CHAPTER 7. RUSLE USER GUIDE

Principal contributors:

D.C. Yoder
A.J. Ketchem
D.A. Whittemore
J.P. Porter
G.A. Weesies
K.G. Renard

CHAPTER 7. CONTENTS

Welcome to the RUSLE Computer Program 259
  Purpose of the User Guide 259
  Introduction to the User Guide 259
Information for First-Time Users 260
  Getting RUSLE To Work on Your Machine 260
    Background 260
    Loading RUSLE From Supplied Diskette(s)
      Onto Computer 260
    Running RUSLE 261
  How RUSLE Works 261
    Background 261
    Description of RUSLE Screen 262
    Input Field 264
    Giving a Command 264
    Calling a Function 265
    Getting Help 265
  How RUSLE Gets, Uses, and Saves Information 265
    Background 265
    Current Input List 265
    Input Files 266
    Databases 266
Program Commands and Control 267
  Giving Program Commands 267
    List of Commands 268
    Giving a Command 271
The Functions: Program Housekeeping 272
  Calling a Function 272
  The FILE Function 272
  The EXIT Function 272
  The HELP Function 272
  The SCREEN Function 273
Help 274
  Help for a Specific Question 274
  General Help 274

Input Lists and Files 275

SAVEing the Input List to an Input File 275

LOADing an Input File into the Input List 276

Deleting Unnecessary Input Files 277

Using an Input File Created Earlier or Elsewhere 277

Databases 278

Validity of Default Database Information 278

Calling the Database Utility Routines 279

Using the Database Utility Routines 279

Editing Existing Sets or Creating New Ones 280

Printing Database Sets 280

Background on Database Files 281

Deleting Unnecessary Database Sets 281

Restoring Information Deleted From a Database 282

Using Database Sets Created Earlier or Elsewhere 282

Using the RUSLE Soil Loss Prediction Table 284

Relationship of Input Lists and Files to Lines of the RUSLE Table 284

Entering Values Directly into the RUSLE Table 285

Warning Footnotes on the RUSLE Table 285

Specific Guidelines and Answers to Common Questions 286

CITY Database 286

Answers to Common Questions About the CITY Database 286

CROP Database 288

Answers to Common Questions About the CROP Database 288

OPERATIONS Database 290

Answers to Common Questions About the OPERATIONS Database . 291

R Factor 294

Options 294

K Factor 295

Options 295

Answers to Common Questions About the K Factor 295

LS Factor 296
Answers to Common Questions About the LS Factor 296
C Factor 297
Options 297
Concepts of the Time-varying C Factor, With Examples 298
Answers to Common Questions About the C Factor 311
P Factor 315
Options 315
Answers to Common Questions About the P Factor 316
System Considerations: Using RUSLE With Your Equipment 318
Using a Printer 318
Making Copies of RUSLE To Share With Others 318
Reporting Problems and Trouble 318
Summaries 320
Command Keys 320
Terminology 323

WELCOME TO THE RUSLE COMPUTER PROGRAM

Purpose of the User Guide

This User Guide is designed to teach the first-time user how to run the RUSLE computer program, and to answer the questions most frequently asked by more experienced users. Because the program itself provides online help screens that describe the expected user responses, this Guide focuses instead on the reasoning and mechanics behind the responses. The background theory and equations used in soil-loss calculations are described in chapters 1 through 6 of this handbook.

Introduction to the User Guide

The first portion of this User Guide is meant for first-time users. It explains in general terms how RUSLE works and introduces the terminology that will be used in the following sections. This introductory section is not long, and we strongly recommend that novice users read it to familiarize themselves with the program.

The remainder of the User Guide gives specific information on the different parts of the program, including answers to frequently asked questions. This reference section of the User Guide assumes familiarity with the terminology of the program, and may be confusing if you have not read the introductory section.

Within the User Guide, a command or entry that you type in is shown in brackets, [ ], and the name given that command or entry is placed in braces, { }.

INFORMATION FOR FIRST-TIME USERS

Getting RUSLE To Work on Your Machine

Background

The RUSLE computer code was written in the C programming language. Since C is portable, the program can be run on a variety of machines. RUSLE has been tested on IBM-compatible machines with 640 kilobytes of RAM memory using the MS-DOS operating system (version 2.0 or later). It has also been used on large systems under the UNIX operating system, and on the ATT 6386 under both DOS and UNIX. The executable program will run only on the type of machine on which it was compiled; if this is something other than DOS, the machine type will be specifically noted on the diskette labels.

The RUSLE executable program is large and does not have the capability of using expanded memory. It therefore may cause problems if used at the same time as other programs that reside in lower memory, possibly including some shell programs or peripheral drivers. This problem most often shows up as an "out of memory" error message. This can be minimized by not using these memory-resident packages when running RUSLE. Consult your DOS or program manuals or your site consultant for more details.

Loading RUSLE From Supplied Diskette(s) Onto Computer

The RUSLE program can be run directly from the supplied diskette in the floppy drive. As with most other programs, however, RUSLE can be run much faster and more efficiently by loading it onto the hard disk drive.

For DOS systems, RUSLE can be loaded by copying all the files from the diskette(s) onto the hard disk. These files include the RUSLE executable program, the Database files, the files used to create the Help screens, and some miscellaneous system files. These files should be copied into a directory created specifically to run RUSLE, as this will speed execution and make the program easier to use. Consult your DOS system manual for instructions on creating a directory and on copying files from the diskette drive to that directory.

Copying all the files onto the hard disk should also be done when using the RUSLE computer program from within a DOS window environment. The files should still be copied from the supplied diskette(s) into a directory, although this will be done through use of a file manager. If desired, the windowing package can then be used to link the execution of the RUSLE code to an icon through use of a command line containing one of the commands specified below.

For UNIX systems, the technique for loading the program from the diskette(s) onto the hard disk depends greatly on the system. Consult your system manuals or specialist for instructions.

Running RUSLE

Once the RUSLE files have been loaded into a directory on the hard drive, begin operation by moving into that directory and typing one of the commands listed below. The command [rusle] puts you into the main program and brings up a menu that lists the options for the program. If you want to run only one of the factors or options, simply enter its command. For example, to have access to the CROP Database, the command is [rusle crop]; to run the C-factor subprogram, the command is [rusle c]. The options for the command string are as follows:

[rusle]	to get the RUSLE Main Menu, giving access to all the options
[util]	the three Database Utility programs
[city]	the CITY Database Utility program
[crop]	the CROP Database Utility program
[op]	the OPERATIONS Database Utility program
[r]	R-factor routine only
[k]	K-factor routine only
[ls]	LS-factor routine only
[c]	C-factor routine only
[p]	P-factor routine only

If you choose any option other than [rusle], you will not be able to reach all the other options.

How RUSLE Works

Background

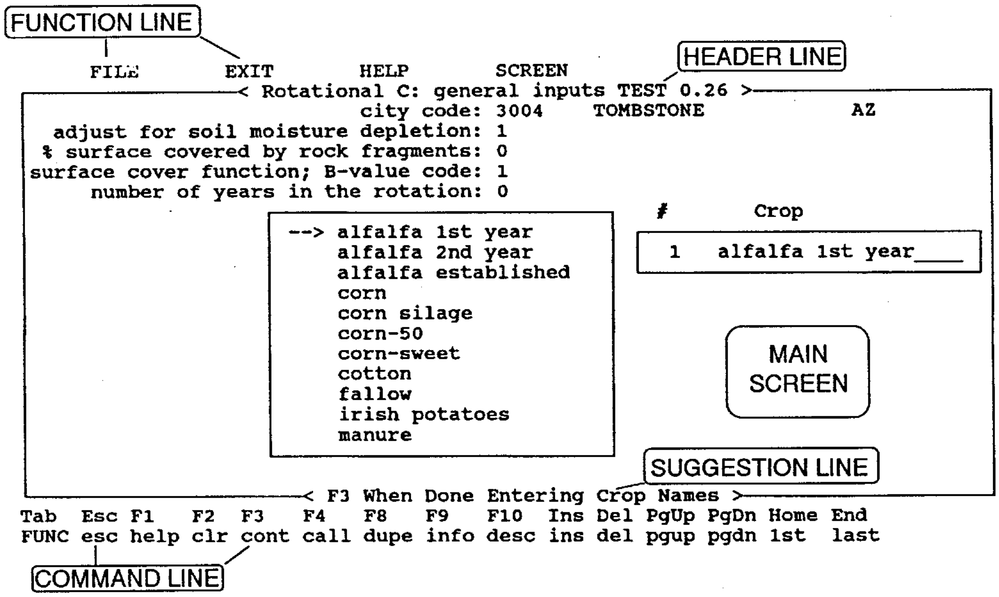
The computer code within RUSLE is responsible only for calculations based on information that you give it; you are responsible for the quality of that information, and most of your time using the program will be spent describing your situation in terms that the program understands.

RUSLE is based on a routine that waits for your response. There are two basic types of responses. The first is to answer the question that the program is asking–a fill-in-the-blank answer. We call this response an "input." The second type of response is a "command," which tells the program what to do next. For example, commands are used to direct the program to carry out the calculations or to move to the next screen. Inputs and commands are often used together. For instance, if the program prompts you for the name of a CROP, you might type in [cotton] as the input, and then give a command by pressing the [ENTER] key.

In most cases, you can respond with either an input or a command, but there are a few cases when only a command is appropriate. For instance, in looking at results, all the program needs to know is when and how you want to move on; the input type of response is not available.

Description of RUSLE Screen

All your interaction with RUSLE will be through screens that have the same general pattern as shown



```
     FILE      EXIT       HELP       SCREEN
 ──────────────< Rotational C: general inputs TEST 0.26 >──────────────────
                    city code: 3004     TOMBSTONE                AZ
   adjust for soil moisture depletion: 1
   % surface covered by rock fragments: 0
  surface cover function; B-value code: 1
       number of years in the rotation: 0
                                                  #        Crop
                   --> alfalfa 1st year
                       alfalfa 2nd year          1   alfalfa 1st year____
                       alfalfa established
                       corn
                       corn silage
                       corn-50
                       corn-sweet
                       cotton
                       fallow
                       irish potatoes
                       manure
 ─────────────────────< F3 When Done Entering Crop Names >──────────────────
 Tab  Esc F1   F2  F3   F4   F8   F9   F10  Ins Del PgUp PgDn Home End
 FUNC esc help clr cont call dupe info desc ins del pgup pgdn 1st  last
```

The screen can be divided into five parts, as follows:

(1) Main Screen

The main portion of the screen is the area in the center, which contains the list of questions and comments. It is here that inputs are entered and results displayed.

(2) Command Line

The bottom two lines of the screen make up the Command Line. Whenever the program is awaiting your response, commands will tell it what to do next. The available commands are listed here on the screen. The upper of these two lines shows which key to press to execute each command; on the lower line below each key is listed a three- or four-letter command description. The section "Giving Program Commands" later in this chapter lists all possible commands and what they do.

(3) Function Line

The top line on the screen is called the Function Line; it defines RUSLE Functions. These are subroutines that perform the housekeeping chores for the program. The FILE Function provides all the options for manipulating Input Files and the Current Input List (see "Input Lists And Files"). The EXIT Function provides options for leaving the program. The HELP Function provides several options for general help on running RUSLE. The SCREEN Function gives the user control over how the screen looks. "The Functions: Program Housekeeping" explains the use of the Function Line.

(4) Header Line

The second line down from the top of the screen is the Header Line, which tells the user the version of RUSLE that is running and the name of the current screen.

(5) Suggestion Line

The third line up from the bottom of the screen is the Suggestion Line, which lists the most likely command or course of action. This displays the suggested response or course of action as you go through the logical sequence of the program.

Input Field

An Input Field is a place where the cursor sits as the program waits for you to answer a question. As mentioned above, your response can be either an answer to the question (input), or a command telling the program what to do next. An input can take any of three forms:

(1) *Filling in a blank.* For example, when the program asks for a CROP name, the appropriate response may be to type [cotton].

(2) *Selecting an option.* The program shows a numbered list of options and asks you to select one. If you have not already chosen one of the options, the entire list will be shown. There are two ways to select an option: Type in the number shown next to the desired option and then press [ENTER] to select it, or use the [ARROW] keys to move through the list and then press [ENTER] to choose the marked option. If one of the options was chosen earlier, the list may not be shown, but the {list} command will display the options again. See "Giving Program Commands" for more information on this command. In some cases the list is always shown, but the marking arrow may not appear within it. In this instance, use of the {list} command will move the arrow into the list.

(3) *Selecting an item from a list.* This works in the same way as the option selection described above, but is based on a list of named items rather than on a numbered list of options. The box in the center of the example screen shown previously contains such a list.

Giving a Command

The second way to respond to an input field is to give the program a command, which tells it what to do next. There are several commands to which the program will always respond, so they are not displayed on the screen. These are the [ARROW] keys and [ENTER]. A list of all possible commands and what they do can be found in the section titled "Program Commands and Controls." Not all of the commands can be used at every place in the program, so the list of commands that are available for use from the current Input Field is shown on the Command Line at the bottom of the screen. A command is given by pressing the key listed on the Command Line.

Calling a Function

One command that is usually available is {FUNC}, which moves the cursor onto the Function Line (the top line of the screen). This allows you to perform the general housekeeping chores shown on that line. These include the FILE routines (see "How RUSLE Gets, Uses, and Saves Information" for an overview or "Input Lists and Files" for complete information), the EXIT routines that allow the user to leave the program, the HELP routines for assistance in using the program (see "Help"), and the SCREEN routine for changing screen colors.

When {FUNC} has been used to move the cursor to the Function Line, use the [RIGHT ARROW] or [LEFT ARROW] keys to move it through the functions. As the cursor moves to a function, the list of associated routines is displayed. [ARROW] up or down through the list until the routine you want is marked, or type in the number of the desired function. Press [ENTER] to select that routine and begin its execution. With the cursor on the Function Line, use the {esc}ape command to return to the Main Screen.

"The Functions: Program Housekeeping" gives more information on calling and using the Function routines.

Getting Help

RUSLE provides two general types of help. The first is additional information on the current input field, describing what sort of input is expected and why. This is always available through the {help} command. The second is made up of general descriptions of RUSLE and its operation, available through the HELP routines of the Function Line.

How RUSLE Gets, Uses, and Saves Information

Background

RUSLE requires a lot of information telling it which calculations to make and what values to use for the variables in each equation. It can get that information from one of three sources: values you enter directly, data that you have stored in Databases, or information that you entered earlier and then stored.

Current Input List

RUSLE maintains a record of all responses to the program questions in a list called the Current Input List. This is updated every time you respond to a question by either entering an input or selecting an option or item. The list does not include the commands given to the program. When the program begins, this list is cleared (everything is set to 0).

The section titled "Input Lists and Files" provides more information on the Current Input List and its manipulation.

Input Files

Entering information by hand is tedious and inefficient. In order to make it easier to enter information and also to provide a record of the values used, you can save the Current Input List into an Input File. Each Input File thus contains a copy of the Current Input List at the time it was saved, along with a unique name and identifying comments. "Input Lists and Files" explains how to save the Current Input List into an Input File, and how to load one of these files back in as the new Current Input List.

Databases

The RUSLE Databases provide another way of entering data more efficiently by associating a large amount of information with a single Identifier. For example, all the weather information for a specific city can be associated with a city code, which from then on serves as the Identifier for all that information. You need to enter the weather information for that city only once; from then on, you can retrieve any part of that information by just giving the Citycode Identifier.

RUSLE contains three Databases. The CITY Database includes all necessary climatic data identified with a specific city or region. The CROP Database provides information on the growth and residue characteristics of specific crops or other vegetation. The OPERATIONS Database defines how field operations affect the soil, crop, and residue and, through those, the erosion rates. General information on using the Databases can be found in the section titled "Databases," and information unique to an individual Database can be found in the sections "CITY Database," "CROP Database," and "OPERATIONS Database."

