PNNL-22699

U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

User Instructions for the CiderF Individual Dose Code and Associated Utility Codes

Paul W. Eslinger
Bruce A. Napier

August 2013

Pacific Northwest
NATIONAL LABORATORY

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
email: orders@ntis.gov <http://www.ntis.gov/about/form.aspx>
Online ordering: http://www.ntis.gov

PNNL-22699

User Instructions for the CiderF Individual Dose Code and Associated Utility Codes

Paul W. Eslinger
Bruce A. Napier

August 2013

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

SUMMARY

Historical activities at facilities producing nuclear materials for weapons released radioactivity into the air and water. Past studies in the United States have evaluated the release, atmospheric transport and environmental accumulation of ^{131}I from the nuclear facilities at Hanford in Washington State and the resulting dose to members of the public (Farris et al. 1994). A multi-year dose reconstruction effort (Mokrov et al. 2004) is also being conducted to produce representative dose estimates for members of the public living near Mayak, Russia, from atmospheric releases of ^{131}I at the facilities of the Mayak Production Association.

The approach to calculating individual doses to members of the public from historical releases of airborne ^{131}I has the following general steps:

- Construct estimates of releases ^{131}I to the air from production facilities.
- Model the transport of ^{131}I in the air and subsequent deposition on the ground and vegetation.
- Model the accumulation of ^{131}I in soil, water and food products (environmental media).
- Calculate the dose for an individual by matching the appropriate lifestyle and consumption data for the individual to the concentrations of ^{131}I in environmental media at their residence location.

A number of computer codes were developed to facilitate the study of airborne ^{131}I emissions at Hanford. The RATCHET code modeled movement of ^{131}I in the atmosphere (Ramsdell Jr. et al. 1994). The DECARTES code modeled accumulation of ^{131}I in environmental media (Miley et al. 1994). The CIDER computer code estimated annual doses to individuals (Eslinger et al. 1994) using the equations and parameters specific to Hanford (Snyder et al. 1994).

Several of the computer codes developed to model ^{131}I releases from Hanford are general enough to be used for other facilities. This document provides user instructions for computer codes calculating doses to members of the public from atmospheric ^{131}I that have two major differences from the Hanford modeling sequence. First, the air transport code HYSPLIT (Draxler et al. 2012) is used instead of the RATCHET code. Second, the new individual dose code CiderF replaces the older CIDER code and five auxiliary codes.

Contents

Tables

Figures

1.0 Introduction

1.1 Background

Historical activities at facilities producing nuclear materials released a number of radionuclides into the air and water. Of the radionuclides released into the air, ^{131}I is one of the major isotopes with the potential to affect the health of members of the public living near nuclear facilities. Past studies in the United States have evaluated the release, transport and environmental accumulation of ^{131}I from the facilities at Hanford in Washington State, and the resulting dose to members of the public (Farris et al. 1994). A multi-year dose reconstruction effort (Mokrov et al. 2004) is also being conducted to produce representative dose estimates for members of the public living near Mayak, Russia, from atmospheric releases of ^{131}I from the Mayak Production Association.

The general approach to calculating individual doses to members of the public from historical releases of airborne ^{131}I has the following general steps:

- Construct estimates of releases ^{131}I to the air from production facilities.
- Model the transport of ^{131}I in the air and subsequent deposition on the ground and vegetation.
- Model the accumulation of ^{131}I in soil, water and food products (environmental media).
- Calculate the dose to individuals by matching the appropriate lifestyle and consumption data for an individual to the concentrations of ^{131}I in environmental media at their residence location.

A number of computer codes were developed to model the transport and environmental accumulation of ^{131}I as part of the Hanford Environmental Dose Reconstruction Project. The equation set and associated parameters specific to Hanford are documented in (Snyder et al. 1994). The air transport code named RATCHET (Regional Atmospheric Transport Code for Hanford Emission Tracking) was used to model the movement of ^{131}I in the air (Ramsdell Jr. et al. 1994). The environmental accumulation code DESCARTES (Dynamic Estimates of Concentrations and Accumulated Radionuclides in Terrestrial Environments) has a published user's guide (Miley et al. 1994). An associated computer code for estimating annual doses to humans named CIDER (Calculation of Individual Doses from Environmental Radionuclides) also has a published user's guide (Eslinger et al. 1994).

Several of the computer codes developed for modeling ^{131}I releases at Hanford are general enough to be used at different release facilities. This document addresses a modeling sequence for computing doses from ^{131}I to members of the public that has two major differences from the published Hanford modeling sequence. First, the publicly available air transport code HYSPLIT (Draxler et al. 2012) is used instead of the RATCHET code. Second, a new individual dose code CiderF (Calculation of Individual Doses from Environmental Radionuclides in Fortran) replaces the earlier CIDER code and five auxiliary codes. However, the new CiderF code solves the same equation set as the older CIDER code.

1.2 Overview of Calculational Sequence

Identification of computer codes that can be used in a dose reconstruction analysis is provided in Table 1.1. The first eleven codes in this sequence are used to produce the concentrations of ^{131}I in environmental media that a human will interact with. Once these eleven codes are finished, the individual dose code CiderF can be run multiple times for different individuals without rerunning the previous codes. The HYSPLIT code can be run on systems utilizing either Linux or Windows operating systems. The Linux version is listed here because we run the computationally intensive code on a Linux cluster.

Table 1.1 Computer Codes Useful for a Dose Reconstruction Analysis for Airborne ^{131}I

Code	Platform	Documentation	Purpose
HYSPLIT	Linux	(Draxler et al. 2012)	Atmospheric transport of unit releases from all facilities
srcSetupRuns	Linux	This Document	Utility code to set up the data files for runs of the HYSPLIT code
QsubRuns	Linux	This Document	Utility code that schedules the large number of runs of the HYSPLIT code
AirGrid	Linux	This Document	Utility code that converts HYSPLIT output files into files registered to the suite of nodes used by the dose codes
AirCombGrid	Linux	This Document	Code that combine time-based release information from facilities with HYSPLIT outputs to estimate ^{131}I air concentration and deposition
AirCombGridView	Windows	This Document	Utility code that aids in visualizing the spatial extent of ^{131}I air concentration and deposition output by the AirCombGrid code
DESCARTES	Windows	(Miley et al. 1994)	Code that uses air concentration and deposition to calculate concentrations of ^{131}I in soil, plants, and animals (food products for humans)
ADIETP	Windows	(Miley et al. 1994)	Utility program to produce animal diets for different feeding regimes
FrostpUno	Windows	This Document	Utility program to develop frost dates needed in DESCARTES
MLOOK	Windows	(Miley et al. 1994)	Utility program to convert binary files output by DESCARTES to text format
RECIPE	Windows	(Miley et al. 1994)	Utility program to produce food production and distribution information for DESCARTES
CiderF	Windows	This Document	Calculate doses to an individual from ^{131}I
CiderView	Windows	This Document	View doses from CiderF in KML format

Some leafy vegetable and milk products consumed by individuals are not grown or produced at the location where potentially exposed individuals live. In addition, some of these products are consumed long after they are produced, thus there is time for radioactive decay to significantly reduce the ^{131}I concentration in these foods. The calculational sequence described in this report implements an optional commercial distribution system for leafy vegetables and milk products. This allows an individual to eat leafy vegetables produced on a farm grown at a different location from their residence location. In addition, a two-step milk product distribution system is implemented. First, a dairy farm produces milk (the dairy farm is called a creamery in the user guide for the DESCARTES code) and then the creamery supplies that milk to residents in other locations (called a grocery in the user guide for the DESCARTES code). This technique models the movement of milk products from farms to consumers. Consumption of locally produced leafy vegetables and milk products is also supported.

The DESCARTES and CiderF codes require external data to implement the equations listed later in sections 2.1 and 2.2. These data include releases from facilities, atmospheric transport, production and distribution of milk and leafy vegetables, growing season dates, animal diets, and information regarding human diets and lifestyle activities. These external data sets and their use with the environmental accumulation and dose codes are summarized below.

Air Transport Data – The atmospheric transport code, HYSPLIT (Draxler et al. 2012), coupled with the AirCombGrid code, provides daily integrated radionuclide air concentrations and surface deposition rates. These data were used as input to the DESCARTES and CiderF codes: the deposition rates and air concentrations are used in the DESCARTES code; the air concentrations are passed on for use in the CiderF code. The meteorological data used by the HYSPLIT code to estimate the air concentrations and deposition rates are not addressed in this report.

Facility Release Data – In this application setting, the HYSPLIT code is used to model transport of unit releases of ^{131}I from a number of facilities. Because the concentration equations are linear, estimates of historical releases from the facilities can be used to scale the transport results to obtain air concentrations and deposition rates. The release data are required inputs to the AirCombGrid code.

Animal Feeding Regimes – Radionuclide concentrations in animal products (cow and goat milk, meat, poultry, and eggs) are estimated as the product of the ingested activity and an animal to food product transfer factor. The equations in DESCARTES are based on the concept of feeding regimes to account for the various types of feeds consumed by livestock. The ADIETP code develops the daily ingestion rates of the various types of vegetation consumed by animals, based on generalized feeding regime data. These daily ingestion rates are then stored in data files. These data sets are accessed during a given run of the DESCARTES computer code and a realistic animal diet is randomly selected for each location and season.

Milk and Leafy Vegetable Production and Distribution – Data about the commercial production and distribution of milk and leafy vegetables within the project domain are used in DESCARTES and CiderF. Much of this information is collected in data files accessed by DESCARTES. These data have been published (Deonigi et al. 1994; Marsh et al. 1992) for the Hanford dose reconstruction effort and also for dose reconstruction activities for Mayak (Mokrov et al. 2007).

Human Dietary Data – Human diets are an important factor in determining doses from airborne ^{131}I. The food types modeled in DESCARTES and CiderF are leafy vegetables, other vegetables, fruit, grain, fresh dairy products, stored dairy products, meat, poultry and eggs. Dietary information and recommended lifestyle and exposure information for Mayak is provided in (Rovny et al. 2009). This information is used to produce reference diets used in CiderF.

The computer codes identified in Table 1.1 must be executed in a specified sequence. The general sequence of steps required to build a new ^{131}I concentration data set for the dose code and also generate doses for individuals is identified in the following paragraphs. Indented bullets are runs of support codes that assist the code in the numbered bullet:

1) Run the HYSPLIT code multiple times to produce air concentration and deposition values. Computer run times for this step may be days to weeks in duration.
    a) Run the srcSetupRuns code to set up the analysis directory structure and the individual HYSPLIT input files
    b) Use the QsubRuns utility program to manage execution of the HYSPLIT runs
    c) Run the AirGrid code to convert potentially large HYSPLIT output files into small files containing concentrations and depositions only at the locations used in the dose codes
    d) Run the AirCombGrid code to model releases from individual facilities and superimpose plumes from daily releases into two files containing the time history of ^{131}I air concentration and deposition rates at multiple locations
    e) Optionally run the AirCombGridView code to produce concentration or deposition maps in KML format for viewing in Google Earth (Google 2013) or raster format for viewing in ArcGis (ESRI 2013)
2) Run the DESCARTES code to produce ^{131}I concentrations in plant product media; animal product media (beef, goat milk, eggs, and poultry); herd and individual cow milk media; creamery and grocery milk media; commercial leafy vegetable media, and surface soil. Computer run times for this step range from a few minutes to a few hours.
    a) Run the FrostpUno code to generate frost date libraries for each modeled year
    b) Run the ADIETP program to generate animal diet libraries
    c) Run the FrostpUno code to generate feeding season date libraries
    d) Run the RECIPE program to generate the recipe for milk cow feeding regimes contributing to creamery milk at each node
    e) Run the RECIPE program to generate the recipe for which nodes supply milk to each creamery
    f) Run the RECIPE program to generate the recipe for which creameries supply milk to grocery stores
    g) Run the RECIPE program to generate the recipe for which nodes supply commercial leafy vegetables to consumption nodes
3) Run the CiderF code to produce dose estimates for individuals. Computer run times for this step range from a few seconds for an individual case to an hour for a map case.
    a) Optionally run the CiderView code to produce contour maps in KML format for viewing in Google Earth.

The sequential relationship between the major computational modules is illustrated in Figure 1.1.

1.3 Dose Domain

The CiderF code calculates doses to individuals from ^{131}I at a set of discrete locations called nodes. These nodes are assigned unique integer numbers and have associated data, such as concentrations of ^{131} in air, soil and food crops. These concentrations are generated by a sequence of calculations using the HYSPLIT, AirGrid, AirCombGrid and DESCARTES codes. The CiderF code uses the node numbers to identify data appropriate to a specific location, but it does not directly utilize any geographic information.

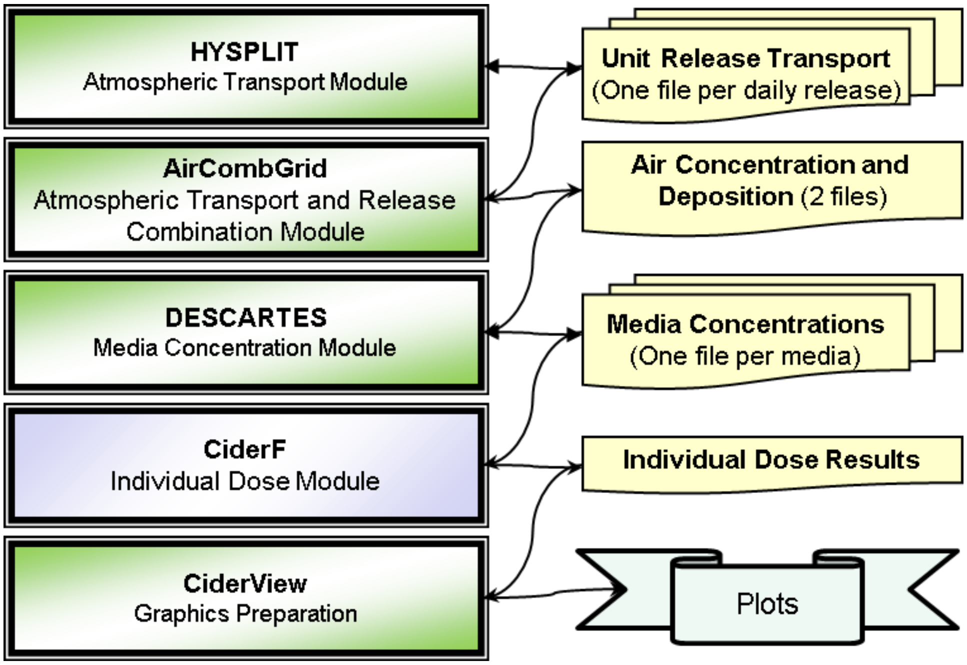

HYSPLIT
Atmospheric Transport Module
AirCombGrid
Atmospheric Transport and Release Combination Module
DESCARTES
Media Concentration Module
CiderF
Individual Dose Module
CiderView
Graphics Preparation
Unit Release Transport
(One file per daily release)
Air Concentration and Deposition (2 files)
Media Concentrations
(One file per media)
Individual Dose Results
Plots

Figure 1.1 Sequential Relationship of Major Computational Modules

The AirGrid code registers the concentrations and depositions produced by the HYSPLIT code on the set of nodes used for the dose codes. Spatial registering can be accomplished by nearest neighbor assignment or distance weighted interpolation. Typically, the nodes used in DESCARTES and CiderF are positioned so they cover a contiguous geographic region. However, they do not need to be equally spaced. If the nodes do cover a contiguous region, the CiderView program can be used to plot dose contours over the domain for different modeling cases.

To ensure the locations for the dose calculations are consistent across all of the computer codes, a grid of locations (a point on this grid is called a node) must be established. An example plot of a region with 618 nodes is provided in Figure 1.2. Several of the codes establish the number of nodes through user inputs, but the DESCARTES and FROSTPUNO codes must be compiled for a specific number of nodes. The current versions of DESCARTES and FROSTPUNO require a dose domain with exactly 618 nodes.

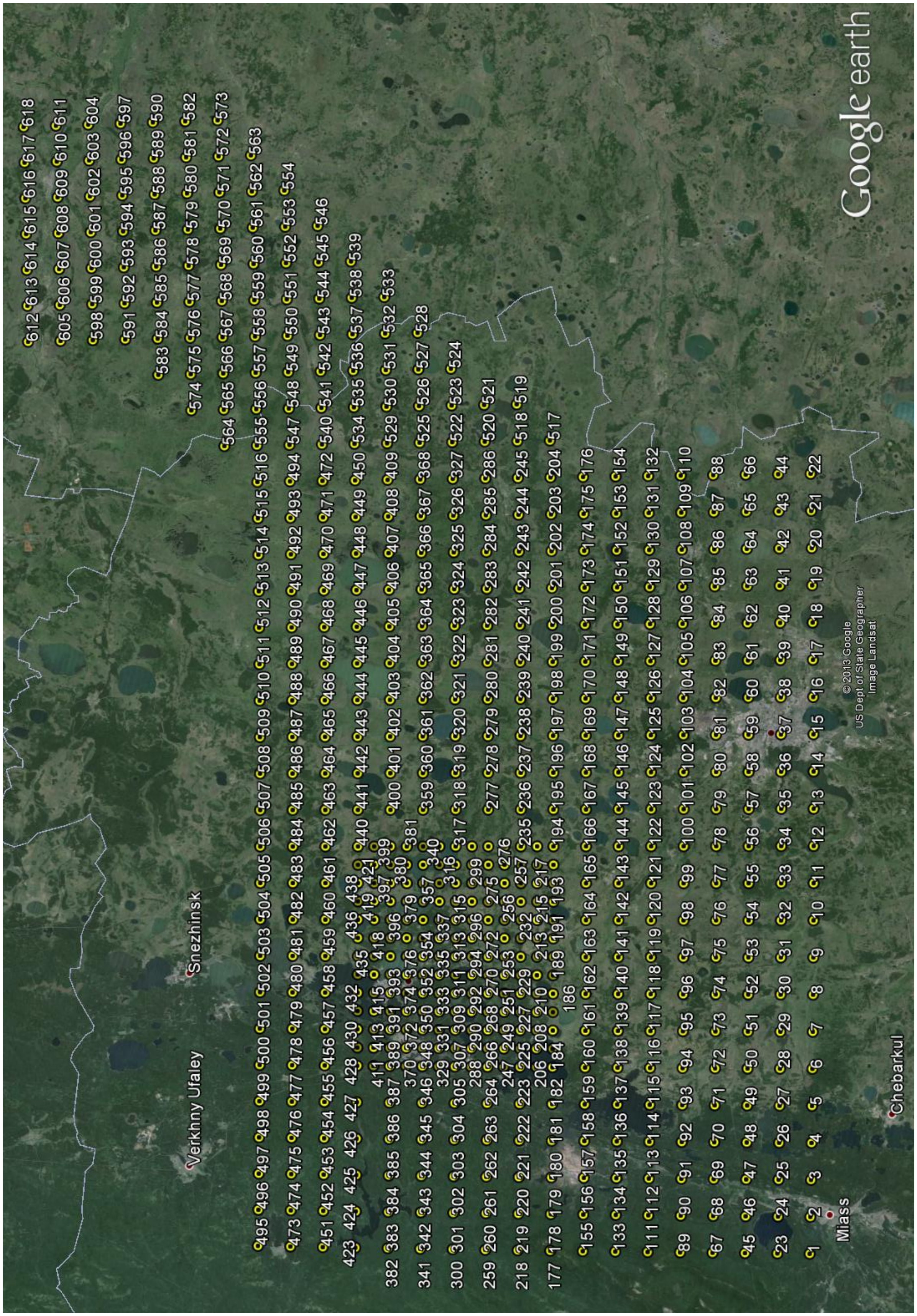


Figure 1.2 Example Node Locations in a Contiguous Geographic Domain

1.4 Other Considerations

1.4.1 Keyword File Use

Several of the codes in the processing sequence use text input files called keyword files. The general purpose of a keyword file is to provide flexible control to a computer program using human readable inputs. The general structure of keywords is defined in Section 11.0. A keyword file can be prepared with any editor that can modify ASCII files without leaving embedded control codes.

Color coding is added to some of the example keyword files in this document to enhance readability. When color is added, red text signifies comments that have no effect on code execution. Blue text signifies entries in quote strings that are input without modification.

1.4.2 Problem Size Limitations

The computer codes described in this document are capable of producing individual dose estimates in a stochastic framework, thereby propagating the uncertainty in the doses. There is always a desire to have information finely resolved in space and also ensure that uncertainty estimates have converged in a statistical sense. This desire leads to selecting many nodes and a large number of realizations, which equates to long computer run times and big data files. In terms of file sizes, the largest files typically produced in the modeling sequence are the air concentration and deposition files used for input into DESCARTES. The size of each of these files, in bytes, can be calculated from the following equation:

$$N_b = \left(14 + N_{nodes} N_{days}\right) \times \max\left(16, \left[6 + 4\,N_{real}\right]\right)$$

Where

N_{nodes} = number of nodes in the domain,
N_{days} = number of days in the time period of interest for calculating doses,
N_{real} = number of stochastic realizations.

Some of the example files discussed in this document are associated with a problem with 618 nodes, 250 realizations, and the time period from June 1, 1948, through December 31, 1972 (8980 days). Thus, the size of each of these two files is 5.58×10^9 bytes, or 5.58 gigabytes. The same two files are 88.8 megabytes in size each when 1 realization is used.

1.4.3 Assumptions for Using Surrogate Meteorological Data

The code named SHUFFL (Miley et al. 1994) performs the major function of preparing input concentration and deposition data files for DESCARTES when RATCHET (Ramsdell Jr. et al. 1994) is used for atmospheric transport. A new technique was developed for cases when the historical meteorological data needed for running RATCHET are not available. The new AirCombGrid code was developed to interface with the HYSPLIT air transport code (Draxler et al. 2012). Although archived meteorological data are available for the HYSPLIT code (GDAS 2012), the data are only available for 2005 and later years. The following assumptions are made to apply the available meteorological data to historical releases:

1) Separate runs of the HYSPLIT code are made for every releasing facility using the assumption of a unit release of contaminant.

2) The separate HYSPLIT runs for each facility are performed for every day for a period of time where meteorological are available. This period of time will typically cover several years, such as 2007 through 2011.

3) A time history of concentrations and depositions for the release from a facility on a given day of the year are computed by multiplying the concentration and deposition history derived from unit releases by the actual facility emissions.

4) The transport of historical release amounts in a month of a past year will use the unit releases calculated for the same month in a year where meteorological data are available. If a single realization (best estimate) run is desired, the effective air concentration and deposition will be the average of the values from the unit releases of the HYSPLIT runs (see step 2) for each day separately in the modeled month. For a stochastic run (more than one realization), the unit releases from step 2 are assigned randomly (on a month by month basis) to the output realization.

5) Plumes from daily releases at one or more facilities are superimposed in space and time to yield the combined effect of all the releases.

This general approach mimics the transport of historical releases by using transport representative of the air movement in the month of the year. In the best estimate case, the approach essentially uses an average χ/Q approach, where the χ/Q is based on the average of several recent years.

Leap years can cause data mismatches in this approach. If the historical release year is a leap year and the year modeled in HYSPLIT is not a leap year, then transport for releases on the 28th of February for the modeled year is replicated for the 29th of February of the historical year. Similarly, if the historical release year is a not leap year and the year modeled in HYSPLIT is a leap year, then the last day of February is dropped for the historical year.

1.4.4 Speciation of Iodine

The released ^{131}I can propagate in organic, elemental or particulate forms (Napier et al. 2008). The HYSPLIT code can model transport of all of these forms, but not the interchange between the forms. Thus, each run of the HYSPLIT code used a unit source of all three forms. The air concentration and ground deposition of each form were calculated and saved for later processing.

The speciation algorithm for ^{131}I embedded in the AirCombGrid code is taken from (Napier et al. 2008). The algorithm has two steps. First, the fraction of ^{131}I in each form is sampled uniformly from within its specific range. Second, the three fractions are normalized so the total equals one. The fractions of each form are defined as follows:

- elemental iodine, uniformly distributed on (0.10, 0.45),
- organic iodine, uniformly distributed on (0.20, 0.35), and
- particulate iodine, uniformly distributed on (0.20, 0.65).

1.4.5 Activity and Dose Units

Historically, the amount of ^{131}I released to the air was quantified in units of Ci. The suite of codes described in this document can handle activity in either Ci or Bq, however, no internal conversions are made between the two measurement systems.

The radiological activity units in HYSPLIT, DESCARTES and CiderF are actually set by the units used to describe the releases from the facilities (see Section 6.3.14). The descriptions of the equation parameters in Section 2.3 all assume that release quantities are measured in Ci. However if the releases are entered in Bq, then the user should substitute Bq for Ci in the equation descriptions.

Similarly, the dose units described in this document are identified as rad or rem. However, the dose factor data are explicitly entered in keyword files (see Section 9.7.15) and they can be adjusted for any desired input or output units. For example, dose factors can be defined as Gy/Ci for the measurement of absorbed dose or Sv/Ci for the measurement of biological risk, assuming the released activity is entered in Ci. If the released activity is entered in Bq, then the modern values for the dose units of Gy/Bq and Sv/Bq can be used directly.

2.0 Mathematical Basis

The computer code DESCARTES estimates the environmental accumulation of ^{131}I in soil, plants, and animal products at multiple locations based on the time history of air concentration and surface deposition. The new computer code CiderF replaces the older CIDER code to calculate the dose to individuals from internal and external radiation sources. The CiderF code uses the same equation set as the CIDER code. The equations implemented in DESCARTES and CIDER codes have been previously published (Snyder et al. 1994). However, they are replicated in this section for completeness.

A number of input variables (parameters) are required to execute the DESCARTES and CiderF computer codes. An alphabetical list of these parameters is provided after all of the equations are presented. Some values for input variables are developed outside of the computer codes and some are calculated as intermediate values. For example, χ, the integrated daily radionuclide air concentration, is calculated from the atmospheric transport code HYSPLIT as processed by AirCombGrid, and C_{ap}, an animal product concentration, is an intermediate value calculated in DESCARTES.

A few parameters listed in the next two sections are used in both of the DESCARTES and CiderF codes. Values for these shared parameters are input into the DESCARTES code. The DESCARTES code then passes the parameter values to CiderF in a data file, thereby ensuring that the same values are used in both sets of equations.

2.1 Environmental Accumulation (DESCARTES) Equations

The primary equations solved by the DESCARTES code are provided in this section. Descriptions of the parameters in the equations are deferred to Section 2.3. User instructions for the DESCARTES code are available in a separate document (Miley et al. 1994).

The biomass rate of change over time (January 1 – June 30) is provided in the following differential equation:

$$\frac{dB}{dt}=\frac{k_g}{2}\left[1\text{-}\cos\left(\frac{2\pi t}{t_{tot}}\right)\right]B\left[\frac{B_{max}\text{-}B}{B_{max}}\right] \qquad \text{(DES-1)}$$

The biomass rate of change over time with senescence (July 1 – December 31) is provided in the following differential equation:

$$\frac{dB}{dt}=\frac{k_g}{2}\left[1\text{-}\cos\left(\frac{2\pi t}{t_{tot}}\right)\right]B\left[\frac{B^*_{max}\text{-}B}{B^*_{max}}\right]\text{-}k_s\left(B\text{-}B_{min}\right) \qquad \text{(DES-2)}$$

The maximum biomass adjusted for senescence is provided in the following equation:

$$B^*_{max}=\frac{k_g\left(B_{max}\right)^2}{B_{max}\left(k_g\text{-}k_s\right)+k_sB_{min}} \qquad \text{(DES-3)}$$

The foliar interception fraction is provided in the following equation:

$$f_v = 1 - e^{-\alpha B} \tag{DES-4}$$

The translocation rate constant is provided in the following equation:

$$\lambda_{trans} = \lambda_{weath} \left(\frac{f_{trans}}{1 - f_{trans}} \right) \tag{DES-5}$$

The upper soil layer activity rate of change is provided in the following differential equation:

$$\frac{dQ_{usl}}{dt} = If_s - Q_{usl} \left(\lambda_{perc} + \lambda_{rad} + \lambda_{splash} \right) + Q_{ov} \lambda_{weath} - R_{resus} + R_{senc,iv} + R_{senc,ov} \tag{DES-6}$$

The root zone activity rate of change is provided in the following differential equation:

$$\frac{dQ_{rz}}{dt} = Q_{usl} \lambda_{perc} - Q_{rz} \left(\lambda_{leach} + \lambda_{rad} \right) - R_{root} \tag{DES-7}$$

The outer vegetation activity rate of change is provided in the following differential equation:

$$\frac{dQ_{ov}}{dt} = If_v - Q_{ov} \left(\lambda_{weath} + \lambda_{rad} + \lambda_{trans} \right) + Q_{usl} \lambda_{splash} - R_{senc,ov} + R_{resus} \tag{DES-8}$$

The inner vegetation activity rate of change is provided in the following differential equation:

$$\frac{dQ_{iv}}{dt} = Q_{ov} \lambda_{trans} - Q_{iv} \lambda_{rad} + R_{root} - R_{senc,iv} \tag{DES-9}$$

The deposition rate of resuspended upper soil layer material is provided in the following equation:

$$R_{resus} = \frac{V_d Q_{sl} ML}{\rho_{usl}} \tag{DES-10}$$

The rate of inner vegetation senescence (July 1 – December 31) is provided in the following equation:

$$R_{senc,iv} = \frac{Q_{iv}}{B} k_s \left(B - B_{min} \right) \tag{DES-11}$$

The rate of outer vegetation senescence (July 1 – December 31) is provided in the following equation:

$$R_{senc,ov} = \frac{Q_{ov}}{B} k_s \left(B - B_{min} \right) \tag{DES-12}$$

The rate of uptake through roots (January 1 – June 30) is provided in the following equation:

$$R_{root}=Q_{rz}\frac{CR}{\rho_{rz}}\left(\frac{dB}{dt}\right) \tag{DES-13}$$

The rate of uptake through roots (July 1 – December 31) with senescence is provided in the following equation:

$$R_{root}=Q_{rz}\frac{CR}{\rho_{rz}}\left[\frac{dB}{dt}+k_s\left(B\text{-}B_{min}\right)\right] \tag{DES-14}$$

The quantity to concentration conversion for other vegetables, grain, pasture, alfalfa and silage (subscript p) and leafy vegetables and fruit (subscripts p,iv and p,ov) is provided in the following equation:

$$C_p=\frac{Q_{iv}+Q_{ov}}{B} \quad C_{p,iv}=\frac{Q_{iv}}{B} \quad C_{p,ov}=\frac{Q_{ov}}{B} \tag{DES-15}$$

The quantity of nuclide consumed by an animal at location l and time t is provided in the following equation:

$$A_{cons}(t,l)=\sum_{v=1}^{V}R_{v\_a}\,C_v(h_v,l)\,e^{-\lambda_{rad}th_s} \tag{DES-16}$$

The animal product concentration at location l and day t is provided in the following equation:

$$C_{ap}(t,l)=TF_{ap}\left\{A_{cons}(t,l)+FS_a\left[\frac{f_{usl}Q_{usl}(t,l)}{\rho_{usl}}+\frac{f_{rz}Q_{rz}(t,l)}{\rho_{rz}}\right]\right\}+TF_{ap}(I)\left[M+\frac{S}{1000}\right] \tag{DES-17}$$

The undecayed concentration in commercially available creamery milk at creamery X and grocery milk at location l for day t is provided in the following equations:

$$\begin{gathered}C_{cream,X}(t)=\sum_{l=1}^{L(X)}f_{cream,X}(t,l)\sum_{r=1}^{4}f_r(t,l)C_r(t,l)\\ C_{groc}(t,l)=\left[\sum_{x=1}^{X(l)}f_{groc,X}(t,l)C_{cream,X}(t)\right]+f_u(t,l)C_u(t,l)+f_{other}(t,l)C_{other}(t,l)\end{gathered} \tag{DES-18}$$

The undecayed concentration in inner (subscript iv) and outer (subscript ov) compartments of commercially available leafy vegetables at location l for day t is provided in the following equations:

$$C_{comlv,iv}(t,l)=\sum_{m=1}^{M(l)} f_{lv}(l,m)C_{lv,iv}(t,m)$$
$$C_{comlv,ov}(t,l)=\sum_{m=1}^{M(l)} f_{lv}(l,m)C_{lv,ov}(t,m) \tag{DES-19}$$

2.2 Individual Dose (CiderF) Equations

The primary equations solved by the CiderF code are provided in this section. Descriptions of the parameters are provided in Section 2.3. User instructions for the CiderF code are provided in Section 9.0.

