

Integrating ENVI with ArcToolbox and ModelBuilder



1 Introduction

Deriving informational products from geospatial data often requires performing both image processing tasks and GIS analysis utilizing separate software applications. Many times these software applications are run concurrently and intermediate products are manually transferred from one application to the other until a finished product is produced. This rather inefficient process is necessary since the user typically can not create the desired output product(s) by using only a single package --- the image processing and GIS software are used in tandem.

Users have expressed a keen interest in improving the interoperability between image processing packages like ENVI and GIS packages like ArcGIS. Improved interoperability increases productivity, reduces training and expertise requirements for multiple software packages, and helps to justify the investments and costs in purchasing these products. In addition, software development costs are reduced since custom software tools are developed once and then leveraged from both applications.

Over the past few years ITT Visual Information Solutions and ESRI have been working towards improving the interoperability between their respective products: ENVI and ArcGIS. This whitepaper discusses a methodology developed jointly by ITT VIS and ESRI that enables ENVI based image and data processing functionality written in IDL to be called directly from ArcToolbox and ModelBuilder.

This is an exciting and powerful development because once ENVI based functionality is made available in ArcToolbox and ModelBuilder, the ArcGIS user can utilize these tools as if they were available natively. An increase in productivity is achieved since the ArcGIS user can stay in a familiar environment and the output products generated by the ENVI routines are pushed directly back into ArcMap and/or a geodatabase.

The ENVI based tools can be easily shared with other users in the same manner as Arc Toolbox tools and Model Builder models are currently shared. Although the ENVI based tools may appear to be native ArcToolbox tools an ENVI license is still required to run them.

This whitepaper will give an overview of the methodology for integrating ENVI capabilities with ArcGIS via ArcToolbox and Model Builder. If further information is required regarding this topic, contact a member of the ITT Visual Information Solutions Sales Team at 303-786-9900 or <http://www.ittvis.com>. For additional information on ArcGIS, ArcToolbox, and Model Builder, contact an ESRI representative at <http://www.esri.com>.

2 Methodology

The goal of this methodology is to provide access to specific ENVI routines from within the Arc Toolbox and ModelBuilder framework thus enabling the ArcGIS user to run these routines either individually or as part of a custom model created in ModelBuilder.

The first step in the methodology is to identify the specific ENVI routine that will be integrated. This routine can be an existing ENVI routine or a custom routine. Ideally, the ENVI routine should be one that requires no user interaction during execution. In the ENVI environment, this type of routine is one that can be run in batch mode. The next step is to create a tool in the ArcToolbox that gathers any inputs from the user, calls the ENVI routine and passes the required inputs to the routine. The following section will briefly describe components of this methodology.

2.1 ENVI routines suitable for integration with ArcToolBox

An ENVI routine suitable for integration with ArcToolbox can either be an existing ENVI routine or a custom routine developed by the user. One of the major strengths of ENVI is the ease in which users can extend the existing capability of the software by developing custom routines and functionality. This is typically done utilizing the Interactive Data Language (IDL), which is one of the core products available from ITT VIS. In fact, most of the ENVI package is written using IDL. IDL is an interpreted language designed to facilitate processing, analysis, and visualization of all sorts of 2D imagery and 3D datasets.

In addition, routines suitable for integration with ArcToolBox are those that can be run in ENVI batch mode, which means that during execution no user interaction is required. The required inputs to the routine are passed via parameter or keyword variables.

The ability to run ENVI routines in batch mode means that external batch scripts, run from the operating system command line, can be created that start ENVI without the user interface, in other words a headless ENVI, then call the routine with its required inputs. Once the ENVI routine is finished ENVI exits on its own.

ENVI is essentially an IDL application and thus its routines are IDL functions or procedures. Therefore, the external batch scripts are really starting IDL and running an IDL routine. Our methodology for integrating ENVI routines with ArcToolbox leverages this ability to call IDL and ENVI from the command line.

Details of how to call IDL from the operating system command line can be found in the IDL help system.

2.2 ArcToolbox and ModelBuilder

ArcToolbox and ModelBuilder are components of the geoprocessing framework in ArcGIS. The purpose of the geoprocessing framework is to enable the user to automate processing tasks and to perform modeling and analysis. The geoprocessing framework provides the basis for spatial analysis and modeling through numerous tools and operators that work on geographic datasets.

