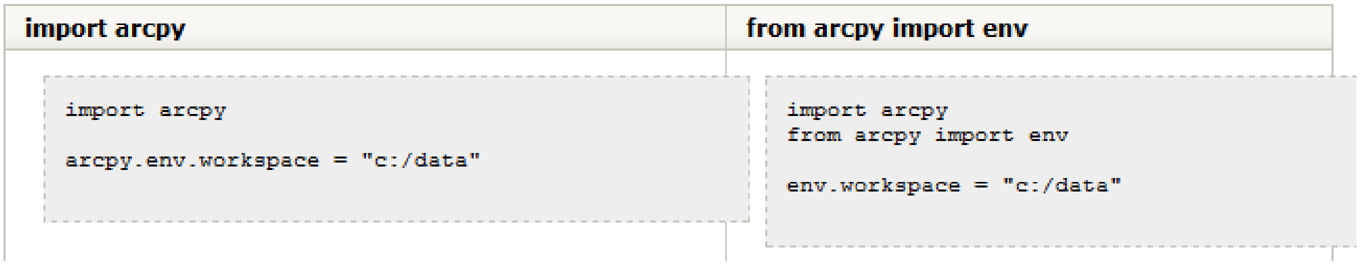

System-wide defaults, saved to geoprocessing settings.
Application settings
(If model is run within ModelBuilder)
Tool settings
Temporarily overrides application settings. Not saved anywhere.
(If model is run as a tool)
Model settings
Can override passed down settings, saved with model.
Script
Can override passed down settings.
Model process settings
Can override passed down settings, saved with model.
Script
Can override passed down settings.

Source: http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//001w00000036000000.htm

- All levels contain the same environment variables and have the same effect on output results. They differ only in how you access and set them.
- The four environment levels form a hierarchy where the application level is highest. In this hierarchy, environment settings are passed down to the next level, as illustrated below. At each level, you can override the passed-down environment settings with another setting.
- The one exception to this hierarchy is with models. If you run the model from the ModelBuilder window, the application environment is passed down. If you run the model using its tool dialog box, the tool environment is passed down.
- There are four levels of environment settings:
  - **Application-level** settings are the default and will be applied to any tool when it is executed.
  - **Tool-level** settings are applied to a single run of a tool and override the application-level settings.
  - **Model-level** settings are specified and saved with a model and override tool- and application-level settings.
  - **Model process-level** settings are specified at the model process level, are saved with the model, and override model-level settings.

5.2 Environment settings in Scripts

- Scripts can be used to both retrieve and set the environment settings
- Can override any default environment settings from within a script.
- In ArcPy, geoprocessing environments are organized as properties under the ArcPy class env. These properties can be used to retrieve the current values or to set them.
- Each environment setting has a name and a label.
- Labels are displayed on the Environment Settings dialog box in ArcGIS.
- Names are used in scripts or at the command line in ArcGIS applications.
- Environment settings are read/write properties of the env class and can be accessed by as arcpy.env.<*environmentName*>
- Before the env class can be used it must first be imported
- Alternatively, instead of prefixing each environment name with arcpy.env, you can simplify your code by taking advantage of Python's from-import statement. This alternative has the advantage of simplifying your code and making it easier to read.



import arcpy	from arcpy import env

`import arcpy`
 	
`arcpy.env.workspace = "c:/data"`
	`import arcpy`
	`from arcpy import env`
	
	`env.workspace = "c:/data"`

- Settings can be set within a script, overriding any passed down environment settings. This ONLY applies to the current execution of the script.
- Settings are NOT ALWAYS passed down
  - Script run from outside ArcGIS application e.g. OS Command prompt
- Script calls another script
  - Calling script does not know if the called script contains the Geoprocessor object

5.4 Exercise 4