The air immersion dose equation is as follows:

$$D_{imm}(t,l)= DF_{imm}\,\chi(t,l)\left[f_{time}+(1\text{-}f_{time})\,Sh1\right]/86{,}400 \tag{CID-1}$$

The groundshine dose equation is as follows:

$$\begin{aligned} &D_{grd}(t,l)= A_{grd}f_{time} + A_{grd}(1\text{-}f_{time})\,Sh1 \\ &\text{where } A_{grd}=Q_{usl}(t,l)\,DF_{usl}+Q_{rz}(t,l)\,DF_{rz} \end{aligned} \tag{CID-2}$$

The inhalation dose equation is as follows:

$$\begin{aligned} &D_{inh}(t,l)= BR\,DF_{inh}\left[f_{time}A_{inh}+\left(1\text{-}f_{time}\right)R_{io}A_{inh}\right] \\ &\text{where } A_{inh}=\frac{\chi(t,l)}{86{,}400}+Q_{usl}(t,l)\frac{ML}{\rho_{usl}} \end{aligned} \tag{CID-3}$$

The equations for ingestion dose for foods with inner and outer compartments concentrations accounted for in combination such as other vegetables and grains (subscript veg1) or existing only as a single concentration compartment for animal products such as meat, milk and eggs (subscript ap) are as follows:

$$\begin{aligned} &D_{ing,veg1}(t,l)= DF_{ing}\sum_{p} C_p(t\text{-}th_p,l)\,R_p\,f_d\,e^{-\lambda_{rad}th_p} \\ &D_{ing,ap}(t,l)= DF_{ing}\sum_{ap} C_{ap}(t\text{-}th_p,l)\,R_p\,e^{-\lambda_{rad}th_p} \end{aligned} \tag{CID-4}$$

The equation for ingestion dose from crops with inner and outer vegetation compartments concentrations accounted for separately (leafy vegetables and fruit) is as follows:

$$D_{ing,veg2}(t,l)= DF_{ing}\sum_{p}\left[C_{p,iv}(t\text{-}th_p,l)+C_{p,ov}(t\text{-}th_p,l)\,L_{proc}\right]R_p\,f_d\,e^{-\lambda_{rad}th_p} \tag{CID-5}$$

The equations for the external exposure dose to mother, fetus and nursing baby from immersion and groundshine are as follows:

$$
\begin{aligned}
&D_{ext,bfeed}(t) = DF_{imm}A_{ext} + DF_{usl}B_{ext} + DF_{rz}C_{ext} \\
&\qquad A_{ext} = \sum_{t=\tau}^{T} \chi(t,l)\left[f_{time} + (1-f_{time})\,Sh1\right] \Big/ 86{,}400 \\
&\text{where} \quad B_{ext} = \sum_{t=\tau}^{T} Q_{usl}(t,l)\left[f_{time} + (1-f_{time})\,Sh1\right] \\
&\qquad C_{ext} = \sum_{t=\tau}^{T} Q_{rz}(t,l)\left[f_{time} + (1-f_{time})\,Sh1\right]
\end{aligned}
\qquad \text{(CID-6)}
$$

The equations for the inhalation dose to mother, fetus and nursing baby are as follows:

$$
\begin{aligned}
&D_{inh,mother}(l) = DF_{inh}A_{inh,mother}(l) \\
&D_{inh,fetus}(l) = DF_{pre}A_{inh,mother}(l) \\
&D_{inh,baby}(l) = DF_{inh,baby}A_{inh,baby}(l) + DF_{nurs}A_{inh,mother}(l) \\
&\text{where } A_{inh,x}(l) = \sum_{t=\tau}^{T} BR\,Q_{usl}(t,l)\,\frac{ML}{\rho_{usl}}\left[f_{time} + (1-f_{time})R_{io}\right] + \\
&\qquad \sum_{t=\tau}^{T} BR\,\frac{\chi(t,l)}{86{,}400}\left[f_{time} + (1-f_{time})R_{io}\right]
\end{aligned}
\qquad \text{(CID-7)}
$$

The equations for the ingestion dose to mother, fetus and nursing baby are as follows:

$$
\begin{aligned}
&D_{ing,mother}(l) = DF_{ing}A_{ing,mother}(l) \\
&D_{ing,fetus}(l) = DF_{ing,pre}A_{ing,mother}(l) \\
&D_{ing,baby}(l) = DF_{inh,baby}A_{ing,baby}(l) + DF_{nurs}A_{ing,mother}(l) \\
&\text{where } A_{ing,x}(l) = \sum_{t=\tau}^{T}\left[C_p(t,l) + C_{ap}(t,l)\right] f_d\, R_p\, e^{-\lambda_{rad} th_p}
\end{aligned}
\qquad \text{(CID-8)}
$$

2.3 Definition of Parameters for DESCARTES and CiderF Equations

The parameters used in the equations presented in Sections 2.1and 2.2 are listed alphabetically and defined below. The descriptions all assume that release quantities are measured in Ci. If the releases are entered in the codes in units of Bq, then the user should substitute Bq for Ci in the equation descriptions.

86,400 conversion factor, s/d

α empirical foliar interception constant, m^2/kg(dry)

λ_{leach} leaching rate from root zone to deep soil, d^{-1}

λ_{perc} percolation rate from upper soil layer to root zone, d^{-1}

λ_{rad} radiological decay constant, d^{-1}

λ_{splash} rain splash rate constant, d^{-1}

λ_{trans} plant translocation rate constant, d^{-1}

λ_{weath} weathering rate constant, d^{-1}

ρ_{rz} root zone soil areal density from 1 mm to 15 cm depth, kg(wet)/m^2

ρ_{usl} root zone soil areal density to a depth of 1 mm, kg(wet)/m^2

τ beginning of the exposure period of fetus, nursing baby, or lactating mother, Julian d

$\chi(t,l)$ integrated daily radionuclide air concentration on day t at location l, Ci-s/(m^3•d)

$A_{ing,x}(l)$ radionuclide intake rate via ingestion for individual x at location l, where x = fetus, nursing baby, or lactating mother, Ci

$A_{cons}(t,l)$ animal radionuclide consumption rate on day t at location l, Ci/d

$A_{ing,x}(l)$ radionuclide intake rate via ingestion for individual x at location l, where x = fetus, nursing baby, or lactating mother, Ci

$A_{inh,x}(l)$ radionuclide intake rate via inhalation for individual x at location l, where x = fetus, nursing baby, or lactating mother, Ci

B current daily biomass, kg(dry)/m^2

B_{max} maximum potential biomass, kg(dry)/m^2

B^*_{max} maximum biomass adjustment factor, kg(dry)/m^2

B_{min} minimum (winter) biomass, kg(dry)/m^2

BR age dependent breathing rate, m^3/d

$C_{ap}(t,l)$ animal product radionuclide concentration on day t at location l, where ap = milk, beef, chicken, eggs, Ci/L (milk) or Ci/kg(wet) (others)

$C_{cream,X}(t)$ radionuclide concentration in milk from creamery X on day t, Ci/L

$C_{groc}(t,m)$ radionuclide concentration for cow milk at all grocery stores at location m on day t, Ci/L

$C_{commlv,iv}(t,l)$ radionuclide concentration in the inner compartment of commercially available leafy vegetables on day t and location l, Ci/kg(wet)

$C_{comlv,ov}(t,l)$ radionuclide concentration in the outer compartment of commercially available leafy vegetables on day t at location l, Ci/kg(wet)

$C_{lv,iv}(t,l)$ radionuclide concentration in the inner compartment of leafy vegetables on day t at location l, Ci/kg(wet)

$C_{lv,ov}(t,l)$ radionuclide concentration in the outer compartment of leafy vegetables on day t at location l, Ci/kg(wet)

C_p radionuclide concentration in food or feed crop, where p = other vegetables, grain, pasture, alfalfa and silage Ci/kg(dry)

$C_{p,iv}$ radionuclide concentration in the inner vegetation compartment, where p = leafy vegetables or fruit Ci/kg(dry)

$C_{p,ov}$ radionuclide concentration in the outer vegetation compartment, where p = leafy vegetables or fruit Ci/kg(dry)

$D_{imm}(t,l)$ air immersion dose on day t at location l, $rad_{thyroid}$ or rem_{EDE}

$C_r(t,l)$ radionuclide concentration in milk from a herd of cows fed according to feeding regime r at location 1 on day t, Ci/L

$C_u(t,l)$ radionuclide concentration in milk of unknown origin at grocery location l on day t (computed as milk from a herd of cows fed according to feeding regime r at location 1 on day t), Ci/L

CR plant to soil concentration ratio, $Ci/kg_{vegetation}$(dry) per Ci/kg_{soil}(wet)

$D_{ext,bfeed}$ external exposure dose to the fetus, nursing baby or lactating mother from immersion and groundshine, $rad_{thyroid}$ or rem_{EDE}

$D_{grd}(t,l)$ dose from groundshine on day t at location l, $rad_{thyroid}$ or rem_{EDE}

$D_{ing}(t,l)$ ingestion dose from local or commercial foods on day t at location l, $rad_{thyroid}$ or rem_{EDE}

$D_{ing,ap}(t,l)$ ingestion dose from animal product ap on day t at location l, where ap = beef, poultry, eggs, or milk, $rad_{thyroid}$ or rem_{EDE}

$D_{ing,veg1}(t,l)$ ingestion dose from local food crops with a single compartment on day t at location l, $rad_{thyroid}$ or rem_{EDE}

$D_{ing,veg2}(t,l)$ ingestion dose from local food crops with a two compartments on day t at location l, $rad_{thyroid}$ or rem_{EDE}

$D_{ing,x}(l)$ ingestion dose to individual x at location l during the exposure period, where x = fetus, nursing baby or lactating mother, $rad_{thyroid}$ or rem_{EDE}

$D_{inh}(t,l)$ inhalation dose from local or commercial foods on day t at location l, $rad_{thyroid}$ or rem_{EDE}

$D_{inh,x}(l)$ inhalation dose to individual x at location l during the exposure period, where x = fetus, nursing baby or lactating mother, $rad_{thyroid}$ or rem_{EDE}

$D_{imm}(t,l)$ air immersion dose on day t at location l, $rad_{thyroid}$ or rem_{EDE}

DF_{imm} immersion dose rate factor, $rad_{thyroid}$/d per Ci/m^3 or rem_{EDE}/d per Ci/m^3

DF_{ing} ingestion dose factor, $rad_{thyroid}/Ci_{ingested}$ or $rem_{EDE}/ Ci_{ingested}$

DF_{inh} inhalation dose factor, $rad_{thyroid}/Ci_{inhaled}$ or $rem_{EDE}/ Ci_{inhaled}$

DF_{pre} dose factor relating the dose to the fetus (prenatal) from intakes of the mother, $rad_{thyroid,fetus}/Ci_{intake,mother}$

DF_{rz} dose rate factor for radionuclides in the root zone soil, rem/d per Ci/m^2

DF_{usl} dose rate factor for radionuclides in the upper soil layer or surface activity, rem/d per Ci/m^2

$f_{cream,X}(t,l)$ fraction of milk at creamery X that came from a dairy at location l on day t, dimensionless

f_d dry-weight to wet-weight conversion factor, kg(dry)/kg(wet)

$f_{groc,X}(t,l)$ fraction of grocery milk at location l on day t that came from creamery X, dimensionless

$f_{lv}(l,m)$ fraction of leafy vegetables purchased from a retail or wholesale distributor at location l that was grown at location m, dimensionless

$f_{other}(t,l)$ fraction of grocery milk at location l on day t that came from a known source outside the study region, dimensionless

$f_r(t,l)$ fraction of milk from a herd of cows fed according to regime r from a dairy at location l on day t, dimensionless

f_{rz} fraction of root –zone soil consumed by an animal in a day, equal to 1-f_{usl}, dimensionless

f_s soil deposition fraction, equal to 1-f_v, dimensionless

FS_a animal soil ingestion rate, where a = poultry, cattle or goat, kg(wet)/d

f_{time} fraction of days spent outdoors, dimensionless

f_{trans} fraction of outer vegetation deposition that translocates (moves) to the inner vegetation compartment, dimensionless

$f_u(t,l)$ fraction of milk at grocery location l on day t that came from an unknown source, dimensionless

f_{usl} fraction of upper-soil layer soil consumed by an animal in a day, equal to 1-f_{rz}, dimensionless

f_v vegetation foliar interception fraction, dimensionless

h_v Julian day on which feed type v was most recently harvested

I areal deposition rate (from air transport model), Ci/(m^2•d)

k_g growth rate constant, d^{-1}

k_s senescence rate constant, d^{-1}

l location of interest

L_{proc} food processing retention fraction, dimensionless

M surface area of cattle manger directly exposed to atmospheric deposition, m^2

ML mass loading factor for local soil in air, kg/m^3

Q_{iv} activity in the inner vegetation compartment, Ci/m^2

Q_{ov} activity in the outer vegetation compartment, Ci/m^2

Q_{rz} activity in the rooting zone soil compartment, Ci/m^2

Q_{usl} activity in the upper soil layer, Ci/m^2

R_{bmilk} breast milk consumption rate for a nursing infant, L/d

R_{io} ratio of indoor air to outdoor air activity, dimensionless

R_p food product consumption rate, where food crop or animal product, kg(wet)/d for all foods except milk and L/d for milk

R_{resus} rate of radionuclide redeposition on vegetation from resuspension of soil, Ci/(m^2•d)

$R_{senc,iv}$ rate of radionuclide transfer from the inner vegetation compartment of plants to the soil by vegetation senescence, Ci/(m^2•d)

$R_{senc,ov}$ rate of radionuclide transfer from the outer vegetation compartment of plants to the soil by vegetation senescence, Ci/(m^2•d)

R_{root} rate of radionuclide uptake through roots, Ci/(m^2•d)

R_{v_a} quantity of feed type v than animal a consumes in a day, kg(dry)/d

S stock tank dilution factor, m^{-1}

Sh1 shielding factor for semi-infinite plumes, dimensionless

t day of interest, Julian day

T end of the exposure period in the year being evaluated, 365 or 366 (leap years)

t_{tot} total number of days in the year being evaluated, 365 or 366 (leap years)

TF_{ap} animal product transfer factor, where ap = goat milk, individual cow milk, herd cow milk, beef, poultry or eggs, d/L for milk and d/kg(wet) for all others

th_p holdup time from collection or harvest to consumption, where p is a food crop or animal product, d

th_s holdup time for stored feed crops, d

V_d local deposition velocity of resuspended soil back to soil or vegetation, m/d

3.0 srcSetupRuns – User Instructions

The purpose of the srcSetupRuns utility code is to automate the setup of directories and input files needed for multiple runs of the HYSPLIT code. The srcSetupRuns code is the first code to be executed when generating a new suite of files containing radionuclide concentrations in environmental media. Both serial and parallel versions of HYSPLIT are available (HYSPLIT 2013). The srcSetupRuns code produces a sequence of commands that can be used to execute the serial version of HYSPLIT.

The srcSetupRuns code was developed for other purposes, but it is general enough to set up HYSPLIT runs for dose reconstruction purposes. Only the functions pertinent for applications suitable for dose reconstruction activities are described here.

3.1 How the Code Is Invoked

The srcSetupRuns code can run under either the Windows 7 or Linux operating systems. Under the Windows 7 operating system, the code executes at a command prompt. A run of srcSetupRuns is initiated by entering the following command line:

```
srcSetupRuns "Keyfilename"
```

Under the Linux operating system, srcSetupRuns is executed through the following Bourne Shell or C Shell commands:

```
srcSetupRuns "Keyfilename"
```

For these commands, srcSetupRuns is the name of the executable program, and “Keyfilename” is the name of a controlling keyword file. Both the name of the executable program and the keyword file may contain path information. The keyword file contains text control information. If the srcSetupRuns code cannot open the keyword file it will write an error message to the standard output device and terminate execution.

3.2 Input and Output Files

The srcSetupRuns code reads three primary input files (control keywords, meteorological file mapping, and facility releases) and writes a large number of output files. These files are described in the following sections.

3.2.1 Input Files

Control of a run of the srcSetupRuns code is achieved by using an input keyword file. An example keyword file is provided in Table 3.1. Detailed descriptions of the individual keywords are described in Section 3.3. Although the srcSetupRuns code will operate under either of the Windows 7 or Linux operating systems, the example input file is prepared for the Linux operating system. The only input differences in this file between the two operating systems is the entry for the OPERATE keyword and the path names for files. The entries in the file that are colored red are comments ignored by the program. Entries colored blue are quote strings that will be read without modification.

Table 3.1 Example srcSetupRuns Keyword File

```
FILE REPORT "Mayak.rpt"
USER "Paul W. Eslinger"
! Define the operating system
OPERATE LINUX ! WINDOWS
! Define mapping to meteorlogical data files
FILE METMAP="/files0/atm/metdata/gdas1/Date_File_Map.txt"
! Define release locations
!FILE LOCATION "/files0/atm/Mayak/setup/Station_List_Reactor80.txt"
!FILE LOCATION "/files0/atm/Mayak/setup/Station_List_Reactor95.txt"
FILE LOCATION "/files0/atm/Mayak/setup/Station_List_Plants150.txt"
! Define release dates
DATE BEGIN="2006-12-29 00:00:00"  END="2011-12-31" DAY
!
! Define information for Hysplit
OPTION HYSPLIT
FILE RUNDIR="/files0/atm/Mayak/Hysplit/" ! Base directory for Hysplit output
HYSPLIT
  HOURS    =   96       ! Number of hours (integer) for the transport run
  CONHOUR =   24       ! Number of hours for concentration accumulation
  SRCHOUR =   24       ! Number of hours for the source
  SRCRATE = 0.0416667  ! Release rate per hour
  DECAY    =   8.0207   ! Halflife of decay in days (omit for no decay)
  LATDEL   =   0.025    ! Latitude  spacing of Hysplit concentration grid
  LONDEL   =   0.025    ! Longitude spacing of Hysplit concentration grid
  LATSPAN =    4.0     ! Latitude  span of Hysplit concentration grid
  LONSPAN =    8.0     ! Longitude span of Hysplit concentration grid
  CENTERDE = 0.025     ! Rounding delta for output coordinate center
  PARTICLE=  250000    ! Number of particles to use
  CONLEVEL= 100.0      ! Number of meters in lower concentration level
  SRCHIGH =  150.0     ! Source height in meters
  EXEC = "/share/apps/hysplit.0113/trunk/exec/hycs_std" ! Hysplit executable
  ASCDATA = "/files0/atm/Mayak/setup/" ! ASCDATA.CFG file (replicate)
  SETUP    = "/files0/atm/Mayak/setup/" ! SETUP.CFG file (mostly replicate)
!
! Define information for Hysplit binary to ascii conversion
OPTION ASCII
FILE ASCEXEC = "/share/apps/hysplit/exec/con2asc" ! Name of executable file
!
END
```

The srcSetupRuns code creates directories if needed and sets up input files for runs of the HYSPLIT code in a number of subdirectories. The HYSPLIT code uses meteorological data (GDAS 2012) that can be downloaded from a web server to your own computer system. The srcSetupRuns code uses a simple file to map meteorological data files to the dates for each specific run of the HYSPLIT code. Part of an example map file is provided in Table 3.2. Each line of this file contains a range of dates, a directory where the meteorological data file resides, and the specific meteorological data file name. File naming conventions are explained in information available on the Air Resources Laboratory website (NOAA 2013). Lines with the entry … denote deleted items.

Table 3.2 Excerpts from a Map File for Meteorological Data

```
"2006-12-29","2006-12-31","/files0/atm/metdata/gdas1/","gdas1.dec06.w5"
"2007-01-01","2007-01-07","/files0/atm/metdata/gdas1/","gdas1.jan07.w1"
"2007-01-08","2007-01-14","/files0/atm/metdata/gdas1/","gdas1.jan07.w2"
"2007-01-15","2007-01-21","/files0/atm/metdata/gdas1/","gdas1.jan07.w3"
"2007-01-22","2007-01-28","/files0/atm/metdata/gdas1/","gdas1.jan07.w4"
"2007-01-29","2007-01-31","/files0/atm/metdata/gdas1/","gdas1.jan07.w5"
"2007-02-01","2007-02-07","/files0/atm/metdata/gdas1/","gdas1.feb07.w1"
"2007-02-08","2007-02-14","/files0/atm/metdata/gdas1/","gdas1.feb07.w2"
"2007-02-15","2007-02-21","/files0/atm/metdata/gdas1/","gdas1.feb07.w3"
"2007-02-22","2007-02-28","/files0/atm/metdata/gdas1/","gdas1.feb07.w4"
"2007-03-01","2007-03-07","/files0/atm/metdata/gdas1/","gdas1.mar07.w1"
...
```

The srcSetupRuns code also requires location information for the releasing facilities. This information is provided in a text file. An example file for two facilities is provided in Table 3.3. Multiple release facilities may be entered in this file. The only constraint is that all facilities in a single release file must have the same release height (see the SRCHIGH modifier description in Section 3.3.4). Facilities with different heights can be handled in different runs of the srcSetupRuns code. The first entry on each line is a unique alphanumeric facility ID that is up to 4 characters in length, enclosed in double quotation marks. The second and third entries are the latitude and longitude of the facility in decimal degrees. The fourth entry is a brief facility description enclosed in double quotation marks. The facility ID will be used in path names for files, thus special characters should be avoided.

Table 3.3 Example Release Station File

```
"B",55.68805556,60.79888889,"B Plant Stacks"
"DB",55.69722222,60.80416667,"DB Plant Stacks"
```

If multiple runs of srcSetupRuns code are made, the user must make a copy of the output run command file (see Table 3.5), which is always named srcRunCommands.cmd, so previous run commands are not overwritten. Multiple saved run command files can be combined with a text editor before using the QsubRuns utility program to control HYSPLIT execution.

3.2.2 Output Files

The srcSetupRuns code always generates a report file. An example report file is provided in Table 3.4 for a successful run of the code. Error messages for run time errors trapped by the srcSetupRuns code are written to this file.

Table 3.4 Example Report File for srcSetupRuns

```
                    srcSetupRuns    Version M.01.001
                     Last Modified on 24 Apr 2013
-------------------------------------------------------------------
               Set up Hysplit and Postprocessing Runs
-------------------------------------------------------------------

    Current Run ID = 20130424114037   User Name = Paul W. Eslinger
         System Date = 04-24-2013   System Time = 11:40:37.389

      The software used to generate this output is experimental
         and has not been formally tested or peer reviewed.
```

```
Met data mappings read for 372 files from
File: /files0/atm/metdata/gdas1/Date_File_Map.txt

Definitions read for 2 locations from
File: /files0/atm/Mayak/setup/Station_List_Plants150.txt

Message originating in routine srcSetupRuns at 04/24/2013 on 11:42:45.754
Message: Normal Termination
```

The srcSetupRuns code also generates a file containing the commands needed to execute all of the HYSPLIT runs. An example of the generated commands for the Linux operating system is provided in Table 3.5. A large number of HYSPLIT runs are required to model daily releases from every facility. An application using five years of meteorological data and releases from eight facilities requires 14,632 separate HYSPLIT runs. A utility program (QsubRuns) is provided to manage the execution of these runs on a multiple CPU Linux cluster. Lines with the entry … denote deleted items.

Table 3.5 Excerpts from the srcRunCommand.cmd file Generated by srcSetupRuns

```
cd /files0/atm/Mayak/Hysplit/20100203/OK/
/share/apps/hysplit.0113/trunk/exec/hycs_std
/share/apps/hysplit/exec/con2asc -icdump -x -s
cd /files0/atm/Mayak/Hysplit/20100204/AV1/
/share/apps/hysplit.0113/trunk/exec/hycs_std
/share/apps/hysplit/exec/con2asc -icdump -x -s
cd /files0/atm/Mayak/Hysplit/20100204/AV2/
/share/apps/hysplit.0113/trunk/exec/hycs_std
/share/apps/hysplit/exec/con2asc -icdump -x –s
...
```

The srcSetupRuns code generates three input files (named CONTROL, SETUP.CFG and ASCDATA.CFG) needed for every run of the HYSPLIT code. See the description of the RUNDIR modifier in Table 3.8 for an explanation of the subdirectory structure used to hold all of the HYSPLIT runs. The format of the three input files are defined in the HYSPLIT documentation (Draxler et al. 2012), so a description is not provided in this document.

One example HYSPLIT control file is provided in Table 3.6 to illustrate the fact that three forms of ^{131}I are tracked. The organic form is denoted by the species name “Orgn”, the elemental form is denoted by the species name “Elem” and the particulate form is denoted by the species name “Part”. All three forms are modeled in the code and the outputs include ground level deposition and air concentration near the ground.

Table 3.6 Example HYSPLIT CONTROL File

```
06 12 29 00
1
 55.70194   60.78306    95.0
96
0
10000.0
2
/files0/atm/metdata/gdas1/
gdas1.dec06.w5
/files0/atm/metdata/gdas1/
gdas1.jan07.w1
3
Orgn
```

```
 4.166670E-02
24
00 00 00 00 00
Elem
 4.166670E-02
24
00 00 00 00 00
Part
 4.166670E-02
24
00 00 00 00 00
1
55.5 61.0
    0.02500    0.02500
    4.00000    8.00000
./
cdump
2
   0.000  100.000
00 00 00 00 00
00 00 00 00 00
00 24 00
3
0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0
 8.020700E+00
0.0
0.0 0.0 0.0
0.01 0.0 0.0 0.0 0.0
0.08 0.0 0.0
 8.020700E+00
0.0
1.0 1.0 1.0
0.001 0.0 0.0 0.0 0.0
0.0 4.0E+04 5.0E-05
 8.020700E+00
0.0
```

3.3 Keyword Definitions for the srcSetupRuns Code

The keywords for the srcSetupRuns code can generally be entered in any order. However, the last keyword in the file must be the END keyword. The keywords used in the srcSetupRuns code are identified in alphabetical order in Table 3.7.

Table 3.7 Summary of Keywords Used in the srcSetupRuns Code

Keyword	Section	Purpose
DATE	3.3.1	The DATE keyword is used to define the range of dates for running the HYSPLIT code.
END	3.3.2	The END keyword signifies the end of data in the file.
FILE	3.3.3	The FILE keyword is used to enter names for input and output files.
HYSPLIT	3.3.4	The HYSPLIT keyword is used to enter data specific to running the HYSPLIT code.
OPERATE	3.3.5	The OPERATE keyword is used to identify the operating system the program is running under.

Keyword	Section	Purpose
OPTION	3.3.6	The OPTION keyword is used to identify the major execution options in the code.
USER	3.3.7	The USER keyword is used to enter the name of the user running the code.
VERBOSE	3.3.8	The optional VERBOSE keyword can be used to increase the amount of information written to standard out as the run progresses.

3.3.1 DATE Keyword

The DATE keyword is used to define the range of dates for running the HYSPLIT code. Input files for the HYSPLIT code will be generated for every day from the beginning date through the end date. The following is this keyword’s syntax:

```
DATE BEGIN=“YYYY-MM-DD HH:00:00” END=“YYYY-MM-DD” DAY
```

The date and time in the quote string associated with the BEGIN modifier indicates the date for the first HYSPLIT runs. Embedded spaces, dashes and colons within the quote string are significant. The date in the quote string associated with the END modifier indicates the date for the last HYSPLIT runs. The data signified by YYYY is the four digit year. The data signified by MM is a two digit month. A leading 0 must be used for months before October. For example, February would be indicated by 02. The data signified by DD is a two digit day. A leading 0 is identified. The first day of the month is indicated by 01. The data signified by HH is a two digit hour. A leading 0 is required for hours less than 10. Midnight is denoted by 00, noon by 12 and a 24 hour clock is used. Dates and hours are universal coordinated times (UTC), not local times.

The modifier DAY is required on this keyword. This keyword has other options that are not applicable for HYSPLIT runs performed for the purpose of dose reconstruction. Those options are not described here. The purpose of the DAY modifier is to tell the code to set up one HYSPLIT run per calendar day.

The following keyword example generates subdirectories and populates them with HYSPLIT input files for every day from December 29, 2006, through December 31, 2011. Each run will start the release at midnight (UTC).

```
DATE BEGIN=“2006-12-29 00:00:00” END=“2011-12-31” DAY
```

3.3.2 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword’s syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

3.3.3 FILE Keyword

The FILE keyword is used to enter names for input and output files. The following is this keyword's syntax:

```
   FILE modifier "quote"
```

The file names are entered in quote strings. Path names up to 256 characters long (name length limitation in Windows 7) are supported. At least one FILE keyword is required for every run of the code. The modifiers associated with the FILE keyword are described in Table 3.8.

