```
Title:                 BaseHarvest README_1.pdf/README_1.txt info
Project:               LANDIS-II Landscape Change Model
Project Component:     Extension-Base-Harvest
Repository:            https://github.com/LANDIS-II-Foundation/Extension-Base-Harvest
Author:                LANDIS-II Foundation
Revision Date:         24 Nov 2016
```

Welcome to the source code repository for Extension-Base-Harvest, a LANDIS-II extension.

This README_1.pdf/README_1.txt 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 programming process for modifying the source of Extension-Base-Harvest;

        3) The basic testing of a freshly (re)built Extension-Base-Harvest .dll

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

The LANDIS-II model uses sets of linked .dll and .exe files to produce science-based output.
The .dll and .exe files (collectively known as 'assemblies') are constructed by compiling
sets of .cs files using the .NET Framework. The .cs files are written in the programming
language, C# and may contain both code and data. The .NET Framework provides the runtime
environment and libraries needed for executing a C# program. (C# code cannot be independently
executed without the help of the .NET Framework because it is so-called 'managed code'.)
The science powering LANDIS-II resides in the .cs files, the so-called source 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 C# programming tasks, which must compile sets of .cs files, Visual Studio
creates 'container' files called 'projects' and 'solutions'. A 'project' is a collection of
source code files that the C# compiler combines to produce a single output; typically either
a library (.dll) or an executable program (.exe). 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 that translates the science into algorithms and then into
    C# script
  ==> a .csproj file is created that links the .cs files together within an IDE
    ==> the .cs files are modified, as needed, to reflect updated science or to create
        a new extension
      ==> the IDE takes the set of modified .cs files and 'builds' an assembly (a .dll or
        .exe file)
        ==> the newly-built .dll file(s) constitute 'the extension' and are packaged into
            a Windows-based installer (Wizard)
          ==> LANDIS-II users run the Wizard to install the extension (the .dll file(s))
                into C:\Program Files\LANDIS-II\v6\bin\extensions

```
############################################################
Building the Extension-Base-Harvest .dll from source code
############################################################

NB. It is recommended that you use Git for version control and change tracking.
This means cloning the Extension-Base-Harvest reposotory 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 straighforward 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 inclsuion 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 will mark references to the .dlls as "unresolved" until the solution is
actually (re)built. During the build process, Visual Studio will automatically download the
required .dlls and put them in the correct folder (.../src/libs/). The .dlls required for
(re)building Extension-Base-Harvest are automatically downloaded during the build process
from the Support-Library-Dlls repo,
https://github.com/LANDIS-II-Foundation/Support-Library-Dlls.



The following references 1) commands within a Windows/Git command-line interface (GitBASH) and
2) Visual Studio (VS) menu-driven options; some outputs are given.

===== STEP1. Clone and .git track a local repository ========================================

        a. Clone the Extension-Base-Harvest repo to your machine to create a LOCAL 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 and load the project =====================================

        a. Open VS and load "base-harvest.csproj"

        b. Select the 'Solution Explorer' tab
Solution 'base harvest' (1 project)
  C# base-harvest
    Prperties
    References
    EventsLog.cs
    InputParametersParser.cs
    MetadataHandler.cs
    packages.config
    PlugIn.cs
    SummaryLog.cs

        b1. Note that VS has added three (3) directories to your local cloned repo:


\.vs
\bin                       # Git does NOT track the \bin folder
\obj
```

c. Change the VS Reference path to ensure the solution builds correctly.

        c1. Select the 'Solution Explorer' tab ==> double-click on Properties (wrench icon)
        c2. A new window opens; note that the tab, 'Reference Paths' (on the left-hand side)
             reports, "(Not set)"
        c3. Click on the 'Reference Paths' tab; paste in the PATH TO THE GIT-TRACKED REPO ON
             YOUR MACHINE in the field under "folder:"; and add the new path as an option by
             using the "Add folder" button
        c4. Two reference paths should now be listed:
"C:\Users\<you>\<your-path-to-the-local-repo>\Extension-Base-Harvest\src\libs\"
"C:\Users\bmarr\Desktop\BaseHarvest_testRepo\Extension-Base-Harvest\src\libs\"

        c5. Delete the path,
        "C:\Users\bmarr\Desktop\BaseHarvest_testRepo\Extension-Base-Harvest\src\libs\" and
        then Save
        c6. Under tab "File", Save as a solution file (.sln)

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

        a. Under the "Build" tab, select "Build base-harvest"
        a1. If the VS build is successful, three (3) new files are created:

Landis.Extension.BaseHarvest-3.0.dll
Landis.Extension.BaseHarvest-3.0.pdb
Landis.Library.Metadata.dll.VisualState.xml

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

==== STEP1. Switch out the Base Harvest .dll file ==…

        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, freshly (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 from the Landis Foundation
           site (http://www.landis-ii.org/extensions). Note that the Base Harvest example
           is automatically installed by installation of Extension_Base-Harvest.

        a1. If the LANDIS-II run is successful, the NEW .dll will have added new output to
            the examples\Base Harvest directory. The new output is a "Metadata" folder which
            houses another folder ("Base Harvest") within which are various .xml files.

a2. The content of the directory, ...\examples\Base Harvest should now look like this:

```
Directory of C:\Program Files\LANDIS-II\v6\examples\Base Harvest
08/03/2015  10:18             1,099 age-only-succession-dynamic-inputs.txt
08/03/2015  10:18               259 age-only-succession.txt
08/03/2015  10:18             2,289 BaseHarvest-v1.2-Sample-Input.TXT
08/03/2015  10:18             9,929 ecoregions.gis
08/03/2015  10:18               202 ecoregions.txt
11/13/2016  15:35    <DIR>          harvest
11/13/2016  15:35             7,007 harvest-event-test-log.csv
08/03/2015  10:18             9,929 initial-communities.gis
08/03/2015  10:18             1,162 initial-communities.txt
11/13/2016  15:35             2,950 Landis-log.txt
08/03/2015  10:18            10,240 management.gis
11/13/2016  15:35    <DIR>          Metadata
08/03/2015  10:18               918 scenario.txt
08/03/2015  10:18               132 SimpleBatchFile.bat
08/03/2015  10:18             1,915 species.txt
08/03/2015  10:18            10,240 stand.gis
```