ArcToolbox is a collection of GIS tools organized into a number of toolboxes and toolsets within each toolbox, thus providing organized access to all of the tools in ArcGIS and a mechanism for finding and executing the desired tools. The typical tool takes ArcGIS datasets as inputs, such as feature classes, rasters, and tables, applies an operation on the data, and creates a new dataset. Figure 1 shows the ArcToolbox dialog containing the standard set of tools along with a custom ENVI Toolbox.

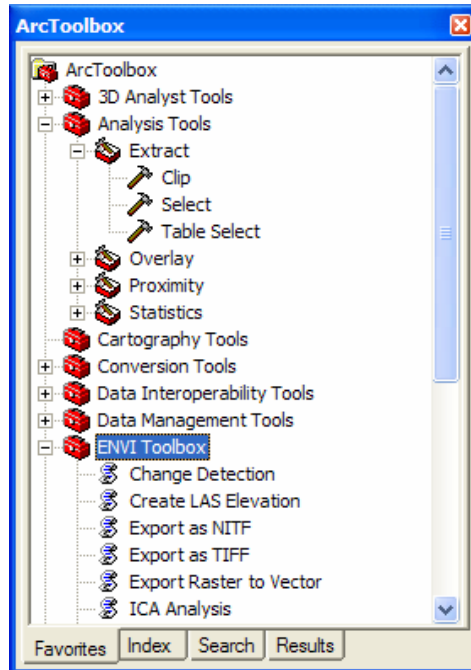


Figure 1: ArcToolbox Window

The ModelBuilder allows the user to graphically create and execute geoprocessing models which are created by dragging and dropping tools from ArcToolbox into the ModelBuilder interface and connecting the tools together to form a sequence of operations. Figure 2 shows an example of a model that consists a several different tools available in ArcToolbox.

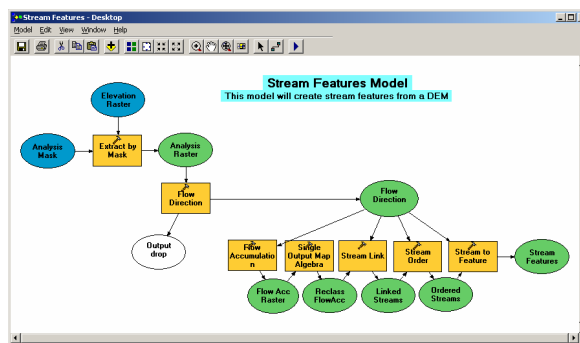


Figure 2: Sample model in ModelBuilder

Functionality that is not available in ArcGIS can be added by developing scripts using the Python programming language. The new functionality then becomes a new tool available within ArcToolbox, which can be used just like any of the pre-defined tools including as a component within a model in ModelBuilder.

More detailed information on the ArcGIS geoprocessing framework can be found in the ArcGIS Desktop Help or on the ArcGIS website at <http://www.esri.com>.

2.3 Creating an ArcToolbox script that calls an ENVI routine

The Python programming language is utilized to create ArcToolbox scripts in the ArcGIS geoprocessing framework. Python is also an interpreted language and available in the open source (<http://www.python.org>). Programmatic access to the geoprocessing framework using Python is available from the ArcObjects component called the geoprocessor object. The geoprocessor object exposes the available geoprocessing functions to Python. The various geoprocessing functions enable the Python script to interact with the user, data, and the ArcGIS user interface.

Providing a detailed description of how to develop a Python script is beyond the scope of this whitepaper. For additional details refer to the ArcGIS Desktop Help system. The Python script will have several basic components:

- Include the required ArcGIS library functions
- Create an instance of the geoprocessor object
- Capture inputs from the GUI that is created to interface with the tool
- Create a command string consisting of the path to the IDL executable, the name of the ENVI routine, and the required input arguments.
- Pass the command string to the operating system for execution.

The Python language provides a function for submitting and executing operating system commands. This is a key feature of our methodology because it provides the mechanism by which the Python script calls the ENVI routine.