You should change the information in the Databases to meet your specific situation and needs. Remember that the Databases exist to make it easier to enter the required information but that the users are still responsible for the validity of the information used in their computations. Appendix D provides valuable advice on how to modify the existing core database information to match your specific situation.

PROGRAM COMMANDS AND CONTROL

From any input field within RUSLE, you have the option of either answering the question or giving the program a command. In general, these commands control the flow of the program by telling it when to move on to the next step or screen, telling it that you want to get back out of the current section, telling it to leave the program altogether, and so on.

Giving Program Commands

There are several commands to which the program will always respond, so these are not shown on the screen. One such command is {ENTER}, given using the [ENTER] key ([RETURN] or [NEW LINE] on some machines). This command tells the program to accept the current value as the answer to the question being asked and to move on to the next question. The other commands not displayed are the [ARROW] commands, which are used to move the cursor. When a [RIGHT ARROW] or [LEFT ARROW] doesn't make sense (as in a vertical list of questions), using either [ARROW] will have the same effect as using [ENTER].

The [ARROW] commands have one additional peculiarity. As described in "Input Field," some questions ask you to choose from a list of options. When no choice has been made, moving the cursor onto that question will automatically display the list and put the cursor in the list. Using [ARROW]'s at this point will move the cursor within the list rather than between questions on the main screen. Once you have answered the question, the list will not be displayed unless you ask for it with the {list} command, so the [ARROW]'s will move the cursor on the Main Screen without interruption.

In addition to [ENTER] and [ARROW], there are usually several other commands to which the program will respond. These vary from screen to screen and question to question, but the available commands will always be listed on the Command Line. The Command Line actually comprises the bottom two lines of the screen. The upper line shows which key to use, and the lower line gives a three- or four-letter description of the command. "Description of RUSLE Screen" shows an example of how this appears on the screen.

List of Commands

A complete list of all possible commands and a brief description of what each does is given below. The command description or name is given within braces, { }, and the associated DOS key for each command is shown within brackets, [ ]. The keys may differ on some machines, but the command description is always as shown here.

{FUNC} [TAB] places the cursor onto the Function Line to let you do the required housekeeping chores (see "Calling a Function" or "The Functions: Program Housekeeping"). When the cursor is on the Function Line, use {esc}ape to bring it back to the Main Screen.

{esc} [ESC] allows you to {esc}ape from the current screen or question without giving an answer. This is most commonly used in three places: (1) if you have gotten into a screen or series of screens and want to get back out; (2) after an error or warning message has been displayed, and you want to continue; (3) to continue the program after a result has been displayed.

{help} [F1] shows one or more screens of additional information to help answer the current question (see "Help"). This usually includes a brief description of how the variable is used in the calculations and also suggestions for possible answers.

{clear} [F2] {clear}s (sets to 0) all variables associated with the current screen. This is most useful when you want to get a fresh start on a screen containing several mistakes.

{cont} [F3] {cont}inues program movement to the next logical screen in the sequence. For example, when you have answered all the questions on one screen, this command moves you to the next screen or initiates the calculations. For screens that require only one input or command, the {cont}inue command acts just like an [ENTER], telling the program to accept the current value and to move on.

{call} [F4] {call}s a subroutine on which the answer might depend. Use of this command will automatically put you into the required subroutine; when you exit from that, you will return to the current question.

The {call} command is used in three different instances.

(1) When an answer is calculated from the results of several different factors. An example is in estimation of soil loss from the factor values. Each factor is {call}ed individually by use of this command.

(2) When the question asks for the Identifier of a Database Set, as when entering a city code, a crop name, or an operation name. Use of the {call} from one of these locations allows you to examine or modify the information within the associated Database. "Databases" gives more detail.

(3) When the calculation requires information entered in another portion of the program. An example of this is the value of average field slope, which is used in several places but is calculated within the LS factor. If this value has already been calculated, it will be shown; if not, the {call} command must be used to move to that calculation.

{list} [F6] displays a list from which you can select an option or item, as explained in "Input Field." If you have previously selected one of the choices, the list may not be shown but this command will appear as an option. Use of this command displays the list and allows you to move through it using the [ARROW] keys.

There are places in the program where the list is also shown but the marker arrow is not visible. In these cases, a {list} command will move the marker arrow into the list, where the marker can be controlled with the [ARROW] keys.

{save} [F7] {save}s the data shown on the screen into a Database Set named by the Identifier at the top of the screen. Changes made within the CITY, CROP, or OPERATIONS Database routines will not be saved into those Databases unless this command is used before exiting those routines. If changes have been made, you will be asked whether or not you want to save them before you are allowed to exit.

{dupe} [F8] used only in the crop listing on the initial C factor input screen, this duplicates an entire operation listing screen into another location within the list. For example, if I {dupe} "corn" from the first place on the list into the fifth, the program will duplicate all operations associated with that first corn onto the fifth screen. It will not change the dates, so this must be done by moving to that screen and modifying them individually.

{info} [F9] gives information on the Current Input List. When you {save} a Current Input List into an Input File, you are also saving a series of comments describing that List. Use of this command allows you to look at (and change) the descriptive information for the Current Input List.

{desc} [F10] gives information on the current CROP or OPERATIONS Database Set. These sets also contain a series of descriptive comments, which can be viewed and/or changed from almost anywhere this command is available.

{ins} [INS] used to insert a line of information just above the line on which the cursor is resting.

{del} [DEL] used to delete the line of information on which the cursor is resting.

{1st} [HOME] used to jump to the first in a series of screens or to the beginning of a list.

{last} [END] used to jump to the last in a series of screens or to the end of a list.

{pgup} [PGUP] used to move forward one screen in a series of screens or up one screen in a long list of information.

{pgdn} [PGDN]	used to move back one screen in a series of screens or down one screen in a long list of information.

Giving a Command

A command is given by pressing the key listed above the desired command. This directs the program to immediately execute that command.

THE FUNCTIONS: PROGRAM HOUSEKEEPING

Several components of the RUSLE program are not part of the actual program workings but are needed to make the program easier to use. These are called Functions and are used to manipulate Input Files, to exit the program, to manipulate the screen coloring, or to get additional help. They are displayed on the Function Line, which is the top line on the screen. See "How RUSLE Works" for explanation of a sample screen.

Calling a Function

Functions are called by use of the {FUNC} command to move the cursor to the Function Line. Then use the [RIGHT ARROW] or [LEFT ARROW] key to move the cursor to the desired Function. The list of routines associated with that Function will be shown automatically. Either use the [UP ARROW] or [DOWN ARROW] key to select a routine or type in the option number. Press [ENTER] to execute the routine.

Once the cursor has been moved to the Function Line, you can return it to the Main Screen by using {esc}.

The FILE Function

The most commonly used Function in RUSLE is the FILE Function. From here, the Current Input List (see "How RUSLE Gets, Uses, and Saves Information" or "Input Lists and Files") can be cleared (all values set to zero), it can be SAVEd into an Input File, or it can be replaced by values LOADed from an Input File. As described in "Input Lists and Files," the purposes of this function are to allow you to save information so that you don't have to reenter it later, and to keep a record of the inputs used to reach a specific answer.

The EXIT Function

The EXIT Function provides a quick and easy way to get out of RUSLE from anywhere in the program. This Function is not the only way out (a series of {esc} commands will give the same result), but it is the fastest.

The HELP Function

The HELP Function provides general help in operating RUSLE, including such options as introductory information for the novice user and definitions of the various commands. This Function does not give information about the current question, because that role is filled by the {help} command. "Help" gives more detail on the HELP Function and the {help} command.

The SCREEN Function

The SCREEN Function gives the user some control over the appearance of the screen, including a color option for computers with color screens and a black-and-white option for monochrome screens.

HELP

Two types of help are available within the RUSLE program. The first is help for a specific question, available by using the {help} command. This is meant to provide you with information on how the variable fits into the general scheme of RUSLE calculations and with an idea of the expected type and size of answer. The second type of help available is more general information on the workings of RUSLE, available through the HELP Function.

Help for a Specific Question

It is often difficult to know what sort of response is expected when faced with a specific question. The {help} command provides that information. Help information is available for almost every response within the program, such as a blank to fill in, a list to select from, or even a warning message. If you don't know how to respond, try the {help} command.

Information on giving commands is found in "Program Commands and Control."

General Help

If you need more general information on running RUSLE, you can almost always call in the HELP Function. This provides information on the following:

- guide to RUSLE
    - how RUSLE works
    - principal contributors
- how to use help
- Command keys
    - ENTER
    - ARROW keys
    - Command keys
- guide for first-time user

Information on using the Functions is found in "The Functions: Program Housekeeping."

INPUT LISTS AND FILES

As explained briefly in "How RUSLE Gets, Uses, and Saves Information," RUSLE maintains a set of the current values of all important variables used by the program, and this is called the Current Input List. This List is updated every time one of these variables is changed.

The calculated values of the R, K, LS, C, and P factors and of the average annual soil loss are also saved in the Current Input List. This can cause a situation in which some of the inputs have been changed but new results have not been calculated. The results in the Current Input List may thus be meaningless, because they no longer correspond to the inputs in that List. RUSLE keeps track of such potential errors and gives warnings in the Soil Loss Prediction Table when they occur.

To facilitate the use of RUSLE, the FILE Function has been included to store the Current Input List for later use. When the Current Input List is stored (SAVEd), it becomes an Input File. This File contains all the values from the Current Input List at the time of SAVEing, plus any descriptive comments that you added through use of the {info} command. This File can later be LOADed back into the program, replacing all values in the Current Input List with the values in the File. This capability allows you to use that Current Input List later as a template, changing only the necessary input values before calculating another answer. It also gives you a record of the inputs associated with a particular result.

SAVEing the Input List to an Input File

The Current Input List is SAVEd to an Input File through the FILE Function of the Function Line. Refer to "Calling a Function" or "The FILE Function" for detailed information on the Function Line and on how to call a Function. Briefly, you move the cursor to the Function Line with the {FUNC} command. Because the FILE Function is the first one listed on the Function Line, its options will be shown automatically. Select the SAVE option either by marking it with the arrow (using the [UP ARROW] and [DOWN ARROW] keys) or by typing in the number of the SAVE option. [ENTER] to begin the SAVE routine.

This routine will bring up a list of the existing Input Files. To SAVE the Current Input List to an existing Input File, type in its name or move the marker with the [UP ARROW] and [DOWN ARROW] keys, and then press [ENTER]. *If you SAVE to an existing file, the information previously in it will be lost.*

You can SAVE the Current Input List to a new Input File by typing in a new name, using a maximum of eight characters. All alphanumeric characters (including the underscore, "_") may be used in the name. The program will not allow you to use blank spaces or a period within the name.

It is important that you choose an Input File name that will give you enough information so that you can pick it out of a list, because you may create many of these Files. This is difficult to do within the eight-character limit, but most users have found it helpful to include single-letter descriptions of each crop (for instance, *c* for corn, *b* for soybeans, *a* for alfalfa) followed by a single letter describing the tillage system. As an example of this sort of scheme, we describe a rotation of 4 yr of alfalfa followed by 1 yr of conventionally tilled corn and then 2 yr of no-till corn. One possible name is "4accc2cn," where the *4a* indicates 4 yr of alfalfa, the first *c* indicates that the alfalfa is planted after conventional tillage, the *cc* stands for conventional corn, followed by *2cn*, which describes the 2 yr of corn under no-till. The name you choose depends on which information that you are saving is most important to you. If you are most concerned with location, then the file name should include that. If you are comparing rotations, the location is probably not as important.

Finally, you will be allowed to enter up to five lines of file description, or to modify the comments entered previously through the {info} command. This may include such information as location, rotation, tillage practices–anything that might help you later to identify the file. Once you have entered all the information you want, use the {cont} command to force the actual SAVE to occur.

LOADing an Input File Into the Input List

A new Current Input List can be LOADed from an existing Input File. This allows you to take a File with values close to the ones you want and to change just the variables that differ. This is especially useful when you are making multiple runs under fairly similar conditions.

An Input File is LOADed into the Current Input List by use of the same procedure as for SAVEing, but by choosing the LOAD rather than the SAVE option of the FILE Function. Refer to "Calling a Function" or "The FILE Function" for detailed information on the Function Line and how to call a Function, or to the section immediately preceding this one for a brief description of the procedure.

LOADing an Input File into the Current Input List will replace all values in the List. These values will be lost unless they were SAVEd earlier.

Deleting Unnecessary Input Files

When Input Files are no longer needed, they can be removed using the Delete routine of the FILE Function. Either type in the file name or [ARROW] to the file you want; then [ENTER] to remove it. You can also use the {del}ete command from inside this routine or the LOAD and SAVE routines to erase files.

Using an Input File Created Earlier or Elsewhere

Input Files can be stored and shared with other users. When it is SAVEd, an Input File is stored in the directory in which the program is being run, and the Input File's name is the one you gave it when SAVEing plus a ".rus" suffix. If, for example, you gave the Input File the name "c-b-conv" when SAVEing, it would show up in the directory as the file "c-b-conv.rus."

The File can then be shared by copying it onto a diskette from which other users can copy it into their RUSLE directory. If your Input File has the same name as one already in their directory, they should change the name before copying it. After it is copied into the RUSLE directory, the new Input File will be treated like any of their other Input Files. Refer to your system manuals if you need help in copying or renaming files.

Files created on an older version of RUSLE can usually be used with newer versions of the program. Changes between versions may alter the number, names, and types of variables used by the program, but the program routines can adapt. If the program is looking for a variable that is not found within the old Input File, the variable will be set to zero or left blank. Extra values within the old Input File are ignored. If you are using an Input File created under an earlier program version, go through the program carefully the first time to make sure that all variables are set correctly.

DATABASES

Much of the information needed by the RUSLE calculations comes from a group of three Databases. The CITY Database contains information on climate, the CROP Database holds the parameters defining the characteristics of vegetative growth and residue, and the OPERATIONS Database defines the effects of field operations on the soil, crop, and residues.

Databases provide a way of associating a large amount of data with a single name or number, called the "Identifier." For instance, the CITY Database takes all the information necessary to describe for RUSLE the weather of a specific city or area, and associates that information with a single code number. One Identifier (city code, crop name, or operation name) is assigned to each Database Set, which includes all associated information.

In most cases, the data in each Database are independent of other available information. For example, the CITY Database's weather information is not likely to be affected by which crop is grown or which field operations are used. This independence holds to a lesser degree for the OPERATIONS Database and CROP Database. For example, a crop's growth characteristics will change with weather but likely will not vary as much from location to location as does weather. A corn Database Set may well apply to a fairly large part of the midwestern United States, although the weather within that area may vary considerably.

One key to using the Databases is to decide how many different Sets are required. For example, for how large an area of the Midwest are a set of corn-growth characteristics appropriate, and when will you have to add another CROP Database Set to show differences? How many CITY Database Sets are required to describe an area, given its particular weather patterns? How significant do crop varietal differences have to be to justify another CROP Set? You must answer these questions, and your responses will likely be based on the available data and the difference these changes seem to make in the final results.

Validity of Default Database Information

The values given in the default CROP and OPERATIONS Databases supplied with the program are supported by published literature cited in chapter 5. These values generally apply to the specific combination of location, crop variety, expected yield, row spacing, planting density, tillage practices, equipment speed, soil conditions, and so forth, used in the studies. The values should therefore be thought of as typical base values designed to give general guidance.

You, the user, are ultimately responsible for ensuring that the information within the Databases fits your local conditions. In general, you should modify the values in the Databases so that the results match your conditions. For instance, if you have seen in field observations and measurements that your soybean residue decays more quickly (or more slowly) than that predicted by RUSLE, you should change the residue decay parameter in the CROP Database until the model matches the information you have. The default Databases provide a strong basis in actual measured values, but these default values should be modified to reflect measured values under your specific conditions. Appendix D describes how to carry out these modifications.

The CITY Database is somewhat different, because the default Database values are based on the same information you are likely to find. You may, however, need to create additional Sets to describe your local conditions.

