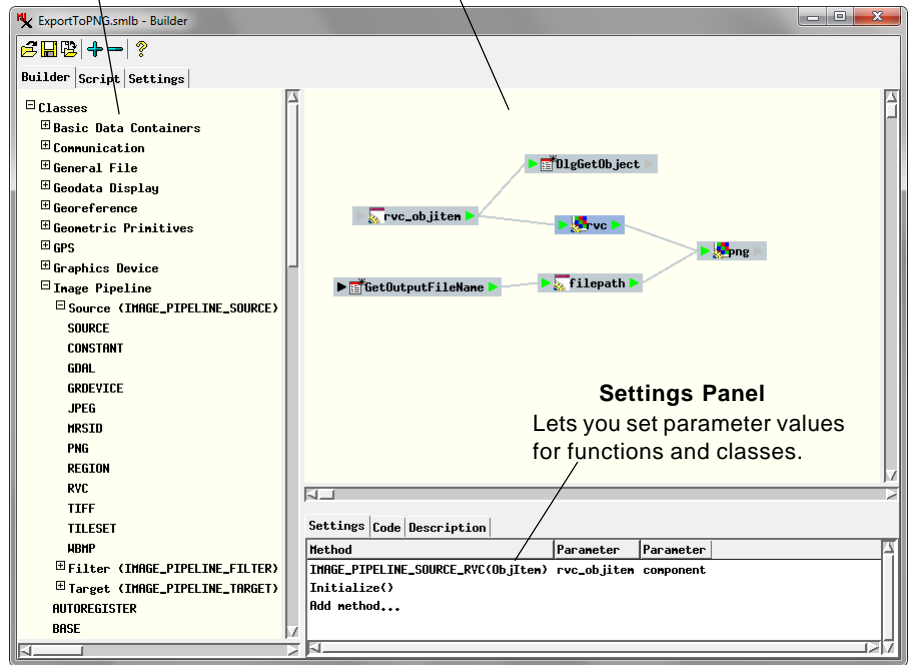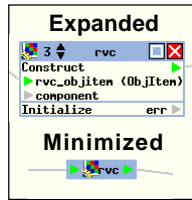# Script Builder Overview

The Script Builder in TNTmips Pro (Script / Builder) provides a graphical design environment for creating standalone processing scripts in the TNTgis geospatial scripting language (SML). The Builder window (shown at right) provides a design canvas and a list from which you can choose classes and functions to place in the canvas. The design canvas shows these script components as boxes that can be moved as needed and minimized or expanded to hide or show the details of the function/class parameters. You create a process flow in the design canvas by graphically connecting the output of one component to the input of the next. These

Class & Function List
Select components from list to place in canvas.

Design Canvas
Functions and classes shown as boxes which you can position and connect to establish processing sequence.

Settings Panel
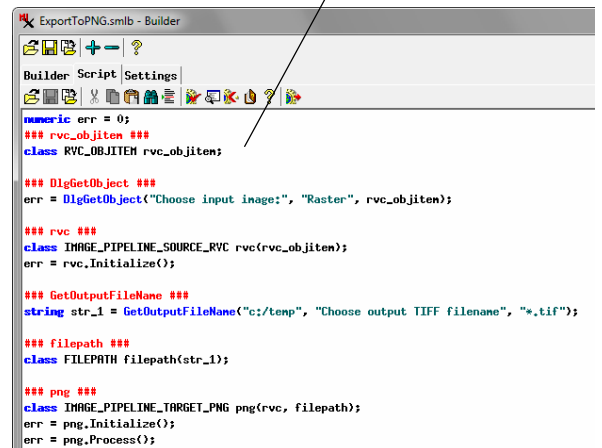Lets you set parameter values for functions and classes.

Class and function boxes can be shown in the design panel expanded to show all parameter details or minimized to provide a simple overview of the process flow.
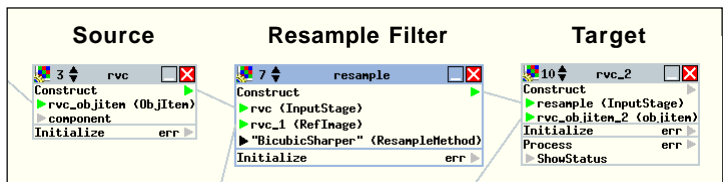
connections are shown by the gray lines in the design canvas, and the component inputs and outputs are represented by arrowheads in the boxes. Connecting components establishes a sequence of execution and automatically generates SML code that can be viewed and run at any time using the Script tabbed panel. Script designs can be saved to, and reloaded from, a file (*.smlb) that stores your selection of script components and their settings, positioning in the design space, and connections.
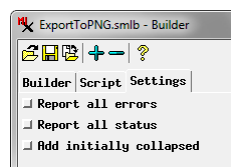
The Script Builder is best suited for creating scripts that use the image processing pipeline classes in SML, which have a modular, object-oriented processing design. The Builder also provides automated operations for adding interactive prompts for input and output objects and files, and settings that provide automatic error and status reporting when the script is run (see illustration below).

Details on builder operations can be found in the Technical Guide entitled *Geospatial Script Builder: Operating the Builder*. A number of sample builder files and accompanying descriptive TechGuides can also be found at microimages.com.

The Settings tabbed panel provides toggles that let you set whether to have code automatically generated to report errors and process status whenever the script is run. You can also set whether graphical design components are initially added in minimized or expanded form.

Script Tabbed Panel
Shows automatically-generated script code and provides controls for checking and executing the script.

```
numeric err = 0;
### rvc_objitem ###
class RVC_OBJITEM rvc_objitem;

### DlgGetObject ###
err = DlgGetObject("Choose input image:", "Raster", rvc_objitem);

### rvc ###
class IMAGE_PIPELINE_SOURCE_RVC rvc(rvc_objitem);
err = rvc.Initialize();

### GetOutputFileName ###
string str_1 = GetOutputFileName("c:/temp", "Choose output TIFF filename", "*.tif");

### filepath ###
class FILEPATH filepath(str_1);

### png ###
class IMAGE_PIPELINE_TARGET_PNG png(rvc, filepath);
err = png.Initialize();
err = png.Process();
```

You design a processing flow by connecting one script component (function or class) to the next component in the process. The example above shows part of the design for a resampling process using Image Pipeline classes. The input stage for the resample filter in the middle has been specified by connecting its input arrow to the output arrow of the RVC image source on the left. The output arrow of the resample filter has been connected to the input stage arrow of the image target on the right.