2.4 Adding the Script to ArcToolbox

Once the Python script has been created, it can be added to ArcToolbox and thus making it available to the user just like any other tool. The basic steps for adding a script to ArcToolbox include:

- Naming the script
- Identifying the Python script that will be run
- Setting the script parameters that correspond to the parameters in the Python script.

Figure 3 shows an example of parameters for a Python script that calls an ENVI routine for computing NDVI. The three parameters for this script are the name of the ENVI routine that will be called, the name of the input file, and the name of the output file.

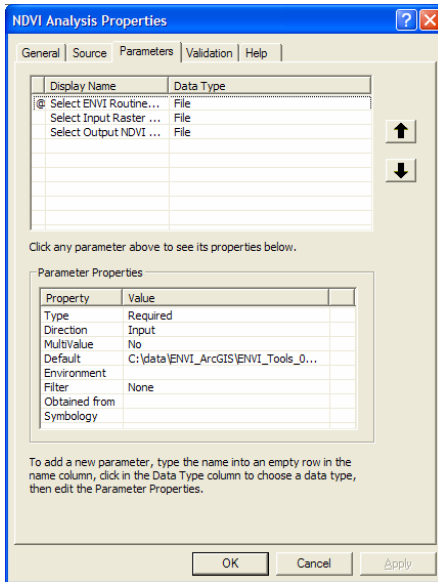


Figure 3: Parameter setting dialog for a script that calls an ENVI routine for computing NDVI. After finishing this dialog, the script is added to ArcToolbox and is now available for use.

2.5 Running the Script

Using the NDVI example, when the user clicks on the NDVI script tool in ArcToolbox, the dialog shown in Figure 4 appears. The three parameters shown in the dialog are the same ones that were set above as shown in Figure 3. One of the parameters is the name of the ENVI routine to run. Our methodology includes this parameter to allow for flexibility when sharing the tool with other users since each user may install the tool in different locations.

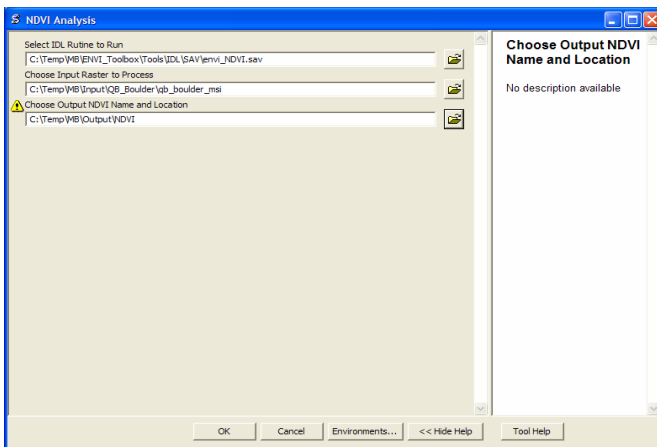


Figure 4: Dialog that appears after selecting the NDVI script tool in ArcToolbox

After setting the parameters to valid values, the script is started when the user presses the OK button. A status dialog, similar to one shown in Figure 5, appears providing the user feedback regarding the progress of the script.

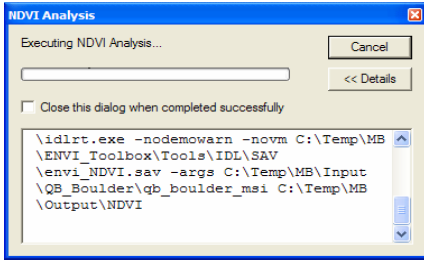


Figure 5: Progress dialog for the NDVI script

Note that while the script is running the user has no direct interaction with ENVI. The ENVI routine executes using the parameter information and then exits. The output of this particular example is shown in Figure 6.

One of the features of our methodology is to ensure that the output products are automatically loaded into ArcMap or geodatabase. Although this strategy is flexible and can be easily changed as desired, it allows the user to begin using the results immediately rather than having to manually load the file into ArcMap.