Calling the Database Utility Routines

The Database Utility routines are used to manipulate the information stored in the Databases. One way to reach these routines is by selecting the Database Utility from the RUSLE Main Menu. More commonly, you can get into the Utility routines for a specific Database from anywhere the Database is used. You can {call} the CITY Database routines from wherever you are asked for a city, the CROP Database routines from wherever you enter a crop name, and the OPERATIONS Database routines from wherever you give a field operation name. Practically speaking, you can {call} the CITY routines from the R, K, C, and P factors, whereas the CROP and OPERATIONS routines can be {call}ed from the input screens of the C factor. "Program Commands and Control" gives details on using commands such as {call}.

When you return from the Utility routines, the program should put you back at the same Input Field from which you {call}ed the routines.

Using the Database Utility Routines

The CITY, CROP, and OPERATIONS Databases are created and maintained by four major subroutines known as Utility routines. There are routines to edit the Database Sets or to create new ones, to print Database Sets, to delete unnecessary Sets from a Database, and to bring in (merge) new Database Sets.

Editing Existing Sets or Creating New Ones

The Edit routine is the most commonly used of the Database routines, and serves to review or revise an existing Database Set or to create a new one. After you choose this option, the program will respond with a list of available Sets, listed by Identifier. Select the desired Database Set by either typing in the Identifier or by [ARROW]ing to the one you want. Then press [ENTER]. The program will respond with a screen of information associated with that Identifier. Use {esc} to return from this option.

To edit an existing Database Set, move the cursor around on the screen to make the changes and then save the changed Set into the Database with the {save} command. See "Program Commands and Control" for how to give commands. Note that the new values will not be kept unless the modified Database Set is specifically {save}d into the Database.

The most efficient way to create a new Database Set is to begin with an existing Set that is similar to the one you want. Type in the new Identifier and then change only the values that need to be changed. To keep the new Database Set, {save} it before you {esc}ape from this Utility routine.

This routine can also be used to rename Database Sets. Call up the Set to be changed, then create a new Database Set by changing the Identifier and {save}ing. This gives you two Sets that are identical except for the Identifiers. You can then bring up the list and {del}ete the original Database Set as described below.

To delete an unnecessary Database Set, type in the name of the Set or use the [UP ARROW] and [DOWN ARROW] keys to move the marker to the Set Identifier; then give the {del}ete command. You will be asked to confirm this request.

Printing Database Sets

The second Utility routine is used to print information in the Database. This gives two options: The first option prints a list of available Sets within the Database, and the second option provides a complete printout of all information associated with a chosen Identifier. If, for some reason, the program cannot find an attached printer or cannot send the information to the printer, a warning will be displayed on the screen.

Background on Database Files

The RUSLE program uses three different computer files for each Database. The Original File contains only those Database Sets supplied with the program. The program will read from this file but cannot write to it, so it remains unchanged. The Working File is the one that you actually use. When you {save} a Set, it is copied here, and the list of Identifiers you usually see is made up from the Sets in this file. Finally, the Delete File contains all sets you didn't need and removed from the Working File. This serves as a file of last resort, allowing you an opportunity to bring back files that you wrongly deleted.

Every time the program is run, it checks to see if a Working File exists. If not, it creates one that is a clone of the Original File. If a Working File does exist, the program leaves it alone; this allows you to put new program versions in the same directory without destroying your Working File.

Deleting Unnecessary Database Sets

The speed and efficiency of the RUSLE program is improved by deleting extraneous information. Since Database Sets are designed to fit local conditions, it is likely that you will want to remove Sets that cannot possibly apply to your conditions. When one of the lists of Database Sets is shown, or when you are within the Create/Edit option of the Database Utility routines, you can do one of the following: Use the [ARROW] keys to mark the correct Set in the list, or type in the name of the Database Set, or move into that Set as if you are going to Edit. Giving the {del}ete command will then bring up a confirmation message, to prevent accidental removals. If you confirm your intentions, the Database Set will be removed from the list. As mentioned in the section below, it is still possible to recover that information.

You may also {del}ete Sets more quickly by using the Delete Database Utility routine. This option is faster because the program does not prompt you to confirm your decision to remove Sets.

Because of the large number of CITY Database Sets and because they are divided by state, the CITY Database Utility contains a special routine that allows you to remove large numbers of Sets from the CITY Database by indicating the names (abbreviations) of the states whose cities you wish to keep. It is then possible to {del}ete specific cities within those states by the procedure described above.

Sets {del}eted from the Working File are simply moved into the Delete File, but Sets removed from the Delete File are permanently deleted.

Restoring Information Deleted From a Database

You can use the "bring new Sets" routine of the Database Utilities to recover Database Sets that were mistakenly {del}eted. Simply specify that you want to bring the Sets in from the list of deleted Database Sets.

Information will be lost if you {save} to an existing Database Set. For example, if you make changes to the "corn" CROP Database Set and then {save} it, the information previously there will be lost, and will be replaced by the new information that will now be stored in the Working File. If there is any chance that you may want to keep this old information, either give the new Set a slightly different name or first {save} the old Set under a different name.

Using Database Sets Created Earlier or Elsewhere

The "bring new Sets" routine of the Database Utilities can also be used to copy Sets created on another computer or by another user. The procedure to do this is as follows:

(1) Before entering RUSLE (from the operating system):

(a) You need to know that the names of the Working, Original, and Delete Files end with the ".dat," ".org," and ".del" suffixes, respectively. The prefixes are "croplist," "oplist," and "citylist" for the CROP, OPERATIONS, and CITY Database Sets, respectively. If you want to copy a CROP Database Set, for example, you are looking for the "croplist.dat" file.

(b) Copy that file into another with a different name and with no suffix.

(c) You need to make this available to your machine. In most cases, this means having a copy of the file on a diskette, but it can also mean having access to it over a network.

(d) Copy that new file into your RUSLE directory. Do NOT copy "croplist.dat" into your RUSLE directory under that name, because doing so will erase your "croplist.dat" file.

(e) Refer to your system manuals for assistance if you have questions about copying files.

(2) After entering RUSLE:

(a) Enter the Utility routines for the Database you want to merge.

(b) Begin the "bring new Sets" routine.

(c) Enter the name of the file from which to merge; this is the name you gave it in step 1.b above. The routine will bring in the Sets within that new file. If a Set in the new file has the same Identifier as one already contained in your Database, you will be asked which to keep.

USING THE RUSLE SOIL LOSS PREDICTION TABLE

Although the individual factors and routines of RUSLE can be run separately, the heart of the erosion prediction package is the RUSLE Soil Loss Prediction Table. This table shows the R, K, LS, C, and P values calculated from the Current Input List as well as the annual soil-loss estimate (A) in tons · acre^{-1}.

Each line of the table corresponds to a single set of inputs. For instance, the top line may contain the results of a conventional tillage rotation on a specific field, the next line may be for the same rotation but using reduced tillage, and the third and fourth lines may examine what happens with different rotations on the same field. As this example shows, one main purpose of the table is to show the effects of alternative management systems, although the table may also be used to make other comparisons.

Relationship of Input Lists and Files to Lines of the RUSLE Table

NOTE: See "How RUSLE Gets, Uses, and Saves Information" and "Input Lists and Files" for a description of the Current Input List and of Input Files.

Each line of the RUSLE table corresponds to a set of inputs. When you enter the table, RUSLE creates a temporary Input File for each table line. This permits you to move freely between lines. The values from the Current Input List will be automatically SAVEd into the temporary Input File for the line you are leaving, and the values from the temporary Input File for the new line are automatically LOADed into the Current Input List.

You can also SAVE the Current Input List into an Input File. This replaces the temporary Input File assigned to that table line with the permanent Input File given whatever name you have assigned.

If you have made changes to the temporary Input Files but have not SAVEd these, the program will warn you of this before allowing you to exit.

You can move the cursor between factors on a single line without changing any values. To compute a factor, move the cursor to that column and use the {call} command. When the computations are complete, the new factor value will be shown on the table.

Entering Values Directly Into the RUSLE Table

You can also enter values into the RUSLE Soil Loss Prediction Table directly. For example, if you know that the R Factor associated with your location is 120, you can type that number in the R Factor column in the table. The program will use any combination of typed and computed results to calculate an estimated annual erosion rate.

Warning Footnotes on the RUSLE Table

Because of the complicated relationships between the RUSLE Soil Loss Prediction Table, the Current Input List, and the Input Files, RUSLE keeps track of potential problems and gives warning messages. These show up as flags placed near the values in the RUSLE table and as warnings in footnotes to the table.

In general, these warnings indicate that the factor values in the table may no longer correspond to the numbers in the Current Input List, which can happen in three instances: (1) if you go into one of the factor routines and make some changes but do not carry through with the calculations, (2) if you enter a factor value directly from the keyboard, and (3) if information used several places in the program is changed in one location without being changed everywhere. For example, the city of interest can be specified within the C factor, but the information is used in the C, R, K, and P factors. If the citycode is changed only in the C factor, the results shown in the other factors may no longer correspond to the Current Input List (with its new citycode). To resolve this, move to each of the other factors and use the {call} command to perform the calculations. A similar difficulty arises when the field slope is changed, because the R, LS, and P factors all use this value.

SPECIFIC GUIDELINES AND ANSWERS TO COMMON QUESTIONS

CITY Database

The CITY Database contains all the climate information used in RUSLE. This information is divided into Database Sets, each of which represents a specific location. You, as the user, must determine the size of area to which that information applies. Areas where climatic patterns change quickly with distance will require more Sets (for instance, in the mountainous regions of the western United States).

Each Set has a unique name called the Citycode Identifier. This is an integer number unique to that Set and may range from 1 to 99,999. Citycode Identifiers may be assigned in any arbitrary order, although a specific ordering scheme is used within the default Database. Each CITY Database Set also has a Block Identifier of two upper-case letters. This is used to group the cities into blocks. In the default Database, the state abbreviation is used.

"How RUSLE Gets, Uses, and Saves Information" gives some background, and "Databases" gives specific details on changing and manipulating this information.

Answers to Common Questions About the CITY Database

How do I name/number a new city?

(1) Select a Citycode Identifier. Although any numbering or naming scheme may be used, the one for the default CITY Database is recommended because changes in the numbering scheme may make it more difficult to share your Database Sets with others, or to incorporate their Sets into your program (see "Using Database Sets Created Earlier or Elsewhere"). The default Database uses a five-digit Citycode Identifier for each Set.

(a) An alphabetical list of states is numbered from 1 (Alabama) to 50 (Wyoming), and the District of Columbia is assigned number 51. The first two digits of the five-digit number indicate the number of the state within that list. Under this scheme, cities in Arizona (the third state in the list) have Citycodes of the form 3XXX (same as 03XXX), and cities in number 14 (Indiana) look like 14XXX.

(b) The three rightmost digits refer to the cities within that state. The default scheme places cities within a state in the order in which they were created, with the first city given the description 001, the next 002, and so on.

(2) Type in the name of the city or area. The name associated with a CITY Database Set is by no means limited to the name of an actual city. In fact, the Set is likely to be applied to a larger area, so the name can be that of a county or anything that identifies the Set. In the default Database, this is the city by which the original weather information was identified.

Note: The default scheme uses the two-letter state abbreviation as the Block Identifier.

When is the Equivalent R value used? Much of the erosion in portions of the Pacific Northwestern Wheat and Range Region is the result of rainfall and runoff on frozen or thawing soils, and erosion rates in that area far exceed those predicted using the standard R values. Also, the distribution of this erosion over the year does not match that expected from the standard EI distribution. These considerations make it necessary to use Equivalent R values to describe the erosivity, rather than the standard values of R.

Within RUSLE, the EI distribution you select for a city determines whether the equivalent R can be used. Figure 2-7 of chapter 2 shows all the distribution areas. The program currently recognizes EI distribution areas 6-10, 14-18, 20-22, 29-41, 45, 58-60, and 63 as those for which the Equivalent R value can be used. If you select one of these areas, you are given the choice of using either the standard EI distribution for that area, a "frozen soil 95-5" default distribution that researchers have found works well for the area, or your own distribution. If you choose one of the options containing a "frozen soil" distribution, an equivalent R value will need to be chosen from figures 2-13 through 2-16 of chapter 2.

How do I select and enter an EI distribution? The EI distribution defines how the precipitation energy-intensity varies over the course of the year. Every area within the continental United States has been assigned a standard distribution, numbered as shown in figure 2-7.

In general, two options are available in the selection of an EI distribution: (1) use of one of the standard distributions, with the number selected from figure 2-7; or (2) creation of a new distribution to meet specific local conditions. If the latter is done, the distribution will not be given a number, as it no longer corresponds to the numbering in the figure. Instead, it will be given the label "NEW." You then need to enter the EI values for the new distribution.

The exception to the rule is for the cities of the Northwestern Wheat and Range Region, where it is possible to include the effect of rainfall and runoff on frozen or thawing soils. In this case, four options are presented: (1), (2) the options mentioned above; (3) use of the standard 95-5 frozen soil distribution, which will be given the label "REQ"; and (4) manually entering a new frozen soil distribution, given the label "NEW REQ." The use of either of these last two options requires selection of an Equivalent R value.

For what is the city elevation used? The CITY Database Set includes an entry for the city elevation. The elevation is for information only and is not currently used within the RUSLE program.

CROP Database

The CROP Database contains all the information on growth and residue for the vegetation of interest. The information is divided up into Database Sets, each representing a specific crop or plant community. You must decide how many Sets are required to adequately reflect the differences caused by region, variety, or crop stress. This decision can be made only by noting the sensitivity of the outputs to differing inputs.

The information in each Set is associated with a Crop Identifier, which may be any name up to 20 characters long. This may be anything that makes the crop unique and describes it in a way that is meaningful to you. You cannot use some special characters reserved by the program or the operating system, but the program will warn you of these or will not allow you to enter them.

"How RUSLE Gets, Uses, and Saves Information" gives some background, and "Databases" gives specific details on changing and manipulating this information.

Answers to Common Questions About the CROP Database

How do I choose a vegetation/land use category? The crop category is critical in deciding how RUSLE gets and treats the crop information. The possible categories and what they mean to the program are:

(1) *time-varying vegetation:* This category is used for all vegetation where seasonal or cultivation effects cause significant changes in root mass,

canopy cover, or canopy height. This should be used for all cases except where the changes are so small or gradual that they can be ignored.

(2) *time-invariant vegetation:* If there are few seasonal changes in cover, residue, or root mass at the site, choosing this category greatly simplifies calculations. In this case, you need to enter only the average annual values. The use of this information is described in more detail in "C factor."

RUSLE uses these categories in displaying lists of crops. For example, in the calculation of an average annual C-factor value, RUSLE will show the list containing only the crops designated as "permanent"; when asking for a time-varying vegetation, RUSLE will show a list containing only those.

How does the database information control residue levels? Vegetative residue is a basic component of erosion control, and the information in the CROP Database Set determines the amount of residue and how it behaves. Each Set contains residue-decay rate parameters, which are used along with weather data to estimate the decay rates of the surface and subsurface residue.

RUSLE keeps track of the amount of residue by weight, although its effectiveness in controlling erosion is computed as a function of the percentage of cover. To convert between the two, each CROP Database Set contains as many as three values defining the relationship between cover and weight. You must supply at least one of these values, but giving two or three values will yield better results. Each supplied value is used to define a relationship between cover and weight; if you give more than one value, the relationships are averaged.

RUSLE contains routines that make it easier for you to decide how much residue has been added to the surface by a harvest operation. Within the CROP Database you enter the harvest yield and a few other constants, from which RUSLE will calculate a residue weight.

Surface residue is added to the field only by field operations (see "OPERATIONS Database") or by senescence, which is described in the next section. Surface residue is removed only through decay or by a field operation. Subsurface residue, on the other hand, can be added in one of two ways: (1) by the burial of surface residue during a tillage operation, or (2) by the death of the root biomass of vegetation.

The RUSLE program looks for two possible scenarios in adding root biomass to the subsurface residue. The first of these occurs when the vegetation is completely killed, as might occur with tillage or application of a knock-down herbicide. The second scenario occurs whenever the CROP Database Set indicates a drop in live root biomass, which the program takes as an indication of the conversion of that much root biomass from live roots to subsurface residue.

