Summary of Century 4.7 command lines

<sch\_file> = schedule file name (do not include the .sch extension on the command line)

<binary\_file> = binary file containing all century output variables (do not include the .bin extension on the command line). Note: This file is not human readable, list100\_47.exe is required to extract output.

<extend\_file> = binary file from a previous simulation used to initialize the model for the subsequent simulation (do not include the .bin extension on the command line)

<lis\_file> = text output file containing output extracted from binary file by list100\_47.exe (do not include the .lis extension on the command line)

<txt\_file> = text file that contains a list of variables to extract from the binary file output (for example, see outvars.txt)

To run the Century model:

**century\_47.exe -s <sch\_file> -n <bin\_file>**

For instance: run example1.sch and save output to example1.bin:

century\_47.exe -s example1 -n example1

To run the Century model extending from a previous simulation:

**century\_47.exe -s <sch\_file> -n <bin\_file> -e <extend\_file>**

For instance: run example2.sch by extending the simulation from example1.bin and save output to example2.bin

century\_47.exe -s example2 -n example2 –e example1

To run List100 by responding to the prompts:

**list100\_47.exe**

To run List100 using a list of output variables:

**list100\_47.exe <bin\_file> <lis\_file> <txt\_file>**

For instance: read binary file example2.bin, save the variables listed in outvars.txt to text file example2.lis

list100\_47.exe example2 example2 outvars.txt

Commonly used DOS commands:

**cd** - **C**hange **D**irectory, displays the name of or changes the current directory

**dir** - **DIR**ectory listing, displays a list of files and subdirectories in a directory

Wildcards:

\* matches any arbitrary character string

? matches any single character

copy - COPY target file to destination file, copies one or more files to another location

erase <filename> - deletes one or more files

del <filename> - deletes one or more files (same as erase)

help <command> - displays help information on that command

cd - **C**hange **D**irectory

cd .. 🡪 move up one level in the directory structure

cd century4.5 🡪 move to a directory named century4.5



Using the \ character you can change more than one directory at a time

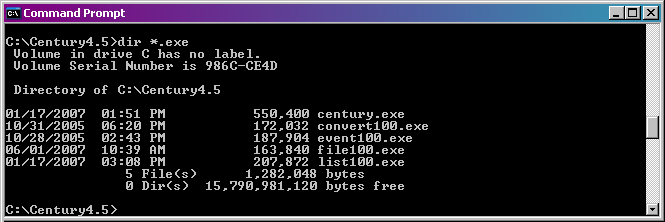


dir - DIRectory listing

dir and wildcards can be used together

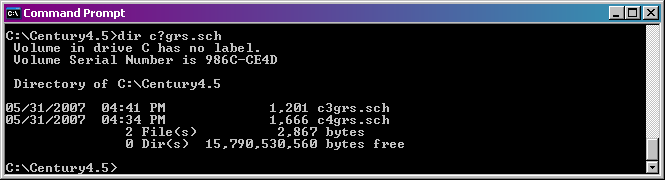
\* wildcard used with dir command

dir \*.exe 🡪 list all files in the current directory with .exe as the last 4 characters



? wildcard used with dir command

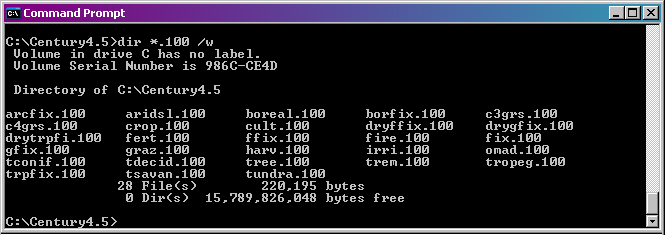
dir c?grs.sch 🡪 list all files in the current directory with c as the first character, grs.sch as the last 7 characters, with any one character in between