Table 3.8 Modifiers Associated with the FILE Keyword for srcSetupRuns

Modifier	Description
ASCEXEC	The quote string associated with the ASCEXEC modifier provides the name (including path) of the program that converts HYSPLIT binary concentration files to ASCII format.
LOCATION	The quote string associated with the LOCATION modifier provides the file name of the text file containing the locations of release facilities (see Table 3.3).
METMAP	The quote string associated with the METMAP modifier provides the name of the input text file containing the map file for meteorological data needed by the HYSPLIT code (see Table 3.2).
REPORT	The quote string associated with the REPORT modifier provides the file name for the output text file that will contain information about the progress of the run. All run time error messages are directed to this file (see Table 3.4).
RUNDIR	The quote string associated with the RUNDIR modifier provides the path that forms the base of the file names for all HYSPLIT runs. The runs are in a directory structure. The top level is the path (including the terminating slash) identified on this keyword. The first level subdirectory under the top level uses the range of dates for the HYSPLIT runs. The second level of subdirectories uses the ID's for every releasing facility.

The following five entries define the required data files:

```
   FILE REPORT   "Mayak.rpt"
   FILE METMAP   "/files0/atm/metdata/gdas1/Date_File_Map.txt"
   FILE LOCATION "/files0/atm/Mayak/setup/Station_List_Plants150.txt"
   FILE RUNDIR   "/files0/atm/Mayak/Hysplit/"
   FILE ASCEXEC  "/share/apps/hysplit/exec/con2asc"
```

An example directory structure on a Linux system for three days of operation for two facilities "A" and "BB" and a RUNDIR entry of of "/files0/atm/Mayak/Hysplit/" is the following:

```
   /files0/atm/Mayak/Hysplit/20070114/A
   /files0/atm/Mayak/Hysplit/20070114/BB
   /files0/atm/Mayak/Hysplit/20070115/A
   /files0/atm/Mayak/Hysplit/20070115/BB
   /files0/atm/Mayak/Hysplit/20070116/A
   /files0/atm/Mayak/Hysplit/20070116/BB
```

3.3.4 HYSPLIT Keyword

The HYSPLIT keyword is used to enter data specific to running the HYSPLIT code. The following is this keyword's syntax:

```
HYSPLIT [modifier1 ("quote1"|N1)] ... [modifier15 ("quote15"|N15)]
```

Only a single HYSPLIT keyword is allowed. The modifiers associated with the HYSPLIT keyword are identified in Table 3.9. The description explains the purpose of the numerical value or quote string associated with each modifier.

Table 3.9 Modifiers Associated with the HYSPLIT Keyword for srcSetupRuns

Modifier	Description
ASCDATA	The quote string associated with the ASCDATA modifier must contain the path and name of the ASCDATA.CFG file needed by HYSPLIT. This file is copied to the directories where HYSPLIT will execute. The file is not modified.
CENTERDE	The output grid for concentration and deposition data from HYSPLIT is nominally centered at the coordinates of the release facility. The grid origin coordinates are rounded to the nearest increment of the numerical value associated with the CENTERDE modifier. This approach forces the HYSPLIT outputs for all runs to be registered to a common grid. This entry and the numerical entries associated with the LATDEL and LONDEL modifiers should all be identical. Units for this entry are decimal degrees.
CONHOUR	The HYSPLIT code outputs contaminant concentrations and depositions that are averaged over a time interval. The numerical entry associated with the CONHOUR modifier specifies the (integer) number of hours in the interval. The only valid entry for this specific application is 24, resulting in one concentration and deposition output per day.
CONLEVEL	The HYSPLIT code can output the air concentrations at different levels. The numerical value associated with the CONLEVEL modifier identifies the height (meters) of the lowest level used to represent ground level air concentrations. A value of 100 meters is suggested.
DECAY	The numerical value associated with the DECAY modifier provides the half-life in days for a radioactive contaminant. This modifier may be omitted if no radioactive decay is modeled.
EXEC	The quote string associated with the EXEC modifier must contain the path and name of the HYSPLIT executable.

Modifier	Description
HOURS	The numerical value associated with the HOURS modifier identifies the (integer) number of hours to track the release in HYSPLIT. The modeling domain is small enough releases can exit the domain within the first day under some weather conditions. For this application, a value of 96 hours is recommended. The value entered should be an integer multiple of the numerical entry associated with the CONHOUR modifier. Under the correct conditions, a plume released on earlier days can return to the release site. However, the assumption is made that concentrations in a returning plume are negligible relative to concentrations from current releases.
LATDEL	The numerical value associated with the LATDEL modifier identifies the latitude spacing of the HYSPLIT concentration grid (decimal degrees). A latitude delta of 0.025 degrees gives a north-south spacing of about 2.75 km.
LATSPAN	The numerical value associated with the LATSPAN modifier identifies the latitude span of the HYSPLIT concentration domain (decimal degrees). A latitude span of 4 degrees covers the domain about 220 km in the north and south directions from the release point.
LONDEL	The numerical value associated with the LONDEL modifier identifies the longitude spacing of the HYSPLIT concentration grid (decimal degrees). A longitude delta of 0.025 degrees gives an east-west spacing of about 2.75 km near the equator and about 1.4 km if the latitude of the release point is near 60 degrees.
LONSPAN	The numerical value associated with the LONSPAN modifier identifies the longitude span of the HYSPLIT concentration domain (decimal degrees). A longitude span of 8 degrees covers the domain about 220 km in the north and south directions from the release point if the latitude is near 60 degrees.
PARTICLE	The numerical value associated with the PARTICLE modifier identifies the number of particles to use in HYSPLIT. More particles result in better resolution of the concentration and deposition values. A value of 250,000 or more particles is recommended.
SETUP	The quote string associated with the SETUP modifier must contain the path and name of a SETUP.CFG file needed by HYSPLIT. This file is copied to the directories where HYSPLIT will execute. The number of particles to use in the runs is modified in this file, if necessary. Other entries in the file are not modified.
SRCHIGH	The numerical value associated with the SRCHIGH modifier identifies the source height in meters. For example, releases from a stack would use the height of the stack.
SRCHOUR	The numerical value associated with the SRCHOUR modifier identifies the (integer) number of hours that the source is active. For this application, daily releases are modeled. The number 24 should be entered.

Modifier	Description
SRCRATE	The numerical value associated with the SRCRATE modifier identifies the hourly release rate for the source. A unit source is defined by entries such that SRCHOUR*SRCRATE=1.0. Thus, the appropriate entry is 0.0416667.

The following example HYSPLIT keyword contains entries for all required modifiers:

```
HYSPLIT
     HOURS   =   96       ! Number of hours (integer) for the transport run
     CONHOUR =   24       ! Number of hours for concentration accumulation
     SRCHOUR =   24       ! Number of hours for the source
     SRCRATE = 0.0416667  ! Release rate per hour
     DECAY   =   8.0207   ! Halflife of radioactive decay in days
     LATDEL  =   0.025    ! Latitude  spacing of Hysplit concentration grid
     LONDEL  =   0.025    ! Longitude spacing of Hysplit concentration grid
     LATSPAN =    4.0     ! Latitude  span of Hysplit concentration grid
     LONSPAN =    8.0     ! Longitude span of Hysplit concentration grid
     CENTERDE = 0.025     ! Rounding delta for output coordinate center
     PARTICLE=  250000    ! Number of particles to use
     CONLEVEL= 100.0      ! Number of meters in lower concentration level
     SRCHIGH =  150.0     ! Source height in meters
     EXEC    = "/share/apps/hysplit.0113/trunk/exec/hycs_std" ! Hysplit executable
     ASCDATA = "/files0/atm/Mayak/setup/" ! ASCDATA.CFG file (replicate)
     SETUP   = "/files0/atm/Mayak/setup/" ! SETUP.CFG file (mostly replicate)
```

3.3.5 OPERATE Keyword

The OPERATE keyword is used to identify the operating system the program is running under. The following is this keyword’s syntax:

```
OPERATE [WINDOWS|LINUX]
```

The following keyword selects the Linux operating system.

```
OPERATE LINUX
```

The following keyword selects the Windows operating system.

```
OPERATE WINDOWS
```

3.3.6 OPTION Keyword

The OPTION keyword is used to identify the major execution options in the code. The following is this keyword’s syntax as it applies to a dose reconstruction problem:

```
OPTION (HYSPLIT) (ASCII)
```

The following keyword selects options to set up HYSPLIT runs and also convert the HYSPLIT output file from binary format to text format. Both options are required for this application. The following keyword selects both options.

```
OPTION HYSPLIT ASCII
```

The above keyword has the same effect as the following two keywords:

```
OPTION HYSPLIT
OPTION ASCII
```

3.3.7 USER Keyword

The USER keyword is used to identify the user of the program. The user name will be written to output files for labeling purposes. The program will error terminate if the user name is not supplied. The following is this keyword's syntax:

```
USER "quote"
```

The user name is entered in a quote string and user names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code:

```
USER "John Q. Public"
```

There are no modifiers associated with the USER keyword.

3.3.8 VERBOSE Keyword

The optional VERBOSE keyword can be used to increase the amount of information written to standard out as the run progresses. It has no effect on the actual contouring calculations. The following is this keyword's syntax:

```
VERBOSE
```

There are no modifiers or quote strings associated with the VERBOSE keyword.

4.0 QsubRuns – User Instructions

The srcSetupRuns utility code sets up the directories and input files needed for multiple runs of the HYSPLIT code. A utility code named QsubRuns is available for Linux systems that are configured as a cluster with a controlling node and auxiliary compute nodes. This utility program will schedule and monitor execution all of the HYSPLIT runs using batch scripts created for the sbatch command (sbatch 2013). No comparable utility program is currently available for Windows systems.

4.1 How the Code Is Invoked

Under the Linux operating system, QsubRuns is executed through the following Bourne Shell or C Shell commands:

```
QsubRuns.exe
```

The executable is named QsubRuns.exe. A path to the executable typically is used. If desired, this utility code can also be run in a batch queue.

4.2 Inputs

The QsubRuns code obtains five separate inputs from standard input. Typically these inputs are entered in a file and then input is redirected to the file. Example inputs are provided in Table 4.1.

Table 4.1 Typical Inputs for the QsubRuns Utility Code

```
3
Mayak
/files0/atm/Mayak/setup/
/files0/atm/Mayak/setup/nodelist.txt
/files0/atm/Mayak/setup/srcRunCommands.cmd
```

The first input is the number of commands associated with each HYSPLIT run in the file name srcRunCommand.cmd produced by the srcSetupRuns code. Excerpts from that file are provided in Table 3.5, and there are 3 commands for every HYSPLIT run. The second entry is a short ID that will be used in script file names. The third entry is a directory where run-time control and signal files will reside. The fourth entry is the name of a text file that contains the names of the compute nodes on the Linux cluster that can be used in this analysis. The fifth entry is the name of the file srcRunCommand.cmd that was produced by the srcSetupRuns code.

4.3 Execution Considerations

Once initiated, this utility program will submit a batch job associated with a single HYSPLIT code run to every compute node identified in the node list file. It will monitor the jobs for completion. Once an individual job completes, QsubRuns will check for additional jobs. If more jobs are required, a new batch job will be submitted. This utility program continues to run until every HYSPLIT run has been started. The user should not modify any of the run-time control or signal files until all of the batch jobs have completed. Once all batch jobs have completed, the run-time control and signal files can be deleted.

5.0 AirGrid – User Instructions

The purpose of the AirGrid code is to register the concentrations and depositions produced by the HYSPLIT code on the set of locations (nodes) to be used in the DESCARTES and CiderF codes. An interpolation algorithm is included to handle the cases where the nodes in the dose domain are not a subset of the output locations used by the HYSPLIT code. This intermediate step greatly reduces the amount of I/O needed in the AirCombGrid code.

5.1 How the Code Is Invoked

The AirGrid code can run under either the Windows 7 or Linux operating systems. Under the Windows 7 operating system, the code executes at a command prompt. A run of AirGrid is initiated by entering the following command line:

```
AirGrid "Keyfilename"
```

Under the Linux operating system, AirGrid is executed through the following Bourne Shell or C Shell commands:

```
AirGrid.exe "Keyfilename"
```

For these commands, AirGrid or “AirGrid.exe” is the name of the executable program, and “Keyfilename” is the name of a controlling keyword file. Both the name of the executable program and the keyword file may contain path information. The keyword file contains text control information describing the run. If AirGrid cannot open the keyword file the code will terminate execution after writing an error message to the standard output device.

5.2 Input and Output Files

The AirGrid code reads a keyword control file to set up the basic problem. Then, it reads from the suite of files generated by the HYSPLIT code. This code writes a report file and a suite of output files. These files are described in the following sections.

5.2.1 Input Files

Excerpts from an input keyword control file for the AirGrid code is provided in Table 5.1. Detailed descriptions of the individual keywords are described in Section 5.3. The input file is the same under the Windows 7 or Linux operating systems. The entries in the file that are colored red are comments ignored by the program. Lines with the entry … denote deleted items.

Table 5.1 Excerpts from an AirGrid Keyword File

```
FILE REPORT "AirGrid.rpt"
USER "Paul W. Eslinger"
FILE LIST  "FileList.txt"  ! Get the file list
VERBOSE
! Define the Hysplit concentration grid for interpolation purposes
GRID CtrLat=55.5 CtrLon=61.0 DelLat=0.025 DelLon=0.025
  MinLat=50.0 MaxLat=60.5  MinLon=51.0 MaxLon=71.0
```

```
! Define the interpolation type
INTERPOLATE DISTANCE
! Define the number of days for HYSPLIT runs and date position
DAYS NUMBER=4 POSITION=27
!Locations at which to extract concentrations
NODE NUMBER=1 LAT=55.1000 LON=60.0000
NODE NUMBER=2 LAT=55.1000 LON=60.1000
NODE NUMBER=3 LAT=55.1000 LON=60.2000
...
NODE NUMBER=514 LAT=55.9500 LON=61.9000
NODE NUMBER=515 LAT=55.9500 LON=62.0000
NODE NUMBER=516 LAT=55.9500 LON=62.1000
END
```

The AirGrid code also reads from the suite of output files written by the HYSPLIT code. Specifically, it reads HYSPLIT output files after conversion from binary to text format. A few lines from one of the cdump.txt files are provided in Table 5.2. The data in the file are tagged with time and geographic location. Surface deposition is given in the columns labeled Orgn0000, Elem0000 and Part0000. Air concentrations are given in the columns labeled Orgn0100, Elem0100 and Part0100. Additional information about the file format is available in the HYSPLIT User's Guide (Draxler et al. 2012).

Table 5.2 Excerpts from a cdump.txt file from HYSPLIT

```
YEAR  MO  DA  HR       LAT        LON    Orgn00000    Orgn00100    Elem00000    Elem00100    Part00000    Part00100
2007   2  16   0   55.6500    60.8000   0.0000E+00   0.5707E-16   0.1564E-17   0.7427E-16   0.0000E+00   0.0000E+00
2007   2  16   0   55.6750    60.8000   0.0000E+00   0.3387E-12   0.2221E-09   0.3447E-12   0.3074E-09   0.3299E-12
2007   2  16   0   55.7000    60.8000   0.0000E+00   0.2033E-12   0.4273E-10   0.2054E-12   0.1853E-09   0.2035E-12
2007   2  16   0   55.6250    60.8250   0.0000E+00   0.1328E-15   0.9049E-19   0.0000E+00   0.6494E-13   0.1157E-15
2007   2  16   0   55.6500    60.8250   0.0000E+00   0.7485E-13   0.4551E-10   0.5969E-13   0.4082E-10   0.6966E-13
2007   2  16   0   55.6750    60.8250   0.0000E+00   0.1498E-11   0.1142E-08   0.1450E-11   0.7888E-09   0.1480E-11
2007   2  16   0   55.7000    60.8250   0.0000E+00   0.2257E-11   0.1557E-08   0.2234E-11   0.6488E-09   0.2252E-11
2007   2  16   0   55.7250    60.8250   0.0000E+00   0.8024E-13   0.5780E-10   0.7566E-13   0.3244E-10   0.7790E-13
2007   2  16   0   55.6000    60.8500   0.0000E+00   0.1704E-15   0.9169E-19   0.1131E-16   0.2245E-13   0.0000E+00
2007   2  16   0   55.6250    60.8500   0.0000E+00   0.1017E-13   0.8298E-11   0.9903E-14   0.6808E-11   0.1061E-13
2007   2  16   0   55.6500    60.8500   0.0000E+00   0.3649E-12   0.2605E-09   0.3216E-12   0.1949E-09   0.3321E-12
2007   2  16   0   55.6750    60.8500   0.0000E+00   0.1004E-11   0.7479E-09   0.9253E-12   0.5199E-09   0.9805E-12
2007   2  16   0   55.7000    60.8500   0.0000E+00   0.1687E-11   0.1224E-08   0.1611E-11   0.4793E-09   0.1669E-11
2007   2  16   0   55.7250    60.8500   0.0000E+00   0.4784E-12   0.3208E-09   0.4495E-12   0.1661E-09   0.4694E-12
2007   2  16   0   55.7500    60.8500   0.0000E+00   0.1562E-13   0.1154E-10   0.1398E-13   0.6538E-11   0.1533E-13
2007   2  16   0   55.5750    60.8750   0.0000E+00   0.0000E+00   0.1428E-17   0.4015E-16   0.0000E+00   0.0000E+00
2007   2  16   0   55.6000    60.8750   0.0000E+00   0.8901E-15   0.1367E-11   0.1445E-14   0.8424E-12   0.1217E-14
2007   2  16   0   55.6250    60.8750   0.0000E+00   0.7997E-13   0.5466E-10   0.6637E-13   0.4773E-10   0.7033E-13
2007   2  16   0   55.6500    60.8750   0.0000E+00   0.4747E-12   0.3229E-09   0.3999E-12   0.2564E-09   0.4461E-12
```

The list of HYSPLIT files for AirGrid to process is provided in an input file. A truncated example file is provided in Table 5.3. Each HYSPLIT output file is provided on a separate line and a full pathname is required. The example file names in this example are for a machine using the Linux operating system.

Table 5.3 Example Input File List file for AirGrid

```
/files0/atm/Mayak/Hysplit/20061229/A/cdump.txt
/files0/atm/Mayak/Hysplit/20061229/AI/cdump.txt
/files0/atm/Mayak/Hysplit/20061229/AV1/cdump.txt
/files0/atm/Mayak/Hysplit/20061229/AV3/cdump.txt
/files0/atm/Mayak/Hysplit/20061229/B/cdump.txt
/files0/atm/Mayak/Hysplit/20061229/DB/cdump.txt
/files0/atm/Mayak/Hysplit/20061229/OK/cdump.txt
/files0/atm/Mayak/Hysplit/20061230/A/cdump.txt
/files0/atm/Mayak/Hysplit/20061230/AV2/cdump.txt
/files0/atm/Mayak/Hysplit/20061230/AV3/cdump.txt
/files0/atm/Mayak/Hysplit/20061230/B/cdump.txt
/files0/atm/Mayak/Hysplit/20061230/DB/cdump.txt
/files0/atm/Mayak/Hysplit/20061230/OK/cdump.txt
```

5.2.2 Output Files

The AirGrid code writes two or more output files. The code always writes a text file (report file) that describes the problem setup and documents the run. The report file contains error messages, if any were generated. In addition, a grid file is written for every HYSPLIT file. Excerpts from a report file are provided in Table 5.4.

Table 5.4 Excerpts from a Report File for AirGrid

```
                           AirGrid Version 1.00.002
                           Last Modified on 20 Jun 2013
        Current Run ID = 20130620114131   User Name = Paul W. Eslinger
             System Date = 06-20-2013   System Time = 11:41:31.105

Output locations defined for 516 nodes
  Lat=  55.1000  Lon=  60.0000 : Node 1
  Lat=  55.1000  Lon=  60.1000 : Node 2
...
  Lat=  55.9500  Lon=  62.0000 : Node 515
  Lat=  55.9500  Lon=  62.1000 : Node 516

Will process data from 14632 ASCII concentration files from Hysplit

Interpolation type is DISTANCE

Latitude grid information
  Center  =   55.5000
  Delta   =    0.0250
  Number  = 421
  Minimum =   50.0000
  Maximum =   60.5000

Longitude grid information
  Center  =   61.0000
  Delta   =    0.0250
  Number  = 803
  Minimum =   50.9750
  Maximum =   71.0250

Interpolation Type: Minimum Distance (4 neighbors)
Index=  1 Point=(  55.1000,  60.0000) Neighbor=(55.1000, 60.0000)
Index=  1 Point=(  55.1000,  60.0000) Neighbor=(55.1000, 60.0000) Weight=1.00000E+00
Index=  1 Point=(  55.1000,  60.0000) Neighbor=(55.1000, 60.0250) Weight=0.00000E+00
Index=  1 Point=(  55.1000,  60.0000) Neighbor=(55.1000, 59.9750) Weight=0.00000E+00
...
Time: 11:41:53.868 File: /files0/atm/Mayak/Hysplit/20061229/A/cdump.txt
Time: 11:41:54.906 File: /files0/atm/Mayak/Hysplit/20061229/AI/cdump.txt
...
Time: 17:35:02.001 File: /files0/atm/Mayak/Hysplit/20111231/DB/cdump.txt
Time: 17:35:02.579 File: /files0/atm/Mayak/Hysplit/20111231/OK/cdump.txt

Message originating in routine AirGrid at 06/20/2013 on 17:35:03.180
Message: Normal Termination
```

A few lines from one of the AirGrid.txt output files are provided in Table 5.5. The data in the file are tagged with time, node number and geographic location. After the three header lines, each line contains a day index (number of days of transport), an effective date, the node number, and then six values. These six values are organic species concentration and deposition (OrgCon and OrgDep), elemental species concentration and deposition (ElemCon and ElemDep) and particulate species concentration and deposition (PartCon and PartDep). All of the lines other than the first three lines are wrapped for display purposes. One of these files is written for every HYSPLIT output file.

Table 5.5 Excerpts from an AirGrid.txt file from AirGrid

```
Program: AirGrid 1.00.002   Run ID: 20130620114131
File: "/files0/atm/Mayak/Hysplit/20070114/B/AirGrid.txt"
516,4,"OrgnCon","OrgnDep","ElemCon",ElemDep","PartCon","PartDep"
...
1,"2007-06-01",99,  55.3000,  61.0000, 0.00000E+00, 0.00000E+00, 0.00000E+00,
0.00000E+00, 0.00000E+00, 0.00000E+00
1,"2007-06-01",100,  55.3000,  61.1000, 0.00000E+00, 0.00000E+00, 0.00000E+00,
0.00000E+00, 0.00000E+00, 0.00000E+00
1,"2007-06-01",101,  55.3000,  61.2000, 1.18500E-15, 0.00000E+00, 6.59100E-16,
3.92300E-13, 9.67700E-16, 5.88100E-14
1,"2007-06-01",102,  55.3000,  61.3000, 1.04000E-14, 0.00000E+00, 9.94700E-15,
7.29600E-12, 1.14800E-14, 7.71600E-13
1,"2007-06-01",103,  55.3000,  61.4000, 2.03500E-14, 0.00000E+00, 1.96700E-14,
1.35400E-11, 1.86200E-14, 1.29700E-12
1,"2007-06-01",104,  55.3000,  61.5000, 1.17100E-14, 0.00000E+00, 1.33300E-14,
9.08400E-12, 1.35800E-14, 9.34500E-13
1,"2007-06-01",105,  55.3000,  61.6000, 7.31400E-15, 0.00000E+00, 6.76800E-15,
4.78000E-12, 7.38300E-15, 4.91000E-13
1,"2007-06-01",106,  55.3000,  61.7000, 5.35800E-15, 0.00000E+00, 5.46300E-15,
3.61800E-12, 4.62200E-15, 3.08200E-13
1,"2007-06-01",107,  55.3000,  61.8000, 3.59000E-15, 0.00000E+00, 3.34200E-15,
2.07100E-12, 4.28000E-15, 3.09300E-13
1,"2007-06-01",108,  55.3000,  61.9000, 2.38600E-15, 0.00000E+00, 2.04200E-15,
1.12500E-12, 2.66200E-15, 1.97400E-13
1,"2007-06-01",109,  55.3000,  62.0000, 2.29200E-15, 0.00000E+00, 2.08200E-15,
1.33300E-12, 1.99900E-15, 1.04500E-13
 ...
```

5.3 Keyword Definitions for AirGrid

The keywords for the AirGrid code can generally be entered in any order. However, the last keyword in the file must be the END keyword. All of the keywords used in the AirGrid code are identified in alphabetical order in Table 5.6.

Table 5.6 Summary of Keywords Used in the AirGrid Code

Keyword	Section	Purpose
DAYS	5.3.1	The DAYS keyword is used to specify information about the HYSPLIT input files.
END	5.3.2	The END keyword signifies the end of all keyword data.
FILE	5.3.3	The FILE keyword is used to enter the names of input and output files.
GRID	5.3.4	The GRID keyword is used to define the output grid used in the HYSPLIT runs so those data can be registered into the nodes used in the DESCARTES and CiderF codes.

Keyword	Section	Purpose
INTERPOL	5.3.5	The INTERPOL keyword is used to define the type of interpolation to use when registering the node locations for DESCARTES and CiderF into the rectangular output grid used in the HYSPLIT runs.
NODE	5.3.6	The NODE keyword is used to define the nodes that will be used in the DESCARTES and CiderF codes.
USER	5.3.7	The USER keyword is used to enter the name of the user running the code.
VERBOSE	5.3.8	The optional VERBOSE keyword can be used to increase the amount of information written to standard out as the run progresses.

5.3.1 DAYS Keyword

The DAYS keyword is used to specify information about the HYSPLIT input files. The following is this keyword's syntax:

```
DAYS NUMBER=N1 POSITION=N2
```

The numerical value associated with the NUMBER parameter identifies the number of days of transport in each HYSPLIT code run. For this application, 4 days are used.

Each input HYSPLIT file (see Table 5.3) has a date embedded in the file name. This date has the form "YYYYMMDD". The numerical value associated with the POSITION modifier gives the index of the start of the date sequence in each file name. The date sequence starts in position 27 in the file names shown in Table 5.3.

An example use of this keyword is the following:

```
DAYS NUMBER=4 POSITION=27
```

5.3.2 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword's syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

5.3.3 FILE Keyword

The FILE keyword is used to enter the names of input and output files. The following is this keyword's syntax:

```
FILE modifier "quote"
```

The file names are entered in quote strings. Path names up to 256 characters long (name length limitation in Windows 7) are supported. The modifiers associated with the FILE keyword are described in Table 5.7.

Table 5.7 Modifiers Associated with the FILE Keyword for AirGrid

Modifier	Description
LIST	The quote string associated with the LIST modifier identifies the name of the input file that will contain names of the HYSPLIT output files to process.
REPORT	The quote string associated with the REPORT modifier contains the name of the output text file containing information about the progress of the run. All error messages are directed to this file.

The following entries define all of the data files identified in Table 5.7.

```
FILE REPORT  “AirGrid.rpt”
FILE LIST  “FileList.txt”
```

5.3.4 GRID Keyword

The GRID keyword is used to define the rectangular output grid used in the HYSPLIT runs. The purpose for defining this grid is to support spatially registering the HYSPLIT results into nodes used in the DESCARTES and CiderF programs. The following is this keyword’s syntax:

```
GRID LATMIN=N1 LATMAX=N2 LATDEL=N3 LONMIN=N4 LONMAX=N5 LONDEL=N6
```

The modifiers associated with the GRID keyword are described in Table 5.8.

Table 5.8 Modifiers Associated with the GRID Keyword for AirGrid

Modifier	Description
CTRLAT	The numerical value associated with the CTRLAT modifier identifies the center latitude (decimal degrees) for the HYSPLIT grid.
CTRLON	The numerical value associated with the CTRLON modifier identifies the center longitude (decimal degrees) for the HYSPLIT grid.
DELLAT	The numerical value associated with the DELLAT modifier identifies the latitude spacing (decimal degrees) for the HYSPLIT grid.
DELLON	The numerical value associated with the DELLON modifier identifies the longitude spacing (decimal degrees) for the HYSPLIT grid.
MAXLAT	The numerical value associated with the optional MAXLAT modifier identifies the maximum latitude (decimal degrees) for the grid. A value of 90.0 is used if this modifier is omitted.
MAXLON	The numerical value associated with optional MAXLON modifier identifies the maximum longitude (decimal degrees) for the grid. A value of 180.0 is used if this modifier is omitted.

Modifier	Description
MINLAT	The numerical value associated with the optional MINLAT modifier identifies the minimum latitude (decimal degrees) for the grid. A value of -90.0 is used if this modifier is omitted.
MINLON	The numerical value associated with the MINLON modifier identifies the minimum longitude (decimal degrees) for the grid. A value of -180.0 is used if this modifier is omitted.

The output locations for the runs of the HYSPLIT code must map into the grid locations defined here. Run times can be negatively impacted if the latitude and longitude spacing is small and the optional MINLAT, MINLON, MAXLAT and MAXLON modifiers are not used. An example GRID keyword entry is the following:

```
GRID CtrLat=55.5 CtrLon=61.0 DelLat=0.025 DelLon=0.025 MinLat=50.0 MaxLat=60.5
   MinLon=51.0 MaxLon=71.0
```

5.3.5 INTERPOL Keyword

The optional INTERPOL keyword is used to define the type of interpolation to use when registering the node locations for DESCARTES and CiderF into the rectangular output grid used in the HYSPLIT runs. The following is this keyword's syntax:

```
INTERPOL [ BILINEAR | DISTANCE | NEAREST ]
```

Three interpolation methods are available. If a node location for DESCARTES and CiderF is coincident with a HYSPLIT grid location then all three methods return the same result. The interpolation methods are the following:

- **BILINEAR**: This interpolation method uses bilinear interpolation on the four points in the HYSPLIT concentration grid that are closest to the desired node location.
- **DISTANCE**: This interpolation method uses inverse squared distance weighting on the four points in the HYSPLIT concentration grid that are closest to the desired node location.
- **NEAREST**: This method selects the single HYSPLIT grid point that is closest to the desired node location.

The default algorithm is the distance weighting algorithm. The default algorithm is used if the INTERPOL keyword is not entered. An example keyword that selects the default inverse-distance weighting algorithm is the following:

```
INTERPOL DISTANCE
```

5.3.6 NODE Keyword

The NODE keyword is used to define the nodes that will be used in the DESCARTES and CiderF codes. The following is this keyword's syntax:

```
NODE NUMBER=N1 LAT=N2 LON=N3
```

The NODE keyword for CiderView has the same definition as the NODE keyword for AirCombGrid (see Section 6.3.7). The modifiers associated with the NODE keyword are described in Table 10.10.

Table 5.9 Modifiers Associated with the NODE Keyword for CiderView

Modifier	Description
LAT	The numerical value associated with the LAT modifier identifies the latitude (decimal degrees) for the node.
LON	The numerical value associated with the LON modifier identifies the longitude (decimal degrees) for the node.
NUMBER	The numerical value associated with the NUMBER modifier identifies the node number. The node number must be a unique integer in the range of 1 to the number of nodes identified with the NODENUM keyword.