What is the crop senescence option? In RUSLE, senescence is defined as leaf loss after the plant has reached maturity. The senescence option in the C-factor calculations calls on the CROP Database Set to provide information that may not be readily apparent. If this option is chosen, the crop is assumed to add residue to the surface when the leaves fall. RUSLE handles this by treating a decrease in canopy cover as a similar increase in surface cover. For example, if the canopy cover decreases from 90% to 75%, the total weight of surface residue is presumed to increase by an amount equal to the weight that would give 15% surface cover as defined by the cover-weight relationship.

Do I have to enter all those values for root mass, canopy cover, and canopy fall height? You do not have to enter values for root mass, canopy cover, and canopy height for the entire year. The program searches for the last value of root mass greater than zero and then assigns all remaining root mass, canopy cover, and fall height to the values they have at that point.

How do I handle crops with growth cycles lasting longer than a year? There are two ways of handling a crop whose growth cycle lasts longer than 1 yr: (1) Leave it as a single crop. When the crop goes beyond 1 yr, the program will continue to use the last values of root mass, canopy cover, and canopy height for the rest of the time. (2) Call in a regrowth file, either under an operation within the original crop or as a completely separate crop.

OPERATIONS Database

Field operations are important to the RUSLE program in how they affect the soil, vegetation, and residue. Operations disturb the soil, begin vegetative growth, kill the vegetation, add residue to the surface, or incorporate residue. They may also affect the way the vegetation grows.

This information is divided into Database Sets, with each Set representing a specific operation. You must decide how many Database Sets are required to adequately reflect the differences in the type, speed, and method of use of implements. This decision can be made only by noting the sensitivity of the outputs to differing inputs.

The information in each Set is associated with an Operation Identifier, which may be any name up to 20 characters long. You cannot use some special characters reserved by the program or the operating system, but the program will warn you of these or will not allow you to enter them. The Operation Identifier may be anything that makes the name of the operation unique and that describes it in a way that is meaningful to you.

The OPERATIONS Database is currently used only in the C-factor calculations.

Answers to Common Questions About the OPERATIONS Database

What are the "effects"? The information in an OPERATIONS Database Set consists of a list of effects of the operation on the soil/vegetation/residue system and any additional information required to define them. The program goes through the effects in the order in which they are given (first #1, then #2, and so on) and calculates the impact of the effect on the soil, vegetation, and residue.

RUSLE allows up to five effects for any field operation, and these are chosen from a list of nine possible effects. If a field operation has more than five effects, these may be split between two operations scheduled to occur on the same day.

The program does not allow you to have more than one residue addition or removal in an operation. For instance, you may not add both residue from the current crop and some other residue within a single operation. This limitation exists because you can enter only a single number to tell the program how much residue you are adding. This also means that you cannot add and remove residue within the same operation.

The nine possible effects are listed below, along with a brief description of their place in the program and the calculations.

(1) *no effect:* The program requires five effects. Actual operation effects are listed first, and any remaining spaces are filled with this null value.

(2) *soil surface disturbed:* A field operation disturbs the soil surface in ways that affect erosion rates: (a) disturbing the soil is the only way to incorporate some of the surface residue; (b) the surface of the soil is loosened, which changes the degree of soil consolidation seen in the prior-land-use subfactor of the soil-loss-ratio (see ch. 5); (c) the surface roughness is altered, changing its impact on erosion rates. Including this effect in the list for an operation automatically brings forward questions that are used to define these changes.

(3) *current crop residue added to surface:* This effect specifies residue from the current crop, which means that the residue and decomposition parameters are taken from the current CROP Database Set. If a regrowth crop has been called in, the parameters are taken from there. If there is no current crop, the program adds residue from the most recent current crop.

The program will ask you to define in one of two ways how much residue is added to the surface. If the operation is a harvest operation (see below), the program will automatically pull in from the CROP Database Set the amount of residue added at harvest. If the operation is not defined as a harvest, you must enter the amount of residue added when you fill in the list of operations within the C-factor inputs.

(4) *other residue added to field:* This effect is used to model the impact of material coming from other sources, such as straw mulch used on a vegetable crop or manure spread on a field. This effect brings up questions asking you to define the cover and decomposition parameters for that residue, as well as what percentage of the applied material is left on the surface.

Because the only way to apply a material to the subsurface region (as with a manure injection) is to disturb the surface, this effect must be followed by a (2) within the same operation. When this is done, the percentage of residue buried by the (2) will be for the residue on the surface before the operation. The (2) effect will not bury any of the residue added to the surface by this operation.

(5) *residue removed/added to field:* This is the only effect that considers the removal of residue from the field. You need to specify whether it is from only the current or most recent crop (for instance, baling of corn stover) or from all previous crops (such as burning of residue). Within the list of operations in the C-factor inputs, you will need to specify how much residue is removed.

(6) *current crop harvested:* Information in the CROP Database Set for each crop indicates how much residue is added to the surface when the crop is harvested. Use of this effect automatically brings in that value as residue added to the surface. Refer to the earlier description of effect (3) for more information.

(7) *crop growth begins:* This effect tells RUSLE to begin the growth cycle found in the CROP Database Set for the vegetation listed at the top of the C-factor input screen, and to set "day 0" in that vegetation's cycle as the date of this operation.

(8) *current crop is killed:* This effect does two things: (a) the canopy cover and fall height are automatically set to 0 because the crop is no longer growing, and (b) the root mass is converted into a subsurface residue and begins to decay. The crop can be killed without adding residue to the surface, but from that point on there is no canopy effect.

(9) *call in a new crop growth set:* This effect begins growth of a regrowth crop whose name you are asked to specify when you define the cropping sequence and field operations. This effect should be used when growth patterns change because of an operation or weather, or simply to bring in a new crop as the current crop. The growth, decomposition, and weight-cover values are all replaced by those of the new growth Set, as are the root mass and canopy values.

In what order should I enter the effects? The effects of an operation must be entered in the order in which they occur. For example, operations that add residue either before or after disturbing the soil will give different results. Adding the residue before the soil disturbance will incorporate some of the residue, whereas adding it afterward will not.

How are residues buried or uncovered? The "% left" variable is defined as the percentage of original residue left on the surface following the operation. This can be defined as a "% wt", which is the percentage of the original surface-residue weight. If defined instead as a "% cov", it is the percentage of surface cover left after the operation. For example, if the residue covered 50% of the surface before the operation and the operation leaves 90% cover, the cover after the operation is 50% · 90% = 45%.

If there is more than one residue type on the surface, it is assumed that the specified "% cov" refers to each residue type individually and not to the overall cover. For example, if before the operation there was on the surface a weight of wheat residue equivalent to 60% cover plus a weight of soybean residue equivalent to 40% cover, then after an operation leaving 30% cover there would remain on the surface weights of residue equivalent to 18% and 12% cover for the wheat and soybeans, respectively. Depending on the specified residue characteristics, this might well yield an overall percent cover not equal to 30% of the original cover.

You can also use this term to uncover buried residue by specifying that more than 100% of the residue be left on the surface. For example, if before the operation the surface has 50% cover and the operation leaves 120%, the surface cover after the operation is 50% • 120% = 60%. The program is set to limit the total amount of residue cover to 99.99%, and will not bring up more residue than actually exists. The option of uncovering residue can be used with either the percent weight or percent cover options.

R Factor

The RUSLE R factor defines the total annual erosive potential that is due to climatic effects. This factor reflects the impact of geographical location on erosion, including such factors as localized impacts of lakes or mountain ranges and the dominance of frontal or cyclonic activity.

Options

Initial R-factor value. The first option in selecting an R value is to take it directly from the isoerodent maps of figures 2-1 through 2-5 of chapter 2. The R values in the default CITY Database Sets are taken from these. This information is currently available for all locations in the contiguous United States and Hawaii.

Adjusted R value. For fields with very low slopes and in areas with high rainfall, the R value will be modified to reflect the absorption of raindrop impact energy by ponded water. Note that this changes the R factor to a value that *applies to only* a specific field rather than to a general geographical region, and does not change the R value in the CITY Database Set.

You are asked to indicate whether or not you want this correction. If your field has a very rough surface or has moderate-to-high ridges, more of the surface will be exposed to raindrop impact, thereby increasing detachment. Under these circumstances, do not use the adjustment.

Equivalent R value. As explained in "Answers to Common Questions About the CITY Database" and chapter 2, erosion in parts of the Northwestern Wheat and Range Region far exceeds the amount predicted by the simple R value because much of the erosion occurs on frozen or thawing soils subjected to gentle rains but erosive runoff. If the CITY Database Set you specify was defined as being for such an area, the Equivalent R value can be entered. You are not required to enter this value, because you are not forced to have this effect for cities in this area. "Answers to Common Questions About the CITY Database" explains how to use this effect.

K Factor

The K factor of RUSLE defines the soil erodibility under a set of standard conditions.

Options

Seasonal K value. The seasonal K value attempts to include the effects of freeze-thaw cycles and other factors affecting the temporal variation of soil erodibility. This option requires an original estimate of K, which may be either entered directly or calculated through use of the soil-erodibility nomograph.

Volcanic K value. Data collected in Hawaii for volcanic soils have shown a somewhat different relationship between soil properties and erodibility. If this option is used, K is calculated for the entered data. No seasonal variation of K is needed for these soils, because they are in general not subject to freeze-thaw cycles. These data have not been tested for soils outside Hawaii.

Answers to Common Questions About the K factor

What are the dates of maximum and minimum K for the seasonal K option? The time-varying K calculations yield a K value for each half-month period through the year, with the calculations made at the middle of each period. The date shown for the maximum or minimum K may not correspond to the period with the highest or lowest value if that date is very early or very late in the period.

Why can't I use the time-varying K for the western United States? As described in chapter 3, the algorithms used to calculate the time-varying K work well for the eastern United States, but not for the area west of approximately longitude 105°W. The RUSLE program determines your location by use of the EI distribution zone number, and will not allow the calculations for any areas west of that line.

How is the K nomograph used? The calculation of seasonal K variability uses an original estimate of K. If a Soil Interpretation Record exists, you may enter the K value from it as your estimate. The other option is to develop an initial estimate of K using the K-nomograph method. You get access to the K-nomograph method by {call}ing the subroutine from the Input Field for the original estimate.

What is the soil hydrologic group? Because the soil hydrologic group is a soil property, it is entered within the inputs for the K factor even though it is not used in the K-factor calculations. It is used solely within the P factor to indicate the effect of runoff on support practices.

How is the # years to consolidate used? The C factor also requires the length of time it takes for the soil to fully reconsolidate in the calculation of the PLU subfactor. Since this is primarily a soil property, it is entered here in the K factor.

How is the % surface covered by rocks used? RUSLE treats rocks on the surface as surface cover rather than through their impact on the K factor. This value is therefore not used in the K-factor calculation, but rather is called in from here for the C-factor routines. However, since surface rock cover is generally a soil characteristic, this input is included here with the other soil inputs.

LS Factor

The RUSLE LS factor accounts for the effects of slope length and slope steepness on soil loss.

Answers to Common Questions About the LS factor

Which LS table should I use? The selection of an LS table depends on the condition of the soil and its susceptibility to rill erosion, presented as a ratio of rill to interrill erosion rates. This susceptibility can be a function of either the innate soil properties (such as texture, aggregation, and structure) or the degree to which the soil is modified by mechanical disturbance. In general, a disturbed soil shows a higher rill-interrill erosion ratio than does an undisturbed soil.

Table 4-1 is used for soils with low rill-interrill erosion ratios, usually including those not disturbed for some time. Table 4-2 applies to soils with a moderate percentage of erosion coming from the formation of rills, including soils that are disturbed relatively frequently. Most agricultural soils fall into this category.

Table 4-3 is used for soils that undergo high degrees of rill erosion, including highly disturbed soils such as those on construction sites. This category also contains agricultural soils that by their nature are susceptible to large amounts of rilling.

Finally, table 4-4 contains LS values for soils subject to thawing, runoff from snowmelt, and rain on frozen soil or snow. This changes the importance of the LS factor in relation to overall erosion rates. In general, these values will be used only for slopes in the Northwestern Wheat and Range Region.

What is the equivalent slope? Several routines within RUSLE require an estimate of the average steepness of the downhill slope, called the "equivalent slope." This estimation is easy for a uniform slope because uniform slope = average slope = equivalent slope. For a complex slope, RUSLE defines the equivalent slope steepness as the uniform slope steepness that gives the same LS value as that calculated for the complex slope. For example, if the complex slope has an overall length of 200 ft and an overall LS = 2.34, the equivalent slope steepness is that for which a uniform slope of 200 ft has an LS = 2.34.

The equivalent slope is calculated automatically in the LS routines. When needed by other parts of the program, these calculations may be {call}ed from there.

How do I get LS printouts? The Print option of the LS factor gives a printout of the tables showing LS values corresponding to a broad range of uniform slope lengths and steepnesses. The output tables are 132 characters wide and do not fit on most printers (which use 8½" x 11" paper), unless the printers can be specially configured to do this. Check your printer manuals.

C Factor

The RUSLE C factor describes the effects of cover and management on average annual soil loss.

Options

Time-varying vs. continuous. The C-factor calculations are used to determine the soil-loss ratio (SLR) subfactors for half-month time periods over which conditions are assumed to remain constant. The exception to this is cases of continuous pasture, meadow, or rangeland, for which conditions are likely to change very little over the course of an entire year. For these cases it would be of little value to make the calculations every half month, because the numbers would change only slightly.

The time-varying option must be used whenever the cropping or plant community changes significantly over a year, or when field operations disturb the soil or plant residues.

Single disturbance vs. rotation. Within the time-varying C option there are also two very different options. The first assumes that there is a single disturbance of the system, with subsequent long-term restabilization. You therefore need to define for the program the condition of the soil, vegetation, and residue immediately after the disturbance, and how the site changes over the years as the soil reconsolidates and vegetation regrows. This alternative fits best for construction sites, mine spoils, or many of the mechanical rangeland improvement techniques.

The single-disturbance option is selected by placing a zero (0) in the field for the number of years in the rotation. This brings up additional questions concerning the final surface roughness expected for the site, since this number will be a function of the expected long-term vegetation. For the rotation option, it is assumed that there is repeated soil disturbance, which keeps this natural long-term roughness from ever having a significant influence.

The other alternative is to assume a rotation, where the operations are repeated in a cycle through many years. For example, the list of operations for a corn-soybean rotation would be only 2 yr long, but this list would be expected to repeat itself every 2 yr. If the rotation option best fits the system you are describing, the RUSLE program runs through the calculations three times. The program uses the results from the first two times as the initial conditions for the third run. The rotation option is chosen by specifying the number of calendar years in the rotation.

Concepts of the Time-Varying C Factor, With Examples

At first glance it may appear complex, but the time-varying option for calculating a C factor is simple if several basic concepts are understood. Since these concepts are crucial to an understanding of the power and flexibility of the C-factor calculations, they are illustrated through the use of examples. *Values and Database Sets included in the examples are meant solely as illustrations.*

Important parameters in the time-varying C option. The time-varying C option is based on a listing of the field operations, which in turn call in information from the CITY, CROP, and OPERATIONS Databases. The combination of information from these sources defines the changes in crop root mass and canopy cover, in soil roughness and consolidation, and especially in surface and subsurface residue.

Defining the current crop in the time-varying C option. The RUSLE program keeps track of only one set of CROP Database parameters at a time: root mass, canopy cover and height, and residue amount and decay variables. These values are taken from a single CROP Database Set, and the Set in use at a specific time is referred to as the current crop. When a CROP Database Set becomes the current crop, the program automatically pulls in all the information associated with that crop name, replacing the root mass, canopy, and residue values that had been linked to the previous current crop.

There are two ways to identify the current crop, and both of these require the use of a field operation. The first option is to have a planting operation, which instructs the program to install as the current crop the Set named at the top of the operations list screen. The second option is to have a field operation that asks for a regrowth crop. The program will prompt you for the name of that new Set, which it then pulls in as the current crop.