switches can be used to change appearance of the dir listing

/w switch 🡪 use wide list format

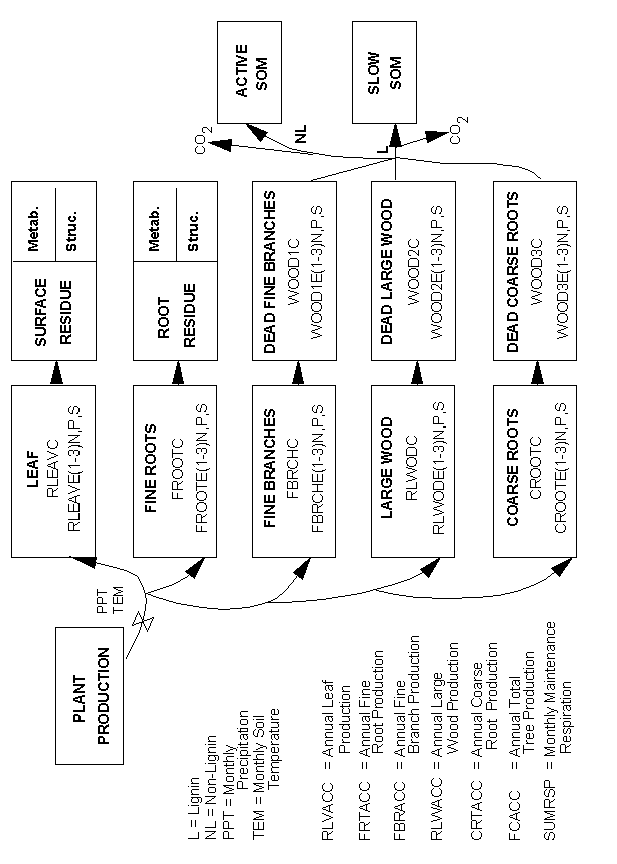
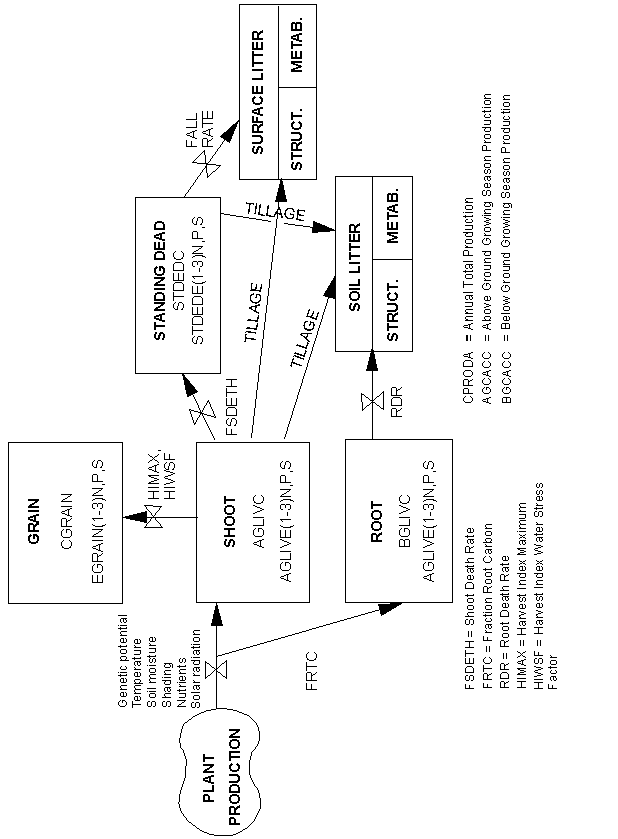