A separate NODE keyword is used for defining each node. The following keywords illustrate the use of the NODE keyword in defining ten nodes.

```
NODE NUMBER=1 LAT=55.1000 LON=60.0000
NODE NUMBER=2 LAT=55.1000 LON=60.1000
NODE NUMBER=3 LAT=55.1000 LON=60.2000
NODE NUMBER=4 LAT=55.1000 LON=60.3000
NODE NUMBER=5 LAT=55.1000 LON=60.4000
NODE NUMBER=6 LAT=55.1000 LON=60.5000
NODE NUMBER=7 LAT=55.1000 LON=60.6000
NODE NUMBER=8 LAT=55.1000 LON=60.7000
NODE NUMBER=9 LAT=55.1000 LON=60.8000
NODE NUMBER=10 LAT=55.1000 LON=60.9000
```

5.3.7 USER Keyword

The USER keyword is used to identify the user of the program. The user name will be written to output files for labeling purposes. The program will error terminate if the user name is not supplied. The following is this keyword’s syntax:

```
USER "quote"
```

The user name is entered in a quote string. User names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code:

```
USER "John Q. Public"
```

There are no modifiers associated with the USER keyword.

5.3.8 VERBOSE Keyword

The optional VERBOSE keyword can be used to increase the amount of information written to standard out as the run progresses. It has no effect on the calculations. The following is this keyword’s syntax:

```
VERBOSE
```

There are no modifiers or quote strings associated with the VERBOSE keyword.

6.0 AirCombGrid – User Instructions

The purpose of the AirCombGrid code is to combine facility release information with air transport information based on runs of the HYSPLIT code to develop the air deposition and air concentration files needed by the DESCARTES code. The AirCombGrid code can only be executed after the entire suite of HYSPLIT runs has completed and the AirGrid code has been run.

6.1 How the Code Is Invoked

The AirCombGrid code can run under either the Windows 7 or Linux operating systems. Under the Windows 7 operating system, the code executes at a command prompt. A run of AirCombGrid is initiated by entering the following command line:

```
AirCombGrid "Keyfilename"
```

Under the Linux operating system, AirCombGrid is executed through the following Bourne Shell or C Shell commands:

```
AirCombGrid.exe "Keyfilename"
```

For these commands, AirCombGrid or “AirCombGrid.exe” is the name of the executable program, and “Keyfilename” is the name of a controlling keyword file. Both the name of the executable program and the keyword file may contain path information. The keyword file contains text control information describing the run. If AirCombGrid cannot open the keyword file the code will terminate execution after writing an error message to the standard output device.

6.2 Input and Output Files

The AirCombGrid code reads a keyword control file to set up the basic problem. Then, it reads from the suite of files generated by the HYSPLIT code. This code writes a report file and a suite of output files. These files are described in the following sections.

6.2.1 Input Files

Excerpts from an input keyword control file for the AirCombGrid code for a stochastic case with 250 realizations is provided in Table 6.1. Detailed descriptions of the individual keywords are described in Section 6.3. Although the AirCombGrid code will operate under either of the Windows 7 or Linux operating systems, the example input file is prepared for the Linux operating system. The only input difference in this file between the two operating systems is the form of the path names for files. The entries in the file that are colored red are comments ignored by the program. Lines with the entry … denote deleted items.

Table 6.1 Excerpts from an AirCombGrid Keyword File

```
!   Stochastic Keyword file for the AirCombGrid program
FILE REPORT "Stochastic.rpt"
USER "Paul W. Eslinger"
TITLE "Stochastic AirCombGrid for Mayak dose reconstruction for members of the public"
```

```
FILE RELEASE "Stochastic_Release.csv"
FILE CONCENTR "Stochastic_Conc.dat"
FILE DEPOSITI "Stochastic_Depo.dat"
FILE SPECIATI "Stochastic_Spec.txt"
FILE PREFIX   "/files0/atm/Mayak/Hysplit/"    ! Prefix for Hysplit files
FILE GRIDPREF "/files0/atm/Mayak/Stochastic/" ! Prefix for monthly grid
!Dates are Gregorian dates in the format "yyyy-mm-dd"
PERIOD START="1948-06-01" STOP="1972-12-31"
! If EXECUTE keyword doesn't exist, then stop before process Hysplit files
EXECUTE
! Number of realizations
REALIZAT 250
NUCLIDE ID="I131" HALFLIFE=8.0207 ! Halflife in days
! Seed for the random number generator
SEED 345335.0D0
! Years that Hysplit models were run
HYSPLIT 2007 2008 2009 2010 2011
! Facility descriptions
FACILITY ID="A"   START="1948-06-01" STOP="1972-12-31" TITLE="Graphite reactor: A"
...
FACILITY ID="DB"  START="1959-10-01" STOP="1972-12-31" TITLE="Radiochemical plant: DB"
! Node definitions
NODENUM 516
NODE NUMBER=1 LAT=55.1000 LON=60.0000
NODE NUMBER=2 LAT=55.1000 LON=60.1000
...
NODE NUMBER=515 LAT=55.9500 LON=62.0000
NODE NUMBER=516 LAT=55.9500 LON=62.1000
! Release keywords - A reactor
STOCHASTIC ID="ReleaseA   194806" DIST="Triangular" PARAM 253.5694 357.14 453.5678
  Units="Ci" TITLE="Monthly release from the B plant stacks"
STOCHASTIC ID="ReleaseA   194807" DIST="Triangular" PARAM 253.5694 357.14 453.5678
  Units="Ci" TITLE="Monthly release from the B plant stacks"
STOCHASTIC ID="ReleaseA   194808" DIST="Triangular" PARAM 253.5694 357.14 453.5678
  Units="Ci" TITLE="Monthly release from the B plant stacks"
STOCHASTIC ID="ReleaseA   1948Total" DIST="Triangular" PARAM 1774.9858 2499.98
  3174.9746 Units="Ci" TITLE="Annual release from the B plant stacks"
...
STOCHASTIC ID="ReleaseDB  197211" DIST="Triangular" PARAM 0.11715 0.165 0.20955
  Units="Ci" TITLE="Monthly release from the DB plant stacks"
STOCHASTIC ID="ReleaseDB  197212" DIST="Triangular" PARAM 0.11715 0.165 0.20955
  Units="Ci" TITLE="Monthly release from the DB plant stacks"
STOCHASTIC ID="ReleaseDB  1972Total" DIST="Triangular" PARAM 1.4058 1.98 2.5146
  Units="Ci" TITLE="Monthly release from the DB plant stacks"
END
```

The AirCombGrid code also reads from the suite of output files written by the AirGrid code. These files exist in a directory structure developed from the meteorological data date and facility ID (see Section 3.3.3) and each file uses the name AirGrid.txt. Internal logic in the AirCombGrid code is used to find the desired file for a given date and facility when needed. A few lines from one of the AirGrid.txt files are provided in Table 5.5.

6.2.2 Output Files

The AirCombGrid code always writes five output files. With some input options, the code writes many more files. The five files always written are the following:

- **Report File** – This text file describes the problem setup and documents the run. It also contains error messages, if any were generated.

- **Speciation Fractions** – This text file contains organic, elemental or particulate iodine fractions for every realization.
- **Release Values** – This text file contains the calculated release values from every facility for every realization.
- **Concentration** – This binary file contains air concentration information structured for use in the DESCARTES code.
- **Deposition** – This binary file contains air deposition information structured for use in the DESCARTES code.

Excerpts from a report file for a best estimate run with 1 realization is provided in Table 6.2. Error messages for run time errors trapped by the AirCombGrid code are written to this file. A few lines in the file are long enough that they wrap to the next line when imported into this table.

Table 6.2 Excerpts from a Report File for AirCombGrid

```
                    AirCombGrid    Version 1.00.02
                    Last Modified on 21 Jun 2013
----------------------------------------------------------------------
      Combine Air Transport Runs with Facility Release for Mayak
----------------------------------------------------------------------

    Current Run ID = 20130621124228   User Name = Paul W. Eslinger
         System Date = 06-21-2013   System Time = 12:42:28.258
...

User:  Paul W. Eslinger
Title: Best Estimate AirCombGrid for Mayak dose reconstruction for members of the
public

Nuclide definition
    I131    : ID
     8.0207 : Halflife (days)

Input Keyword File Name
File: Best.kwd
...
Simulation time period controls
    1948-06-01 : Start date
    1972-12-31 : Stop date
          8979 : Number of days
           295 : Number of months
            25 : Number of years

Number of realizations: 1

Seed for random number generator:     345335.0

Number of Hysplit Model Years: 5
     2007 2008 2009 2010 2011

Number of Facilities: 8
    ID   Start      Stop       Description
    ---- ---------- ---------- ----------------------------------------
    B    1948-12-01 1967-05-31 Radiochemical plant: B
    DB   1959-10-01 1972-12-31 Radiochemical plant: DB
    A    1948-06-01 1972-12-31 Graphite reactor: A
    AV1  1950-07-01 1972-12-31 Graphite reactor: AV-1
    AV2  1951-03-01 1972-12-31 Graphite reactor: AV-2
    AV3  1952-09-01 1972-12-31 Graphite reactor: AV-3
    AI   1951-12-01 1972-12-31 Graphite reactor: AI
    OK   1951-10-01 1972-12-31 Heavy Water Reactors; OK Complex

Number of Nodes: 516
    Index Latitude   Longitude
    ----- ---------- ----------
        1   55.10000   60.00000
        2   55.10000   60.10000
...

Definition of the underlying data grid
    Latitude in degrees (stored North to South) 57.5000 to 53.4750 in steps of  0.025
    Longitude in degrees (stored West to East)  57.0000 to 65.0250 in steps of  0.025
      162 steps in latitude
      322 steps in longitude
    52164 total points

  Atmospheric data file identification information
   Fixed Record Length        :    16
   Realizations in File       :     1
   Nodes in File              :   516
   First Year in File         :  1948
   First Julian Date in File  :   153
   Last Year in File          :  1972
   Last Julian Date in File   :   366
   Nuclide Record             : I131
   Data type in air File      : CONCENTR
   Title                      : Best Estimate AirCombine for Mayak dose members of the
public
   Program Name               : AirCombine
   Program Version            : 1.00.05
...
Map of Hysplit model years by simulated month and realization
    "1948-06" average of  2007 2008 2009 2010 2011
    "1948-07" average of  2007 2008 2009 2010 2011
    "1948-08" average of  2007 2008 2009 2010 2011
    "1948-09" average of  2007 2008 2009 2010 2011
...

  Atmospheric data file identification information
   Fixed Record Length        :    16
   Realizations in File       :     1
   Nodes in File              :   516
   First Year in File         :  1948
   First Julian Date in File  :   153
   Last Year in File          :  1972
   Last Julian Date in File   :   366
   Nuclide Record             : I131
   Data type in air File      : DEPOSITI
   Title                      : Best Estimate AirCombine for Mayak dose reconstruction
for members of the public
   Program Name               : AirCombGrid
   Program Version            : 1.00.02

Map of Hysplit model years by simulated month and realization
    "1948-06" average of  2007 2008 2009 2010 2011
    "1948-07" average of  2007 2008 2009 2010 2011
    "1948-08" average of  2007 2008 2009 2010 2011
...
Message: Normal Termination
```

Another text file written by the AirCombGrid code contains information about the speciation fractions for iodine transport. Excerpts from an iodine speciation fraction file for a best estimate run with 1 realization are provided in Table 6.3. Many of the lines in the file are long enough that they wrap to the next line when imported into this table. This file identifies the fraction for each form of ^{131}I in each realization of generated concentrations and depositions.

Table 6.3 Excerpts from an Output Speciation Fraction File from AirCombGrid

```
AirCombGrid Version 1.00.02 (21 Jun 2013)
Keyword File: Best.kwd
Run: 06-21-2013 12:42:28.258

Species fractionization for iodine by realization
Real  Particle   Organic Elemental
---- --------- --------- ---------
   1  0.440000  0.290000  0.270000

Summary statistics for variable: "SpecParticleFrac" (Data units are None)
   Title: Particle Fraction from Iodine Speciation
    Minimum    1% Level    5% Level   10% Level   25% Level     Median    75% Level
90% Level   95% Level   99% Level     Maximum      Mean      St. Dev.
 ----------- ----------- ----------- ----------- ----------- ----------- ----------- -
---------- ----------- ----------- ----------- ----------- -----------
  4.4000E-01  4.4000E-01  4.4000E-01  4.4000E-01  4.4000E-01  4.4000E-01  4.4000E-01
4.4000E-01  4.4000E-01  4.4000E-01  4.4000E-01  4.4000E-01  0.0000E+00

Summary statistics for variable: "SpecOrganicFrac" (Data units are None)
   Title: Organic Fraction from Iodine Speciation
    Minimum    1% Level    5% Level   10% Level   25% Level     Median    75% Level
90% Level   95% Level   99% Level     Maximum      Mean      St. Dev.
 ----------- ----------- ----------- ----------- ----------- ----------- ----------- -
---------- ----------- ----------- ----------- ----------- -----------
  2.9000E-01  2.9000E-01  2.9000E-01  2.9000E-01  2.9000E-01  2.9000E-01  2.9000E-01
2.9000E-01  2.9000E-01  2.9000E-01  2.9000E-01  2.9000E-01  0.0000E+00

Summary statistics for variable: "SpecElementalFrac" (Data units are None)
   Title: Elemental Fraction from Iodine Speciation
    Minimum    1% Level    5% Level   10% Level   25% Level     Median    75% Level
90% Level   95% Level   99% Level     Maximum      Mean      St. Dev.
 ----------- ----------- ----------- ----------- ----------- ----------- ----------- -
---------- ----------- ----------- ----------- ----------- -----------
  2.7000E-01  2.7000E-01  2.7000E-01  2.7000E-01  2.7000E-01  2.7000E-01  2.7000E-01
2.7000E-01  2.7000E-01  2.7000E-01  2.7000E-01  2.7000E-01  0.0000E+00
```

Another text file output by the AirCombGrid code contains the generated release (Ci) from every facility for every month of operation. Also included is annual output total by facility. This file is a text file written in comma separated variables format, so it can be directly imported into a spreadsheet or database. Zero releases are not output to this file. Excerpts from a facility release file for a best estimate run with 1 realization are provided in Table 6.4. A few lines in the file are long enough that they wrap to the next line when imported into this table. Lines containing … denote items deleted for presentation purposes.

Table 6.4 Excerpts from an Output Source Term File from AirCombGrid

```
"ID","Year","Month","Realizations"
"A",1948,06, 3.57140E+02
"A",1948,07, 3.57140E+02
"A",1948,08, 3.57140E+02
"A",1948,09, 3.57140E+02
```

```
...
"B",1949,10, 2.59380E+04
"A",1949,10, 2.27269E+02
...
"B",1954,06, 7.99807E+03
"A",1954,06, 4.20000E-02
"AV1",1954,06, 4.16700E+00
"AV2",1954,06, 4.16700E+00
"AV3",1954,06, 4.16700E+00
"AI",1954,06, 4.20000E-02
"OK",1954,06, 1.00000E-01
...
"DB",1972,Total, 1.98000E+00
"A",1972,Total, 9.96000E-02
"AV1",1972,Total, 9.96000E-02
"AV2",1972,Total, 9.96000E-02
"AV3",1972,Total, 9.96000E-02
"AI",1972,Total, 9.96000E-02
"OK",1972,Total, 3.60000E-02
"ID","Year","Month","Min","1 Pct","5 Pct","10 Pct","25 Pct","50 Pct","75 Pct","90
Pct","95 Pct","99 Pct","Max","Mean","StDev"
"A",1948,06, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02,
3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02,
3.57140E+02, 0.00000E+00
"A",1948,07, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02,
3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02,
3.57140E+02, 0.00000E+00
"A",1948,08, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02,
3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02, 3.57140E+02,
3.57140E+02, 0.00000E+00
```

The AirCombGrid also writes two binary files. One file contains air concentration information structured for use in the DESCARTES code. The other file contains air deposition information structured for use in the DESCARTES code. The format of these files is not provided here, but the utility code MLOOK (Miley et al. 1994) is available to convert portions or all of the file to text format. These files can be large, possibly several gigabytes in size. See Section 1.4.2 for a file size algorithm.

The AirCombGrid code can also write a series of optional files. Activation of the optional outputs is performed with the FILE keyword in the control keyword file. Specifically, check the description of the GRIDPREF modifier for the FILE keyword in Section 6.3.5. These optional files contain monthly average air concentrations and month end deposition values on the set of nodes defined for DESCARTES. Two files (one for concentration and one for deposition) are written for every month. Data are output only for the first realization, even if more than one realization is generated.

Excerpts from a monthly average air concentration file for a best estimate run with 1 realization is provided in Table 6.5 This text file contains concentrations information on the set of nodes. After three header lines, the data lines contain the latitude (degrees), longitude (degrees) and air concentration (Ci-s/m^3). A daily air concentration can be obtained by dividing these values by 86,400 s/d. The structure of a deposition file is exactly the same as the structure of the concentration file, except that the data units are Ci/m^2.

Table 6.5 Excerpts from a Monthly Concentration Grid File Written by AirCombGrid

```
Program: AirCombGrid 1.00.02 Run ID: 20130621124228
File: "/files0/atm/Mayak/Best/grid_1948_08_conc.txt"
516
  55.1000,  60.0000, 2.83282E-09
  55.1000,  60.1000, 3.23815E-09
  55.1000,  60.2000, 3.85818E-09
  55.1000,  60.3000, 4.26457E-09
  55.1000,  60.4000, 4.65304E-09
  55.1000,  60.5000, 4.88396E-09
  55.1000,  60.6000, 4.62181E-09
```

6.3 Keyword Definitions for AirCombGrid

The keywords for the AirCombGrid code can generally be entered in any order. However, the last keyword in the file must be the END keyword. The keywords used in the AirCombGrid code are identified in alphabetical order in Table 6.6.

Table 6.6 Summary of Keywords Used in the AirCombGrid Code

Keyword	Section	Purpose
BEST	6.3.1	The optional BEST keyword signifies that best estimate calculations are to be made.
END	6.3.2	The END keyword signifies the end of data in the file.
EXECUTE	6.3.3	The optional EXECUTE keyword signifies that output calculations are to be made. Without this keyword, the code will check all inputs but it will not generate files for use in DESCARTES.
FACILITY	6.3.4	The FACILTY keyword is used to define the facilities that release ^{131}I to the air at some time during a run of the code.
FILE	6.3.5	The FILE keyword is used to enter file names for the code to use.
HYSPLIT	6.3.6	The HYSPLIT keyword is used to define the years where HYSPLIT runs were performed with unit releases.
NODE	6.3.7	Multiple NODE keywords are used to define the nodes that will be used in the DESCARTES and CiderF codes.
NODENUM	6.3.8	The NODENUM keyword is used to identify the total number of nodes in data files produced for use in the DESCARTES code.
NUCLIDE	6.3.9	The NUCLIDE keyword is used to identify the nuclide and associated radioactive half-life.
PERIOD	6.3.10	The PERIOD keyword is used to start and stop dates used in data files produced by the DESCARTES code.
REALIZAT	6.3.11	The REALIZAT keyword is used to define the number of realizations of data in the files output for use in DESCARTES.
SEED	6.3.12	The SEED keyword is used to set the random seed for stochastic variable generation.

Keyword	Section	Purpose
SPECIATI	6.3.13	The optional SPECIATI keyword can be used to override the stochastic definition of ^{131}I speciation and use fixed speciation fractions.
STOCHAST	6.3.14	The STOCHAST keyword is used to enter data for variables that can be defined by statistical distributions.
TITLE	6.3.15	The TITLE keyword is used to enter a descriptive title for the code run.
USER	6.3.16	The USER keyword is used to enter the name of the user running the code.

6.3.1 BEST Keyword

The optional BEST keyword signifies that best estimate calculations are to be made. Omit this keyword if best estimate calculations are not to be used. The following is this keyword's syntax:

```
BEST
```

There are no modifiers or quote strings associated with the BEST keyword. The BEST keyword can only be used for cases when the number of realizations is 1. If BEST is used, then air concentrations and depositions from HYSPLIT are averaged over all available model runs for a given day. Otherwise, air concentrations and depositions from HYSPLIT are assigned randomly by realization.

6.3.2 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword's syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

6.3.3 EXECUTE Keyword

The optional EXECUTE keyword signifies that output calculations are to be made. Without this keyword, the code will check all inputs but it will not generate files for use in DESCARTES. The AirCombGrid code is very I/O intensive and run times can be several days to a couple of weeks in length for a large problem definition. This optional keyword allows the user to complete error checking on inputs before committing to a long run. The following is this keyword's syntax:

```
EXECUTE
```

There are no modifiers or quote strings associated with the EXECUTE keyword.

6.3.4 FACILITY Keyword

The FACILTY keyword is used to define the facilities that release ^{131}I to the air at some time during a run of the code. The following is this keyword's syntax:

```
FACILITY ID=“quote1” START=“quote2” STOP=“quote3” TITLE=“quote4”
```

All four modifiers are required on this keyword. A separate FACILTY keyword is required for every facility. Explanation of the modifiers and associated data is provided in Table 6.7

Table 6.7 Modifiers Associated with the FACILITY Keyword for AirCombGrid

Modifier	Description
ID	The quote string associated with the ID modifier provides the facility ID. The ID is a unique character string of no more than 4 characters in length. This ID must match with the facility ID’s used in the release station file for the srcSetupRuns code (see Table 3.3).
START	The quote string associated with the START modifier identifies the first day the facility releases ^{131}I to the air. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example for June 7, 1955, is the following: “1955-06-07”.
STOP	The quote string associated with the STOP modifier identifies the last day the facility releases ^{131}I to the air. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example for December 31, 1972, is the following: “1972-12-31”.
TITLE	The quote string associated with the TITLE modifier contains a description that is used for labeling purposes. It can contain up to 72 characters.

The following set of FACILITY keywords identifies four releasing facilities.

```
FACILITY ID="B"   START="1948-12-01" STOP=“1967-05-31"
   TITLE= “Radiochemical plant B"

FACILITY ID="DB"  START="1959-10-01" STOP="1972-12-31"
   TITLE="Radiochemical plant: DB"

FACILITY ID="A"   START="1948-06-01" STOP="1972-12-31"
   TITLE="Graphite reactor: A"

FACILITY ID="AI"  START="1951-12-01" STOP="1972-12-31"
   TITLE="Graphite reactor: AI"
```

6.3.5 FILE Keyword

The FILE keyword is used to enter the names of input and output files. The following is this keyword’s syntax:

```
FILE modifier "quote"
```

The file names are entered in quote strings. Path names up to 256 characters long (name length limitation in Windows 7) are supported. At least one FILE keyword is required for every run of the code. The modifiers associated with the FILE keyword are described in Table 6.8.

Table 6.8 Modifiers Associated with the FILE Keyword for AirCombGrid

Modifier	Description
CONCENTR	The quote string associated with the CONCENTR modifier contains the name of the output file that will contain concentration data structured for use in DESCARTES.
DEPOSITI	The quote string associated with the DEPOSITI modifier contains the name of the output file that will contain deposition data structured for use in DESCARTES.
GRIDPREF	The quote string associated with the optional GRIDPREF modifier contains the path name of the directory where the monthly concentration and deposition files will be written. The terminating slash in the path name must be included. Omit the GRIDPREF modifier and path name if the output files are not desired.
PREFIX	The quote string associated with the PREFIX modifier contains the path name of the root directory where all the HYSPLIT runs are stored. The terminating slash in the directory name must be included. An example quote string for a Linux system is "/files0/atm/Mayak/Hysplit/".
RELEASE	The quote string associated with the RELEASE modifier contains the name of the output text file that will contain the generated monthly releases from each facility. See Table 6.4 for an example of the contents of this file.
REPORT	The quote string associated with the REPORT modifier contains the name of the output text file containing information about the progress of the run. All error messages are directed to this file.
SPECIATI	The quote string associated with the SPECIATI modifier contains the name of the output text file that will contain the generated fractions of iodine speciation. See Table 6.3 for an example of the contents of this file.

The following entries define all of the data files identified in Table 6.8.

```
FILE REPORT   "Best.rpt"
FILE RELEASE  "Best_Release.csv"
FILE CONCENTR "Best_Conc.dat"
FILE DEPOSITI "Best_Depo.dat"
FILE SPECIATI "Best_Spec.txt"
FILE PREFIX   "/files0/atm/Mayak/Hysplit/"  ! (Required) Prefix for Hysplit runs
FILE GRIDPREF "/files0/atm/Mayak/Best/" ! (Optional) prefix for monthly grid files
```

6.3.6 HYSPLIT Keyword

The HYSPLIT keyword is used to define the years where HYSPLIT runs were performed with unit releases. The following is this keyword’s syntax:

```
HYSPLIT N1 N2 ... Nn
```

Each year that HYSPLIT was run should be entered separately as a four digit integer. The assumption is made that HYSPLIT was run with unit releases from all facilities for every day of every year identified with this keyword.

The following keyword identifies that HYSPLIT runs were made for every facility for every day in 2007, 2008, 2009, 2010 and 2011.

```
HYSPLIT 2007 2008 2009 2010 2011
```

6.3.7 NODE Keyword

The NODE keyword is used to define the nodes that will be used in the DESCARTES and CiderF codes. The following is this keyword’s syntax:

```
NODE NUMBER=N1 LAT=N2 LON=N3
```

The modifiers associated with the NODE keyword are described in Table 6.9. The node locations must be a subset of the locations identified by the GRID keyword (see Section 0).

Table 6.9 Modifiers Associated with the NODE Keyword for AirCombGrid

Modifier	Description
LAT	The numerical value associated with the LAT modifier identifies the latitude (decimal degrees) for the node.
LON	The numerical value associated with the LON modifier identifies the longitude (decimal degrees) for the node.
NUMBER	The numerical value associated with the NUMBER modifier identifies the node number. The node number must be a unique integer in the range of 1 to the number of nodes identified with the NODENUM keyword.

A separate NODE keyword is used for defining each node. The following keywords illustrate the use of the NODE keyword in defining ten nodes.

```
NODE NUMBER=1 LAT=55.1000 LON=60.0000
NODE NUMBER=2 LAT=55.1000 LON=60.1000
NODE NUMBER=3 LAT=55.1000 LON=60.2000
NODE NUMBER=4 LAT=55.1000 LON=60.3000
NODE NUMBER=5 LAT=55.1000 LON=60.4000
NODE NUMBER=6 LAT=55.1000 LON=60.5000
NODE NUMBER=7 LAT=55.1000 LON=60.6000
NODE NUMBER=8 LAT=55.1000 LON=60.7000
NODE NUMBER=9 LAT=55.1000 LON=60.8000
NODE NUMBER=10 LAT=55.1000 LON=60.9000
```

6.3.8 NODENUM Keyword

The NODENUM keyword is used to define the total number of nodes that will be used in the DESCARTES and CiderF codes. The following is this keyword’s syntax:

```
NODENUM N1
```

The number of nodes must be entered as an integer. The following keyword identifies that a total of 516 nodes will be used.

```
NODENUM 516
```

6.3.9 NUCLIDE Keyword

The NUCLIDE keyword is used to define the nuclide used in the modeling codes. The following is this keyword's syntax:

```
NUCLIDE ID=“quote” HALFLIFE=N1
```

The quote string associated with the ID modifier identifies the radionuclide of interest. For this application, the only valid entry is “I131”. The numerical value associated with the HALFLIFE modifier is the radioactive half-life of the nuclide in days. The following keyword identifies ^{131}I as the nuclide of interest and provides the 8.0207 day half-life.

```
NUCLIDE ID="I131" HALFLIFE=8.0207
```

6.3.10 PERIOD Keyword

The PERIOD keyword is used to define the time period for the simulation. The following is this keyword's syntax:

```
PERIOD START=“quote1” STOP=“quote1”
```

Explanation of the modifiers and associated data for the PERIOD keyword is provided in Table 6.10.

Table 6.10 Modifiers Associated with the PERIOD Keyword for AirCombGrid

Modifier	Description
START	The quote string associated with the START modifier identifies the first day for the simulation. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example for June 7, 1955, is the following: “1955-06-07”.
STOP	The quote string associated with the STOP modifier identifies the last day for the simulation. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example for December 31, 1972, is the following: “1972-12-31”.

The start and stop dates on this period keyword must match exactly with the start and stop dates in a subsequent run of DESCARTES. The following PERIOD keyword identifies a simulation that runs from June 1, 1948 through December 31, 1972.

```
PERIOD START="1948-06-01" STOP="1972-12-31"
```

6.3.11 REALIZAT Keyword

The REALIZAT defines the number of realizations to process. The following is this keyword's syntax:

```
REALIZAT N1
```

The number of realizations must be entered as an integer. The following keyword identifies that a total of 250 realizations will be used.

```
REALIZAT 250
```

The following keyword identifies that 1 realization will be used. Use of the BEST modifier requires that only one realization be run.

```
REALIZAT 1
```

6.3.12 SEED Keyword

The SEED keyword sets the value for the seed for the random number generator. The following is this keyword's syntax:

```
SEED Value1
```

The value for Value1 must be a number in the range 1.0 to 2147483646.0, in whole number increments. The following are examples of the keyword record:

```
SEED 5
SEED 101.0
SEED 2147483645.0D0
```

There are no quote strings or modifiers associated with the SEED keyword.

6.3.13 SPECIATI Keyword

The optional SPECIATI keyword can be used to override the stochastic definition of ^{131}I speciation (see 1.4.4) and use fixed speciation fractions. Do not enter this keyword if the stochastic definition is desired. The following is this keyword's syntax:

```
SPECIATI PFRAC=N1 OFRAC=N2 EFRAC=N3
```

Explanation of the modifiers and associated data for the SPECIATI keyword is provided in Table 6.11.

Table 6.11 Modifiers Associated with the SPECIATI Keyword for AirCombGrid

Modifier	Description
PFRAC	The number associated with the PFRAC modifier is the fraction of iodine that occurs in particulate form. A value in the range 0 to 1 is required.
OFRAC	The number associated with the PFRAC modifier is the fraction of iodine that occurs in organic form. A value in the range 0 to 1 is required.
EFRAC	The number associated with the PFRAC modifier is the fraction of iodine that occurs in the elemental form. A value in the range 0 to 1 is required.