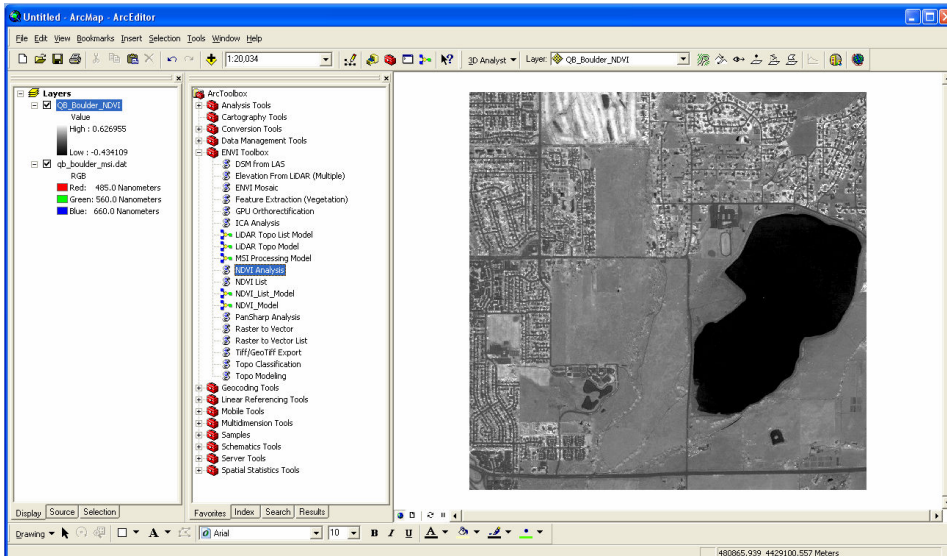


Figure 6: Output of NDVI script computed using the ENVI NDVI procedure displayed in ArcMap

2.6 A ModelBuilder Example

Once a script is added to ArcToolbox it can be used just like any other tool, including as part of the model created using ModelBuilder. Figure 7 shows an example of a model that takes 6 input lidar files computes a digital elevation model (DEM) for each dataset, mosaics the 6 DEMs and then computes topographic features from the DEM. The final products are then loaded directly into ArcMap as shown in Figure 8.

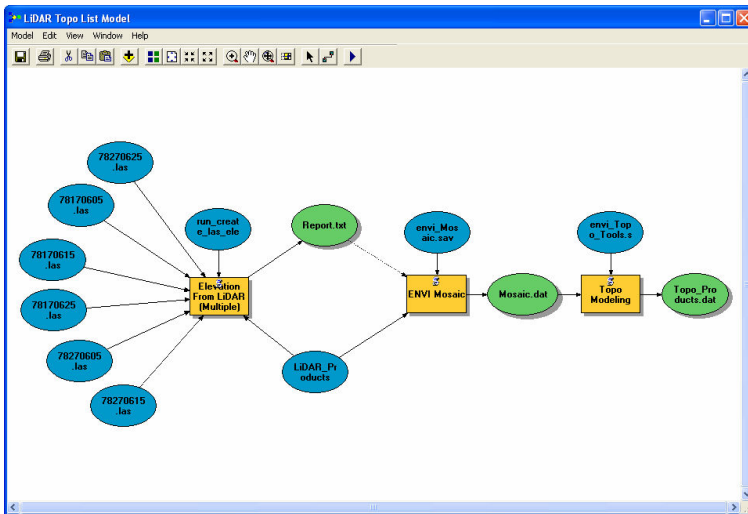


Figure 7: Example model that utilizes several scripts that each call a separate ENVI routine.

The yellow boxes represent the scripts that will be run as part of this model. The blue boxes are the input parameters required by each script. The green boxes indicate the outputs.