Note that the crop listed at the top of the screen never becomes the current crop unless an operation tells the program to begin its growth. In spite of this, when the results are shown, they will be displayed under this crop name. This means that the crops you list on the initial C-factor screen and that are shown atop each operations list screen serve two purposes: (1) they are used in accounting, defining over what time period the SLR values are summed; and (2) if called on by a planting operation, they can become the current crop.

It is essential that you keep track of which is the current crop, as this defines all the crop and residue parameters. For instance, unless you specifically direct the program otherwise, any residue added to the surface will have the decay and cover characteristics associated with the CROP Database Set for the current crop.

Defining a Set as the current crop causes one additional hidden impact. The date of the operation defining a new current crop (either a planting operation or one requesting a new Set) automatically becomes day zero in the growth cycle in that Set.

Use of regrowth crops within the time-varying C option. Some events drastically alter the growth patterns of the vegetation without killing it completely, such as mowing hay or cutting rangeland brush. In RUSLE, these changes in growth patterns are handled through CROP Database Sets called in as regrowth crops. The regrowth Set is meant to reflect the crop growth and residue patterns as the crop rebounds from the effects of some operation or event that drastically changes those patterns. The date when an operation calls in a regrowth crop becomes day zero in the crop growth cycle, and that Set becomes the current crop.

When you use regrowth crops, you must make sure that proper transition is made between crops. For example, if there is a sudden drop between the last root mass value for the first Set and the first root mass for the second, the program will assume that you meant for this drop to occur, and will add that difference to the subsurface residue. Changes in residue cover and decay parameters as you move from one crop to the next indicate that subsequent residue additions will have those new characteristics.

How the time-varying option handles root mass. The root mass for the current crop begins growing at either the time of planting or when the crop is called in as a regrowth file. When an operation is defined as killing the current crop, the root mass becomes part of the subsurface residue and begins to decompose at a rate controlled by the decay parameter from the CROP Database Set in which the root mass was defined.

The program will also recognize a drop in root mass either within a CROP Database Set or between crops, as in the transition to a regrowth crop. Any drop in root mass is taken as a similar increase to the subsurface residue pool. If the drop in root mass occurs between Sets, the decay parameter for this residue will be that from the first Set.

Definition of canopy cover in the time-varying option. For the most part, the handling of canopy cover is very straightforward; the program simply takes the values from the current crop Set and uses them to make the calculations. The single possible complication occurs when you choose the senescence option for a crop. This indicates that a drop in canopy cover should cause an increase in residue on the soil surface. The program calculates this effect by looking at the percentage drop in residue cover, and then adding to the surface cover a mass of residue equal to that which would give the same percentage of cover to a bare surface. This amount is defined by the residue cover parameters from the CROP Database Set.

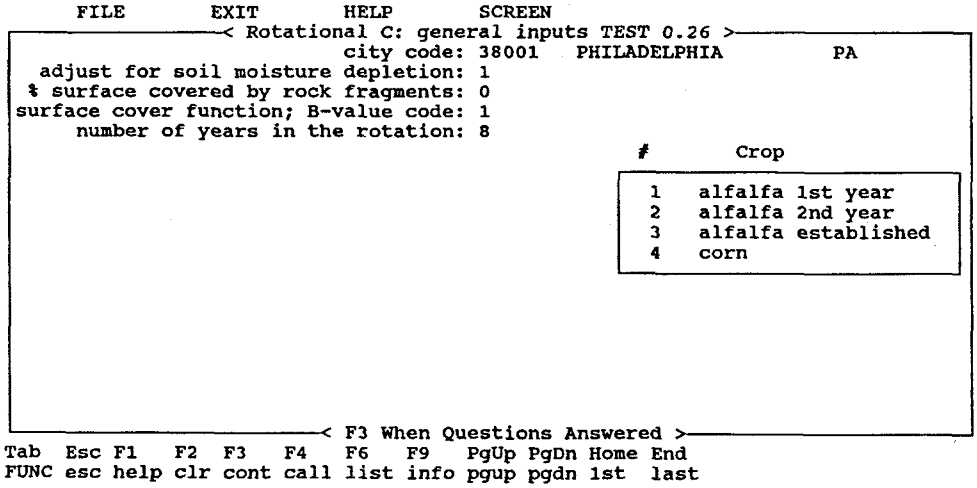
The program does not keep track of drops in canopy cover when you switch to a regrowth crop, because this type of change is usually caused by an operation, which often involves removing canopy material from the field.

How RUSLE handles surface residues. RUSLE keeps track individually of every residue you add to the field, and calculates its cover and decay relationships based on the specific parameters assigned to that residue when it was still associated with a current crop or when it was added as a "foreign" residue. Most residue additions are for the current crop, so you must be aware of which crop that is and whether its residue parameters are the ones you want.

Residue additions to the soil surface are specified on the screen where you list the operations. In general, you are responsible for telling the program how much residue is being added to the surface, but there is one exception. If you specify that the operation harvests the current crop, this tells the program to look in the CROP Database Set associated with that crop to find the residue added at harvest. The program then automatically enters that value into the correct place in the list of operations.

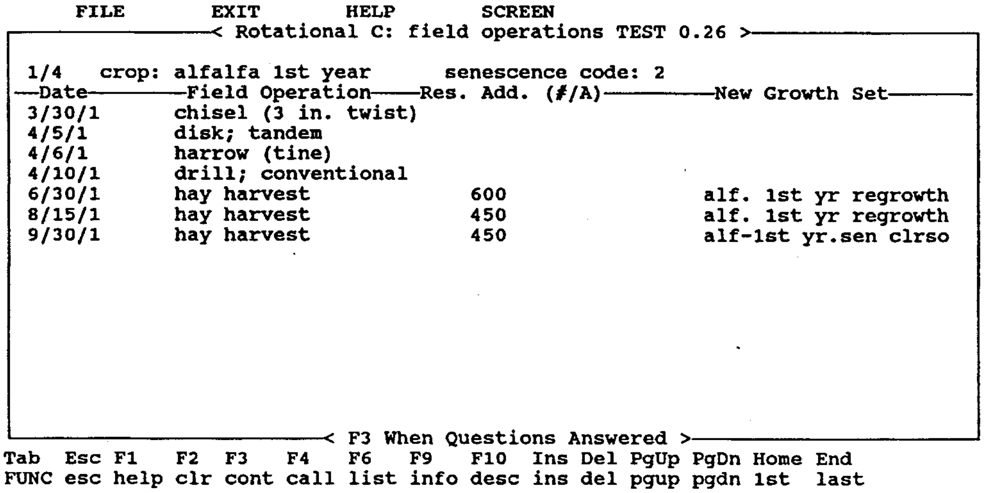
Example 1: Five years of Eastern alfalfa cut for hay followed by 3 yr of corn. Many of these concepts become more clear when dealing with specific examples. The first example is for an alfalfa-corn rotation in eastern Pennsylvania, with 5 yr of alfalfa followed by 3 yr of corn.

The first screen describes the location and gives general information about the rotation. The number of years in the rotation is set at 8 since, as we will see later, the first operation in the rotation will not be repeated for 8 yr. The crops indicated in the list were selected from the CROP Database File, and for each crop in the list the program will display a screen on which you will be asked to enter the field operations associated with that crop. The crops in this list need not correspond to calendar years; there can be either more or fewer crops in the list than years in the rotation. This list also does not need to include all CROP Database Sets used in the rotation.



```
      FILE       EXIT       HELP       SCREEN
 ──────────────< Rotational C: general inputs TEST 0.26 >─────────────
                          city code: 38001   PHILADELPHIA        PA
   adjust for soil moisture depletion: 1
  % surface covered by rock fragments: 0
 surface cover function; B-value code: 1
      number of years in the rotation: 8
                                                #       Crop

                                                1   alfalfa 1st year
                                                2   alfalfa 2nd year
                                                3   alfalfa established
                                                4   corn

 ─────────────────────< F3 When Questions Answered >──────────────────
Tab  Esc F1   F2  F3   F4   F6   F9   PgUp PgDn Home End
FUNC esc help clr cont call list info pgup pgdn 1st  last
```

Below is the first screen of information describing the field operations associated with the rotation. All operations that we want included in the calculation for "alfalfa 1st year" are listed here.

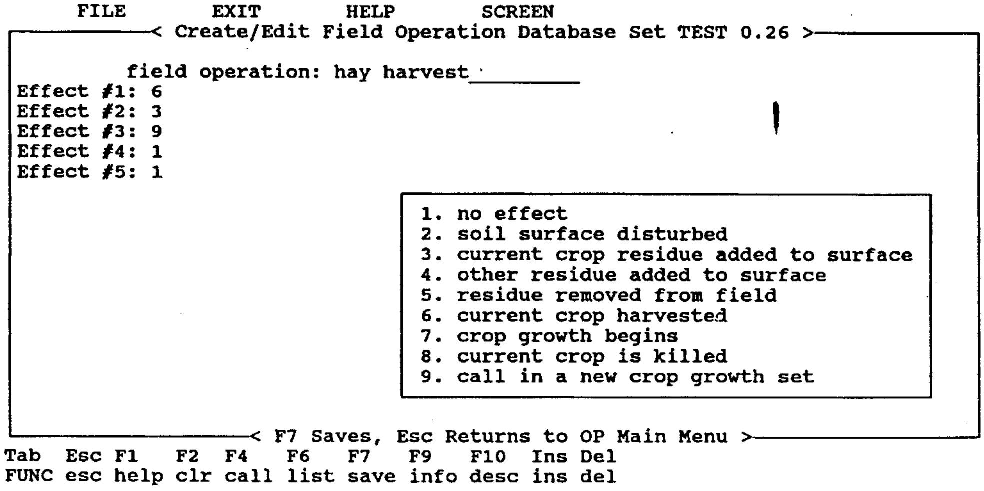


```
      FILE       EXIT       HELP       SCREEN
┌────────────────< Rotational C: field operations TEST 0.26 >─────────────────┐
│                                                                             │
│ 1/4   crop: alfalfa 1st year      senescence code: 2                        │
│──Date──────────Field Operation────Res. Add. (#/A)──────────New Growth Set───│
│ 3/30/1      chisel (3 in. twist)                                            │
│ 4/5/1       disk; tandem                                                    │
│ 4/6/1       harrow (tine)                                                   │
│ 4/10/1      drill; conventional                                             │
│ 6/30/1      hay harvest           600                  alf. 1st yr regrowth │
│ 8/15/1      hay harvest           450                  alf. 1st yr regrowth │
│ 9/30/1      hay harvest           450                  alf-1st yr.sen clrso │
│                                                                             │
│                                                                             │
│                                                                             │
│                                                                             │
│                                                                             │
│                                                                             │
│                                                                             │
└───────────────────────< F3 When Questions Answered >────────────────────────┘
Tab  Esc F1   F2  F3   F4   F6   F9   F10  Ins Del PgUp PgDn Home End
FUNC esc help clr cont call list info desc ins del pgup pgdn 1st  last
```

The first three operations in this list are tillage operations, which disturb the soil surface and incorporate some of the surface residue. Since the corn crop came just before these operations in the rotation, the operations will be burying corn residue. The tillage operations will also affect the soil surface random roughness.

The drill operation on April 10 includes an effect entitled "crop growth begins." This takes the CROP Database Set listed at the top of the screen (alfalfa 1st year), installs it as the current crop, and begins its growth cycle.

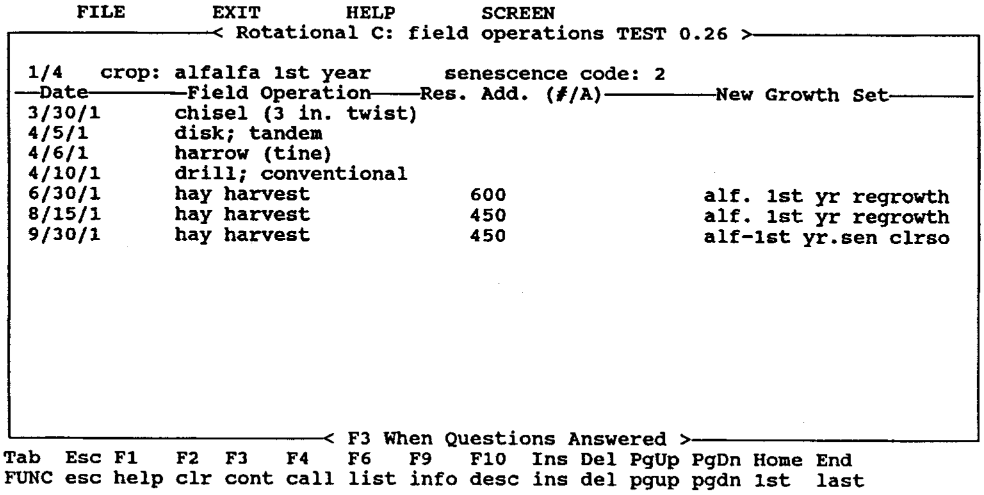
The hay harvests scheduled in this first year do two things: (1) they completely change the alfalfa growth characteristics, requiring a regrowth Database Set to show how the crop responds after being cut, and (2) they add some alfalfa residue to the field through wastage in the harvest process. There are two ways of handling this residue addition. The first way is as shown above, which uses the OPERATIONS Database Set below.



```
     FILE      EXIT      HELP      SCREEN
        < Create/Edit Field Operation Database Set TEST 0.26 >

         field operation: hay harvest
Effect #1: 6
Effect #2: 3
Effect #3: 9
Effect #4: 1
Effect #5: 1
                                      1. no effect
                                      2. soil surface disturbed
                                      3. current crop residue added to surface
                                      4. other residue added to surface
                                      5. residue removed from field
                                      6. current crop harvested
                                      7. crop growth begins
                                      8. current crop is killed
                                      9. call in a new crop growth set

                    < F7 Saves, Esc Returns to OP Main Menu >
Tab  Esc F1   F2  F4   F6   F7   F9   F10  Ins Del
FUNC esc help clr call list save info desc ins del
```

This Set specifies that the operation harvests the current crop, which causes the program to automatically go into the CROP Database Set for the current crop (in this case, "alfalfa 1st year") and to extract the information required to calculate residue added at harvest. The OPERATIONS Set also specifies that the operation adds residue from the current crop to the field: this is redundant because this is assumed by the program when it sees the harvest effect, but this may be included for the sake of completeness. Finally, the Set states that the operation will significantly affect the crop growth characteristics, requiring a new CROP Database Set to model that regrowth.

The operations list is repeated below for convenience. In it we see the regrowth CROP Database Sets listed in the far right column. Note that from 6/30/1 to 8/15/1, the current crop is "alf. 1st yr regrowth," which once again becomes the current crop with the hay harvest of 8/15/1. This means that on 8/15, the program brings this Set in to replace itself, and once again sets the days of growth to zero and restarts the growth cycle all over again.