The value for each of the three speciation fractions must be in the range 0 to 1. In addition, the three fractions must sum to 1. The following is an example of the SPECIATI keyword:

```
SPECIATI PFRAC=0.4391 OFRAC=0.2953 EFRAC=0.2656
```

6.3.14 STOCHAST Keyword

Stochastic input variables are defined by the STOCHAST keyword. The following is this keyword’s syntax:

```
STOCHAST [ID=“Quote1”] [TITLE=“Quote2”] [UNITS=“quote3”] [DIST=“Quote4”]
   PARAM N1,…Nn (TRUNCATE U1 U2)
```

The function of all of the modifiers on the STOCHAST keyword is explained in detail in Section 12.1 and is not repeated here. This description focuses on the how to define different values for the quote string associated with the ID modifier so the code can internally access the correct data.

The AirCombGrid code uses the assumption that facility release values are available on both a monthly basis and an annual basis. Monthly release values are calculated and then scaled so the sum of the monthly values matches the annual release values for a facility. The release definitions are all entered using STOCHAST keywords. Potentially, a large number of STOCHAST keywords will be needed. A single STOCHAST keyword defines either a monthly or annual release from a single facility. A separate STOCHAST keyword is required for every month a facility has releases. In addition, a separate STOCHAST keyword is required for every year a facility has releases.

The quote string associated with the ID for stochastic releases consists of three pieces, all of which are case sensitive. The first 7 characters contain the word “Release”. The next four characters contain the facility ID. The facility ID must start in position 8 and it must be blank filled if it is less than 4 characters in length. If the release is a monthly release, then characters 12 through 17 contain the four digit year and the two digit month. Months 1 through 9 must contain a leading 0. If the release is a yearly total, then characters 12 through 20 contain the four digit year followed by the word “Total”. The following eight example STOCHAST keywords define some monthly and yearly releases for the “A” and “DB” facilities.

```
STOCHAST ID="ReleaseA   194806" DIST="Triangular" PARAM 253.5694 357.14 453.5678
   Units="Ci" TITLE="Monthly release from the A reactor"

STOCHAST ID="ReleaseA   194807" DIST="Triangular" PARAM 253.5694 357.14 453.5678
   Units="Ci" TITLE="Monthly release from the A reactor"

STOCHAST ID="ReleaseA   194812" DIST="Triangular" PARAM 253.5694 357.14 453.5678
   Units="Ci" TITLE="Monthly release from the A reactor"

STOCHAST ID="ReleaseA   1948Total" DIST="Triangular" PARAM 1774.9858 2499.98
   3174.9746 Units="Ci" TITLE="Annual release from the A reactor"

STOCHAST ID="ReleaseDB  197201" DIST="Triangular" PARAM 0.11715 0.165 0.20955
   Units="Ci" TITLE="Monthly release from the DB plant stacks"

STOCHAST ID="ReleaseDB  197202" DIST="Triangular" PARAM 0.11715 0.165 0.20955
   Units="Ci" TITLE="Monthly release from the DB plant stacks"

STOCHAST ID="ReleaseDB  197212" DIST="Triangular" PARAM 0.11715 0.165 0.20955
   Units="Ci" TITLE="Monthly release from the DB plant stacks"

STOCHAST ID="ReleaseDB  1972Total" DIST="Triangular" PARAM 1.4058 1.98 2.5146
   Units="Ci" TITLE="Annual release from the DB plant stacks"
```

6.3.15 TITLE Keyword

The TITLE keyword is used to define a single-line problem title. The problem title will be written to output files. The AirCombGrid program will error terminate if the title is not supplied. The following is this keyword's syntax:

```
TITLE "quote"
```

The title is entered in a quote string. Titles up to 80 characters long are supported. The following example defines a title for a run of the code:

```
TITLE "Example title line for the AirCombGrid code."
```

There are no modifiers associated with the TITLE keyword.

6.3.16 USER Keyword

The USER keyword is used to identify the user of the program. The user name will be written to output files for labeling purposes. The program will error terminate if the user name is not supplied. The following is this keyword's syntax:

```
USER "quote"
```

The user name is entered in a quote string. User names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code:

```
USER "John Q. Public"
```

There are no modifiers associated with the USER keyword.

7.0 AirCombGridView – User Instructions

The AirCombGridView utility code reads grid files produced by the AirCombGrid code and writes data files formatted for import into commercial plotting programs. The output files can either be viewed in Google Earth (Google 2013) or in ArcGis (ESRI 2013). The files for display in Google Earth are written in KML format. The files for ArcGis are written in ESRI raster format.

7.1 How the Code is Invoked

The AirCombGridView code runs under the Windows operating system (Release 7). The code executes at a command prompt. A run of AirCombGridView is initiated by entering the following command line:

```
AirCombGridView keyfilename
```

For this command, AirCombGridView is the name of the executable program and keyfilename is the name of a keyword file. Both the name of the executable program and the keyword file may contain path information. If AirCombGridView is invoked without entering the name of the keyword file name, the code will prompt the user for the file name. The keyword file contains text control information describing the run. If AirCombGridView cannot open the keyword file, then the code will terminate execution after writing an error message to the standard output device.

7.2 Input and Output Files

The AirCombGridView code reads either two or three files and writes two, three or four files, depending on the options chosen. These input and output files are defined in the following sections.

7.2.1 Input Files

The AirCombGridView code always reads a control keyword file. An example keyword file for the AirCombGridView code is provided in Table 7.1. Detailed descriptions of the individual keywords are described in Section 7.3.

Table 7.1 Example Keyword File for AirCombGridView

```
USER "Paul W. Eslinger"
FILE DATA "grid_1949_01_conc.txt"
FILE REPORT "Test.rpt"
FILE KML    "Test.kml"
FILE RASTER "Test.asc"
VIEW LATMIN=55.07 LATMAX=56.0 LONMIN=59.95 LONMAX=62.15 STEP=0.01
INTERPOL DISTANCE IGNORE=7.5
SCALE 4.28241E5  ! Optional scale to Bq  (4.28241E5 to convert Ci-s/m^3 to Bq/m^3)
KMLBLANK    FILE="D:\Fortran\HEDR\AirCombGridView\Test\Mayak_Polygon.kml"
KMLLEGEND   FILE="D:\Fortran\HEDR\AirCombGridView\Test\Legend.jpg"
KMLDOCUMENT "Conc (Bq/m^3)"
KMLCOLOR Color_01="B30000FF" Color_02="B33399FF" Color_03="B300FFFF"
  Color_04="B3B4FFFF" Color_05="B300FF00"Color_06="B310FFA8" Color_07="B3FF0000"
  Color_08="B3FF9900" Color_09="B3FFFF00" Color_10="B3FFFFCC"
KMLLEVEL 100 50 10 5 1 0.5 0.1 0.05 0.01 0.005 ABSOLUTE ! Contouring levels
END
```

The AirCombGridView code also reads a monthly concentration or deposition file produced by the AirCombGrid code. A brief description of these files is provided in Section 0 and Table 6.5.

7.2.2 Output Files

The AirCombGridView code always writes a report file. An example report file is provided in Table 7.2 for a run that produces both KML and raster outputs. Error messages for run time errors trapped by the code, if any, are written to this file.

Table 7.2 Example Report File for AirCombGridView

```
                           AirCombGridView  Version 1.00.005
                          Last Modified on 20 May 2013
...

User:  Paul W. Eslinger

Input Keyword File Name
File: conc.kwd

File Name for this file
File: grid_1948_12_conc.rpt

File Name for Input Data
File: grid_1948_12_conc.txt

File Name for Output KML Code
File: grid_1948_12_conc.kml

File Name for Output Raster
File: grid_1948_12_conc.asc

Output data are scaled. Multiplicative factor is  4.282410E+05

KML file domain limits (decimal degrees)
       53.500 : Latitude minimum
       57.500 : Latitude maximum
        0.025 : Latitude step
       57.000 : Longitude minimum
       65.000 : Longitude maximum
        0.025 : Longitude step

The 52164 data values range from  0.000000E+00 to  1.564000E+02

KML Countour definitions
  1 : Contour level type (1=Absolute,2=Fraction of Maximum value)
  10 : Number of output contour levels
 Index    Entry          Data Value   Color
    1    1.00000E+02    1.00000E+02 "B30000FF"
    2    5.00000E+01    5.00000E+01 "B33399FF"
...

Raster file domain limits (decimal degrees)
       53.500 : Latitude minimum
       57.500 : Latitude maximum
...

Message originating in routine AirCombGridView at 05/20/2013 on 15:14:04.334
Message: Normal Termination
```

Describing the contents of a KML file is beyond the scope of this document. Interested readers are referred to online descriptions of the Keyhole Markup Language (Google 2012). Instead, a snapshot of an output KML file viewed using Google Earth is provided in Figure 7.1. This plot is provided for illustration purposes only and does not represent real conditions. The dots on the figure show the underlying data grid for the dose codes. These data are interpolated and converted to contours for display purposes.

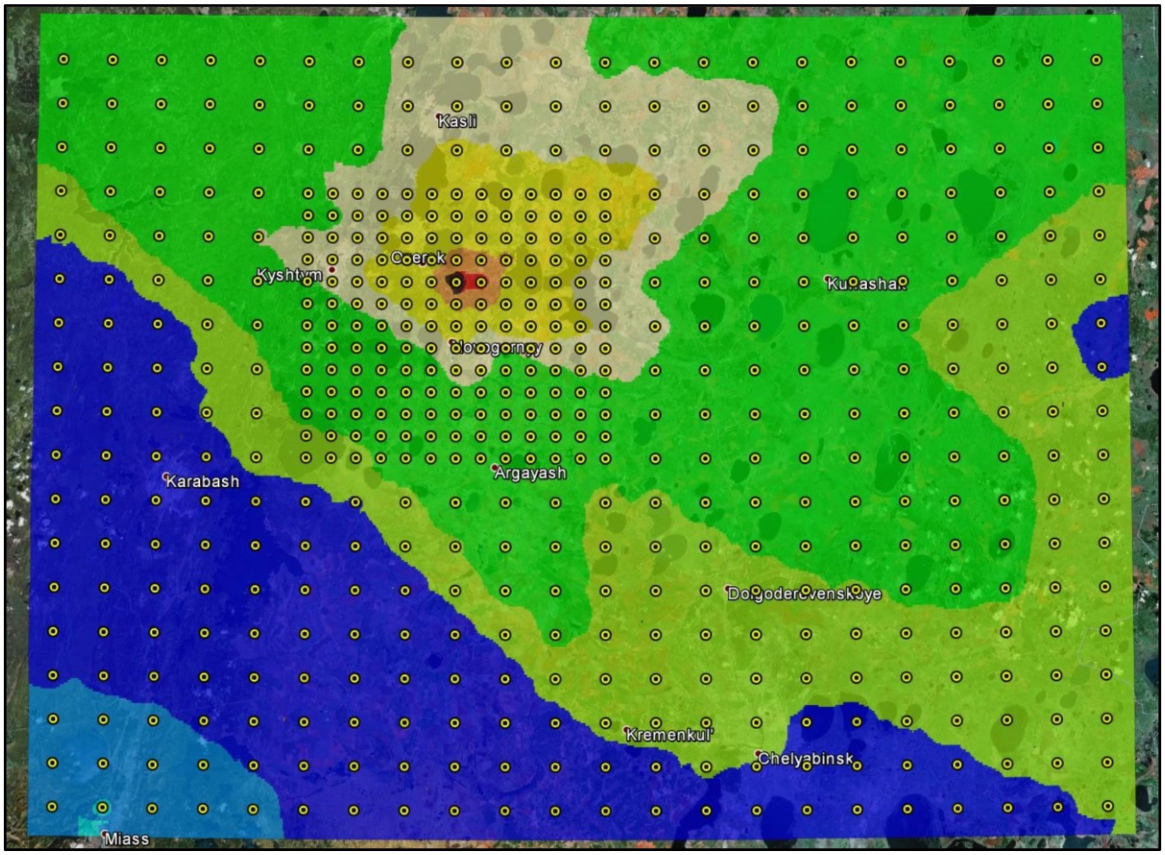


Figure 7.1 Example Plot from Google Earth using a KML File produced by the AirCombGridView Code

Describing the contents of a raster file is beyond the scope of this document. Interested readers are referred to ArcGis documentation (ESRI 2013). For raster file output, a projection file is also written. This file has the same base name as the raster file, but it ends in the extension “.prj” rather than “.asc”. The name of the file is derived from the raster file name rather than being a separate input. The purpose of the projection file is to tell ArcGis what coordinate system is used for the data in the raster file. The assumption is made that the WGS84 coordinate system (NGA 2013) is used and location data are entered in decimal degrees.

7.3 Keyword Definitions for AirCombGridView

The keywords for the AirCombGridView code can generally be entered in any order. However, the last keyword in the file must be the END keyword. All of the keywords used in the AirCombGridView code are identified in alphabetical order in Table 7.3.

Table 7.3 Summary of Keywords Used in the AirCombGridView Code

Keyword	Section	Purpose
END	7.3.1	The END keyword signifies the end of data in the file.
FILE	7.3.2	The FILE keyword is used to enter names for input and output files.
INTERPOL	7.3.3	The optional INTERPOL keyword is used to modify the default interpolation scheme used to produce contour data.
KMLBLANK	7.3.4	The optional KMLBLANK keyword is used to identify an input polygon file in KML format. Use of this file will cause Google Earth to display the polygon on the dose map.
KMLCOLOR	7.3.5	The optional KMLCOLOR keyword is used to modify the default colors for contours in a KML file.
KMLDOCUM	7.3.6	The KMLDOCUM keyword is used to define the KML "document name" that will display in the navigation pane when the KML file is opened in Google Earth.
KMLLEGEN	7.3.7	The optional KMLLEGEN keyword is used to identify a graphics file that will display a legend when the KML file is opened in Google Earth.
KMLLEVEL	7.3.8	The KMLLEVEL keyword is used to identify the contour levels in the output KML file.
KMLLOGO	7.3.9	The optional KMLLOGO keyword is used to identify a graphics file that will display a logo when the KML file is opened in Google Earth.
SCALE	7.3.10	The optional SCALE keyword defines a multiplicative scale factor that is applied to the data to be plotted.
USER	7.3.11	The USER keyword is used to enter the name of the user running the code.
VERBOSE	7.3.12	The optional VERBOSE keyword can be used to increase the amount of information written to standard out as the run progresses.
VIEW	7.3.13	The VIEW keyword is used to identify the viewing region for the data contours.

7.3.1 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword's syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

7.3.2 FILE Keyword

The FILE keyword is used to enter the names of input and output files. The following is this keyword's syntax:

```
FILE modifier "quote"
```

The file names are entered in quote strings. Path names up to 256 characters long (name length limitation in Windows 7) are supported. At least one FILE keyword is required for every run of the code. The modifiers associated with the FILE keyword are described in Table 7.4.

Table 7.4 Modifiers Associated with the FILE Keyword for AirCombGridView

Modifier	Description
DATA	The quote string associated with the DATA modifier contains the name of the input file that contains concentration or deposition data on a grid. These files are produced by the AirCombGrid code (see Table 6.5).
KML	The quote string associated with the KML modifier contains the name of the output file that will contain plot data structured as KML for use in Google Earth. This output file is optional. Do not enter this modifier if the KML file is not desired. If this file is not defined, then the KMLCOLOR, KMLDOCUM, KMLLEGEN and KMLLEVEL keywords are not needed.
RASTER	The quote string associated with the RASTER modifier contains the name of the output file that will contain plot data in a raster file suitable for viewing in ArcGis. This output file is optional. Do not enter this modifier if the raster file is not desired. The raster file name must end in ".asc" so ArcGis knows how to process the file.
REPORT	The quote string associated with the REPORT modifier contains the name of the output file containing information about the progress of the run. All error messages are directed to this file.

The following entries define all of the data files identified in Table 7.4.

```
FILE REPORT "Example.rpt"
FILE DATA   "grid_1948_12_conc.txt"
FILE KML    "grid_1948_12_conc.kml"
FILE RASTER "grid_1948_12_conc.asc"
```

The ArcGis program uses a "projection" file to identify which coordinate system applies to the data in the raster file. The AirCombGridView file writes a projection file for ArcGis that assumes the WGS84 coordinate system (NGA 2013) is used with units of decimal degrees. This file has the same name as the raster file, but it ends in the extension ".prj". If the raster file has the name **`grid_1948_12_conc.asc`**, then the projection file has the name **`grid_1948_12_conc.prj`**.

7.3.3 INTERPOL Keyword

The optional INTERPOL keyword is used to modify the default interpolation method used to map the input data onto a regularly spaced grid for the contouring algorithm. The following is this keyword's syntax:

```
INTERPOL [NEAREST|DISTANCE] (IGNORE=N1)
```

The default method interpolation uses inverse distance weighting (squared distance) on only the four input data points closest to the contouring point. This method is selected by entering the modifier DISTANCE. As an alternative, a closest neighbor algorithm is also provided and it can be selected using the NEAREST modifier. No interpolation is done in the nearest neighbor algorithm. If a point on the contouring grid is nearly coincident with an input data point, then both algorithms return the data value rather than an interpolated value.

The numerical value associated with the optional modifier DISTANCE implements a further distance restriction. The numerical value contains a distance with units of kilometers. Data points selected for inclusion in the interpolation that are further than N1 km from the point being interpolated are excluded from the interpolation. Small values can lead to situations where no data values are retained in the interpolation algorithm. An interpolated value of zero is returned in this situation. Large values can result in the nearest neighbor being a long distance from the interpolated point, especially if the contour grid is larger than the span of the input data. For the application using the grid shown in Figure 1.2, a value of about 7.5 km is recommended, and it is the default value in the code.

The INTERPOL keyword can be omitted if the default algorithm is desired. The following example keyword would implement the default algorithm:

```
INTERPOL DISTANCE IGNORE=7.5
```

7.3.4 KMLBLANK Keyword

The optional KMLBLANK keyword is used to identify an input polygon file in KML format. Use of this file will cause Google Earth to display the polygon on the dose map.

```
KMLBLANK “quote”
```

The quote string on this keyword contains the file name. The full pathname, including the drive letter, should be used. An example polygon file is provided in Table 10.2. An example use of this keyword is the following:

```
KMLBLANK “d:\mayak\best\polygon.html”
```

7.3.5 KMLCOLOR Keyword

The optional KMLCOLOR keyword is used to modify the default colors for contours in a KML file. This keyword has no effect and is not needed if a KML file is not written. Up to 10 contours may be used. The following is this keyword's syntax:

```
KMLCOLOR modifier1 “quote1” ... (modifier10 “quote10”)
```

The modifiers associated with the KMLCOLOR keyword are described in Table 7.5.

Table 7.5 Modifiers Associated with the KMLCOLOR Keyword for AirCombGridView

Modifier	Description
COLOR_01	The quote string associated with the COLOR_01 modifier contains the information to set the opacity and color for contour level 1 (the largest values). The quote string must be 8 characters long and contain four hexadecimal numbers. The first value is the opacity of the contour. A value of "B3" is recommended. Smaller values will be less opaque and show more of the underlying map layer. The second value is the intensity of the color blue. The third value is the intensity of the color green. The fourth value is the intensity of the color red. An example for a red contour is "B30000FF". An example for a green contour is "B300FF00".
COLOR_02	The quote string associated with the COLOR_02 modifier contains the information to set the opacity and color for contour level 2.
COLOR_03	The quote string associated with the COLOR_03 modifier contains the information to set the opacity and color for contour level 3.
COLOR_04	The quote string associated with the COLOR_04 modifier contains the information to set the opacity and color for contour level 4.
COLOR_05	The quote string associated with the COLOR_05 modifier contains the information to set the opacity and color for contour level 5.
COLOR_06	The quote string associated with the COLOR_06 modifier contains the information to set the opacity and color for contour level 6.
COLOR_07	The quote string associated with the COLOR_07 modifier contains the information to set the opacity and color for contour level 7.
COLOR_08	The quote string associated with the COLOR_08 modifier contains the information to set the opacity and color for contour level 8.
COLOR_09	The quote string associated with the COLOR_09 modifier contains the information to set the opacity and color for contour level 9.
COLOR_10	The quote string associated with the COLOR_10 modifier contains the information to set the opacity and color for contour level 10 (smallest values).

The following keyword entry redefines all ten default contour colors.

```
KMLCOLOR Color_01="B30000FF" Color_02="B33399FF"
   Color_03="B300FFFF" Color_04="B3B4FFFF" Color_05="B300FF00"
   Color_06="B310FFA8" Color_07="B3FF0000" Color_08="B3FF9900"
   Color_09="B3FFFF00" Color_10="B3FFFFCC"
```

The following keyword entry redefines the default colors only for contours 2 and 4.

```
KMLCOLOR Color_02="B33399FF" Color_04="B3B4FFFF
```

7.3.6 KMLDOCUM Keyword

The KMLDOCUM keyword is used to define the KML "document name" that will display in the navigation pane when the KML file is opened in Google Earth. This keyword has no effect and is not needed if a KML file is not written. The following is this keyword's syntax:

```
KMLDOCUM "quote"
```

The document name is entered in a quote string. Up to 256 characters can be entered, but short names are recommended. The following keyword example identifies that the file contains concentrations of ^{131}I in units of Bq/m^3 for December of 1948.

```
KMLDOCUM="I131 Conc 1948_12 Bq/m^3"
```

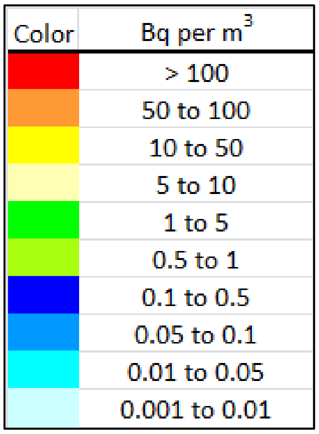
7.3.7 KMLLEGEN Keyword

The optional KMLLEGEN keyword is used to identify a graphics file that will display a legend in the upper left corner of the screen when the KML file is opened in Google Earth. This keyword has no effect and is not needed if a KML file is not written. The following is this keyword's syntax:

```
KMLLEGEN "filename"
```

The file name is entered in a quote string. Path names up to 256 characters long (name length limitation in Windows 7) are supported. A path name, including the drive letter, must be used for the file name so Google Earth can locate the file.

Legends are not required for the KML output option, but they enhance the interpretation of the map. An optional legend is included in the KML file if this keyword is used to point to a file on the user's hard drive that contains the graphic legend in *.gif or *.jpg format. This legend file must be prepared using an external program. An example legend file is shown in Figure 7.2.



Color	Bq per m^3
🟥	> 100
🟧	50 to 100
🟨	10 to 50
🟨	5 to 10
🟩	1 to 5
🟩	0.5 to 1
🟦	0.1 to 0.5
🟦	0.05 to 0.1
🟦	0.01 to 0.05
⬜	0.001 to 0.01

Figure 7.2 Example Optional Legend File for the AirCombGridView Code

The following example keyword identifies the above graphics file as a legend file, assuming that the file name is Conc_Contours.gif. A full pathname, including the drive letter, is recommended.

```
KMLLEGEN "H:\Mayak\Best\AirCombGrid\Conc_Contours.gif"
```

7.3.8 KMLLEVEL Keyword

The KMLLEVEL keyword is used to identify the contour levels in the output KML file. This keyword has no effect and is not needed if a KML file is not written. The following is this keyword's syntax:

```
KMLLEVEL N1 N2 (N3) ... (N10) [ABSOLUTE |RELATIVE]
```

This keyword defines the contouring levels in terms of absolute values (modifier ABSOLUTE) or fractions of the maximum data values (modifier RELATIVE). Data values below the minimum contour level will not show on the contour plot. Data values at or above the highest contour level will show in the highest contour. At least one contour level must be entered. A maximum of ten contours can be defined.

An example KMLLEVEL keyword that matches the contour levels shown in the legend file in Figure 7.2 is the following:

```
KMLLEVEL 100 50 10 5 1 0.5 0.1 0.05 0.01 0.001 ABSOLUTE
```

The report file (see Table 7.2) contains information on the selected contours as well as the range of input data. An example KMLLEVEL keyword that bases the contour levels on a relative fraction of the maximum data value is the following:

```
KMLLEVEL 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 RELATIVE
```

7.3.9 KMLLOGO Keyword

The optional KMLLOGO keyword is used to identify a graphics file that will display in the upper right corner of the screen when the KML file is opened in Google Earth. This keyword has no effect and is not needed if a KML file is not written. The following is this keyword's syntax:

```
KMLLOGO "filename"
```

The file name is entered in a quote string. Path names up to 256 characters long (name length limitation in Windows 7) are supported. A path name, including the drive letter, must be used for the file name if the output KML file resides on a different drive than the drive where Google Earth is installed.

The following example keyword identifies a graphics file that will be displayed as a logo: A full pathname, including the drive letter, is recommended.

```
KMLLOGO "D:\Fortran\HEDR\AirCombGridView\Test\PNNL_Color_Logo_Vertical.jpg"
```

7.3.10 SCALE Keyword

The optional SCALE keyword defines a multiplicative scale factor that is applied to the data to be plotted. The following is this keyword's syntax:

```
SCALE N1
```

This keyword is useful for changing units on the data in the grid file. For example, the following keyword converts the data in air concentration grid files output by the AirCombGrid code from units of Ci-s/m^3 to units of Bq/m^3.

```
SCALE 4.28241E5
```

7.3.11 USER Keyword

The USER keyword is used to identify the user of the program. The user name will be written to output files for labeling purposes. The program will error terminate if the user name is not supplied. The following is this keyword's syntax:

```
USER "quote"
```

The user name is entered in a quote string. User names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code:

```
USER "John Q. Public"
```

There are no modifiers associated with the USER keyword.

7.3.12 VERBOSE Keyword

The optional VERBOSE keyword can be used to increase the amount of information written to standard out as the run progresses. It has no effect on the actual contouring calculations. The following is this keyword's syntax:

```
VERBOSE
```

There are no modifiers or quote strings associated with the VERBOSE keyword.

7.3.13 VIEW Keyword

The VIEW keyword is used to define the grid used for outputting contours to the KML and raster files. The following is this keyword's syntax:

```
VIEW LATMIN=N1 LATMAX=N2 LONMIN=N3 LONMAX=N4 STEP=N5
```

The modifiers associated with the VIEW keyword are described in Table 7.6.

Table 7.6 Modifiers Associated with the VIEW Keyword for AirCombGridView

Modifier	Description
LATMIN	The numerical value associated with the LATMIN modifier identifies the minimum latitude (decimal degrees) for the grid.
LATMAX	The numerical value associated with the LATMIN modifier identifies the maximum latitude (decimal degrees) for the grid.
LATDEL	The numerical value associated with the LATDEL modifier identifies the latitude spacing (decimal degrees) for the grid.
LONMIN	The numerical value associated with the LONMIN modifier identifies the minimum longitude (decimal degrees) for the grid.
STEP	The numerical value associated with the STEP modifier identifies the spacing (decimal degrees) to use on the viewing grid.

The grid used for determining contours in the output KML and raster files is defined differently than the grid used by the AirCombGrid, DESCARTES and CiderF codes. It is a rectangular region specified by minimum and maximum latitude and longitude values. It has an internal latitude and longitude spacing defined by the STEP modifier.

The following example VIEW keyword was used in preparing the example file shown in Figure 7.1.

```
   VIEW LATMIN=55.07 LATMAX=56.0 LONMIN=59.95 LONMAX=62.15 STEP=0.01
```

Large values of the grid spacing (STEP modifier) lead to a very coarse plot. Smaller values give a more visually pleasing plot, but at the expense of a larger KML file. A value near 0.0025 is a reasonable compromise for the example grid shown in Figure 7.1.

8.0 FROSTPUNO – User Instructions

The DESCARTES code requires stochastic inputs for the last spring frost date and the first fall frost date at every node in the modeling domain. These dates are used to define the local vegetation growing season. This section describes a utility code FROSTPUNO that builds the data files needed by DESCARTES.

The FROSTPUNO utility code performs the same functions as the FROSTP (Miley et al. 1994) utility code with two major simplifications. First, the same statistical distribution of frost dates is used at every node rather than potentially using a different statistical distribution at each node. Second, the same frost date statistical distribution applies to all years in the run by DESCARTES rather than changing the statistical distribution for every year.

If detailed historical weather data are available across the modeling domain, then the FROSTP utility code should be used. If summary frost date ranges are available only for one location in a small modeling domain, then the FROSTPUNO utility code can be used. The FROSTP utility code could still be used, but a large number of input keyword files would have to be generated rather than a single short keyword file.

8.1 How the Code Is Invoked

The FROSTPUNO code runs under the Windows 7 operating system and it executes at a command prompt. A run of srcSetupRuns is initiated by entering the following command line:

```
FROSTPUNO "Keyfilename"
```

For this command, FROSTPUNO is the name of the executable program, and “Keyfilename” is the name of a controlling keyword file. Both the name of the executable program and the keyword file may contain path information. The keyword file contains text control information for the code. If the FROSTPUNO code cannot open the keyword file it will terminate execution after writing an error message to the standard output device.

8.2 Input and Output Files

The FROSTPUNO code reads one file and writes two or more files, depending on the inputs. These input and output files are defined in the following sections.

8.2.1 Input Files

An example input keyword control file for the FROSTPUNO code is provided in Table 8.1. Detailed descriptions of the individual keywords are described in Section 8.3.

Table 8.1 Example Keyword File for FROSTPUNO

```
REPORT "FrostStoc.rpt"
TITLE "Mayak Frost Date Library for 1948 (Baladino Airport, Chelyabinsk)"
USER "Paul W. Eslinger"
FROST "FrostStoc"
SEED 1948
YEARS 1948 1972
REALIZATIONS 250
ECHO
SPRING 3  89 102 122
FALL   3 275 297 314
END
```

8.2.2 Output Files

The FROSTPUNO code always writes a report file. Excerpts from a report file associated with the example keyword file in Table 8.1 are provided in Table 8.2. Error messages for run time errors trapped by the code are written to this file.

Table 8.2 Excerpts from a Report File Written by FROSTPUNO

```
                                FrostpUno 4.0.0
                          Last Modified on 21 May 2013

            Current Run ID = 20130521124146   User Name = Paul W. Eslinger
                 System Date = 05-21-2013   System Time = 12:41:46

Title: Mayak Frost Date Library for 1948 (Baladino Airport, Chelyabinsk)

 250 Realizations are being generated

Definition for variable NODE_xxxx_SPRING
   The distribution type is Discrete Uniform
   The lower limit is      89
   The upper limit is     122

Definition for variable NODE_xxxx_FALL
   The distribution type is Discrete Uniform
   The lower limit is     275
   The upper limit is     314

                       Summary statistics for Frost Dates
  Year  Time      Average     St. Dev.      Minimum       Median      Maximum
  1948 Spring   1.0550E+02   9.8238E+00          89          105          122
  1948 Fall     2.9449E+02   1.1568E+01         275          294          314
  1949 Spring   1.0548E+02   9.8320E+00          89          105          122
  1949 Fall     2.9448E+02   1.1565E+01         275          294          314
  1950 Spring   1.0550E+02   9.8283E+00          89          105          122
  1950 Fall     2.9450E+02   1.1579E+01         275          294          314
  1951 Spring   1.0548E+02   9.8300E+00          89          105          122
  1951 Fall     2.9450E+02   1.1560E+01         275          294          314
  1952 Spring   1.0550E+02   9.8345E+00          89          105          122
  1952 Fall     2.9450E+02   1.1566E+01         275          294          314
```

A frost date file is written by the FROSTPUNO code for every year in the problem definition for the DESCARTES run. Although the typical user will never modify these files, the first 20 lines of one of the files are provided in Table 8.3. There are 7 header lines. After that, each data line contains a node number, a realization number, a spring frost day (Julian day) and a fall frost day (Julian day). All realizations for one node are written before the data for the next node are output. This file was developed for a run with 516 nodes and 250 realizations and it is 129007 lines long.

