

Title: README\_BHv1.1 -- Repository Info  
Project: LANDIS-II Landscape Change Model  
Project Component: Extension-Base-Harvest Repository  
Component Deposition: <https://github.com/LANDIS-II-Foundation/Extension-Base-Harvest>  
Author: LANDIS-II Foundation  
Origin Date: 24 Nov 2016  
Revision Date: 14 Jan 2017

Welcome to the source code repository for Extension-Base-Harvest, a LANDIS-II extension.  
This README file provides the following info:

- 1) The basic relationship between 'the science' (various biological, geological, geochemical, climatological, ecological, spatial, and landscape ecological mechanisms) and 'the model' (LANDIS-II);
- 2) The basic process for modifying and subsequently compiling the source code (written in C#) into a new, Extension-Base-Harvest.dll library.
- 3) The basic process for testing a new Extension-Base-Harvest.dll library.

#### ##### The Science and the Model #####

The science powering the LANDIS-II model ultimately resides in .cs files, written in the C# programming language. The collection of .cs files associated with the LANDIS-II Core Model or with any LANDIS-II extension is the so-called source code. Using the source code and the .NET Framework, the actual libraries (.dll files) and executables (.exe files) of the LANDIS-II model are constructed. The LANDIS-II model then uses various sets of libraries and executables to produce process-based output.

The .NET Framework provides the runtime environment needed for executing C# source code. Executing C# source code means that the source code is compiled to produce an assembly, either a library (.dll file) or an executable (.exe file). The C# code in .cs files cannot be independently executed; the use of the .NET Framework is required because the C# programming language is so-called 'managed code'.

Integrated development environments (IDEs) are used to assist in compiling .cs files into assemblies. Visual Studio and MonoDevelop are two useful IDEs for the C# programming language. To help with tracking the set of .cs files that are to be compiled, Visual Studio creates 'container' files called 'projects' and 'solutions'. A 'project' is the collection of source code files that the C# compiler will combine to produce a single output (an assembly). A Visual Studio project file is designated with a .csproj extension. A 'solution' is a set of one or more .csproj files. A Visual Studio solution file is designated with a .sln extension.

The process of building 'the science' into 'the model' is done via a LANDIS-II extension.  
The process looks like this:

- ==> a set of .cs files is created or modified that translates process-based science into algorithms, and from the algorithms, into C# source code (script)
- ==> a .csproj file is created that links the various .cs files together within an IDE
- ==> the IDE takes the set of .cs files plus the .NET Framework and 'builds' the requisite assemblies: libraries (.dll files) and executables (.exe files). LANDIS-II extensions consist ONLY of libraries.
- ==> the newly-built assemblies constitute 'the extension' and are packaged into a Windows-based installer (a Wizard)
- ==> LANDIS-II users run the Wizard which installs the extension (a set of assemblies) into the following directory: "C:\Program Files\LANDIS-II\v6\bin\extensions\"

```
#####
Building a new or modified extension from source code:
NBs
#####
```

NB. It is recommended that you use Git for version control and change tracking. This means cloning the Extension-Base-Harvest repository to your local machine. Help with Git functionality can be found in the ProGit Manual (freely available) as a .pdf (<https://git-scm.com/book/>). A very straightforward Windows/Git interface is "git BASH" (available at <https://git-for-windows.github.io/>)

NB. Should you want the LANDIS-II Foundation to consider your changes for inclusion in the LANDIS-II Foundation's main GitHub repository (<https://github.com/LANDIS-II-Foundation/>) you will need to submit a Pull request.

NB. Visual Studio (VS) may mark references to some libraries as "unavailable" until the solution is actually (re)built. During the build process, VS will automatically retrieve any requisite libraries (assemblies) from the Support-Library-Dlls repository, located at <https://github.com/LANDIS-II-Foundation/Support-Library-Dlls>. Retrieval of requisite libraries is done by running the script, "install-libs.cmd" as a pre-build event.

NB. Libraries such as "System" and "System.Core" are assemblies that should be available on your machine as part of the .Net Framework. For example, examining "System" and "System.Core" in References (Solution Explorer ==> References) yields the following output in an Object Browser window in VS,

```
Assembly System
C:\Windows\Microsoft.NET\Framework\v2.0.50727\System.dll
```

```
Assembly System.Core
C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\v3.5\System.Core.dll
```

```
#####
Building a new or modified extension from source code:
Step-by-step
#####
```

The following steps use both commands and window-driven options. The commands are Git commands which are available within a Git command-line interface (GitBASH for Windows). The menu-driven options are those available in Visual Studio (VS).

The example given below is a specific one using "Extension-Base-Harvest". The repo for this extension is available at <https://github.com/LANDIS-II-Foundation/Extension-Base-Harvest>.