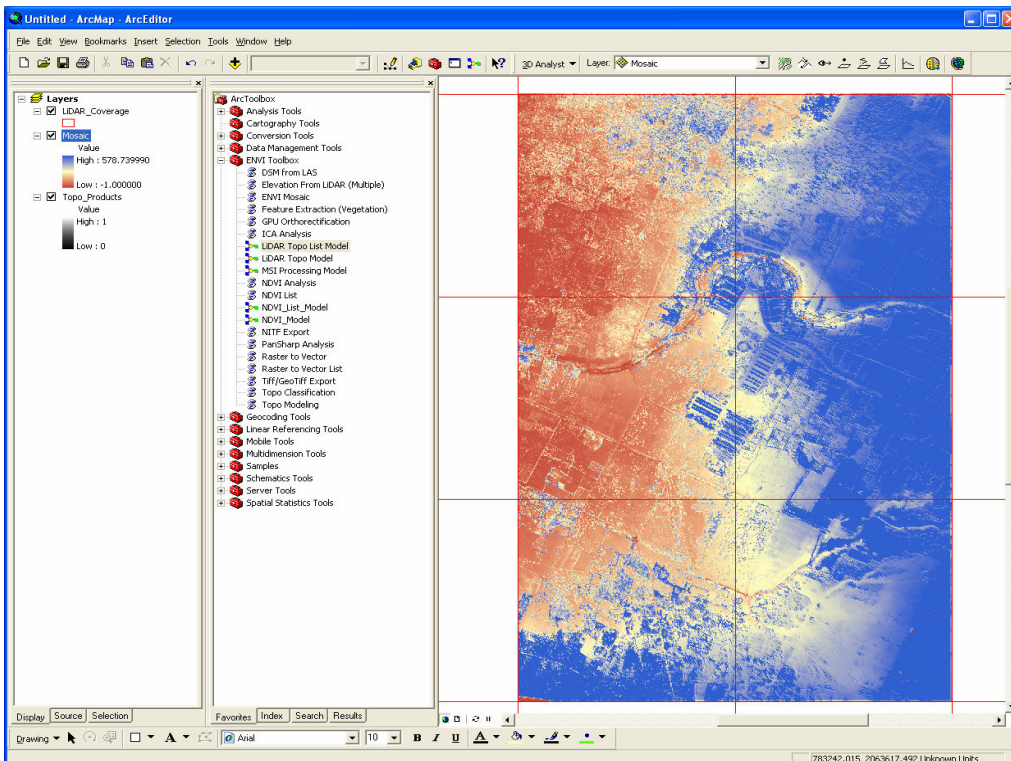


Figure 8: Output produced by model shown in Figure 7.

This particular example illustrates the power and flexibility of this methodology. The model combines 3 different scripts, that each call a separate ENVI routine, into a single workflow that is run on multiple files and without user interaction. The user runs this

entire workflow from the ArcGIS environment and obtains that the output products from the workflow directly into the open ArcMap session. In the background though, ArcGIS is leveraging functionality from ENVI that either doesn't already exist in ArcGIS or is more efficiently performed using ENVI's implementation.

3 General Requirements

The methodology described in this whitepaper was developed using ENVI v4.7 and ArcGIS Desktop v9.3 on Windows XP. Python v2.5, which is shipped with ArcGIS, was used to develop the scripts.

4 Summary

This whitepaper describes an approach developed jointly by ITT VIS and ESRI that integrates ENVI capability into the ArcGIS environment via ArcToolbox and ModelBuilder. There are several advantages for utilizing this approach.

- This approach demonstrates true integration between ENVI and ArcGIS and provides access to ENVI's powerful image and data processing capabilities to a class of ArcGIS users that may not have ever considered using ENVI in the past.
- Facilitates adding functionality available in ENVI that either is not available in ArcGIS or is not as efficient as the ENVI implementation.
- Once added to ArcToolbox, the scripts that call ENVI routines can be used just like any other tool in ArcToolbox, as well as, inclusion into custom models in ModelBuilder.
- The ENVI capability is accessed from within the ArcGIS environment and the output products are pushed directly into ArcMap and/or a geodatabase. This feature increases productivity since the user can access this capability from one software package environment.
- Users experience a greater return on their investment in both ENVI and ArcGIS since more sophisticated tools can be developed that leverage the strengths of each software package. In addition, the same functionality developed in ENVI can also be accessed from ArcGIS with minimal effort and can be easily shared with other users.
- The methodology is flexible and robust enough to rapidly integrate any existing or custom ENVI routines, especially those that can be run in ENVI batch mode.

To obtain further information regarding this methodology contact your local ITT VIS sales representative

5 Acknowledgments

ITT VIS would like to acknowledge the various ITT employees that made significant contributions to the development of this methodology. In addition, we also acknowledge the contributions of Anthony Calamito and Dan Zimble of ESRI for their contributions.