Table 8.3 Excerpts from a Frost Date File Written by FROSTPUNO

```
Mayak Frost Date Library for 1948 (Baladino Airport, Chelyabinsk)
FrostpUno
4.0.0
Paul W. Eslinger
12:41:46
05-21-2013
 250
   1    1  89 280
   1    2  96 298
   1    3 120 310
   1    4 105 282
   1    5 117 277
   1    6 107 282
   1    7 112 294
   1    8 104 282
   1    9  98 306
   1   10 115 296
   1   11  90 289
   1   12 116 313
   1   13 113 280
```

8.3 Keyword Descriptions for FROSTPUNO

The keywords for the FROSTPUNO code can generally be entered in any order. However, there are two restrictions. The first keyword in the file must be the REPORT keyword. The last keyword in the file must be the END keyword. Any keyword or data entered after an END keyword is ignored.

Because the FROSTPUNO code is a slight modification of the FROSTP code, it uses the older keyword decoding routines and stochastic variable generations in the original DESCARTES framework. The DESCARTES user guide (Miley et al. 1994) should be consulted for those older descriptions if run-time errors are encountered in FROSTPUNO. All of the keywords used in the FROSTPUNO code are identified in alphabetical order in Table 8.4.

Table 8.4 Summary of Keywords Used in the FROSTPUNO Code

Keyword	Section	Purpose
ECHO	8.3.1	The optional ECHO keyword is used to invoke writing the definitions of the statistical distribution for every node to the report file.
END	8.3.2	The END keyword signifies the end of all keyword data.
FALL	8.3.3	The FALL keyword is used to define the statistical distribution for the first frost date (Julian day) in the fall.
FROST	8.3.4	The FROST keyword is used to enter the base name for the output frost date files.
REALIZAT	8.3.5	The REALIZAT keyword defines the number of realizations to process.

Keyword	Section	Purpose
REPORT	8.3.6	The REPORT keyword is used to enter the name of the output report file. It must be the first keyword in the file.
SEED	8.3.7	The SEED keyword defines the seed for the random number generator.
SPRING	8.3.8	The SPRING keyword is used to define the statistical distribution for the last frost date (Julian day) in the spring.
TITLE	8.3.9	The TITLE keyword is used to define a single-line problem title.
USER	8.3.10	The USER keyword is used to identify the user of the program.
YEARS	8.3.11	The YEARS keyword is used to define the range of years for output of frost date files for use in DESCARTES.

8.3.1 ECHO Keyword

The optional ECHO keyword is used to invoke writing the definitions of the statistical distribution for every node to the report file. An abbreviated report file will be written if this keyword is not entered. The following is this keyword's syntax:

```
ECHO
```

There are no modifiers or quote strings associated with the ECHO keyword.

8.3.2 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword's syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

8.3.3 FALL Keyword

The FALL keyword is used to define the statistical distribution for the first frost date (Julian day) in the fall. The following is this keyword's syntax:

```
FALL 3 N1 N2
```

The number 3 denotes the discrete uniform distribution (see appendix D of (Miley et al. 1994)) and is the only recommended distribution. The numbers N1 and N2 denote the range of days (Julian days) for the range of first fall frost dates. The following example keyword indicates that the first fall frost occurs randomly between day 275 and day 314.

```
FALL 3 275 314
```

8.3.4 FROST Keyword

The FROST keyword is used to enter the base name for the output frost date files. The following is this keyword's syntax:

```
FROST "quote"
```

The base file name is entered in a quote string. As many frost date files will be written as years are defined on the YEARS keyword (see Section 8.3.11). The name of any specific output file is generated by appending the 4 digit year and the string ".dat" to the base file name. An example for this keyword is the following:

```
FROST “C:\Projects\Tmp\Frost\Base”
```

If above keyword is matched with the following YEARS keyword,

```
YEARS 1948 1952
```

then, five output frost date files with the following names would be written:

```
C:\Projects\Tmp\Frost\Base1948.dat
C:\Projects\Tmp\Frost\Base1949.dat
C:\Projects\Tmp\Frost\Base1950.dat
C:\Projects\Tmp\Frost\Base1951.dat
C:\Projects\Tmp\Frost\Base1952.dat
```

8.3.5 REALIZAT Keyword

The REALIZAT keyword defines the number of realizations to process. The following is this keyword's syntax:

```
REALIZAT N1
```

The number of realizations must be entered as an integer. The following keyword identifies that a total of 250 realizations will be used.

```
REALIZAT 250
```

The following keyword identifies that 1 realization will be used.

```
REALIZAT 1
```

8.3.6 REPORT Keyword

The REPORT keyword is used to enter the name of the output report file. It must be the first keyword in the file. The following is this keyword's syntax:

```
REPORT "quote"
```

The report file name is entered in a quote string. Path names up to 256 characters long (name length limitation in Windows 7) are supported.

8.3.7 SEED Keyword

The SEED keyword defines the seed for the random number generator. The following is this keyword's syntax:

```
SEED N1
```

The value for N1 must be an integer or whole number in the range 1 to 999999. Even though the random number generation algorithm can handle larger seeds, the RDBLK routines from the DESCARTES framework cannot correctly decode numerical entries that contain more than 10 digits. The following example defines a seed with a value of 33431.

```
SEED 33431
```

8.3.8 SPRING Keyword

The SPRING keyword is used to define the statistical distribution for the last frost date (Julian day) in the spring. The following is this keyword's syntax:

```
SPRING 3 N1 N2
```

The number 3 denotes the discrete uniform distribution (see appendix D of (Miley et al. 1994)) and is the only recommended distribution. The numbers N1 and N2 denote the range of days (Julian days) for the range of last spring frost dates. The following example keyword indicates that the last spring frost occurs randomly between day 89 and day 122.

```
SPRING 3 89 122
```

8.3.9 TITLE Keyword

The TITLE keyword is used to define a single-line problem title up to 72 characters in length. The problem title will be written to output files. The program will error terminate if the title is not supplied. The following is this keyword's syntax:

```
TITLE "quote"
```

The title is entered in a quote string. The following example defines a title for a run of the code:

```
TITLE "Example title line"
```

There are no modifiers associated with the TITLE keyword.

8.3.10 USER Keyword

The USER keyword is used to identify the user of the program. The user name will be written to output files for labeling purposes. The program will error terminate if the user name is not supplied. The following is this keyword's syntax:

```
USER "quote"
```

The user name is entered in a quote string. User names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code:

```
USER "John Q. Public"
```

There are no modifiers associated with the USER keyword.

8.3.11 YEARS Keyword

The YEARS keyword is used to define the range of years for output of frost date files for use in DESCARTES. The following is this keyword’s syntax:

```
YEARS N1 N2
```

The numbers N1 and N2 denote the range of years for generating output files. The last year must be equal to or greater than the first year.

The following example keyword indicates that files should be output for all years from 1948 through 1972.

```
YEARS 1948 1972
```

The following example keyword indicates that files should be output for only the year 1965.

```
YEARS 1965 1965
```

9.0 CiderF – User Instructions

9.1 Overview

The CiderF code computes doses to individuals using a suite of concentration data files generated by the DESCARTES code. Dose calculations are divided into two modes (see Section 9.6.2). The first mode, called the individual mode, applies to individuals living at specific locations. Example applications of this mode would be to compute the doses to a mother, father and their children that lived at one location for a number of years. The second mode is called a map mode. In map mode, one definition of an individual is applied for all locations. For example, map mode could be used to produce the dose a five-year old child would have received in 1965 at every node in the domain.

Although the CiderF code can use both individual or map modes, only one mode is allowed for any single run of the code. In the map mode, individuals are assumed to live at the same location in the model domain for an entire year. In the individual mode, the individual can move between locations in a given year, and they can even exit the domain for part of the year.

Concentrations produced by the DESCARTES code can be output using daily, weekly or monthly time steps. Consequently, doses are computed internally by CiderF on a daily basis. This approach allows individuals to change locations on specific days and it supports radioactive decay calculations for food holdup times. Daily, annual and total doses can be output for cases that span multiple years. Output files containing daily doses can be huge and typically only annual and total doses are output. The individual could live at up to 100 different locations in a specific dose case.

The CiderF code supports age dependent data for diets, exposure data and dose conversion factors. Doses can be calculated for prenatal infants (see Section 2.2), young children, or for the entire lifetime of a person. Doses calculated for prenatal and nursing infants also depend on the ^{131}I intake of their mother. For the purposes of the calculations, the gestation period for prenatal infant is considered to be exactly 266 days in length.

9.2 How the Code is Invoked

The CiderF code runs under the Windows operating system (Release 7). The code executes at a command prompt. A run of CiderF is initiated by entering the following command line:

```
   CiderF Keyfilename
```

For this command, CiderF is the name of the executable program and Keyfilename is the name of a keyword file containing text control information describing the run. Both the name of the executable program and the keyword file may contain path information. If CiderF is invoked without entering the name of the keyword file, the code will prompt the user for the file name. If CiderF cannot open the keyword file, then the code will terminate execution after writing an error message to the standard output device.

9.3 Memory and Disk Space Requirements

The CiderF code performs dynamic memory allocation, so different runs will use different amounts of memory. As an example, a run of individual cases using 1 realization utilized 36 Mb of memory. In contrast, a map run of 250 realizations for a domain with 516 nodes utilized 220 Mb of memory.

The CiderF code can be executed multiple times after the air transport and environmental accumulation codes have finished without rerunning the previous codes. Data files output by DESCARTES are required as input to CiderF. The disk space requirements for these data files depend on the options selected in DESCARTES. Disk space requirements for several cases where DESCARTES output concentration data on monthly intervals are as follows:

- A single realization case with 516 nodes in the model domain from 1948 through 1972 requires about 81 Mb of disk space for CiderF input files.
- A single realization case with 516 nodes in the model domain from 1948 through 1972 requires about 221 Mb of disk space for input files for both DESCARTES and CiderF.
- A 250 realization case with 516 nodes in the model domain from 1948 through 1972 requires about 3.33 Gb of disk space for CiderF input files.
- A 250 realization case with 516 nodes in the model domain from 1948 through 1972 requires about 12 Gb of disk space for input files for both DESCARTES and CiderF.

9.4 Data Categorization

A number of tokens are used to identify different categories of information in the preparation of inputs for CiderF. These tokens are identified in Table 9.1. The tokens are case sensitive.

Table 9.1 Tokens used in Preparing CiderF Keyword Files

Token	Purpose
Tokens used to identify food types Used in defining reference diets, special diets, diet scaling factors, and identifying the holdup times between harvest and consumption	
beef	The food type is meat from cattle.
eggs	The food type is eggs from poultry.
f_milk	The food type is fresh milk.
s_milk	The food type is stored milk (including cheese).
fruit	The food type is fruit.
grain	The food type is grain.
l_veg	The food type is leafy vegetables.
o_veg	The food type is other vegetables.
poultry	The food type is poultry (chicken).

Token	Purpose
Tokens used to identify food distribution types Used in identifying the holdup times between harvest and consumption	
distributed	Food production and consumptions occur in different locations and the holdup time includes transportation time.
local	Food production and consumption occurs at the same location.
Tokens used to identify seasons of the year Used in defining reference diets and the amount of time spent outdoors	
winter	The winter season. Index = 1 for reference diets.
spring	The spring season. Index = 2 for reference diets.
fall	The fall season. Index = 3 for reference diets.
summer	The summer season. Index = 4 for reference diets.
Tokens used to identify sex (gender) Used in defining dose factors, reference diets and the amount of time spent outdoors	
male	The individual is a male. Index = 1 for reference diets.
female	The individual is a female. Index = 2 for reference diets.
Tokens used to identify lifestyle types Used in defining reference diets and the amount of time spent outdoors	
rural	The individual lives in a rural setting. Index = 1 for reference diets.
urban	The individual lives in an urban setting. Index = 2 for reference diets.
Tokens used to identify internal exposure pathway types Used in defining dose factors	
ing	Exposure to contaminants due to ingestion.
inh	Exposure to contaminants due to inhalation.
Tokens used to identify external exposure pathway types Used in defining dose factors	
imm	Exposure to contaminants due to immersion in air.
u_soil	Exposure from contaminant in the upper soil layer, or contaminant lying on the ground surface.
rz_soil	Exposure from contaminant in the root zone soil layer.
Tokens used to identify lifestyle activity types	
normal	Normal activity for females and males of all ages. Index = 1 for reference diets.
suckling	Activity denoting that a child is suckling from the mother. Index = 2 for reference diets.
preg_nurse	Activity denoting that a woman is either pregnant or nursing a child. Index = 3 for reference diets.
prenatal	Activity denoting a fetus (prenatal child).

Token	Purpose
Tokens used to identify organs Used in defining dose factors	
thyroid	The dose factors apply to the thyroid.
whole_body	The dose factors apply to the whole body.
Tokens used to identify food types Used in defining special and reference diets	
milk_1i	Milk consumed comes from an individual cow on feeding regime 1.
milk_1h	Milk consumed comes from a herd cow on feeding regime 1.
milk_2i	Milk consumed comes from an individual cow on feeding regime 2.
milk_2h	Milk consumed comes from a herd cow on feeding regime 2.
milk_3i	Milk consumed comes from an individual cow on feeding regime 3.
milk_3h	Milk consumed comes from a herd cow on feeding regime 3.
milk_4i	Milk consumed comes from an individual cow on feeding regime 4.
milk_4h	Milk consumed comes from a herd cow on feeding regime 4.
milk_gro	Milk consumed is cow's milk from a grocery store.
milk_cre	Milk consumed is cow's milk from a specific creamery.
cream_nu	Number identifying the specific creamery to use for consuming creamery milk.
milk_goa	Milk consumed comes from a local goat.
vegloc	Leafy vegetables consumed come from local produce (rather than commercial produce).

9.5 File Definitions

A run of the CiderF code uses two input keyword files, 21 or more input media concentration files and one input data file used to transfer the values of common parameters from DESCARTES to CiderF. The CiderF code always writes two output files and it may write an additional two files depending on the options that are selected.

9.5.1 Input Files

A summary of the CiderF input files and their functions is provided in Table 9.2.

Table 9.2 Summary of CiderF Input Files

File	Function
Case keyword file	The case keyword file contains the minimal set of control information needed to define individual dose cases or map cases. This file is in text format and is always required. The name and location of the case keyword file are provided on the command line when starting CiderF, or are provided in response to a prompt if omitted from the command line. Entries in this file are described in Section 9.6.
Factors keyword file	The factors keyword file contains control information that does not vary even if doses are calculated for more than one individual or one map. This file is in text format and is always required. The name and location of the case keyword file are provided in the case keyword file. Entries in this file are described in Section 9.7.
Air concentration	The air concentration file is a binary file written by DESCARTES that contains time varying concentrations in air at every node in the domain. This file is identified by the MED_AIRC modifier (see Section 9.7.10). This file is always required.
Air deposition	The air deposition file is a binary file written by DESCARTES that contains time varying air deposition at every node in the domain. This file is identified by the MED_AIRD modifier (see Section 9.7.10). This file is always required.
Beef concentration	The beef concentration file is a binary file written by DESCARTES that contains time varying concentrations in beef (meat) at every node in the domain. This file is identified by the MED_BEEF modifier (see Section 9.7.10). This file is always required.
Egg concentration	The egg concentration file is a binary file written by DESCARTES that contains time varying concentrations in eggs at every node in the domain. This file is identified by the MED_EGG modifier (see Section 9.7.10). This file is always required.
Fruit concentration for inner compartment	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in the inner compartment of fruit at every node in the domain. This file is identified by the MED_FRTI modifier (see Section 9.7.10). This file is always required.
Fruit concentration for outer compartment	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in the outer compartment of fruit at every node in the domain. This file is identified by the MED_FTRO modifier (see Section 9.7.10). This file is always required.
Goat milk concentration	The air concentration file is a binary file written by DESCARTES that contains time varying concentrations in goat milk at every node in the domain. This file is identified by the MED_GOAT modifier (see Section 9.7.10). This file is always required.
Grain concentration	The grain concentration file is a binary file written by DESCARTES that contains time varying concentrations in grain at every node in the domain. This file is identified by the MED_GRN modifier (see Section 9.7.10). This file is always required.

File	Function
Grocery milk concentration	The air concentration file is a binary file written by DESCARTES that contains time varying concentrations in grocery milk at every node in the domain. This file is identified by the MED_MGRO modifier (see Section 9.7.10). This file is always required.
Local leafy vegetables concentration for inner compartment	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in the inner compartment of local leafy vegetables at every node in the domain. This file is identified by the MED_LVIL modifier (see Section 9.7.10). This file is always required.
Local leafy vegetables concentration for outer compartment	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in outer compartment of local leafy vegetables at every node in the domain. This file is identified by the MED_LVOL modifier (see Section 9.7.10). This file is always required.
Commercial leafy vegetables concentration for inner compartment	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in the inner compartment of commercial leafy vegetables at every node in the domain. This file is identified by the MED_LVIC modifier (see Section 9.7.10). This file is always required.
Commercial leafy vegetables concentration for outer compartment	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in the outer compartment of commercial leafy vegetables at every node in the domain. This file is identified by the MED_LVOC modifier (see Section 9.7.10). This file is always required.
Creamery milk concentration	The creamery milk concentration file is a binary file written by DESCARTES that contains time varying concentrations in creamery milk for every creamery. This file is identified by the MED_MCRM modifier (see Section 9.7.10). This file is always required.
Herd cow milk concentration for feeding regime 1	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for a herd cow in feeding regime 1. This file is identified by the MED_MR1H modifier (see Section 9.7.10). This file is always required.
Herd cow milk concentration for feeding regime 2	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for a herd cow on feeding regime 2. This file is identified by the MED_MR2H modifier (see Section 9.7.10). This file is optional.
Herd cow milk concentration for feeding regime 3	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for a herd cow on feeding regime 3. This file is identified by the MED_MR3H modifier (see Section 9.7.10). This file is optional.
Herd cow milk concentration for feeding regime 4	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for a herd cow on feeding regime 4. This file is identified by the MED_MR4H modifier (see Section 9.7.10). This file is optional.

File	Function
Individual cow milk concentration for feeding regime 1	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for an individual cow on feeding regime 1. This file is identified by the MED_MR1I modifier (see Section 9.7.10). This file is always required
Individual cow milk concentration for feeding regime 2	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for an individual cow on feeding regime 2. This file is identified by the MED_MR2I modifier (see Section 9.7.10). This file is optional.
Individual cow milk concentration for feeding regime 3	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for an individual cow on feeding regime 3. This file is identified by the MED_MR3I modifier (see Section 9.7.10). This file is optional.
Individual cow milk concentration for feeding regime 4	This concentration file is a binary file written by DESCARTES that contains time varying concentrations in milk at every node in the domain for an individual cow on feeding regime 4. This file is identified by the MED_MR4I modifier (see Section 9.7.10). This file is optional.
Other vegetable concentration	The other vegetable concentration file is a binary file written by DESCARTES that contains time varying concentrations in other vegetables at every node in the domain. This file is identified by the MED_OVG modifier (see Section 9.7.10). This file is always required.
Poultry concentration	The poultry concentration file is a binary file written by DESCARTES that contains time varying concentrations in poultry at every node in the domain. This file is identified by the MED_POUL modifier (see Section 9.7.10). This file is always required.
Root zone soil concentration	The root zone concentration file is a binary file written by DESCARTES that contains time varying concentrations in root zone soil at every node in the domain. This file is identified by the MED_RZ modifier (see Section 9.7.10). This file is always required.
Upper soil layer concentration	The upper soil layer concentration file is a binary file written by DESCARTES that contains time varying concentrations in the upper soil layer at every node in the domain. This file is identified by the MED_USL modifier (see Section 9.7.10). This file is always required.
Common parameters	The common parameters file is a text file written by DESCARTES that contains parameters common to DESCARTES and CiderF. This file is identified by the USLR modifier (see Section 9.7.10). This file is always required.

The keyword inputs for CiderF are divided into two files. One file (factors keyword file) contains information that does not vary even if doses are calculated for more than one individual. The other file (case keyword file) contains the minimal set of keywords needed to define individual doses cases or a map case. An example keyword file containing two individual mode cases is provided in Table 9.3. The entries in the file that are colored red are comments ignored by the program. Detailed definitions of the entries in this file are provided in Section 9.6.

Table 9.3 Example CiderF Keyword File Containing Two Individual Mode Cases

```
USER "Paul W. Eslinger"
! Required Files
FILE REPORT "Child5.rpt"
FILE RESULT "Child5.csv"
FILE FACTORS "FactorsBest.kwd"
! Number of realizations
REALIZAT 1
! Random number seed
SEED 1223.
VERBOSE !MEDIA
ORGANS THYROID !WHOLEBODY
! Possible dose pathways
DOSEPATH EXTERNAL INHALATION BEEF LEAFYVEG OTHERVEG FRUIT GRAIN
  POULTRY EGGS MILK TOTAL !ANNUAL DAILY
! Special diets must explicitly define a milk source
DIETSPEC ID="SpecDiet01" milk_gro=1 vegloc=1 TITLE="First special diet"
!
CASE INDIVIDUAL ID="C5-48-Rur"
START   "1948-06-01"
FINISH  "1948-12-31"
SEX     "male"
BIRTH   "1943-01-01"
LIFESTYLE TYPE="urban" BEGIN="1948-06-01" FINISH="1972-12-31"
DIET ID="SpecDiet01"   BEGIN="1948-06-01" FINISH="1972-12-31"
NODE NUMBER=391        BEGIN="1948-06-01" FINISH="1972-12-31"
ENDCASE
!
CASE INDIVIDUAL ID="C5-49-Rur"
START   "1949-01-01"
FINISH  "1949-12-31"
SEX     "male"
BIRTH   "1944-01-01"
LIFESTYLE TYPE="urban" BEGIN="1948-01-01" FINISH="1972-12-31"
DIET ID="SpecDiet01"   BEGIN="1948-06-01" FINISH="1972-12-31"
NODE NUMBER=391        BEGIN="1948-06-01" FINISH="1972-12-31"
ENDCASE
!
END
```

Excerpts from a factors keyword file for CiderF are provided in Table 9.4.   Entries in the file that are colored red are comments ignored by the program. Detailed definitions of the entries in this file are provided in Section 9.7.

Table 9.4 Excerpts from a CiderF Factors Keyword File

```
! Nuclide
NUCLIDE ID="I131" HALFLIFE=8.0207 ! Half life in days
! Time span for data in the media files (must match with DESCARTES TIME keyword)
TIME 06:01:1948 12:31:1972
! Time step used for DESCARTES outputs
STEP MONTH
! Number of nodes
NUMNODES 516
! Creameries defined
CREAMERY 4
! Season mapping by date ranges (Julian day of year)
SEASON WINTER 1   102
SEASON SPRING 103 163
SEASON SUMMER 164 297
SEASON FALL   298 366
!-- Media File Definitions
FILE MED_AIRC "D:\Mayak\Best\ConcenBest.med"
FILE MED_AIRD "D:\Mayak\Best\DepositBest.med"
...
FILE MED_USL  "D:\Mayak\Best\SoilUpBest.med"
FILE USLR     "D:\Mayak\Best\PlantsBest.dat"
! ===> Define the age categories that apply to breathing rate
AGEBREAT 0.0 0.5 2.0 7.0 12.0 17.0 999.0 !Units = years
!  Age category 1 is  0.0 to   0.5
!  Age category 2 is  0.5 to   2.0
!  Age category 3 is  2.0 to   7.0
!  Age category 4 is  7.0 to  12.0
!  Age category 5 is 12.0 to  17.0
!  Age category 6 is 17.0 to 999.0
! ===> Define the age categories that apply to the fraction of time spent outdoors
AGEFRACO 0.0 2.0 17.0 999.0 !Units = years
! ===> Define the age categories that apply to internal exposure
AGEINT 0.0 0.25 1.0 5.0 10.0 15.0 999.0 !Units = years
! ===> Define the age categories that apply to diets
AGEDIET 0.0 0.5 2.0 7.0 12.0 17.0 999.0 !Units = years
! ===> Breathing rates in m^3/day (Data varies by age)
STOCHASTIC ID="BREATHE01" DIST="constant" PARAM 1.62 UNITS="m^3/day"
  TITLE="Breathing rate for age: 0 to 6 months"
...
! ===> Fraction of time spent outdoors
!      Data varies by age, lifestyle, season, and sex
STOCHASTIC ID="FRACOUT01ruralwintermale  " DIST="Constant" PARAM 0
  UNITS="none" TITLE="Fraction time outdoors for age 1 rural ruralwinter male  "
...
! ===> Dose factors for internal exposure
!      Data varies by pathway, organ, sex, age and realization
STOCHASTIC ID="INTERNALing01thyroid   male  " DIST="Constant" PARAM 1.100E+07
  UNITS="rad/Ci" TITLE="Dose factor for age 1 ing thyroid    male  "
...
! ===> Transfer data for prenatal infants
!      Data varies by organ and realization
STOCHASTIC ID="PRENATALthyroid   " DIST="Constant" PARAM 2.30E+6 UNITS="rad/Ci"
  TITLE="Transfer PRENATAL thyroid
STOCHASTIC ID="PRENATALwhole_body" DIST="Constant" PARAM 2.30E+6 UNITS="rem/Ci"
  TITLE="Transfer PRENATAL whole_body"
! ===> Transfer data for nursing infants
!      Data varies by organ and realization
STOCHASTIC ID="NURSINGthyroid   " DIST="Constant" PARAM 2.40E+06 UNITS="rad/Ci"
  TITLE="Transfer NURSING thyroid   "
STOCHASTIC ID="NURSINGwhole_body" DIST="Constant" PARAM 7.00E+04 UNITS="rem/Ci"
  TITLE="Transfer NURSING whole_body"
! ===> Dose factors for external exposure
!      Data varies by pathway, organ and realization
STOCHASTIC ID="EXTERNALimm    thyroid   " DIST="Constant" PARAM 5680
  UNITS="(rad/day)/(ci/m^3)" TITLE="Dose factor EXTERNAL imm     thyroid   "
STOCHASTIC ID="EXTERNALu_soil thyroid   " DIST="Constant" PARAM 102
  UNITS="(rad/day)/(ci/m^2)" TITLE="Dose factor EXTERNAL u_soil  thyroid   "
...
! Indoor to outdoor air activity ratios
STOCHASTIC ID="RIO" DIST="Constant" PARAM 0.68 UNITS="none"
  TITLE="Indoor to outdoor ratio"
STOCHASTIC ID="SHIELD" DIST="Constant" PARAM 0.5 UNITS="none"
  TITLE="Indoor to outdoor ratio"
! Holdup times in days (Data varies by food type and distribution type)
STOCHASTIC ID="HOLDUPbeef    local      " DIST="Constant" PARAM 14   UNITS="day"
  TITLE="Holdup time local       beef
STOCHASTIC ID="HOLDUPl_veg   local      " DIST="Constant" PARAM 3.5  UNITS="day"
```

```
  TITLE="Holdup time local       l_veg
...
! Ratio of dry weight to wet weight (Data varies by food type)
STOCHASTIC ID="DRYWETl_veg   " DIST="Constant" PARAM 0.07  UNITS="kg(dry)/kg(wet)"
  TITLE="Ratio (dry to wet weight)  for l_veg   "
STOCHASTIC ID="DRYWETo_veg   " DIST="Constant" PARAM 0.15  UNITS="kg(dry)/kg(wet)"
  TITLE="Ratio (dry to wet weight)  for o_veg   "
...
! Food processing retention fraction (Data varies by food type)
STOCHASTIC ID="RETAINl_veg   " DIST="Constant" PARAM 0.45 UNITS="none"
  TITLE="Food processing retention fraction for l_veg   "
STOCHASTIC ID="RETAINfruit   " DIST="Constant" PARAM 0.45 UNITS="none"
  TITLE="Food processing retention fraction for fruit   "
! Reference diet specifications
DIETREF ID="RefDiet001" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0
  milk_3h=0 milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0
  vegloc=1 TITLE="Child, 0 to 6 months"
DIETREF ID="RefDiet014" milk_1i=0 milk_1h=1 milk_2i=0 milk_2h=0 milk_3i=0
  milk_3h=0 milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0
  vegloc=1 TITLE="Adult, urban, pregnant/nursing female"
...
! f_milk (fresh milk) consumption for each reference diet
STOCHASTIC ID="RefDiet001f_milk " DIST="constant" PARAM 0.0383 UNITS="kg/day"
  TITLE="RefDiet001 f_milk  consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014f_milk " DIST="constant" PARAM 0.4675 UNITS="kg/day"
  TITLE="RefDiet014 f_milk  consumption" ! Adult, urban, pregnant/nursing female
...
! s_milk (stored milk) consumption for each reference diet
STOCHASTIC ID="RefDiet001s_milk " DIST="constant" PARAM 0.0068 UNITS="kg/day"
  TITLE="RefDiet001 s_milk  consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014s_milk " DIST="constant" PARAM 0.0825 UNITS="kg/day"
  TITLE="RefDiet014 s_milk  consumption" ! Adult, urban, pregnant/nursing female
...
! l_veg (leafy vegetable) consumption for each reference diet
STOCHASTIC ID="RefDiet001l_veg  " DIST="constant" PARAM 0      UNITS="kg/day"
  TITLE="RefDiet001 l_veg   consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014l_veg  " DIST="constant" PARAM 0.0825 UNITS="kg/day"
  TITLE="RefDiet014 l_veg   consumption" ! Adult, urban, pregnant/nursing female
...
! o_veg (other vegetable) consumption for each reference diet
STOCHASTIC ID="RefDiet001o_veg  " DIST="constant" PARAM 0      UNITS="kg/day"
  TITLE="RefDiet001 o_veg   consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014o_veg  " DIST="constant" PARAM 0.0825 UNITS="kg/day"
  TITLE="RefDiet014 o_veg   consumption" ! Adult, urban, pregnant/nursing female
...
! fruit consumption for each reference diet
STOCHASTIC ID="RefDiet001fruit  " DIST="constant" PARAM 0      UNITS="kg/day"
  TITLE="RefDiet001 fruit   consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014fruit  " DIST="constant" PARAM 0.022  UNITS="kg/day"
  TITLE="RefDiet014 fruit   consumption" ! Adult, urban, pregnant/nursing female
...
! grain consumption for each reference diet
STOCHASTIC ID="RefDiet001grain  " DIST="constant" PARAM 0      UNITS="kg/day"
  TITLE="RefDiet001 grain   consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014grain  " DIST="constant" PARAM 0.561  UNITS="kg/day"
  TITLE="RefDiet014 grain   consumption" ! Adult, urban, pregnant/nursing female
...
! egg consumption for each reference diet
STOCHASTIC ID="RefDiet001eggs   " DIST="constant" PARAM 0      UNITS="kg/day"
 TITLE="RefDiet001 eggs    consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014eggs   " DIST="constant" PARAM 0.0392 UNITS="kg/day"
  TITLE="RefDiet014 eggs    consumption" ! Adult, urban, pregnant/nursing female
...
```

```
! beef consumption for each reference diet
STOCHASTIC ID="RefDiet001beef   " DIST="constant" PARAM 0      UNITS="kg/day"
  TITLE="RefDiet001 beef    consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014beef   " DIST="constant" PARAM 0.209  UNITS="kg/day"
  TITLE="RefDiet014 beef    consumption" ! Adult, urban, pregnant/nursing female
...
! poultry consumption for each reference diet
STOCHASTIC ID="RefDiet001poultry" DIST="constant" PARAM 0      UNITS="kg/day"
  TITLE="RefDiet001 poultry consumption" ! Child, 0 to 6 months
STOCHASTIC ID="RefDiet014poultry" DIST="constant" PARAM 0.0105 UNITS="kg/day"
  TITLE="RefDiet014 poultry consumption" ! Adult, urban, pregnant/nursing female
...
! Mapping for reference diets by category
! rural, winter, male, normal, 0 to 6 months
DIETMAP Lifestyle=1 Season=1 Sex=1 Activity=1 Age=1 ID="RefDiet001"
! rural, winter, female, normal, 0 to 6 months
DIETMAP Lifestyle=1 Season=1 Sex=2 Activity=1 Age=1 ID="RefDiet001"
...
!Optional reference diet scaling by year
DIETSCAL YEAR = 1948 f_milk = 1.000 s_milk = 1.000 l_veg = 1.000 o_veg = 1.000
  fruit = 1.000 grain = 1.000 eggs = 1.000 beef = 1.000 poultry = 1.000
DIETSCAL YEAR = 1949 f_milk = 2.000 s_milk = 1.000 l_veg = 1.000 o_veg = 1.000
  fruit = 1.000 grain = 1.000 eggs = 1.000 beef = 1.000 poultry = 1.000
END
```

9.5.2 Output Files

Excerpts from a CiderF report file utilizing a single realization are provided in Table 9.5.