===== STEP1. Clone and .git track a local version of an extension repository =====

- a. Clone the extension repository (repo) of interest to your machine.
- a1. The COPY of the extension repo on your machine is the LOCAL repo;  
the SOURCE of the cloned repo is the REMOTE repo

```
$ git clone https://github.com/LANDIS-II-Foundation/Extension-Base-Harvest.git
Cloning into 'Extension-Base-Harvest'...
remote: Counting objects: 429, done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 429 (delta 6), reused 0 (delta 0), pack-reused 410
Receiving objects: 100% (429/429), 2.54 MiB | 1.09 MiB/s, done.
Resolving deltas: 100% (263/263), done.
```

===== STEP2. Setup Visual Studio (VS) =====

- a. Open VS and load the "<name>.csproj" file
- Project ==> Open ==> Project/Solution

b. Select the 'Solution Explorer' tab to see .cs files, References  
Solution 'base harvest' (1 project)

```
C# base-harvest
  Properties
  References
  EventsLog.cs
  InputParametersParser.cs
  MetadataHandler.cs
  packages.config
  PlugIn.cs
  SummaryLog.cs
```

c. VS has added three (3) directories to your LOCAL repo:

```
...src\.vs\
...src\bin\
...src\obj\
```

d. Save all files as a solution (.sln) in the LOCAL repo before performing the  
(re)build

d1. from the pull down menus: File ==> Save All

d2. Save as

C:\Users\...\Extension-Base-Harvest\src\base-harvest.sln

===== STEP3. (Re)build the project =====

a. Under the "Build" tab, select "Build base-harvest"

b. If the VS build is successful, two new files will have been created:

Landis.Extension.BaseHarvest-3.0.dll #the functional extension to be used by the LANDIS-II  
Core-Model

Landis.Extension.BaseHarvest-3.0.pdb #a program database(.pdb) file for storing debugging  
information about the .dll; created from source files  
during compilation and not needed by LANDIS-II

b1. Note that ... \Extension-Base-Harvest\src\obj\Debug has a text file listing the  
files created by the (re)build (see <name>.csproj.FileListAbsolute.txt)

b2. Note that ... \Extension-Base-Harvest\src\bin\Debug is now populated with various  
reference libraries required for the (re)build. A copy of the newly built,  
"Landis.Extension.BaseHarvest-3.0.dll" library is also found here.

```
#####
Testing the (re)built extension
#####
```

==== STEP1. Switch out extensions =====

a. Remove the OLD "Landis.Extension.BaseHarvest-3.0.dll" from the directory,  
C:\Program Files\LANDIS-II\v6\bin\extensions\

b. Add the NEW, (re)built "Landis.Extension.BaseHarvest-3.0.dll" to the  
directory, C:\Program Files\LANDIS-II\v6\bin\extensions\

==== STEP2. Perform a test run =====

a. Run LANDIS-II with the Base Harvest example provided with the Base Harvest  
extension

- a1. Note that the Base Harvest example is installed along with Base Harvest extension which is available from the LANDIS-II Foundation website (<http://www.landis-ii.org/extensions>)

```
C:\Program Files\LANDIS-II\v6\examples\Base Harvest>call landis-ii scenario.txt
LANDIS-II 6.2 (beta release 1)
```

```
Loading scenario from file "scenario.txt" ...
Initialized random number generator with seed = 1,613,941,518
Loading species data from file "species.txt" ...
Loading ecoregions from file "./ecoregions.txt" ...
Initializing landscape from ecoregions map "./ecoregions.gis" ...
Cell length = 100 m, cell area = 1 ha
Map dimensions: 99 rows by 99 columns = 9,801 cells
Sites: 9,801 active (100.0 %), 0 inactive (0.0 %)
  reading in ecoregion from ./ecoregions.gis
Loading Age-only Succession extension ...
  Registering Data: Succession.AgeCohorts.
  Loading dynamic input data from file "age-only-succession-dynamic-inputs.txt" ...
  Dynamic Input Parser: Add new year = 0.
  Registering Data: TimeOfLastSuccession.
  Registering Data: Shade.
  Creating Dispersal Neighborhood List.
  Dispersal: NeighborRadius=5050, CellLength=100, numCellRadius=50
  Loading initial communities from file "./initial-communities.txt" ...
  Reading initial communities map "./initial-communities.gis" ...
Loading Base Harvest extension ...
  Registering Data: Harvest.PrescriptionName.
  Registering Data: Harvest.TimeOfLastEvent.
  Registering Data: Harvest.CohortsDamaged.
  Generating event table...
  Generating summary table...
  Reading management-area map ./management.gis ...
  Inactive management areas: 3
  Reading stand map ./stand.gis ...
Current time: 10
Running Base Harvest ...
dataset created: harvest/prescripts-10.img
Running Age-only Succession ...
Ageing cohorts ...
% done:  0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Computing shade ...
% done:  0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Cohort reproduction ...
% done:  0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Current time: 20
Running Base Harvest ...
dataset created: harvest/prescripts-20.img
Running Age-only Succession ...
Ageing cohorts ...
% done:  0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Computing shade ...
% done:  0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Cohort reproduction ...
% done:  0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Current time: 30
Running Base Harvest ...
dataset created: harvest/prescripts-30.img
```

```
Running Age-only Succession ...
Ageing cohorts ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Computing shade ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Cohort reproduction ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Current time: 40
Running Base Harvest ...
dataset created: harvest/prescripts-40.img
Running Age-only Succession ...
Ageing cohorts ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Computing shade ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Cohort reproduction ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Current time: 50
Running Base Harvest ...
dataset created: harvest/prescripts-50.img
Running Age-only Succession ...
Ageing cohorts ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Computing shade ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Cohort reproduction ...
% done: 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
|----|----|----|----|----|----|----|----|----|----|
Progress: ++++++
Model run is complete.
```