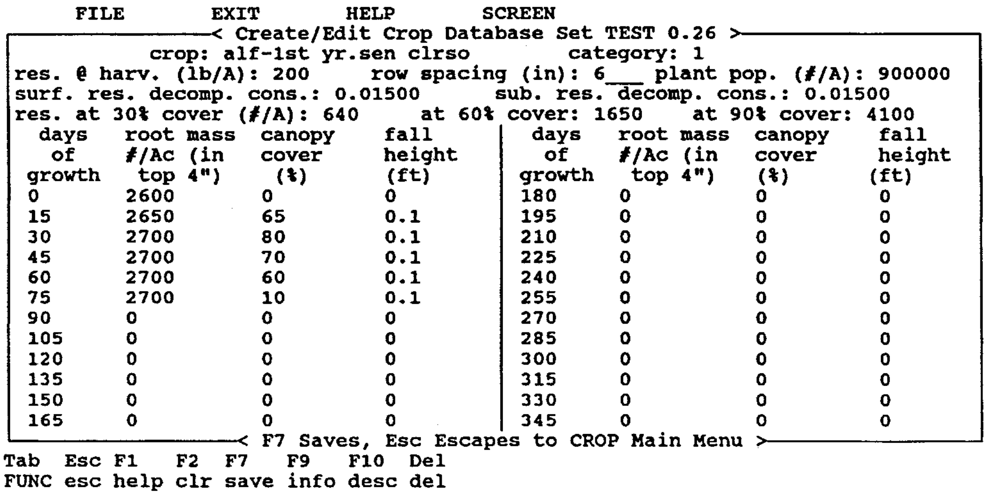


```
     FILE       EXIT       HELP       SCREEN
┌────────────< Rotational C: field operations TEST 0.26 >──────────────┐
│                                                                       │
│ 1/4   crop: alfalfa 1st year     senescence code: 2                   │
│─Date───────Field Operation───Res. Add. (#/A)─────────New Growth Set──│
│ 3/30/1     chisel (3 in. twist)                                       │
│ 4/5/1      disk; tandem                                               │
│ 4/6/1      harrow (tine)                                              │
│ 4/10/1     drill; conventional                                        │
│ 6/30/1     hay harvest             600               alf. 1st yr regrowth │
│ 8/15/1     hay harvest             450               alf. 1st yr regrowth │
│ 9/30/1     hay harvest             450               alf-1st yr.sen clrso │
│                                                                       │
│                                                                       │
│                                                                       │
│                                                                       │
│                                                                       │
└──────────────────< F3 When Questions Answered >───────────────────────┘
Tab  Esc F1   F2  F3   F4   F6   F9   F10  Ins Del PgUp PgDn Home End
FUNC esc help clr cont call list info desc ins del pgup pgdn 1st  last
```

The values in the third column, Res. Add., indicate the amount of residue added to the surface by each operation. Since the "hay harvest" Set listed the operation as a harvest, these values are calculated automatically from information in the CROP Database Set for the current crop. The 600 is calculated from the Set for "alfalfa 1st year" since it is the current crop to that point, and the 450 comes from the "alf. 1st yr regrowth" Set.

The second way of handling the residue additions is to specify the hay harvest simply as adding residue to the surface but not harvesting the crop. The program would then ask you to enter a value in the Res. Add. column rather than automatically calculating a value.

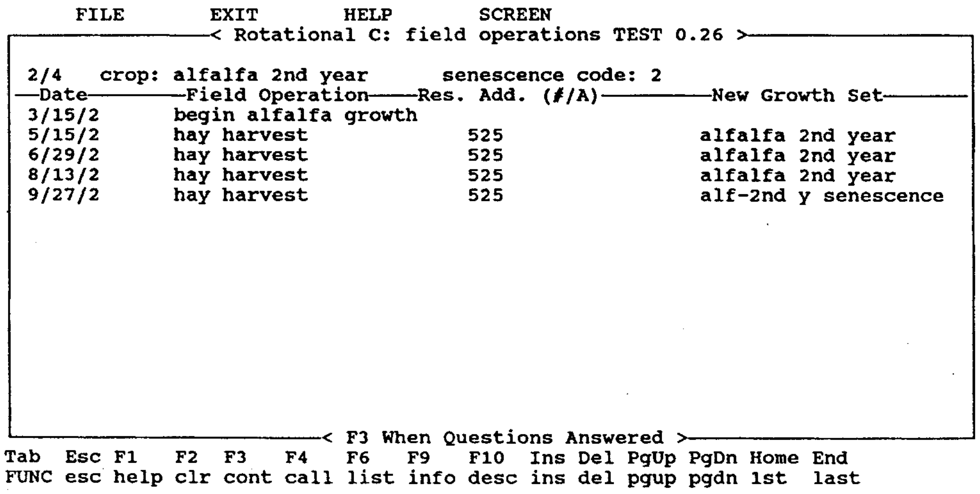
The senescence code for this crop is set at 2, which indicates that the program should consider a drop in canopy cover as an increase in surface cover. The CROP Database Set "alf-1st yr.sen clrso" includes such an effect to model the impact of winter on alfalfa growth, as shown below.



```
     FILE        EXIT        HELP        SCREEN
┌───────────────< Create/Edit Crop Database Set TEST 0.26 >──────────────┐
│            crop: alf-1st yr.sen clrso          category: 1             │
│res. @ harv. (lb/A): 200     row spacing (in): 6___ plant pop. (#/A): 900000
│surf. res. decomp. cons.: 0.01500      sub. res. decomp. cons.: 0.01500 │
│res. at 30% cover (#/A): 640     at 60% cover: 1650    at 90% cover: 4100
│  days    root mass  canopy    fall    │  days    root mass  canopy    fall
│   of     #/Ac (in   cover     height  │   of     #/Ac (in   cover     height
│ growth    top 4")    (%)      (ft)    │ growth    top 4")    (%)      (ft)
│ 0        2600       0         0       │ 180      0          0         0  │
│ 15       2650       65        0.1     │ 195      0          0         0  │
│ 30       2700       80        0.1     │ 210      0          0         0  │
│ 45       2700       70        0.1     │ 225      0          0         0  │
│ 60       2700       60        0.1     │ 240      0          0         0  │
│ 75       2700       10        0.1     │ 255      0          0         0  │
│ 90       0          0         0       │ 270      0          0         0  │
│ 105      0          0         0       │ 285      0          0         0  │
│ 120      0          0         0       │ 300      0          0         0  │
│ 135      0          0         0       │ 315      0          0         0  │
│ 150      0          0         0       │ 330      0          0         0  │
│ 165      0          0         0       │ 345      0          0         0  │
└──────────────────< F7 Saves, Esc Escapes to CROP Main Menu >───────────┘
Tab  Esc F1    F2  F7    F9    F10  Del
FUNC esc help clr save info desc del
```

The drop in canopy cover from 70% to 60% from day 45 to day 60 demonstrates the use of the senescence option. The program will use the residue weight/cover relationship from this Set to determine the weight of residue equivalent to 70 - 60 = 10% cover. This weight will then be added to the surface, with the addition distributed evenly over the 15 d in the period.

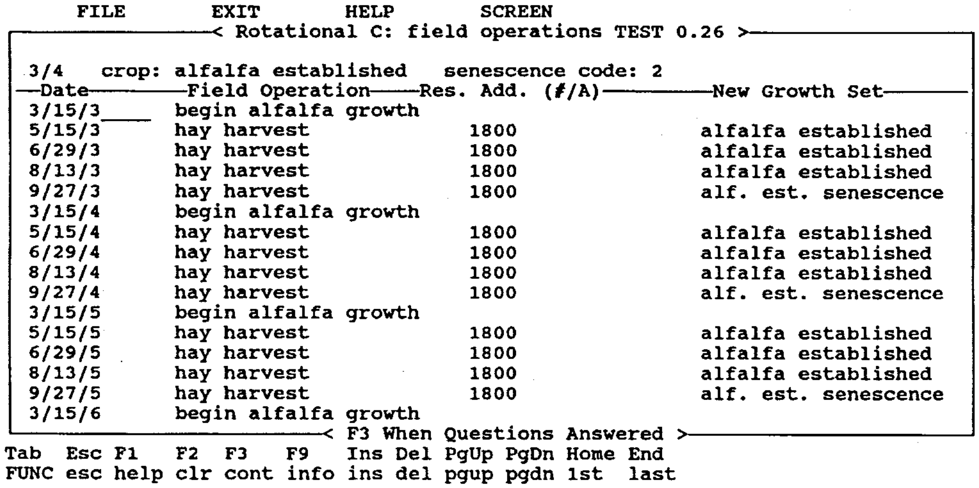
This completes the information required for the first crop listed on the first general information screen. The second crop listed is "alfalfa 2nd year," which requires the operations listing shown below.



```
     FILE        EXIT        HELP        SCREEN
┌───────────────< Rotational C: field operations TEST 0.26 >─────────────┐
│                                                                        │
│ 2/4   crop: alfalfa 2nd year      senescence code: 2                   │
│──Date─────────Field Operation─────Res. Add. (#/A)──────────New Growth Set──
│ 3/15/2        begin alfalfa growth                                     │
│ 5/15/2        hay harvest              525                alfalfa 2nd year
│ 6/29/2        hay harvest              525                alfalfa 2nd year
│ 8/13/2        hay harvest              525                alfalfa 2nd year
│ 9/27/2        hay harvest              525                alf-2nd y senescence
│                                                                        │
└────────────────────────────< F3 When Questions Answered >──────────────┘
Tab  Esc F1   F2  F3   F4   F6   F9   F10  Ins Del PgUp PgDn Home End
FUNC esc help clr cont call list info desc ins del pgup pgdn 1st  last
```

The "begin alfalfa growth" in this listing calls in "alfalfa 2nd year" as the current crop and begins its growth. The hay harvest operations are the same as in the first screen, adding residue to the surface and calling in regrowth crops. As before, the hay harvest operation calculates the amount of residue added based on information in the CROP Database Set of the current crop.

Up to this point we have had a single year's worth of operations on each screen, but this is not a requirement. The screen below shows 3 yr of crops and operations intended to model the alfalfa crop once it has reached fairly stable growth.



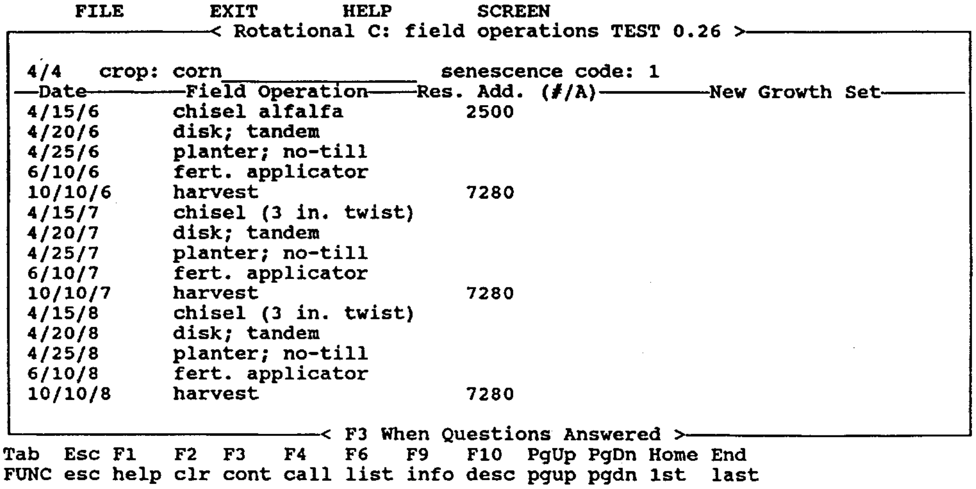
```
    FILE        EXIT        HELP        SCREEN
 ───────────────< Rotational C: field operations TEST 0.26 >────────────────
  3/4   crop: alfalfa established   senescence code: 2
 ─Date────────Field Operation────Res. Add. (#/A)──────────New Growth Set────────
  3/15/3____  begin alfalfa growth
  5/15/3      hay harvest             1800                alfalfa established
  6/29/3      hay harvest             1800                alfalfa established
  8/13/3      hay harvest             1800                alfalfa established
  9/27/3      hay harvest             1800                alf. est. senescence
  3/15/4      begin alfalfa growth
  5/15/4      hay harvest             1800                alfalfa established
  6/29/4      hay harvest             1800                alfalfa established
  8/13/4      hay harvest             1800                alfalfa established
  9/27/4      hay harvest             1800                alf. est. senescence
  3/15/5      begin alfalfa growth
  5/15/5      hay harvest             1800                alfalfa established
  6/29/5      hay harvest             1800                alfalfa established
  8/13/5      hay harvest             1800                alfalfa established
  9/27/5      hay harvest             1800                alf. est. senescence
  3/15/6      begin alfalfa growth
 ──────────────────────────< F3 When Questions Answered >───────────────────
 Tab  Esc F1   F2  F3   F9   Ins Del PgUp PgDn Home End
 FUNC esc help clr cont info ins del pgup pgdn 1st  last
```

A single screen can contain any number of years' worth of operations, as long as the total number of operations on the screen does not exceed 16. Similarly, a screen may contain operations representing a fraction of a year. The only difference comes in the final accounting. The program calculates an overall rotation C factor and a C factor for the time associated with each screen. For this rotation, the program would calculate separate C factors for each of the first 2 yr, but would lump these 3 yr together into a single C factor.

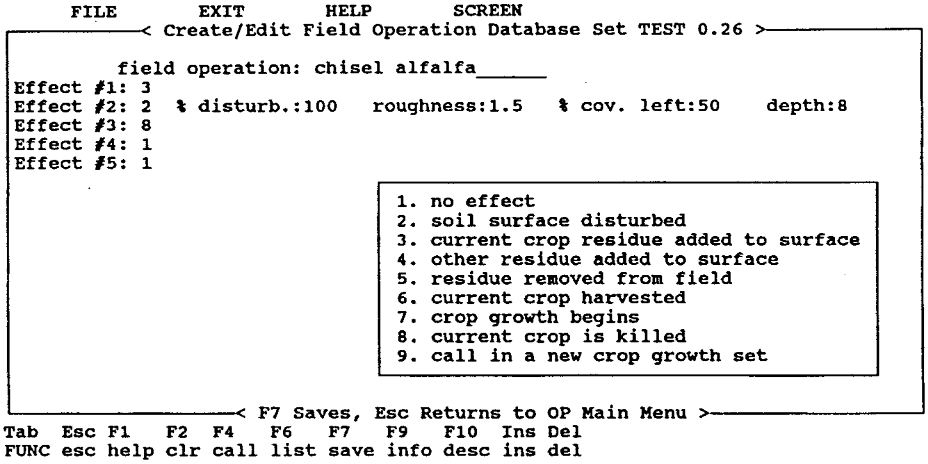
Note also that the program is not limited to working with calendar years; but it will allow you to divide the time up in any convenient fashion simply by which screen contains which operations.

The screens shown above take care of the 5 yr of alfalfa in the rotation, but we must still handle the 3 yr of corn. In the initial crop listing there was only one corn, so all operations associated with these 3 yr will also have to be lumped together on a single screen, as shown below.

The first operation in this listing is used to kill the alfalfa and also to prepare the soil for the corn-planting operation. The information for this operation is listed below.



```
        FILE        EXIT        HELP        SCREEN
┌────────────────< Rotational C: field operations TEST 0.26 >─────────────────┐
│                                                                              │
│ 4/4    crop: corn________________     senescence code: 1                    │
│─Date──────────Field Operation────Res. Add. (#/A)──────────New Growth Set─────│
│ 4/15/6       chisel alfalfa          2500                                    │
│ 4/20/6       disk; tandem                                                    │
│ 4/25/6       planter; no-till                                                │
│ 6/10/6       fert. applicator                                                │
│ 10/10/6      harvest                 7280                                    │
│ 4/15/7       chisel (3 in. twist)                                            │
│ 4/20/7       disk; tandem                                                    │
│ 4/25/7       planter; no-till                                                │
│ 6/10/7       fert. applicator                                                │
│ 10/10/7      harvest                 7280                                    │
│ 4/15/8       chisel (3 in. twist)                                            │
│ 4/20/8       disk; tandem                                                    │
│ 4/25/8       planter; no-till                                                │
│ 6/10/8       fert. applicator                                                │
│ 10/10/8      harvest                 7280                                    │
│                                                                              │
└──────────────────────────< F3 When Questions Answered >──────────────────────┘
 Tab  Esc F1   F2  F3   F4   F6   F9   F10  PgUp PgDn Home End
 FUNC esc help clr cont call list info desc pgup pgdn 1st  last
```



```
        FILE        EXIT        HELP        SCREEN
┌──────────< Create/Edit Field Operation Database Set TEST 0.26 >──────────┐
│                                                                           │
│          field operation: chisel alfalfa______                            │
│Effect #1: 3                                                               │
│Effect #2: 2  % disturb.:100   roughness:1.5   % cov. left:50    depth:8   │
│Effect #3: 8                                                               │
│Effect #4: 1                                                               │
│Effect #5: 1                                                               │
│                                      ┌───────────────────────────────────┐│
│                                      │ 1. no effect                      ││
│                                      │ 2. soil surface disturbed         ││
│                                      │ 3. current crop residue added to surface│
│                                      │ 4. other residue added to surface ││
│                                      │ 5. residue removed from field     ││
│                                      │ 6. current crop harvested         ││
│                                      │ 7. crop growth begins             ││
│                                      │ 8. current crop is killed         ││
│                                      │ 9. call in a new crop growth set  ││
│                                      └───────────────────────────────────┘│
│                                                                           │
└───────────────────< F7 Saves, Esc Returns to OP Main Menu >───────────────┘
 Tab  Esc F1   F2  F4   F6   F7   F9   F10  Ins Del
 FUNC esc help clr call list save info desc ins del
```

This case is one where the order of the operation's effects is critical. This operation first adds residue to the soil surface. Since the operation is not specified as a harvest, the amount added will not be calculated from information in the CROP Database Set, but will instead have to be entered manually in the Res. Add. column of the operations listing.