Table 9.5 Example Report File from CiderF

```
            ----------------------------------------------------
                           CiderF  Version 1.03.015
                         Last Modified on  7 Jun 2013
            ----------------------------------------------------
              Compute Stochastic Doses for Individuals or Maps
            ----------------------------------------------------
...
User:  Paul W. Eslinger

Nuclide to be processed is I131
       8.0207 : Halflife in days
 8.641978E-02 : Decay constant in 1/days

Input Keyword File Name    : "Child5_Ozyorsk.kwd"
Input Factors Keyword File : "D:\Mayak\kwd\RefDiet\MayakFactorsBest.kwd"
File Name for this file    : "Child5_Ozyorsk.rpt"
File Name for the results  : "Child5_Ozyorsk.csv"

Compute doses for: Thyroid
Output doses for pathway:  external
...
Number of reference diets: 14
Number of special diets:   1
Number of days of data in the DESCARTES media files: 8980

Media file opened for: AIRCON    File: D:\Mayak\Best\ConcenBest.med
Media file opened for: AIRDEP    File: D:\Mayak\Best\DepositBest.med
...
Processing case "C5-48-Urb" on 06/11/2013 at 15:54:32.03
!
```

```
! Echo of the Case definition keywords
CASE INDIVIDUAL ID="C5-48-Urb"
START    "1948-06-01"
FINISH   "1948-12-31"
SEX      "male"
BIRTH    "1943-01-01"
WASHED   NO
NODE     BEGIN="1948-06-01" FINISH="1972-12-31" NODE=391
LIFESTYL BEGIN="1948-06-01" FINISH="1972-12-31" TYPE="urban"
DIET     BEGIN="1948-06-01" FINISH="1972-12-31" ID="SpecDiet01"
ENDCASE
Case "C5-48-Urb" has 214 total days

Message originating in routine CiderF at 06/11/2013 on 15:54:32.643
Message: Normal Termination
```

A dose file from CiderF for a single individual case that uses 1 realization is provided in Table 9.6. Multiple individual cases can be output in the same file. The file is structured in comma separated variables format, so it is suitable for importing into a spreadsheet or database. The file contains a single header line. Each subsequent line identifies the case ID, the year for the annual dose, the dose pathway, and then the dose.

Table 9.6 Example Dose File from CiderF for an Individual Case

```
"CaseID","Date","Pathway","Organ","Realizations"
"C5-48-Rur",1948,"total","thyroid", 1.35685E+00
"C5-48-Rur",1948,"total_external","thyroid", 9.82763E-04
"C5-48-Rur",1948,"total_inhalation","thyroid", 1.23090E-01
"C5-48-Rur",1948,"total_ingest_beef","thyroid", 1.90539E-01
"C5-48-Rur",1948,"total_ingest_leafyveg","thyroid", 6.90065E-03
"C5-48-Rur",1948,"total_ingest_otherveg","thyroid", 1.54365E-03
"C5-48-Rur",1948,"total_ingest_fruit","thyroid", 9.10785E-03
"C5-48-Rur",1948,"total_ingest_grain","thyroid", 1.30116E-03
"C5-48-Rur",1948,"total_ingest_poultry","thyroid", 8.25756E-05
"C5-48-Rur",1948,"total_ingest_eggs","thyroid", 1.71009E-02
"C5-48-Rur",1948,"total_ingest_milk","thyroid", 1.00620E+00
```

Excerpts from a dose file from CiderF for a map case that uses 250 realizations is provided in Table 9.7. Multiple map cases can be output in the same file. The file is structured in comma separated variables format, so it is suitable for importing into a spreadsheet or database. The file contains a single header line. Each subsequent line identifies the case ID, the year for the annual dose, the node number, the dose pathway, and then the doses.

Table 9.7 Excerpts from a Dose File from CiderF for a Map Case

```
"CaseID","Date","Node","Pathway","Organ","Realizations"
"C5-48-Rur",1948,1,"total","thyroid", 2.89368E-03, 1.62902E-02, 2.03445E-03, ...
"C5-48-Rur",1948,2,"total","thyroid", 3.84594E-03, 2.19780E-02, 2.55246E-03, ...
"C5-48-Rur",1948,3,"total","thyroid", 4.47756E-03, 1.96501E-02, 3.44041E-03, ...
"C5-48-Rur",1948,4,"total","thyroid", 6.92033E-03, 1.68041E-02, 4.78864E-03, ...
"C5-48-Rur",1948,5,"total","thyroid", 9.46710E-03, 1.78436E-02, 6.03638E-03, ...
"C5-48-Rur",1948,6,"total","thyroid", 9.64654E-03, 2.05667E-02, 6.08737E-03, ...
"C5-48-Rur",1948,7,"total","thyroid", 1.19381E-02, 2.17170E-02, 4.96074E-03, ...
"C5-48-Rur",1948,8,"total","thyroid", 9.51707E-03, 1.75421E-02, 5.23027E-03, ...
"C5-48-Rur",1948,9,"total","thyroid", 9.61222E-03, 1.51009E-02, 5.47338E-03, ...
```

9.6 Keywords in the CiderF Case Keyword File

The keywords used in the CiderF code are identified in alphabetical order in Table 9.8, along with which of the two keyword files they are used in. The END, FILE and STOCHAST keywords appear in both of the input keyword files.

Table 9.8 Summary of Keywords Used in the CiderF Code

Keyword	Section	File	Purpose
AGEBREAT	9.7.1	Factors	The AGEBREAT keyword is used to define the age intervals for age dependent breathing rates.
AGEDIET	9.7.2	Factors	The AGEDIET keyword is used to define the age intervals for age dependent reference diets.
AGEFRACO	9.7.3	Factors	The AGEFRACO keyword is used to define the age intervals for age dependent fraction of time spent outdoors.
AGEINT	9.7.4	Factors	The AGEINT keyword is used to define the age intervals for age dependent internal dose conversion factors.
BIRTH	9.6.1	Case	The BIRTH keyword identifies the date of birth for the individual for whom doses are to be calculated.
CASE	9.6.2	Case	The CASE keyword is used to begin definition of an individual or map mode dose case.
CREAMERY	9.7.5	Factors	The CREAMERY keyword is used to identify the total number of creameries in data files produced by the DESCARTES code.
DEBUG	9.6.3	Case	The DEBUG keyword is used to activate voluminous additional intermediate outputs to the report file.
DIET	9.6.4	Case	The DIET keyword defines a time period when an individual consumes a special diet.
DIETMAP	9.7.6	Factors	The DIETMAP keyword uses characteristics of an individual to map into a reference diet.
DIETMOTH	9.6.5	Case	The DIETMOTH keyword defines a time period when the mother of a prenatal or suckling individual consumes a special diet.

Keyword	Section	File	Purpose
DIETREF	9.7.7	Factors	The DIETREF keyword is used to define consumption sources for all of the reference diets.
DIETSCAL	9.7.8	Factors	The DIETSCAL keyword defines optional scaling factors by year for reference diets.
DIETSPEC	9.6.6	Case	The DIETSPEC keyword is used to define special diets that modify or replace reference diets.
DOSEPATH	9.6.7	Case	The DOSEPATH keyword is used to control which dose calculations are written to output files.
END	9.6.8 9.7.9	Case Factors	The END keyword signifies the end of data in the file.
ENDCASE	9.6.9	Case	The ENDCASE keyword is used to terminate definition of an individual or map mode dose case.
FILE	9.6.10 9.7.10	Case Factors	The FILE keyword is used to enter file names for the code to use.
FINISH	9.6.11	Case	The FINISH keyword is used to define the last day of a time period for a dose calculation.
LIFESTYL	9.6.12	Case	The LIFESTYL keyword is used to lifestyle type for an individual in a dose calculation.
MOTHERBI	9.6.13	Case	The MOTHERBI keyword is used to define the day of birth for the mother of an individual in a dose calculation.
NODE	9.6.14	Case	The NODE keyword is used to identify the locations where the individual for whom doses are to be calculated lived by date.
NODENUM	9.7.11	Factors	The NODENUM keyword is used to identify the total number of nodes in data files produced by the DESCARTES code.
NUCLIDE	9.7.12	Factors	The NUCLIDE keyword is used to identify the nuclide and associated radioactive half-life.
ORGANS	9.6.15	Case	The ORGANS keyword is used to activate calculations of doses to the thyroid, whole body, or both.
OUTPUT	9.6.16	Case	The OUTPUT keyword is used to activate optional output of values and information on generated stochastic variables.
PREGNURS	9.6.17	Case	The PREGNURS keyword is used to define the time period where a mother is pregnant or nursing in a dose calculation.
REALIZAT	9.6.18	Case	The REALIZAT keyword is used to define the number of realizations of data in the input media files from DESCARTES.
SEASON	9.7.13	Factors	The SEASON keyword is used to define the days in a year that are associated with each of the four seasons.

Keyword	Section	File	Purpose
SEED	9.6.19	Case	The SEED keyword is used to set the random seed for stochastic variable generation.
SEX	9.6.20	Case	The SEX keyword is used to define the sex (gender) of an individual in a dose calculation.
START	9.6.21	Case	The START keyword is used to define the first day of a time period for a dose calculation.
STEP	9.7.14	Factors	The STEP keyword is used to identify the averaging interval used in data files produced by the DESCARTES code.
STOCHAST	9.6.22 9.7.15	Case Factors	The STOCHAST keyword is used to enter data for variables that can be defined by statistical distributions.
TIME	9.7.16	Factors	The TIME keyword is used to start and stop dates used in data files produced by the DESCARTES code.
USER	9.6.23	Case	The USER keyword is used to enter the name of the user running the code.
VERBOSE	9.6.24	Case	The VERBOSE keyword is used to activate additional outputs to the standard output and the report file.
WASHED	9.6.25	Case	The WASHED keyword is used to activate the option where the individual washes fruit and leafy vegetables prior to consumption.
WEANED	9.6.26	Case	The WEANED keyword identifies the date of weaning for a suckling individual for whom doses are to be calculated.

The keywords used in a CiderF case keyword file are described in the following sections.

9.6.1 BIRTH Keyword

The BIRTH keyword is used to enter the date of birth for an individual. The BIRTH keyword is only used in the context of a dose case (see Section 9.6.2) and every case must have one BIRTH keyword. The birth date is used to correctly assign age specific data. The following is this keyword's syntax:

```
BIRTH "quote"
```

The birth date is entered in a quote string. The format of the quote string is "YYYY-MM-DD" where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example date string for a person born on June 7, 1955, is the following: "1955-06-07" and the associated keyword is the following:

```
BIRTH “1955-06-07”
```

9.6.2 CASE Keyword

The CASE keyword is used to start the definition of a dose case. The keywords used within a dose case in a CiderF case keyword file are identified in Table 9.9. The first keyword in a case definition is the

CASE keyword and the last keyword in a case definition is an ENDCASE keyword. The other keywords identified in Table 9.9 have no meaning outside a pair of CASE and ENDCASE keywords.

Table 9.9 Summary of Keywords Associated With a CASE Definition in CiderF

Keyword	Purpose
CASE	The CASE keyword begins the definition of a specific dose case.
BIRTH	The BIRTH keyword identifies the date of birth for the individual for whom doses are to be calculated. This information is used to correctly index into data that are age specific.
DIET	The optional DIET keyword is used to apply specialized diets instead of reference diets for a range of dates for the individual for whom doses are to be calculated.
DIETMOTH	The optional DIETMOTH keyword is used to apply specialized diets instead of reference diets for a range of dates for the mother of the prenatal or suckling individual for whom doses are to be calculated.
FINISH	The FINISH keyword identifies the last date for calculation of doses for this specific case.
LIFESTYL	The LIFESTYL keyword is used to identify the lifestyle (rural or urban) of the individual for whom doses are to be calculated by date.
MOTHERBI	The MOTHERBI keyword identifies the date of birth for the mother of a prenatal or suckling individual for whom doses are to be calculated. This information is used to correctly index into data that are age specific.
NODE	The NODE keyword is used to identify the location or locations where the individual for whom doses are to be calculated lived by date.
PREGNURS	The optional PREGNURS keyword identifies the range of dates where an individual for whom doses are to be calculated was pregnant or nursing.
SEX	The sex keyword is used to identify the sex (gender) of the individual for whom doses are to be calculated.
START	The START keyword identifies the first date for calculation of doses for this specific case.
WASHED	The optional WASHED keyword activates the option that washing the outer compartment of two compartment food crops (leafy vegetables and fruit) removes some of the activity on the outer compartment.
WEANED	The WEANED keyword identifies the date of weaning for a suckling individual for whom doses are to be calculated.
ENDCASE	The ENDCASE keyword terminates the definition of a specific dose case.

The following is the CASE keyword syntax:

```
CASE ID="quote" [MAP | INDIVIDU]
```

The quote string associated with the ID modifier contains a case identification that is up to 10 characters in length. Generally, the case identification is used for labeling purposes. The case identification is not required to be unique when multiple cases are identified in a single keyword file. However, output data are labeled with this identification string, thus the use of unique values is recommended. If the CiderView program is used, then the case ID is as part of the file name for files output by CiderView. Thus, the user should select case ID's that are valid entries in file names.

The CiderF code can operate in two different modes: individual mode and map mode. In the individual mode, the individual can move between locations in a given year, and is even allowed to exit the domain for a period of time. In map mode, the same definition for an individual is applied at all nodes and doses are calculated at all nodes. Although CiderF can operate in both individual and map modes, only one mode is allowed for any single run of the code. The following example keyword identifies a CASE for the individual dose mode.

```
CASE INDIVIDUAL ID="C5-50-Rur"
```

The following example keyword identifies a CASE using map mode:

```
CASE MAP ID="F1-53-Urb"
```

The following set of keywords defines a case using all of the keywords listed in Table 9.9 that are pertinent to a prenatal and suckling child in an individual mode case that covers two years of exposure.

```
CASE INDIVIDUAL ID="C5-50-Rur"
START    "1950-01-01"
FINISH   "1951-12-31"
SEX      "male"
BIRTH    "1950-09-15"
MOTHERBI "1929-10-25"
WEANED   "1952-03-15" ! Weaned at 18 months
WASHED
LIFESTYL TYPE="urban"    BEGIN="1948-01-01" FINISH="1972-12-31"
DIET     ID="SpecDiet01" BEGIN="1952-03-16" FINISH="1972-12-31"
DIETMOTH ID="SpecDiet21" BEGIN="1951-01-01" FINISH="1972-12-31"
NODE     NUMBER=391      BEGIN="1948-06-01" FINISH="1972-12-31"
ENDCASE
```

9.6.3 DEBUG Keyword

The optional DEBUG keyword is used to activate writing intermediate calculations to the report file. It should be used sparingly and with only one or two realizations. Otherwise, the volume of output could be immense. The following is this keyword's syntax:

```
DEBUG [modifier 1] (modifier 2) ... (modifier k)
```

Multiple DEBUG keywords can be entered with combinations of modifiers, or a single keyword can be entered containing all of the modifiers. The modifiers can be entered in any order. Table 9.10 describes the modifiers associated with the DEBUG keyword.

Table 9.10 Modifiers Associated with the DEBUG Keyword for CiderF

Modifier	Description
BEEF	Intermediate outputs on calculations for the beef food type.
CASE	Intermediate outputs on calculations for each case.
CREAM	Intermediate outputs on calculations for the creamery milk food type.
DIET	Intermediate outputs on calculations for reference and special diets.
EGGS	Intermediate outputs on calculations for the eggs food type.
EXTERNAL	Intermediate outputs on calculations for external doses.
FRUIT	Intermediate outputs on calculations for the fruit food type.
GRAIN	Intermediate outputs on calculations for the grain food type.
INHALATI	Intermediate outputs on calculations for dose from inhalation.
LVEG	Intermediate outputs on calculations for the leafy vegetables food type.
MEDIA	Intermediate outputs on data from media files written by DESCARTES.
MILK	Intermediate outputs on calculations for the milk food type.
OVEG	Intermediate outputs on calculations for the other vegetables food type.
POULTRY	Intermediate outputs on calculations for the poultry food type.

The following entries provide examples of the use of the DEBUG keyword:

```
DEBUG BEEF
DEBUG DIET
DEBUG FRUIT GRAIN OVEG CASE
```

There are no quote strings associated with the DEBUG keyword.

9.6.4 DIET Keyword

The optional DIET keyword is used to signify that an individual eats a specialized diet for a period of time. The individual eats a reference diet for every date where no specialized diets are identified. The DIET keyword is only used in the context of a dose case (see Section 9.6.2). The following is this keyword’s syntax:

```
DIET ID="quote1" BEGIN="quote2" FINISH="quote3"
```

Explanation of the modifiers and associated data for this keyword is provided in Table 9.11.

Table 9.11 Modifiers Associated with the DIET Keyword for CiderF

Modifier	Description
ID	The quote string associated with the ID modifier identifies a specialized diet. See the DIETSPEC keyword in Section 9.6.6 for an explanation of how to define a specialized diet.
BEGIN	The quote string associated with the BEGIN modifier identifies the first day the individual eats this specialized diet. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10.
FINISH	The quote string associated with the FINISH modifier identifies the last day the individual eats this specialized diet. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10.

The dates on the DIET keywords identify the date range that the specialized diet is eaten. Multiple DIET keywords can be used with the same or different diets. Dates before the start date and after the end date of the case are allowed. If date ranges overlap on multiple DIET keywords, then the data for the first keyword is used. Data for specialized diets should be entered without overlapping dates. Two example DIET keywords are the following:

```
DIET BEGIN="1948-06-01" FINISH="1948-09-01" ID="SpecDiet01"
DIET BEGIN="1948-09-02" FINISH="1949-12-31" ID="SpecDiet02"
```

9.6.5 DIETMOTH Keyword

The optional DIETMOTH keyword is used to signify that the mother of a prenatal or suckling individual eats a specialized diet for a period of time. It has the same structure as the DIET keyword. The mother eats reference diets if no specialized diets are identified. The DIETMOTH keyword is only used in the context of a dose case (see Section 9.6.2). The following is this keyword’s syntax:

```
DIETMOTH ID="quote1" BEGIN="quote2" FINISH="quote3"
```

Explanation of the modifiers and associated data for this keyword is provided in Table 9.12.

Table 9.12 Modifiers Associated with the DIETMOTH Keyword for CiderF

Modifier	Description
ID	The quote string associated with the ID modifier identifies a specialized diet. See the DIETSPEC keyword in Section 9.6.6 for an explanation of how to define a specialized diet.
BEGIN	The quote string associated with the BEGIN modifier identifies the first day the mother eats this specialized diet. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10.

Modifier	Description
FINISH	The quote string associated with the FINISH modifier identifies the last day the mother eats this specialized diet. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10.

The dates on the DIETMOTH keywords identify the date range that the specialized diet is eaten. Multiple DIETMOTH keywords can be used with the same or different diets. Dates before the start date and after the end date of the case are allowed. If date ranges overlap on multiple DIETMOTH keywords, then the data for the first keyword is used. Data for specialized diets should be entered without overlapping dates. Two example DIETMOTH keywords are the following:

```
DIETMOTH BEGIN="1948-06-01" FINISH="1948-09-01" ID="SpecDiet27"
DIETMOTH BEGIN="1948-09-02" FINISH="1949-08-15" ID="SpecDiet03"
```

9.6.6 DIETSPEC Keyword

The optional DIETSPEC keyword is used to define a specialized diet. Specialized diets can modify or totally replace reference diets (see Sections 9.7.7 and 9.7.15.12 for instructions on defining reference diets). The same specialized diet can be used in multiple dose cases. The following is this keyword’s syntax:

```
DIETSPEC ID=“quote1” TITLE=“quote2” (MILK_1I=N1) (MILK_1H=N2) (MILK_2I=N3)
    (MILK_2H=N4) (MILK_3I=N5) (MILK_3H=N6) (MILK_4I=N7) (MILK_4H=N8) (MILK_GRO=N9)
    (MILK_GOA=N10) (MILK_CRE=N11) (CREAM_NU=N12) (VEGLOC=N13)
```

Specialized diets can modify reference diets in two ways. First, the specialized diet can assign the milk and leafy vegetable consumption fractions to different sources than are used for a reference diet. The description of modifiers and associated data provided in Table 9.13 perform this function. The intake fractions identified in the table apply only over the time period that the specialized diet is applied. In addition, a specialized diet can use different intake rates than defined in a reference diet for one or more foods. If an intake rate is not specified in a special diet, then the intake rate for the reference diet is used. Any new intake rates for specialized diets are entered using the STOCHAST keyword (see Section 9.6.22). Some examples of new intake rates are provided in this section.

Table 9.13 Modifiers Associated with the DIETSPEC Keyword for CiderF

Modifier	Description
ID	The quote string associated with the ID modifier identifies a specialized diet. Up to 10 characters can be used in the quote string. The ID’s for multiple specialized diets must be unique.
TITLE	The quote string associated with the TITLE modifier identifies a descriptive title used for labeling purposes. Up to 72 characters can be used in the title.

Modifier	Description
MILK_1I	The numerical value associated with the MILK_1I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 1. The sum of milk intake fractions over all milk sources must equal 1.
MILK_1H	The numerical value associated with the MILK_1H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 1. The sum of milk intake fractions over all milk sources must equal 1.
MILK_2I	The numerical value associated with the MILK_2I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 2. The sum of milk intake fractions over all milk sources must equal 1.
MILK_2H	The numerical value associated with the MILK_2H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 2. The sum of milk intake fractions over all milk sources must equal 1.
MILK_3I	The numerical value associated with the MILK_3I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 3. The sum of milk intake fractions over all milk sources must equal 1.
MILK_3H	The numerical value associated with the MILK_3H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 3. The sum of milk intake fractions over all milk sources must equal 1.
MILK_4I	The numerical value associated with the MILK_4I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 4. The sum of milk intake fractions over all milk sources must equal 1.
MILK_4H	The numerical value associated with the MILK_4H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 4. The sum of milk intake fractions over all milk sources must equal 1.
MILK_GRO	The numerical value associated with the MILK_GRO modifier identifies the fraction of milk intake that comes from grocery stores. The sum of milk intake fractions over all milk sources must equal 1.
MILK_GOA	The numerical value associated with the MILK_GOA modifier identifies the fraction of milk intake that comes from a local goat. The sum of milk intake fractions over all milk sources must equal 1.
MILK_CRE	The numerical value associated with the MILK_CRE modifier identifies the fraction of milk intake that comes from a specific creamery. If this modifier is used, then the creamery number must be provided with the CREAM_NU modifier. Only one creamery can be identified in a single specialized diet. The sum of milk intake fractions over all milk sources must equal 1.
CREAM_NU	The numerical value associated with the CREAM_NU identifies the creamery number for intake of creamery milk identified with the MILK_CRE modifier. If this modifier is not used if the MILK_CRE modifier is not used.

Modifier	Description
VEGLOC	The numerical value associated with the VEGLOC modifier identifies the fraction of leafy vegetable consumption that is locally produced (a garden, for example). A value between 0 and 1 is required. Any vegetables consumed that are not locally produced are assumed to be commercially produced. A value of 0.2 for this entry means that 20% of vegetables consumed are locally produced and 80% are commercially produced.

The following example keyword identifies a specialized diet where all milk comes from a grocery store and only locally produced vegetables are consumed. No changes are made to the reference intake amounts.

```
DIETSPEC ID="SpecDiet01" milk_gro=1 vegloc=1 TITLE="First special diet"
```

The following example keyword identifies a specialized diet where half of milk consumption comes from a grocery store and half of milk consumption comes from a specific creamery (creamery number 2). In addition, 25% of leafy vegetables consumed come from local sources (such as a garden). A number of associated STOCHAST keywords (see Section 9.6.22) are provided to show how the specialized diet can replace the reference diet consumption amounts. The reference diet consumption amount is used if no special diet entry is made for that food type. The tokens used to identify each food type are discussed in Table 9.1.

```
DIETSPEC ID="SpecDiet02" milk_gro=0.5 milk_cre=0.5 cream_nu=2 vegloc=0.25
   TITLE="Second special diet"

STOCHAST ID="SpecDiet02f_milk " DIST="uniform" PARAM 0.0000  0.1005 UNITS="L/day"
   TITLE="SpecDiet02 f_milk  consumption"

STOCHAST ID="SpecDiet02s_milk " DIST="uniform" PARAM 0.0133  0.4641 UNITS="L/day"
   TITLE="SpecDiet02 s_milk  consumption"

STOCHAST ID="SpecDiet02l_veg  " DIST="uniform" PARAM 0.0000  0.0121 UNITS="kg/day"
   TITLE="SpecDiet02 l_veg   consumption"

STOCHAST ID="SpecDiet02o_veg  " DIST="uniform" PARAM 0.0000  0.0244 UNITS="kg/day"
   TITLE="SpecDiet02 o_veg   consumption"

STOCHAST ID="SpecDiet02fruit  " DIST="uniform" PARAM 0.0101  0.1035 UNITS="kg/day"
   TITLE="SpecDiet02 fruit   consumption"

STOCHAST ID="SpecDiet02grain  " DIST="uniform" PARAM 0.0014  0.0099 UNITS="kg/day"
   TITLE="SpecDiet02 grain   consumption"

STOCHAST ID="SpecDiet02eggs   " DIST="uniform" PARAM 0.0000  0.0049 UNITS="kg/day"
   TITLE="SpecDiet02 eggs    consumption"

STOCHAST ID="SpecDiet02beef   " DIST="uniform" PARAM 0.0000  0.0214 UNITS="kg/day"
   TITLE="SpecDiet02 beef    consumption"

STOCHAST ID="SpecDiet02poultry" DIST="uniform" PARAM 0.0000  0.0010 UNITS="kg/day"
   TITLE="SpecDiet02 poultry consumption"
```

9.6.7 DOSEPATH Keyword

The DOSEPATH keyword is used to activate output of calculated dose values to the results file. The following is this keyword's syntax:

```
DOSEPATH [modifier 1] (modifier 2) ... (modifier 14)
```

Multiple DOSEPATH keywords can be entered with combinations of modifiers, or a single keyword can be entered containing all of the modifiers. The modifiers can be entered in any order. Table 9.14 describes the modifiers associated with the DOSEPATH keyword. There are no quote strings associated with the DOSEPATH keyword. The selections on this keyword apply to all organs specified by the ORGANS keyword (see Section 9.6.15).

Table 9.14 Modifiers Associated with the DOSEPATH Keyword for CiderF

Modifier	Description
EXTERNAL	Output doses associated with the external exposures.
INHALATI	Output doses associated with the inhalation.
INGESTIO	Output doses summed over all ingestion pathways.
BEEF	Output doses associated ingestion of beef.
LEAFYVEG	Output doses associated ingestion of leafy vegetables.
OTHERVEG	Output doses associated ingestion of other vegetables.
FRUIT	Output doses associated ingestion of fruit.
GRAIN	Output doses associated ingestion of grain.
POULTRY	Output doses associated ingestion of poultry.
EGGS	Output doses associated ingestion of eggs.
MILK	Output doses associated ingestion of milk.
DAILY	Output doses on an annual basis. This option should be used sparingly because a large amount of output can occur. Daily outputs are not available for map cases.
ANNUAL	Output doses on an annual basis.
TOTAL	Output doses totaled over the entire time period and all pathways.

The DAILY, ANNUAL and TOTAL modifiers identify the time periods when the dose values will be output. The other modifiers select the dose pathway. Therefore, one of the DAILY, ANNUAL or TOTAL modifiers must be selected for each run of the code. In addition, one or more of the exposure pathways must be identified. Daily outputs are not available for map cases due to the potentially large volume of outputs associated with map cases.