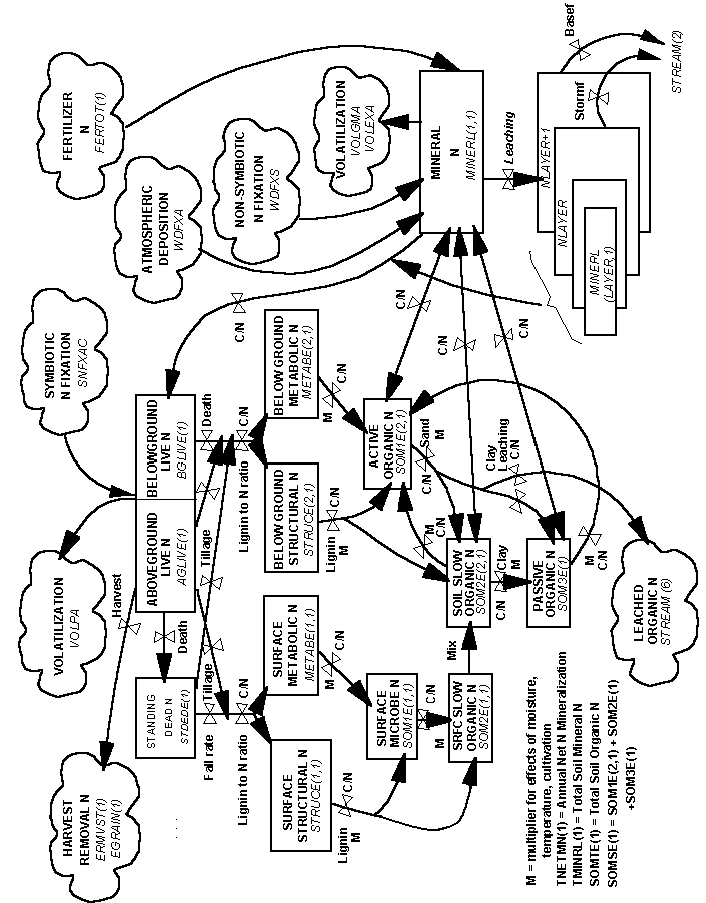
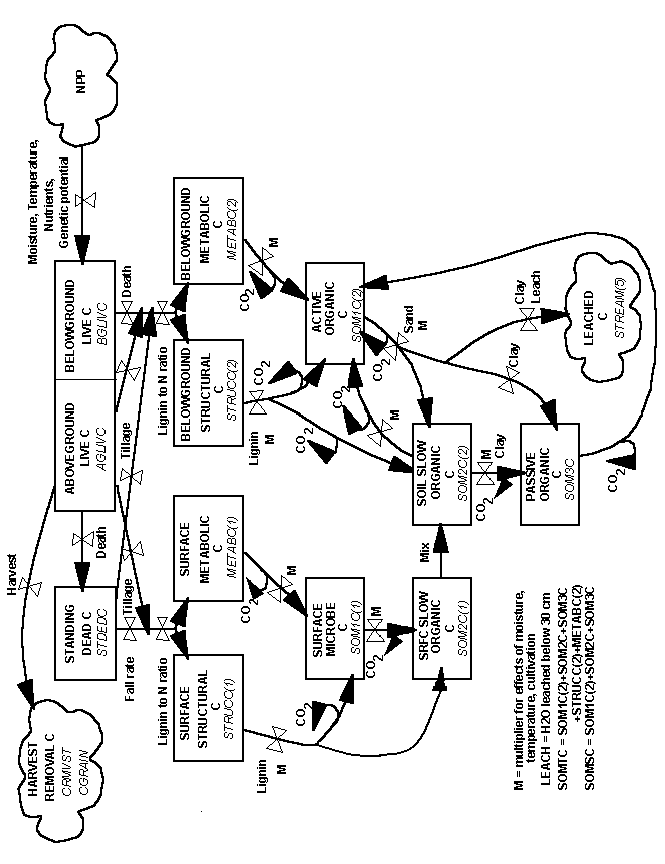
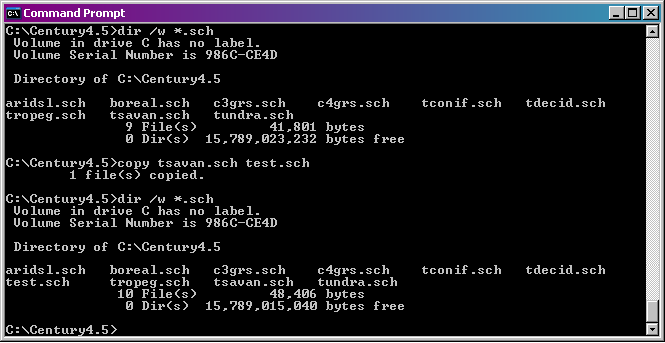
/p switch 🡪 pause after each screen full of information