After the residue is added, the soil surface is disturbed, thereby incorporating some of that newly added residue. If these effects had been entered in the opposite order, none of this new residue would have been buried by the operation, because the soil disturbance would have occurred before the residue was added.

Finally, the current crop is killed, immediately removing the effects of crop canopy and turning the root mass into subsurface residue.

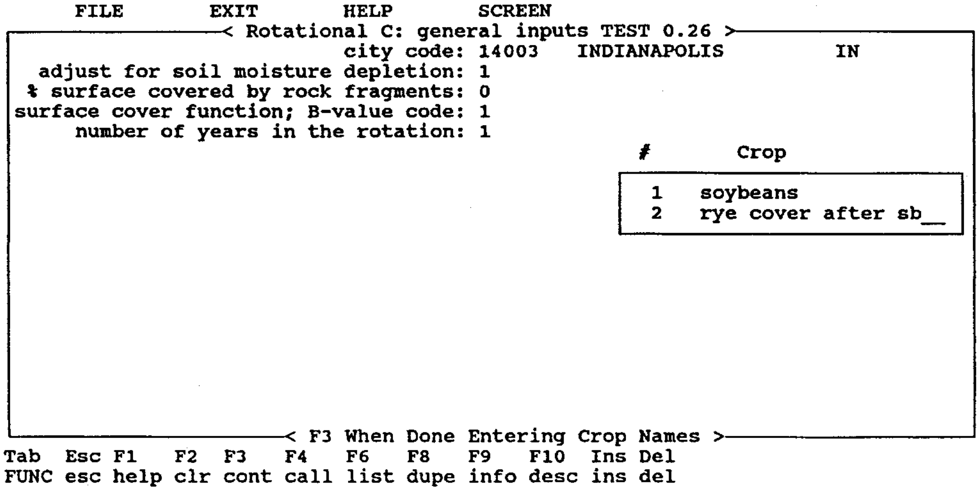
The remainder of the operations in the "corn" listing are relatively straightforward, and are shown below. The "harvest" operation behaves like the "hay harvest" we saw earlier, in that it calculates the amount of residue added to the surface from the CROP Database Set. It does not call in a regrowth Set because harvesting corn also kills it.

The information needed by the rotation is now complete, because the next operation in the series would be the primary tillage before planting alfalfa, which is a repeat of the first operation in the sequence. Note that this would occur on 3/30 of year 9, which is 8 yr after its first occurrence. This is what determined the rotation length of 8 yr. If the rotation length had been entered as 9 yr, the program would assume a fallow period from 10/10/8 to 3/30/10.

Example 2: Continuously cropped conventional soybeans in Indiana with an aerially seeded rye winter cover crop. This example demonstrates how to handle two crops grown simultaneously. The example is based on a continuous conventionally tilled soybean crop, into which rye is aerially seeded when the soybeans are mature. The rye grows as an understory to the soybeans until they are harvested, at which time the rye begins vigorous growth until the onset of winter. The rye is tilled under in the spring before the planting of the next crop of soybeans. Because every operation is repeated every year, the rotation is 1 yr long.

The critical concept is that RUSLE can handle only one current crop, so the CROP Database Set associated with that crop must reflect everything that is growing in the field.

The general information screen for this rotation is shown below.

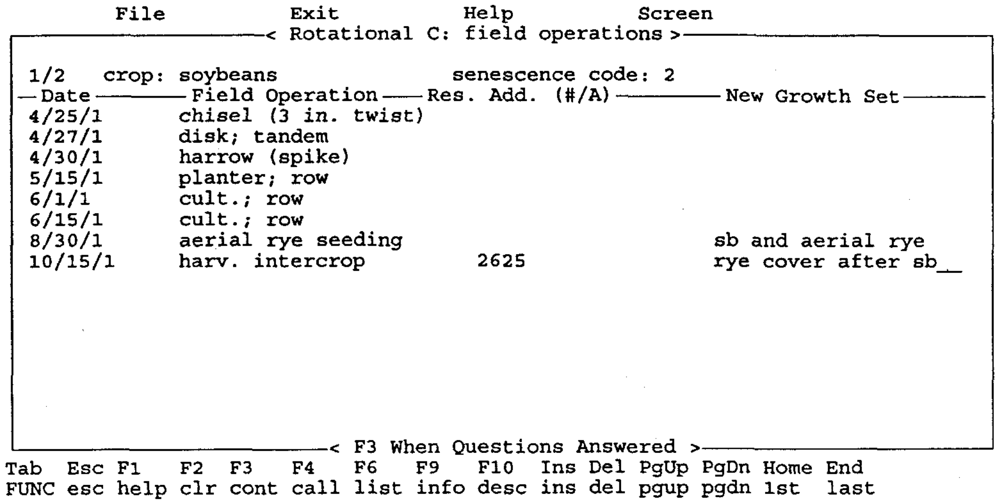


```
        FILE       EXIT       HELP       SCREEN
 ────────────────< Rotational C: general inputs TEST 0.26 >──────────────────
                         city code: 14003   INDIANAPOLIS         IN
   adjust for soil moisture depletion: 1
   % surface covered by rock fragments: 0
  surface cover function; B-value code: 1
       number of years in the rotation: 1
                                                          #       Crop
                                                         1   soybeans
                                                         2   rye cover after sb__




 ─────────────────────< F3 When Done Entering Crop Names >──────────────────
 Tab  Esc F1   F2  F3   F4   F6   F8   F9   F10  Ins Del
 FUNC esc help clr cont call list dupe info desc ins del
```

We could vary the list of crops, depending on how we want the accounting to be done. For instance, if we want only a total rotation value, we could list just soybeans. Another option is as shown above, where we separate the time that the soybeans are growing from the rest of the time.

The operations listing associated with the soybean accounting time is shown below. This includes the planting of the standard soybean file and its growth up to the time of the aerial seeding of the rye. From this point on, the current crop must reflect the root mass and canopy characteristics of not only the soybeans but the combined soybean/rye mixture. These characteristics will be shown in the CROP Database Set "sb and aerial rye" as an increase in root mass and canopy cover and a decrease in canopy height to show the effect of the low-growing rye. The residue characteristics of this Set will still be those of the soybeans, as this is the type of residue that will be added.

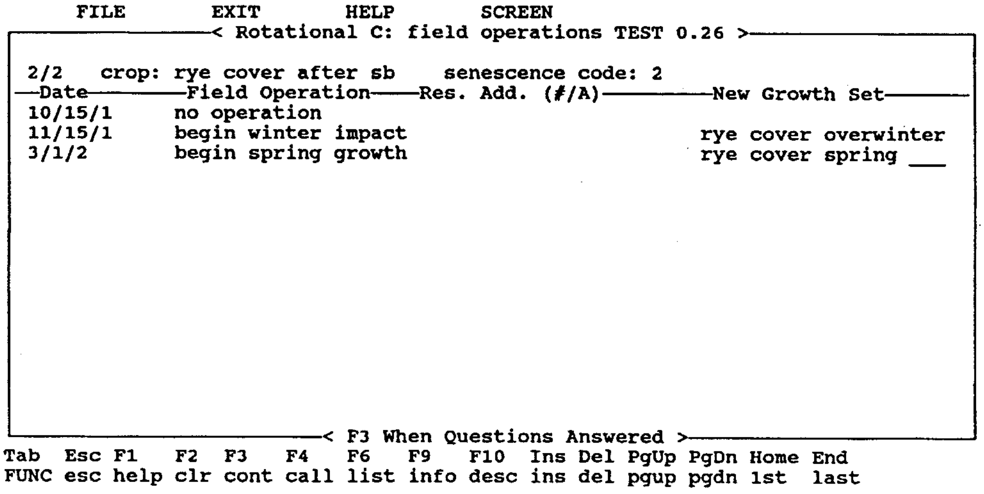
At the time of harvest we must not kill the crop, because the rye will continue to grow. Instead, we add the soybean residue to the surface and then call in a regrowth crop to simulate the vigorous rye growth after soybean harvest. The root mass value in this Set will be much lower than that in the combined soybean/rye Set, which the program will recognize as an addition of the soybean roots to the subsurface soybean residue pool. This new Set will have the residue characteristics of the rye, because any residue added to the surface now will be rye residue.



```
        File          Exit          Help          Screen
 ──────────────────< Rotational C: field operations >──────────────────
  1/2   crop: soybeans              senescence code: 2
 ─Date──────────Field Operation───Res. Add. (#/A)──────────New Growth Set──────
  4/25/1      chisel (3 in. twist)
  4/27/1      disk; tandem
  4/30/1      harrow (spike)
  5/15/1      planter; row
  6/1/1       cult.; row
  6/15/1      cult.; row
  8/30/1      aerial rye seeding                          sb and aerial rye
  10/15/1     harv. intercrop        2625                 rye cover after sb__




 ─────────────────────────< F3 When Questions Answered >─────────────────────────
 Tab  Esc F1   F2  F3   F4   F6   F9   F10  Ins Del PgUp PgDn Home End
 FUNC esc help clr cont call list info desc ins del pgup pgdn 1st  last
```

The rye crop is called in as the current crop with the last operation of the soybean listing, but recall that we wanted to account for the time from the soybean harvest to the next planting of soybeans under the rye cover crop. We do this with the first operation shown on the screen above. This "no operation" has no effects on the soil, crop, or residue, but is used to tell the program to begin the accounting under this crop name on this date. (See next figure.)

There are two ways of modeling the impact of a natural phenomenon like winter or drought conditions. First, if the impact is very sudden (as perhaps the first frost that kills tomatoes), you will want to model the impact as an operation. On the other hand, if the effect is more gradual (as it would be for the rye example), it is probably better to develop a CROP Database Set that shows the reduction in canopy cover and live root mass that can be caused by winter damage. Note in the listing below that nowhere within this screen is there an operation that begins the growth of the "rye cover after sb" crop. A current crop is still growing based on information in the previous screen, so there is no need to plant this crop.



```
        FILE       EXIT       HELP       SCREEN
┌───────────────< Rotational C: field operations TEST 0.26 >──────────────┐
│                                                                          │
│ 2/2   crop: rye cover after sb    senescence code: 2                     │
├─Date────────Field Operation────Res. Add. (#/A)─────────New Growth Set───┤
│ 10/15/1     no operation                                                 │
│ 11/15/1     begin winter impact                      rye cover overwinter│
│ 3/1/2       begin spring growth                      rye cover spring ___│
│                                                                          │
│                                                                          │
└─────────────────────────< F3 When Questions Answered >───────────────────┘
Tab  Esc F1   F2  F3   F4   F6   F9   F10  Ins Del PgUp PgDn Home End
FUNC esc help clr cont call list info desc ins del pgup pgdn 1st  last
```

Example 3: Grazing of rangeland or pasture. The grazing of rangeland or pasture can be handled as the impact of winter on crops was handled. If the grazing is high intensity and can be thought of as occurring at a specific time, it can be treated as an operation, as with the hay harvests shown above. The vegetation would be harvested by the cattle, leaving a rough surface and trampling some residue onto the soil. The grazed area would then begin to regrow, so the high-intensity grazing operation would call in a regrowth crop Set.

On the other hand, if the grazing is long term and low intensity, it is probably best to treat it as a change in the crop growth patterns and to develop a CROP Database Set to reflect these conditions.

Answers to Common Questions About the C Factor

For a single disturbance, how do I define the initial conditions? Remember the factors that you must define: (1) amount of residue on the surface; (2) amount of buried residue; (3) surface roughness; and (4) any vegetative regrowth. This is done by beginning the growth of vegetation whose CROP Database Set reflects that found at the site before its disturbance. This should be grown for several years to reflect the soil consolidation found before the disturbance. The operations used in the disturbance are then listed, making sure that the effects of each operation reflect all of its impacts on the factors listed above.

How do I specify that the operations sequence is a rotation? You command the program to assume a rotation by specifying on the initial C-factor screen the number of years in the rotation. You calculate this by determining how many years it takes before the first operation will occur again. For example, if the first operation in the rotation takes place on May 1 of the first year and will not occur again until May 1 of the fourth year, then there will be 4 - 1 = 3 yr in the rotation. If you want a single disturbance rather than a rotation, simply place a zero in the input field for number of years in the rotation.

For the time-invariant C option, should I use the CROP Database or enter values directly? The C calculations for RUSLE require values from the CROP Database for canopy cover, fall height, and root mass. For continuous crops, these values are assumed to remain relatively constant through the year, so the only entries needed are average annual values. To specify these values, you can either enter them directly within the C factor or use the CROP Database Utility routines (see "Databases" and "CROP Database") to create a Set with these values. Do this with the "permanent" option for the crop/land use category, and then {save} the set under some unique Identifier name.

If you decide to use the CROP Database, the list of all {save}d "continuous" crops will be shown. Select an Identifier to automatically bring in the values from that Set for root mass, canopy cover, and fall height.

Within this time-invariant C option, the values of surface cover and roughness must be typed in directly.

What inputs are required for the time-invariant C option? The variables of surface cover, canopy cover, and fall height are relatively easy to deduce from experience. It is much harder to estimate the mass of roots in the top 4 in of soil. If you choose to not put the required data into a CROP Database Set (see section above), RUSLE will go through a series of questions to help you define the root-mass variable. You will be asked to define the type of plant community and the annual site production potential. The program will use these to calculate a root-mass value directly and, in that case, the program will calculate a corresponding site potential.

What general information is required for the time-varying C option? The first screen within the time-varying C-factor option defines all the general information for the rotation. Soil-moisture depletion should be taken into account only for those areas in which the rainfall is of low amounts and intensities and the soil surface characteristics do not limit infiltration. Thus far, this is supported only by data from the Northwestern Wheat and Range Region, although there may be other areas to which this option can be applied. The "rock cover" variable should reflect the presence of all rocks and similar "permanent" surface cover, and is changed by {call}ing the K factor. Choose the b-value option that best describes your conditions. For well-consolidated soils dominated by interrill erosion, a low b value should be used; for highly disturbed or thawing soils dominated by rilling, a high b value is more appropriate. All other cases should use the moderate b value.

The crop list on the first screen of this option makes it easy to change an existing Input File. Crops can be {dupe}licated to new spots in the rotation, old crops can be {del}eted from the list, and new crops can be {Ins}erted. When you move a crop within the list, its operations will tag along, but the dates of the operations may need to be changed to show their new place within the rotation.

What information is needed on the operations screens of the time-varying C option? The rotation information screen contains general information about the crop and a list of all associated field operations. Also included is a question on whether a decrease in canopy cover should be seen as a contribution to surface cover. If you specified in the general information that you wanted to account for soil moisture depletion, the depletion rate value associated with the crops on this screen is entered here.

You must enter the dates (using xx/xx/xx to represent month/day/year) and names of all field operations, as well as any additional information required by the effects listed in the OPERATIONS Database Sets for those operations. You have great flexibility in entering the year. You can enter it as a calendar year (1995), as an abbreviated calendar year (95), or as the number of the year in the crop sequence. The program calculates the relative time elapsed, so all of these will be treated the same.

