Developer quick-start guide v1.0

Lesley Bross

Robert M. Scheller

Portland State University

Last Revised: August 14, 2014

# Table of Contents

[1 Downloading and viewing the source code 2](#_Toc395788054)

[2 Landis.RasterIO.Gdal 3](#_Toc395788055)

[3 Directory structure 4](#_Toc395788056)

[4 Packaging for deployment 5](#_Toc395788057)

[4.1 Inno Setup 5](#_Toc395788058)

[4.2 LANDIS-II SDK 5](#_Toc395788059)

[5 Extension patterns 6](#_Toc395788060)

[5.1 Using the model core class 6](#_Toc395788061)

[5.2 Validating input parameters 6](#_Toc395788062)

[6 Helpful tools 7](#_Toc395788063)

[6.1 Fusion Log Viewer 7](#_Toc395788064)

[7 Acknowledgments 8](#_Toc395788065)

# Downloading and viewing the source code

1. Install LANDIS-II and extensions as recommended on the LANDIS-II installation page: <https://sites.google.com/site/landismodel/install>
2. LANDIS-II is an open source project hosted by google code in a subversion (SVN) repository. Install Tortoise SVN: <http://tortoisesvn.net/> to access the repository
3. The urls for LANDIS-II source code:

|  |  |  |
| --- | --- | --- |
| Name | Url | .dll’s |
| Core | http://[landis-extensions.googlecode.com/svn/model](http://landis-extensions.googlecode.com/svn/model) | Landis.Console-6.0.exe Landis.Core.dll Landis.Core.Implementation.dll Landis.Extensions.Dataset.dll  Landis.Extensions.exe |
| Extensions | http://landis-extensions.googlecode.com/svn | various; These are all the extension .dll’s |
| Utility Library | http://landis-util.googlecode.com/svn | Landis.Utilities.dll. (replaces Edu.Wisc.Forest.Flel.Util.dll) |
| Spatial Modeling Library | http://landis-spatial.googlecode.com/svn | Landis.Landscapes.dll  Landis.RasterIO.dll Landis.RasterIO.Gdal.dll Landis.SpatialModeling.dll |

1. Don’t forget to use https:// in the url when checking out code if you plan to make changes. Contact the LANDIS-II repository administrator to request committer permissions.
2. Some LANDIS-II projects use premake scripts to generate the .sln/.prj files, create the project folder structure, and download supporting libraries. Download the latest version of pre-make from here: <http://industriousone.com/premake>, and place it in a folder on your system path.
3. Premake supports Microsoft Visual Studio 2008/2010, and MonoDevelop so install one of those IDE’s. At this time, premake does not officially support Visual Studio 2013.
4. Premake scripts are written in Lua. You will find Lua scripts at the root of the src folder in some projects. To run a Lua script, open a command prompt in the directory containing the script and type ‘premake4 vs2010’ where premake4 is the name of the executable you downloaded in step 5.
5. The references may not be configured correctly when you open the project file due to local file system differences. All LANDIS-II references should be pointed at the .dll files that were installed in step 1. They are in the C:\Program Files\LANDIS-II\v6\bin folder.

# Landis.RasterIO.Gdal

1. Landis.RasterIO.Gdal is a LANDIS-II wrapper around the GDAL geospatial abstraction library. If you need to use classes from this library in your project some additional configuration may be required.
2. Add a project reference to C:\Program Files\LANDIS-II\v6\bin\6.0\gdal\_csharp.dll. This is the c\_sharp wrapper for the GDAL libraries.
3. Add a binding redirect element to the app.config file for this particular version of gdal\_csharp.dll. This entry maps any version of the .dll required to the version installed by LANDIS-II. Note that the specific version numbers may change if there is a new release of LANDIS-II. This sample entry is for LANDIS-II 6.0.

<dependentAssembly>

<assemblyIdentity name="gdal\_csharp" publicKeyToken="DB5A52B08DC5B321" culture="neutral"/>

<bindingRedirect oldVersion="0.0.0.0-1.0.4704.32492" newVersion="1.0.4704.32492"/>

</dependentAssembly>

1. Set the build platform target to AnyCpu in Visual Studio.
2. Make sure that the path to the LANDIS-II GDAL directory is prepended to the system path. Example: C:\Program Files\LANDIS-II\GDAL\1.9. If you run your extension using the LANDIS console or the Widgets GUI, the executables will do this for you.

# Directory structure

1. According to SVN convention, the current release version of the code is found in the trunk folder. Branches and tags folders contain code that is in process and not yet part of the main code-line. It is recommended that you create a branch or tag for the working version of your code. See SVN documentation for specifics on how to do this.
2. Most LANDIS-II extension projects contain the following folders:
   1. deploy
      1. Inno setup .iss scripts used to generate the installation executable
      2. .txt helper file used by .iss scripts to customize the installer
      3. A copy of the .exe installer file created by the Inno setup script
      4. docs folder containing the documentation that is distributed with the extension
      5. examples folder containing example scenario scripts that are distributed with the documentation
      6. May include premake.lua files
   2. src
      1. Contains the source code for the extension
      2. May also include the .sln or .prj files if they are published
   3. tests
      1. Contains test data to be used by NUnit tests associated with the extension

# Packaging for deployment

## Inno Setup

LANDIS-II uses Inno Setup to package extensions for deployment. Download and install the client from <http://www.jrsoftware.org/isinfo.php>. Note that you do need to install the compiler to work with the existing .iss scripts.

## LANDIS-II SDK

There is a developer SDK available at <https://code.google.com/p/landis-extensions/wiki/SoftwareDevelopmentKit> to assist with creating and updating .iss scripts.  
  
The SDK provides a template and settings specific to an extension are specified in a .txt file.

# Extension patterns

## Using the model core class

An instance of the model core is passed to an extension in the LoadParameters method of an extension’s PlugIn class. This instance may be used by the extension to access methods and properties of the model core such as the random number generators and timestamp functionality. Details on this pattern may be found on the landis-extensions wiki at <https://code.google.com/p/landis-extensions/wiki/ModelClass>

## Validating input parameters

The LoadParameters method of an extension PlugIn may be used to validate the contents of a scenario file. This method loads the parameters using an InputParametersParser object. The InputParametersParser should be customized for an extension and contains the validation code.

# Helpful tools

## Fusion Log Viewer

This tool from Microsoft reports errors that occur when assembly binding fails at run-time. The following links offer additional documentation and an installation hack if you need to install it:  
<http://msdn.microsoft.com/en-us/library/e74a18c4(v=vs.90).aspx>  
<http://stackoverflow.com/questions/1012252/using-fuslogvw-exe-on-a-machine-with-no-visual-studio-installed>

# Acknowledgments

Funding for the writing of the LANDIS-II Developer quick start guide was supplied by Google as part of the Google Summer of Code 2014. <https://www.google-melange.com/gsoc/homepage/google/gsoc2014>