copy - COPY target file to destination file

copy <target file> <destination file>

For example, copy tsavan.sch to a file named test.sch



**Step by Step: A Century Tutorial**

Prepared by Cindy Keough and Melannie Hartman (Colorado State University)

Step 1: Collect your site data

a. If desired, create a monthly weather data file using the format outlined in section 4.9., “Generating Weather Statistics”, in the Century 4.0 manual.

b. Enter site specific parameters into an existing <site>.100 file or create a new <site>.100 file with your site specific parameter values. The site file can be edited using any text editor such as Windows Notepad or Windows WordPad. Required site specific parameters are:

* latitude of site - SITLAT (decimal degrees)
* longitude of site - SITLNG (decimal degrees)
* fraction of sand in the soil - SAND (0.0-1.0)
* fraction of silt in the soil - SILT (0.0-1.0)
* fraction of clay in the soil - CLAY (0.0-1.0)
* bulk density of soil - BULKD (g cm-3)
* number of soil layers to simulate - NLAYER (1-9)
* pH of the soil

c. If you will be using a weather data file for your simulation (step a), use option 13 in the file100 utility to create weather statistics for your site file (step b) using the weather data file created in step a.

Step 2: Create Site Specific Event Options

a. Options specific to your site and simulation can be entered (or modified) in the crop.100, cult.100, fert.100, fire.100, graz.100, harv.100, irri.100, omad.100, tree.100, and trem.100 files. These files can be edited using a text editor such as Windows Notepad or Windows WordPad. Using the file100 utility to update the \*.100 files is not recommended.

b. When creating a new option in a \*.100 file the first 5 characters of the abbreviation for the option are used for entering the event into a schedule file. These abbreviations must be unique for each option in the file. *The definitions for the parameters in the \*.100 file are contained in the associated \*.def file.*

Step 3: Determine the order and types of events that you want to include in the schedule file for your simulation.

Modify example schedule files using a text editor.

Step 4: Run your simulation

a. The command line for running the Century model is:

century\_47 -s <sch\_file> -n <bin\_file>

* where sch\_file in the name of your schedule file without the .sch extension
* bin\_file is the name of the binary file of monthly output that will be created without the .bin extension.

For organizational purposes, it is easier to keep track of which schedule file and binary file go together if the same or similar file name is used for both files. For example, to run the corn.sch file to produce a binary output file named corn.bin use the command line:

century\_47 -s corn -n corn

b. Century can also be run using information from the end of a previous simulation as a starting point for a new simulation. This option is used when the model has been run to simulate a site up to a specific time period and you wish to run the model forward in time using different options to examine how different management practices will affect the model output. For example, traditional cropping until 2005 and then 4 simulations from 2006 to 2100 one with no-till, one with a reduced amount of fertilizer applied, one using a different crop rotation, and one where the site is allowed to go back to native conditions.

To run the Century model extending from a previous simulation use the command line:

century\_47 -s <sch\_file> -n <bin\_file> -e <extend\_file>

* where sch\_file in the name of your schedule file without the .sch extension,
* bin\_file is the name of the binary file of monthly output that will be created without the .bin extension, and
* extend\_file is the name of the binary file that will be read as a starting point without the .bin extension.

Using the example from step a above and extending from a file named historic.bin the command line is as follows:

century\_47 -s corn -n corn -e historic

Step 5. Examine the model output

a. It is important to check the net primary productivity (NPP) that the model is predicting for your site. If the NPP for your site is not correct then none of the other model output can be expected to be representative of the conditions at your site.

b. Use the list100 utility to extract output variables from the \*.bin file. To run list100 use the command line:

list100\_47

and follow the on-screen prompts.

The list100 utility can also be run using an input file containing a list of output variables that you wish to extract from the file. This option can be useful when running the model using a batch file or script.

list100\_47 <bin\_file> <lis\_file> <txt\_file>

* where bin\_file is the \*.bin file from which you wish to extract data without the .bin extension,
* lis\_file is the output file that will be created by the list100 utility without the .lis extension, and
* txt file is a text file with a list of output variables one per line.

For example, to extract information from the corn.bin file to create a corn.lis file with the output variables in the outvars.txt file use the command line:

list100\_47 corn corn outvars.txt

c. For binary output file variable definitions see Appendix 2.13., “Output variables”, in the Century 4.0 manual and the Century\_4.7\_ModelDevelopmentNotes.txt file.

## Complete List of Event Commands used in Century 4.7 Schedule Files

Below is a complete list of all schedule file event options available. The value of <argument> is an abbreviation ≤ 5 characters.

**CROP <argument>** – Selects the type of crop/grass to grow. Only one crop/grass can be grown at a time. One crop/grass can also be growth with one tree type in a savanna.

<argument> : An acceptable abbreviation from the *crop.100* file.

Format: The year within the block, month, and the word "CROP", followed by the crop type.