There are some limitations on the operations that you can enter, as follows:

(1) You can have only one current crop. Since there will be interactions between interplanted crops, RUSLE does not allow you to use two separate CROP Database Sets and to have them both growing at once. You must instead combine the Sets into one that reflects the total values of the combined crops. The residue parameters should be for the crop whose residue is added to the surface by a harvest operation; you can call in a regrowth CROP Database Set containing the parameters of the second crop before harvesting it.

(2) If an operation has a "harvest" effect, the program will automatically enter the weight of residue added. For other effects, you will be asked to specify the amount of residue added.

(3) The operation can also ask for the name of a regrowth crop. In this case, a list of the crops is shown, and you must either pick from this list or go into the CROP Database Utility routines to create a new crop.

(4) There are few restrictions on the order in which the operations may be listed, but one restriction is that the current crop must exist in order to be able to do anything with it. The program will issue a warning if you try to do anything with the current crop without having specified something to tell the program just what that crop is.

How are all the different roughness values set and used? The time-varying C-factor calculations use three different random roughness values. The first is the roughness immediately after a soil disturbance operation; this value is defined within the OPERATIONS Database Set for that operation, and is a function of soil condition and of the implement that is used.

The second roughness value is that to which the tillage roughness decays as it is acted upon by raindrop impact and surface flow. The default setting for this roughness is 0.24 in, which is the roughness found on experimental fallow plots exposed to natural rainfall. If desired, this value can be set to reflect continuous re-roughening of the surface, as can happen with cattle grazing. In this case, the initial and final roughness values can be set equal to indicate a constant roughness, or can be adjusted as desired to indicate increasing or decreasing roughness. The final decay roughness value is set in the OPERATIONS Database Set.

The third random roughness value used in this part of RUSLE reflects the impact of the vegetation on soil surface roughness; this will be a function of protruding roots, basal mounding, and so on. This variable represents the site roughness several years in the future (as defined by the number of years to soil consolidation), under the assumption that the site will not be disturbed during that time. The value, therefore, also reflects the assumption of some vegetative community that will come to dominate the undisturbed site over this time period. The program uses this value only for runs with a single disturbance, and uses a sigmoidal growth curve to increase the natural roughness to this value from the minimum value defined in the previous paragraph. The long-term site roughness is set in the general information screen of the time-varying C-factor option if the length of the rotation is set to zero.

What outputs are available for the time-varying C option? The options for showing the results of the rotational C-factor calculations vary primarily in the degree of detail shown. The "Rotation C" option divides the results into the SLR · %EI associated with each crop, whereas the "Operation C" option breaks this down further into the SLR · %EI associated with each operation. The "Half-Month Subfactor" option shows the finest division, displaying each of the SLR subfactors for each calculation period.

There is one difference between the "Operation C" results and the "Half-Month" results that might cause confusion. In Operation C, the values shown for percent surface cover are calculated immediately after the operation, whereas the Half-Month surface cover values are calculated in the middle of the time period.

P Factor

The RUSLE P factor reflects the impact of support practices on the average annual erosion rate.

Options

As with the other factors, the P factor differentiates between cropland and rangeland or permanent pasture. Both options allow for terracing or contouring, but the cropland option contains a stripcropping routine whereas the rangeland/permanent-pasture option contains an "other mechanical disturbance" routine. For the purposes of this factor, the rangeland/permanent-pasture option is based on the support operation being performed infrequently, whereas in the cropland option, the support operation is part of the annual management practice.

One variable seen in the contouring, mechanical-disturbance, and stripcropping routines is a site description. RUSLE asks you to choose from a list of descriptions the one that best fits your conditions. This information is used in several ways, as follows:

(1) The site description is used to assign a runoff index and roughness value to the situation. The runoff index is a measure of the percentage of available precipitation that will be seen as runoff, and is a function of soil type, soil structure, soil surface condition, and surface vegetative cover. The runoff index value is similar to the curve number used in the NRCS Direct Runoff method of calculating runoff volume, and is used here for roughly the same purpose. Runoff index values associated with specific soil hydrologic groups and site descriptions can be changed by

using the {call} command from the Input Field where the site description is requested.

(2) The site description is also used to estimate the effect of surface roughness and vegetative cover on erosion and transport from the slope. A qualitative view of the roughness is given in the site descriptions, but the actual roughness values associated with the descriptions are assigned internally.

Answers to Common Questions About the P Factor

What is the variable ridge height option on the contouring subfactor (cropland or pasture/rangeland option)? This option allows you to define how the contour ridge heights change with time during the rotation. The program calculates a P value for each ridge height and then multiplies that by the percentage of annual EI associated with each period to derive an average annual contouring subfactor.

What is the critical slope length in the contouring subfactor (cropland or pasture/rangeland option)? The effectiveness of contouring breaks down when the slope is so long that runoff causes break-over and subsequent gullying. This point is defined by the critical slope length, which is a function of the slope steepness, ridge height, residue cover, and runoff potential. When the slope is longer than this calculated maximum, the contour credit applies only to the portion above the critical length. The portion below has a P contour subfactor of 1. RUSLE will give you a warning that this is occurring, and will adjust the overall P factor accordingly.

Why are there dual site descriptions in the contour subfactor (pasture/rangeland option only)? After a disturbance, a pasture/rangeland soil will take some years to reconsolidate to a point similar to that before the disturbance. In order to look at this consolidation effect, it is necessary to detail just how the site responds to consolidation and what effect this has on runoff. This information is given by describing the site both immediately after disturbance and after enough time has passed for relatively complete consolidation. The program then assumes an exponential decay function, and allows you to look at any year within that range.

What is the stripcropping subfactor (cropland option only)? The stripcropping P subfactor defines the effectiveness of contoured or cross-slope strips in causing deposition where it might benefit future productivity, and in slowing and spreading the runoff. These strips can be any combination of crops and/or tillage practices that would affect the surface roughness, infiltration rates, or hydraulic roughness seen by the runoff.

Why does the stripcropping subfactor require {call}ing the contouring routine (cropland option only)? Because stripcropping is closely linked to contouring, the RUSLE program uses the contouring routines to provide information on row grade and critical slope length. If it is needed, the call to the contouring subroutine occurs automatically. When the contouring routine is called from the stripcropping routine, the site description used should be for the "smoothest" strip within the stripcropping rotation.

What is the stripcropping subfactor strip width (cropland option only)? The stripcropping subfactor program requires a complete description of the slope, including the condition and width of each strip. There are two ways of specifying strip width. The first is to indicate the position of the bottom of the strip as a percentage of the total slope length. For example, if a 120-ft slope is divided into four 30-ft strips, their relative positions would be at (30/120), (60/120), (90/120), and (120/120), or 25%, 50%, 75%, and 100%, respectively. The other option is to specify the actual strip widths in feet, which in the above example would be 30, 30, 30, and 30. The disadvantage of this second style is that the total slope length has already been specified in the LS factor. If the total length calculated by adding these strip widths varies from that entered in LS by more than 10%, the RUSLE program issues a warning and requires a response before proceeding.

What is the Other Mechanical Disturbance subfactor (pasture/rangeland option only)? Mechanical disturbance of the soil takes two possible forms: (1) disturbance on the contour, which results in roughness on the contour and redirection of runoff along the rows; and (2) a more random mechanical disturbance, which results in a rough surface that can slow runoff and in depressional areas that can store runoff and increase infiltration. The option chosen depends on whether the roughness that is left after the disturbance is primarily oriented or random. If the roughness has a definite orientation (ridges), choose the contouring option, even though the ridges may not be directly on the contour. If the roughness is more random, choose the Other Mechanical Disturbance option.

SYSTEM CONSIDERATIONS: USING RUSLE WITH YOUR EQUIPMENT

All computer requirements and instructions for loading and running RUSLE are described in detail in "Getting RUSLE To Work on Your Machine."

Using a Printer

RUSLE is meant primarily to be a tool to make it easy to compare cropping and management alternatives, and to speed up these repeated calculations. Although RUSLE will print out some results and inputs, it is not designed to generate documents.

For machines that use DOS, RUSLE looks for an attached printer, as it does also for the 3B2 and 6386 machines running either DOS or UNIX. When run under BSD UNIX, the print routines will look in a file named "printer.use" for the name of the network printer. Check your manuals or consult with a site specialist if problems arise.

Most of the printouts are meant to fit on 8½" x 11" paper. The exception is the printout of LS tables, which may be up to 132 characters wide. If your printer cannot be configured to handle this many characters within 8½ in, you must use wider paper. Note that copies of the tables are available in chapter 4 of this handbook.

Making Copies of RUSLE To Share With Others

RUSLE contains a routine that allows you to copy the program onto a diskette(s) to share with others. This routine is available from the RUSLE Information option of the RUSLE Main Menu, and works only on DOS. This option is not available in UNIX.

Refer to "Using an Input File Created Earlier or Elsewhere" and to "Using Database Sets Created Earlier or Elsewhere" for information on how to send information you have entered to another user.

Reporting Problems and Trouble

RUSLE is a relatively new and complex computer program. A user may find problems that did not arise in preliminary testing. It will greatly help the developers and programmers of RUSLE if you report problems that you find, giving as much information as possible about when and how the problems occurred. This information includes the following:

- screen you were on and question being answered
- command you gave just before the problem showed up
- what (if anything) showed up on the screen
- how the problem manifested itself (program locked up, gave unreasonable answer, and so on.)

Please send this information to:

George R. Foster
USDA-ARS, National Sedimentation Laboratory
P.O. Box 1157, 598 McElroy Drive
Oxford, Mississippi 38655
Telephone (601) 232-2900
FAX (601) 232-2915

Glenn A. Weesies
USDA-NRCS, National Soil Erosion Research Laboratory
Purdue University, Building SOIL
West Lafayette, Indiana 47907
Telephone (317) 494-8692

Daniel C. Yoder
Department of Agricultural Engineering
University of Tennessee
PO Box 1071
Knoxville, Tennessee 37901-1071
Telephone (615) 974-7266

or

Kenneth G. Renard
USDA-ARS, Southwest Watershed Research Center
2000 E. Allen Road
Tucson, Arizona 85719-1596
Telephone (520) 670-6481
FAX (520) 670-5550

SUMMARIES

Command Keys

A complete list of all possible commands and a brief description of what each does are given below. The command description or name is given in braces, { }, and the associated DOS key for each command is given in brackets, [ ]. The keys may differ on some machines, but the command description is always as shown here.

{FUNC} [TAB] places the cursor onto the Function Line to let you do the required housekeeping chores (see "Calling a Function" or "The Functions: Program Housekeeping"). When the cursor is on the Function Line, use {esc}ape to bring it back to the Main Screen.

{esc} [ESC] allows you to {esc}ape from the current screen or question without giving an answer. This is most commonly used in three places: (1) if you have gotten into a screen or series of screens and want to get back out; (2) after an error or warning message has been displayed, and you want to continue; and (3) to continue the program after a result has been displayed.

{help} [F1] shows one or more screens of additional information to help answer the current question (see "Help"). This usually includes a brief description of how the variable is used in the calculations and also suggestions for possible answers.

{clear} [F2] {clear}s (sets to 0) all variables associated with the current screen. This is most useful when you want to get a fresh start on a screen containing several mistakes.

{cont} [F3] {cont}inues program movement to the next logical screen in the sequence. For example, when you have answered all the questions on one screen, this command moves you to the next screen or initiates the calculations. For screens that require only one input or command, the {cont}inue command acts just like an [ENTER], telling the program to accept the current value and to move on.

{call} [F4] {call}s a subroutine on which the answer might depend. Use of this command will automatically put you into the required subroutine; when you exit from that, you will return to the current question.

The {call} command is used in three different instances, as follows:

(1) When an answer is calculated from the results of several different factors. An example is in estimation of soil loss from the factor values. Each factor is {call}ed individually by use of this command.

(2) When the question asks for the Identifier of a Database Set, as when entering a city code, a crop name, or an operation name. Use of the {call} from one of these locations allows you to examine or modify the information within the associated Database. "Databases" gives more detail.

(3) When the calculation requires information entered in another portion of the program. An example of this is the value of average field slope, which is used in several places but is calculated within the LS factor. If this value has already been calculated, it will be shown; if not, the {call} command must be used to move to that calculation.

{list} [F6] displays a list from which you can select an option or item, as explained in "Input Field." If you have previously selected one of the choices, the list may not be shown but this command will appear as an option. Use of this command displays the list and allows you to move through it using the [ARROW] keys.

There are places in the program where the list is also shown but the marker arrow is not visible. In these cases a {list} command will move the marker arrow into the list, where the marker can be controlled with the [ARROW] keys.

{save} [F7] {save}s the data shown on the screen into a Database Set named by the Identifier at the top of the screen. Changes made within the CITY, CROP, or OPERATIONS

Database routines will not be saved into those Databases unless this command is used before exiting those routines. If changes have been made, you will be asked whether or not you want to save them before you are allowed to exit.

{dupe} [F8] is used only in the crop listing on the initial C-factor input screen; this duplicates an entire operation listing screen into another location within the list. For example, if I {dupe} "corn" from the first place on the list into the fifth, the program will duplicate all operations associated with that first corn onto the fifth screen. It will not change the dates, so this must be done by moving to that screen and modifying them individually.

{info} [F9] gives information on the Current Input List. When you {save} a Current Input List into an Input File, you are also saving a list of comments describing that List. Use of this command allows you to look at (and change) the descriptive information for the Current Input List.

{desc} [F10] gives information on the current CROP or OPERATIONS Database Set. These Sets also contain a series of descriptive comments, which can be viewed or changed from anywhere this command is available.

{ins} [INS] is used to insert a line of information just above the line on which the cursor is resting.

{del} [DEL] is used to delete the line of information on which the cursor is resting.

{1st} [HOME] is used to jump to the first in a series of screens or to the beginning of a list.

{last} [END] is used to jump to the last in a series of screens or to the end of a list.

{pgup} [PG UP] is used to move to the previous screen in a series of screens or up one screen in a long list of information.

{pgdn} [PG DN] is used to move to the next screen in a series of screens or down one screen in a long list of information.

Terminology

command : is an instruction given to RUSLE to control the flow of the program. A *command* is usually given by pressing a key. At any point in the program, the available *commands* are shown on the *Command Line.*

Command Line: is made up of the bottom two lines of the screen, and contains a list of all the available *commands* at any point in the program. The bottom line shows the three- or four-letter description of the *command*, and the upper line shows the keystroke used to execute it.

Current Input List: is a list of the current values for all answers to the questions. This is updated every time a new value is entered or a new selection is made from a list, such as the entering or selection of a new city.

Database Set: is all of the CROP, CITY, or OPERATIONS information associated with a specific *Identifier.* For instance, for the CITY Database, this is all the climatic data for a specific city or region.

file(s): is an operating system file. See your manual or site specialist for a more complete description.

Function: is a set of routines used by RUSLE to perform housekeeping tasks. For RUSLE, these consist of the FILE Function to move information back and forth between the *Current Input List* and the *Input Files*, the EXIT Function to leave the program quickly, the HELP Function to obtain general information about the program, and the SCREEN Function to change screen coloring.

Function Line: is a list of the available *Functions,* shown on the top line of the screen. These can be reached with the {FUNC} *command* to move the cursor to the *Function Line*, using the [ARROW] keys to get the correct *Function* and option, and pressing [ENTER].

Header Line: is the top line of the *Main Screen*, and the second line down from the top of the actual screen. This shows the program version and the screen name.

Identifier: is a name or number with which to associate the information in a particular *Database Set.* When a list of the names or numbers is shown, selection of the one you want brings in all associated information.

input: is a response typed into an *Input Field* or chosen from a list in response to a program question.

Input Field: is the place where the cursor rests while waiting for your response to a question. You can respond with either an *input* or a *command.*

Input File: is a copy of the *Current Input List* that has been SAVEd. This copy is stored as a file with the name it was given in SAVEing, along with a ".rus" suffix.

Main Screen: is the central part of the screen. It contains all the questions and areas for response.

Suggestion Line: is seen within the line forming the bottom of the *Main Screen.* This line displays the suggested *command* or course of action.