Example:

**1 5 CROP ALF**

1 5 PLTM

1 9 HARV H

1 10 LAST

1 10 SENM

**PLTM** – Marks the month in which the current annual crop/grass is planted and starts growing. Use FRST for a perennial crop/grass.

Format: The year within the block, month, and the word "PLTM".

Example:

1 5 CROP ALF

**1 5 PLTM**

1 9 HARV H

1 10 LAST

1 10 SENM

**HARV <argument>** – Schedules a harvest of the current crop/grass and designates which type of harvest to use.

<argument> : An acceptable abbreviation from the *harv.100* file.

Format: The year within the block, month, and the word "HARV", followed by the harvest method.

Example:

**1 9 HARV G**

**FRST** – Marks the month that growth commences for a perennial crop/grass. Use PLTM for an annual crop/grass.

Format: The year within the block, month, and the word "FRST".

Example:

1 1 CROP TMC4

**1 2 FRST**

1 10 LAST

1 10 SENM

**LAST** – Marks the last month of growth for a crop/grass. Applies to both annual and perennial crops/grasses.

Format: The year within the block, month, and the word "LAST".

Example:

1 1 CROP TMC4

1 2 FRST

**1 10 LAST**

1 10 SENM

**SENM** – Marks the month of senescence (death) for crops.

Format: The year within the block, month, and the word "SENM".

Example:

1 1 CROP TMC4

1 2 FRST

1 10 LAST

**1 10 SENM**

**FERT** **<argument>** – Schedules a fertilization event for the specified month.

<argument>: An acceptable abbreviation from the *fert.100* file.

Format: The year within the block, month, and the word "FERT", followed by the fertilization type.

Examples:

**1 5 FERT N5** #N5 must be found in FERT.100

**CULT** **<argument>** – Schedules a cultivation (tillage) event for the specified month.

<argument>: An acceptable abbreviation from the *cult.100* file.

Format: The year within the block, month, and the word "CULT", followed by the cultivation method.

Example:

**1 4 CULT K**

**OMAD <argument>** – Schedules an organic matter addition for the specified month.

<argument>: An acceptable abbreviation from the *omad.100* file.

Format: The year within the block, month, and the word "OMAD", followed by the type of organic matter addition.

Example:

**1 4 OMAD M14**

**IRRI** **<argument>** – Schedules an irrigation event for the specified month.

<argument>: An acceptable abbreviation from the *irri.100* file.

Format: The year within the block, month, and the word "IRRI", followed by the irrigation method.

Example:

**1 5 IRRI A100**

**GRAZ** **<argument>** – Schedules a grazing event for the specified month.

<argument>: An acceptable abbreviation from the *graz.100* file.

Format: The year within the block, month, and the word "GRAZ", followed by the grazing type.

Example:

**1 9 GRAZ GM**

**EROD <argument>** – Schedules an erosion event for the specified month.

<argument>: The amount of soil loss (kg m-2 month-1).

Format: The year within the block, month, and the word "EROD", followed by the amount.

Example:

**1 9 EROD 10.1**

**FIRE <argument>** – Schedules a fire for the specified month. This event burns herbaceous crops/grasses, litter, and dead wood on the ground. Use a TREM event to burn live tree biomass.

<argument>: An acceptable abbreviation from the *fire.100* file.

Format: The year within the block, month, and the word "FIRE", followed by the type of fire.

Example:

**1 10 FIRE HOT**

**TREE** **<argument>** – Selects the type of tree to grow. Only one tree can be grown at a time. One tree can also be growth with one crop/grass type in a savanna.

<argument>: An acceptable abbreviation from the *tree.100* file.

Format: The year within the block, month, and the word "TREE" followed by the type of tree.

Example:

**1 1 TREE TMSH**

1 1 TFST

1 12 TLST

**TREM** **<argument>** – Schedules a tree removal event (fire, cutting, or wind damage) for the specified month.

<argument>: An acceptable abbreviation from the *trem.100* file.

Format: The year within the block, month, and the word "TREM" followed by the type of tree removal method.

Example:

**28 4 TREM BURN**

**TFST** – Marks the month that growth commences for a forest.

Format: The year within the block, month, and the word "TFST".

Example:

1 1 TREE TMSH

**1 1 TFST**

1 12 TLST

**TLST** – Marks the last month of growth for a forest.

Format: The year within the block, month, and the word "TLST".

Example:

1 1 TREE TMSH

1 1 TFST

**1 12 TLST**