The following DOSEPATH keyword will output doses from beef ingestion on an annual basis for all years defined in the specific case:

```
DOSEPATH BEEF ANNUAL
```

The following DOSEPATH keyword will output doses from external exposure, inhalation and all ingestion totaled over all days in the specific case:

```
DOSEPATH EXTERNAL INHALATI INGESTION TOTAL
```

9.6.8 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword's syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

9.6.9 ENDCASE Keyword

The ENDCASE keyword signifies the end of the definition for a dose case. The following is this keyword's syntax:

```
ENDCASE
```

There are no modifiers or quote strings associated with the ENDCASE keyword. Every CASE keyword must be paired with a subsequent ENDCASE keyword.

9.6.10 FILE Keyword

The FILE keyword is used to enter the names of input and output files. The following is this keyword's syntax:

```
FILE modifier "quote"
```

The file names are entered in quote strings. Path names up to 256 characters long (name length limitation in Windows 7) are supported. At least one FILE keyword is required for every run of the code. The modifiers associated with the FILE keyword are described in Table 9.15.

Table 9.15 Modifiers Associated with the FILE Keyword for CiderF Case Keywords

Modifier	Description
FACTORS	The quote string associated with the FACTORS modifier contains the name of the factors keyword file. An example file is provided in Table 9.4.
REPORT	The quote string associated with the REPORT modifier contains the name of the output file containing information about the progress of the run. All error messages are directed to this file. An example file is provided in Table 9.5.
RESULT	The quote string associated with the RESULT modifier contains the name of the output file of dose results. Example files are provided in Table 9.6 and Table 9.7.

Modifier	Description
STOCSTAT	The quote string associated with the STOCSTAT modifier identifies the name of an output file that will contain statistical distribution definitions and summary statistics for all stochastic variables generated in CiderF. This file is only required if the optional STOCSTAT modifier is entered on the OUTPUT keyword (see Section 9.6.16). The information in this optional file is useful for diagnostic purposes.
STOCVALU	The quote string associated with the STOCVALU modifier identifies the name of an output file that will contain values of all stochastic variables generated in CiderF. This file is only required if the optional STOCVALU modifier is entered on the OUTPUT keyword (see Section 9.6.16). The information in this optional file is useful for conducting sensitivity analyses of inputs against outputs.

The following FILE keyword entries define all of the data files identified in Table 9.15.

```
FILE REPORT "Child5.rpt"
FILE RESULT "Child5_Result.csv"
FILE FACTORS "D:\Test\RefDiet\TestFactors.kwd"
FILE STOCVALU "Child5_Valu.txt"
FILE STOCSTAT "Child5_Stat.txt"
```

9.6.11 FINISH Keyword

The FINISH keyword is used to enter the ending (finish) date for a dose case. The FINISH keyword is only used in the context of a dose case (see Section 9.6.2) and every case must have one FINISH keyword. The following is this keyword's syntax:

```
FINISH “quote”
```

The FINISH date is entered in a quote string. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example FINISH date keyword is the following:

```
FINISH “1956-12-31”
```

The finish date must be equal to or after the start date and it must fall in the range of dates in the concentration data files produced by DESCARTES. If an invalid finish date is entered, the run will terminate after writing an error message to the report file. An example error message is the following:

```
Error number 5 encountered in routine CaseCheck at 05/22/2013 on 14:05:29.909
Message: Case finish date is not in the range of DESCARTES run dates
         Case finish date: 1928-12-31
         DESCARTES Date Range: 1948-06-01 to 1972-12-31

Error number 6 encountered in routine CaseCheck at 05/22/2013 on 14:05:29.909
Message: Case finish date is not after the case start date
         Case start date:  1948-06-01
         Case finish date: 1928-12-31
```

9.6.12 LIFESTYL Keyword

The LIFESTYL keyword is used to define the lifestyle for the individual for a period of time. The LIFESTYL keyword is only used in the context of a dose case (see Section 9.6.2). The following is this keyword's syntax:

```
LIFESTYL TYPE="quote1" BEGIN="quote2" FINISH="quote3"
```

A separate LIFESTYL keyword is required for change of lifestyle. Explanation of the modifiers and associated data for this keyword is provided in Table 9.17.

Table 9.16 Modifiers Associated with the LIFESTYL Keyword for CiderF

Modifier	Description
TYPE	The quote string associated with the TYPE modifier provides the lifestyle type. Valid entries are “rural” or “urban” (see Table 9.1). This lifestyle type is used in defining reference diets and the amount of time spent outdoors.
BEGIN	The quote string associated with the BEGIN modifier identifies the first day the individual had this lifestyle type. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example for June 7, 1955, is the following: “1955-06-07”.
FINISH	The quote string associated with the FINISH modifier identifies the last day the individual had this lifestyle type. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example for December 31, 1972, is the following: “1972-12-31”.

The dates on the LIFESTYL keywords must cover every day between the start date (see Section 9.6.21) and the finish date (see Section 9.6.11) of the case. Dates before the start date and after the end date of the case are allowed. An example keyword for an individual who lived in a city for 40 years is the following:

```
LIFESTYL TYPE=“urban” BEGIN=“1935-01-01” FINISH=“1974-12-31”
```

The following sequence of LIFESTYL keywords would indicate the following lifestyle information for an individual:

- lived on a farm before July 22, 1950
- lived in a city from July 22, 1950 through September 4, 1952
- lived on a farm from September 5, 1952, through January 2, 1954
- lived in a city from January 3, 1954 through December 31, 1972

```
LIFESTYL TYPE=“rural” BEGIN=“1900-01-01”  FINISH=“1950-07-21”
LIFESTYL TYPE=“urban” BEGIN=“1950-07-22”  FINISH=“1952-09-04”
LIFESTYL TYPE=“rural” BEGIN=“1952-09-05”  FINISH=“1954-01-02”
LIFESTYL TYPE=“urban” BEGIN=“1954-01-03”  FINISH=“1972-12-31”
```

9.6.13 MOTHERBI Keyword

The MOTHERBI keyword is used to enter the date of birth for the mother of an individual for whom doses are being calculated. The MOTHERBI keyword is only used in the context of a dose case (see Section 9.6.2) where the individual is prenatal or suckling during the time period defined for that specific case. The birth date is used to correctly assign data that are age specific to the mother. The following is this keyword's syntax:

```
MOTHERBI "quote"
```

The birth date is entered in a quote string. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example keyword for a mother born on January 16, 1928, is the following:

```
MOTHERBI “1928-01-16”
```

9.6.14 NODE Keyword

The NODE keyword is used to define the residence location for the individual for a period of time. The NODE keyword is only used in the context of a dose case (see Section 9.6.2). The following is this keyword's syntax:

```
NODE NUMBER=N1 BEGIN="quote1" FINISH="quote2"
```

A separate NODE keyword is required for every place the individual lives for an individual mode case. This keyword is not needed for map mode cases. Explanation of the modifiers and associated data for this keyword is provided in Table 9.17.

Table 9.17 Modifiers Associated with the NODE Keyword for CiderF

Modifier	Description
NUMBER	The numerical value associated with the NUMBER modifier provides the node number where the individual lives. If the individual is not living in the dose domain, then a node number of 0 must be used. Otherwise, the node number is used to identify the concentration data for the location nearest to where the individual resides. A node number must be in the range of 0 to the number of nodes in the domain (see Sections 6.3.7 and 9.7.11).
BEGIN	The quote string associated with the BEGIN modifier identifies the first day the individual lived at this node. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zero's must be used for month and day entries less than 10. An example for June 7, 1955, is the following: “1955-06-07”.

Modifier	Description
FINISH	The quote string associated with the FINISH modifier identifies the last day the individual lived at this node. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zero’s must be used for month and day entries less than 10. An example for December 31, 1972, is the following: “1972-12-31”.

The dates on the NODE keywords must cover every day between the start date (see Section 9.6.21) and the finish date (see Section 9.6.11) of the case. Dates before the start date and after the end date of the specific dose case are allowed. An example keyword for an individual residing at the same location (node 416) for 40 years is the following:

```
NODE NUMBER=416 BEGIN=“1935-01-01” FINISH=“1975-01-01”
```

The following sequence of NODE keywords would indicate the following residence information for an individual:

- lived outside the dose domain before July 22, 1950
- lived at node 114 from July 22, 1950 through September 4, 1952
- lived outside the dose domain from September 5, 1952, through January 2, 1954
- lived at node 223 from January 3, 1954 through December 31, 1972

```
NODE NUMBER=  0 BEGIN=“1900-01-01” FINISH=“1950-07-21”
NODE NUMBER=114 BEGIN=“1950-07-22” FINISH=“1952-09-04”
NODE NUMBER=  0 BEGIN=“1952-09-05” FINISH=“1954-01-02”
NODE NUMBER=223 BEGIN=“1954-01-03” FINISH=“1972-12-31”
```

9.6.15 ORGANS Keyword

The ORGANS keyword is used to define the categories of doses to calculate. The following is this keyword’s syntax:

```
ORGANS (THYROID) (WHOLEBOD)
```

The dose specific to the thyroid as an organ will be calculated if the THYROID modifier is entered. The whole body effective dose will be calculated if the WHOLEBOD modifier is entered. At least one of THYROID and WHOLEBOD is required. Both dose types can be calculated and output in the same run of the CiderF code. An example keyword for calculating just the thyroid dose is the following:

```
ORGANS THYROID
```

An example keyword for calculating both the thyroid dose and the whole body dose is the following:

```
ORGANS THYROID WHOLEBOD
```

9.6.16 OUTPUT Keyword

The OUTPUT keyword is used to activate optional outputs associated with the definition and generation of stochastic variables. The following is this keyword’s syntax:

```
OUTPUT (STOCDEFS) (STOCSTAT) (STOCVALU) (ROW|COLUMN)
```

Explanation of the modifiers for this keyword is provided in Table 9.18. All of the modifiers can be entered on one OUPUT keyword or multiple OUPUT keywords can be used.

Table 9.18 Modifiers Associated with the OUTPUT Keyword for CiderF

Modifier	Description
STOCDEFS	Presence of the STOCDEFS modifier will cause the definitions of all stochastic variables to be written to a text output file. The name of the output file is defined using the STOCSTAT modifier on the FILE keyword (see Table 9.15). The outputs initiated by the STOCDEFS and STOCSTATS modifiers go to the same output file.
STOCSTAT	Presence of the STOCSTAT modifier will cause the sumary statistics for all stochastic variables to be written to a text output file. The name of the output file is defined using the STOCSTAT modifier on the FILE keyword (see Table 9.15). The outputs initiated by the STOCDEFS and STOCSTATS modifiers go to the same output file.
STOCVALU	Presence of the STOCVALU modifier will cause the generated values of all stochastic variables to be written to a text output file. The name of the output file is defined using the STOCVALU modifier on the FILE keyword (see Table 9.15).
COLUMN	The COLUMN modifier affects the output of the generated values written when the STOCVALU modifier is present. If the COLUMN modifier is present, then variables will be written out in columns where the variable name is the column header. Only one of the COLUMN or ROW modifiers can be entered. The default when neither COLUMN nor ROW is entered is the same as if ROW were entered.
ROW	The ROW modifier affects the output of the generated values written when the STOCVALU modifier is present. If the ROW modifier is present, then variables will be written out in rows where the variable name is the first value on the row. Only one of the COLUMN or ROW modifiers can be entered. The default when neither COLUMN nor ROW is entered is the same as if ROW were entered.

An example keyword that outputs only the generated values, formatted in columns by variable, is the following:

```
OUTPUT STOCVALUE COLUMN
```

An example keyword that outputs the variable definitions and summary statistics for each variable is the following:

```
OUTPUT STOCDEFS STOCSTAT
```

9.6.17 PREGNURS Keyword

The optional PREGNURS keyword is used to identify the time interval where a woman whose dose is being computed is either pregnant or nursing a child. Thus, this keyword is never used for a case where doses are being computed for a male. It is only used when computing the dose to a mother. The PREGNURS keyword is only used in the context of a dose case (see Section 9.6.2). The following is this keyword's syntax:

```
PREGNURS BEGIN="quote2" FINISH="quote3"
```

Explanation of the modifiers and associated data for this keyword is provided in Table 9.19.

Table 9.19 Modifiers Associated with the PREGNURS Keyword for CiderF

Modifier	Description
BEGIN	The quote string associated with the BEGIN modifier identifies the day the individual becomes pregnant. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10.
FINISH	The quote string associated with the FINISH modifier identifies the day the individual weans a suckling child. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10.

A different PREGNURS keyword must be used for every time the woman is pregnant. The following example illustrates the use of this keyword for a woman who did not nurse her child after it was born:

```
PREGNURS BEGIN="1948-01-01" FINISH="1948-09-22"
```

The following example illustrates the use of this keyword for a woman who had two children and who nursed each child for approximately one year after it was born:

```
PREGNURS BEGIN="1948-01-01" FINISH="1949-09-22"
PREGNURS BEGIN="1951-03-01" FINISH="1952-01-27"
```

9.6.18 REALIZAT Keyword

The REALIZAT keyword defines the number of realizations to generate. The following is this keyword’s syntax:

```
REALIZAT N1
```

The integer N1 has a minimum value of 1 and a maximum of the number of realizations used in the DESCARTES code to define the set of environmental concentration data. If the number of realizations is less than the number of realizations in the concentration data files, then only the first realizations will be used. The following keyword sets the number of realizations to 10:

```
REALIZAT 10
```

9.6.19 SEED Keyword

The SEED keyword sets the value for the seed for the random number generator. The following is this keyword’s syntax:

```
SEED Value1
```

The value for Value1 must be a number in the range 1.0 to 2147483646.0, in whole number increments. The following are example uses of this keyword:

```
SEED 5
SEED 101.0
SEED 2147483645.0D0
```

There are no quote strings or modifiers associated with the SEED keyword.

9.6.20 SEX Keyword

The SEX keyword is used to enter the sex (gender) of the individual for a dose case. The SEX keyword is only used in the context of a dose case (see Section 9.6.2) and every case must have one SEX keyword. The following is this keyword's syntax:

```
SEX "quote"
```

The sex of the individual is entered in a quote string. The only valid entries (see Table 9.1) are "male" and "female". Any other entry will result in a run time error. An example keyword for a male is the following:

```
SEX "male"
```

9.6.21 START Keyword

The START keyword is used to enter the start date for a dose case. The START keyword is only used in the context of a dose case (see Section 9.6.2) and every case must have one START keyword. The following is this keyword's syntax:

```
START "quote"
```

The START date is entered in a quote string. The format of the quote string is "YYYY-MM-DD" where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example start date keyword is the following:

```
START “1955-06-07”
```

The start date must fall in the range of dates in the concentration data files produced by DESCARTES. If a start date is too early, the run will terminate after writing an error message to the report file. An example error message is the following:

```
Error number 4 encountered in routine CaseCheck at 05/22/2013 on 14:00:32.702
Message: Case start date is not in the range of DESCARTES run dates
         Case start date: 1928-06-01
         DESCARTES Date Range: 1948-06-01 to 1972-12-31
```

9.6.22 STOCHAST Keyword

Stochastic input variables are defined by the STOCHAST keyword. The following is this keyword's syntax:

```
STOCHAST [ID=“Quote1”] [TITLE=“Quote2”] [UNITS=“quote3”] [DIST=“Quote4”]
   PARAM N1,…Nn (TRUNCATE U1 U2)
```

The function of the modifiers on the STOCHAST is explained in detail in Section 12.1 and is not repeated here. This description focuses on the how to define different values for the quote string associated with the ID modifier so the code can internally access the correct data.

The only information in the case keyword file for CiderF that is entered using STOCHAST keywords is data defining intake rates for special diets (see the description of the DIETSPEC keyword in Section 9.6.6). The tokens defining the intake rates for the nine possible food types for diets are provided in Table 9.1.

The quote string associated with the ID modifier on a STOCHAST keyword is exactly 17 characters long for specialized diet information. The first 10 characters must contain the ID of the specialized diet defined using a SPECDIET keyword. Trailing spaces must be entered if necessary so the length is exactly 10 characters. The next 7 characters is one of the food type tokens from Table 9.1. The food type tokens are case sensitive. Trailing spaces after the food type token can be entered, but is not required.

The following nine keywords define the consumption rates for all food types for the specialized diet identified as “SpecDiet02”. Each food type is entered using a separate STOCHAST keyword. The quote strings associated with the ID modifier are highlighted in blue for presentation purposes.

```
STOCHAST ID="SpecDiet02f_milk " DIST="uniform" PARAM 0.0000  0.1005 UNITS="L/day"
   TITLE="SpecDiet02 fresh milk consumption"

STOCHAST ID="SpecDiet02s_milk " DIST="uniform" PARAM 0.0133  0.4641 UNITS="L/day"
   TITLE="SpecDiet02 stored milk consumption"

STOCHAST ID="SpecDiet02l_veg  " DIST="uniform" PARAM 0.0000  0.0121 UNITS="kg/day"
   TITLE="SpecDiet02 leafy vegetable consumption"

STOCHAST ID="SpecDiet02o_veg  " DIST="uniform" PARAM 0.0000  0.0244 UNITS="kg/day"
   TITLE="SpecDiet02 other vegetable consumption"

STOCHAST ID="SpecDiet02fruit  " DIST="uniform" PARAM 0.0101  0.1035 UNITS="kg/day"
   TITLE="SpecDiet02 fruit consumption"

STOCHAST ID="SpecDiet02grain  " DIST="uniform" PARAM 0.0014  0.0099 UNITS="kg/day"
   TITLE="SpecDiet02 grain consumption"

STOCHAST ID="SpecDiet02eggs   " DIST="uniform" PARAM 0.0000  0.0049 UNITS="kg/day"
   TITLE="SpecDiet02 eggs consumption"

STOCHAST ID="SpecDiet02beef   " DIST="uniform" PARAM 0.0000  0.0214 UNITS="kg/day"
   TITLE="SpecDiet02 beef consumption"

STOCHAST ID="SpecDiet02poultry" DIST="uniform" PARAM 0.0000  0.0010 UNITS="kg/day"
   TITLE="SpecDiet02 poultry consumption"
```

9.6.23 USER Keyword

The USER keyword is used to identify the user of the program. The user name will be written to output files. The program will error terminate if the user name is not supplied. The following is this keyword’s syntax:

```
USER "quote"
```

The user name is entered in a quote string. User names up to 16 characters long are supported. The following example keyword defines John Q. Public as the user running the code:

```
USER "John Q. Public"
```

There are no modifiers associated with the USER keyword.

9.6.24 VERBOSE Keyword

The optional VERBOSE keyword can be used to output run progress information. The following is this keyword's syntax:

```
VERBOSE (MEDIA)
```

Presence of the VERBOSE keyword causes more run time information to be written to the standard output device than if the keyword is omitted. The additional information can sometimes be useful in estimating the completion time of a lengthy set of dose calculations. If the optional MEDIA modifier is entered, then the header lines of the media files from DESCARTES are written to the report file. The following is an example use of the VERBOSE keyword:

```
VERBOSE
```

The following is an example use of the VERBOSE keyword where an echo of the information from the media header files to the report file is desired:

```
VERBOSE MEDIA
```

9.6.25 WEANED Keyword

The WEANED keyword is used to define the date that a suckling individual is weaned. The WEANED keyword is only used in the context of a dose case (see Section 9.6.2). The following is this keyword's syntax:

```
WEANED “quote”
```

The quote string on the WEANED keyword identifies the date the individual is weaned. The format of the quote string is “YYYY-MM-DD” where YYYY is a four digit year, MM is a two digit month and DD is a two digit day. Leading zeros must be used for month and day entries less than 10. An example keyword entry for a weaned date on June 7, 1955, is the following:

```
WEANED “1955-06-07”
```

There is no default date for weaning. A wean date must be entered to terminate suckling.

9.6.26 WASHED Keyword

The optional WASHED keyword is used to activate the option where the individual consumes fruit and leafy vegetables that have been washed prior to consumption. Washing (or food processing) is assumed to remove part of the ^{131}I associated with the outer compartment of the food. The WASHED keyword is only used in the context of a dose case (see Section 9.6.2). The following is this keyword's syntax:

```
WASHED [YES|NO]
```

The default action if the WASHED keyword is omitted is that no washing or food processing that would remove contaminants takes place. The default action is the same as the action when the following entry is used:

```
WASHED NO
```

The following keyword entry should be used to activate the washing option:

```
WASHED YES
```

9.7 Keywords in the CiderF Factors Keyword File

The keywords used in a CiderF factors keyword file are described in the following sections. A summary of all keywords used by the CiderF code is provided in Table 9.8.

9.7.1 AGEBREAT Keyword

The AGEBREAT keyword is used to define the age categories for age dependent breathing rates used in the calculation of inhalation doses. The following is this keyword's syntax:

```
AGEBREAT N1 N2 (N3) ... (Nn)
```

The numerical values on this keyword are the breakpoints between different age categories. The values all have units of years. The first category is the age range from the numerical values of N1 to N2. Similarly, the second age category is the age range from N2 to N3. In this application, the lowest age typically is set to 0 years and the highest age is set to a large number. Up to 99 age categories can be defined.

The following keyword entry defines age dependent breathing rates for six age categories. Age category 1 is 0.0 to 0.5 years; category 2 is 0.5 to 2.0 years; category 3 is 2.0 to 7.0 years; category 4 is 7.0 to 12.0 years; category 5 is 12.0 to 17.0 years; and category 6 is 17.0 to 999.0 years.

```
AGEBREAT 0.0 0.5 2.0 7.0 12.0 17.0 999.0
```

9.7.2 AGEDIET Keyword

The AGEDIET keyword is used to define the age categories for age dependent reference diets. The following is this keyword's syntax:

```
AGEDIET N1 N2 (N3) ... (Nn)
```

The numerical values on this keyword are the breakpoints between different age categories. The values all have units of years. The first category is the age range from the numerical values of N1 to N2. Similarly, the second age category is the age range from N2 to N3. In this application, the lowest age typically is set to 0 years and the highest age is set to a large number. Up to 99 age categories can be defined.

The following keyword entry defines age dependent reference diet categories for six age categories. Age category 1 is 0.0 to 0.5 years; category 2 is 0.5 to 2.0 years; category 3 is 2.0 to 7.0 years; category 4 is 7.0 to 12.0 years; category 5 is 12.0 to 17.0 years; and category 6 is 17.0 to 999.0 years.

```
AGEDIET 0.0 0.5 2.0 7.0 12.0 17.0 999.0
```

9.7.3 AGEFRACO Keyword

The AGEFRACO keyword is used to define the age categories for age dependent fraction of time spent outdoors. The following is this keyword’s syntax:

```
AGEFRACO N1 N2 (N3) ... (Nn)
```

The numerical values on this keyword are the breakpoints between different age categories. The values all have units of years. The first category is the age range from the numerical values of N1 to N2. Similarly, the second age category is the age range from N2 to N3. In this application, the lowest age typically is set to 0 years and the highest age is set to a large number. Up to 99 age categories can be defined.

The following keyword entry defines age dependent fraction of time spent outdoors for three age categories. Age category 1 is 0.0 to 2.0 years; category 2 is 2.0 to 17.0 years; and category 3 is 17.0 to 999.0 years.

```
AGEFRACO 0.0 2.0 17.0 999.0
```

9.7.4 AGEINT Keyword

The AGEINT keyword is used to define the age categories for age dependent internal dose conversion factors. The following is this keyword’s syntax:

```
AGEINT N1 N2 (N3) ... (Nn)
```

The numerical values on this keyword are the breakpoints between different age categories. The values all have units of years. The first category is the age range from the numerical values of N1 to N2. Similarly, the second age category is the age range from N2 to N3. In this application, the lowest age typically is set to 0 years and the highest age is set to a large number. Up to 99 age categories can be defined.

The following keyword entry defines age dependent internal dose categories for six age categories. Age category 1 is 0.0 to 0.25 years; category 2 is 0.25 to 1.0 years; category 3 is 1.0 to 5.0 years; category 4 is 5.0 to 10.0 years; category 5 is 10.0 to 15.0 years; and category 6 is 15.0 to 999.0 years.

```
AGEINT 0.0 0.25 1.0 5.0 10.0 15.0 999.0
```

9.7.5 CREAMERY Keyword

The CREAMERY keyword is used to define the number of creameries present in the concentration data produced by the DESCATES code. The number of creameries is needed to correctly access the data in the binary creamery media file written by DESCARTES. The following is this keyword’s syntax:

```
CREAMERY N1
```

The value for N1 must be an integer greater than 0. An example keyword for a run of DESCARTES with 4 creameries is the following:

```
CREAMERY 4
```

9.7.6 DIETMAP Keyword

The DIETMAP keyword is used to define which reference diet to apply to an individual when calculating doses. The following is this keyword's syntax:

```
DIETMAP LIFESTYLE=N1 SEASON=N2 SEX=N3 ACTIVITY=N4 AGE=N5 ID="quote"
```

Five pieces of information are used to choose the reference diet for an individual on a given day. Explanation of the modifiers and associated data for this keyword is provided in Table 9.20. The categories of lifestyle, season, sex and activity and their associated indices are defined in Table 9.1.

Table 9.20 Modifiers Associated with the DIETMAP Keyword for CiderF

Modifier	Description
ACTIVITY	The numerical value associated with the ACTIVITY modifier is the index for the lifestyle activity type. For example, the index for a suckling child is 2.
AGE	The numerical value associated with the AGE modifier is the index for the age category. The categories for age are defined using the AGEDIET keyword (see Section 9.7.2). Using the example keyword provided in that section, category 3 would apply individuals between 2 and 7 years of age.
ID	The quote string associated with the ID modifier identifies a reference diet defined using the DIETREF keyword.
LIFESTYLE	The numerical value associated with the LIFESTYLE modifier is the index for the lifestyle. For example, the index for a rural lifestyle is 1.
SEASON	The numerical value associated with the SEASON modifier is the index for the season of year. For example, the index for spring is 2.
SEX	The numerical value associated with the SEX modifier is the index for the sex (gender) of the individual. For example, the index for a male is 1.

Multiple DIETMAP keywords are required. An upper bound on the number of separate DIETMAP keyword is the product of the number of activities (3), the number of lifestyles (2), the number of sexes (2), the number of activities (3 – prenatal is not an activity for diet purposes) and the number of age categories for diets. However, if one assumes that all children are weaned by age 2 and that no females under age 12 are pregnant (using the example AGEDIET keyword provided in Section 9.7.2), then only 136 DIETMAP keywords are required. The following entries illustrate the use of the DIETMAP keyword:

```
DIETMAP Lifestyle=1 Season=1 Sex=1 Activity=1 Age=1 ID="RefDiet001"
   ! rural, winter, male, normal, 0 to 6 months
DIETMAP Lifestyle=1 Season=1 Sex=2 Activity=1 Age=1 ID="RefDiet001"
   ! rural, winter, female, normal, 0 to 6 months
```

```
DIETMAP Lifestyle=1 Season=2 Sex=1 Activity=1 Age=1 ID="RefDiet001"
   ! rural, spring, male, normal, 0 to 6 months
DIETMAP Lifestyle=1 Season=2 Sex=2 Activity=1 Age=1 ID="RefDiet001"
   ! rural, spring, female, normal, 0 to 6 months
DIETMAP Lifestyle=1 Season=2 Sex=2 Activity=1 Age=6 ID="RefDiet009"
   ! rural, spring, female, normal, >17 years
DIETMAP Lifestyle=1 Season=3 Sex=1 Activity=1 Age=6 ID="RefDiet009"
   ! rural, summer, male, normal, >17 years
DIETMAP Lifestyle=1 Season=3 Sex=2 Activity=1 Age=6 ID="RefDiet009"
   ! rural, summer, female, normal, >17 years
DIETMAP Lifestyle=1 Season=4 Sex=1 Activity=1 Age=6 ID="RefDiet009"
   ! rural, fall, male, normal, >17 years
DIETMAP Lifestyle=1 Season=1 Sex=1 Activity=2 Age=1 ID="RefDiet011"
   ! rural, winter, male, suckling, 0 to 6 months
DIETMAP Lifestyle=1 Season=1 Sex=2 Activity=2 Age=1 ID="RefDiet011"
   ! rural, winter, female, suckling, 0 to 6 months
DIETMAP Lifestyle=1 Season=2 Sex=2 Activity=3 Age=6 ID="RefDiet013"
   ! rural, spring, female, preg_nurse, >17 years
DIETMAP Lifestyle=1 Season=3 Sex=2 Activity=3 Age=6 ID="RefDiet013"
   ! rural, summer, female, preg_nurse, >17 years
DIETMAP Lifestyle=1 Season=4 Sex=2 Activity=3 Age=6 ID="RefDiet013"
   ! rural, fall, female, preg_nurse, >17 years
DIETMAP Lifestyle=2 Season=1 Sex=2 Activity=3 Age=6 ID="RefDiet014"
   ! urban, winter, female, preg_nurse, >17 years
DIETMAP Lifestyle=2 Season=2 Sex=2 Activity=3 Age=6 ID="RefDiet014"
   ! urban, spring, female, preg_nurse, >17 years
```

9.7.7 DIETREF Keyword

The DIETREF keyword is used to define reference diets. Multiple entries of this keyword are required. The following is this keyword's syntax:

```
DIETREF ID=“quote1” TITLE=“quote2” (MILK_1I=N1) (MILK_1H=N2) (MILK_2I=N3)
   (MILK_2H=N4) (MILK_3I=N5) (MILK_3H=N6) (MILK_4I=N7) (MILK_4H=N8) (MILK_GRO=N9)
   (MILK_GOA=N10) (MILK_CRE=N11) (CREAM_NU=N12) (VEGLOC=N13)
```

Reference diets have two aspects. First, the reference diet assigns milk and leafy vegetable consumption fractions to different possible sources. The description of modifiers and associated data provided in Table 9.21 performs this function. In addition, a reference diet specifies intake rates. However, the intake rates for reference diets are entered using the STOCHAST keyword (see Section 9.7.15).

Table 9.21 Modifiers Associated with the DIETREF Keyword for CiderF

Modifier	Description
ID	The quote string associated with the ID modifier identifies a reference diet. Up to 10 characters can be used in the quote string. The ID's for reference diets must be unique.

Modifier	Description
TITLE	The quote string associated with the TITLE modifier identifies a descriptive title used for labeling purposes. Up to 72 characters can be used in the title.
MILK_1I	The numerical value associated with the MILK_1I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 1. The sum of milk intake fractions over all milk sources must equal 1.
MILK_1H	The numerical value associated with the MILK_1H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 1. The sum of milk intake fractions over all milk sources must equal 1.
MILK_2I	The numerical value associated with the MILK_2I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 2. The sum of milk intake fractions over all milk sources must equal 1.
MILK_2H	The numerical value associated with the MILK_2H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 2. The sum of milk intake fractions over all milk sources must equal 1.
MILK_3I	The numerical value associated with the MILK_3I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 3. The sum of milk intake fractions over all milk sources must equal 1.
MILK_3H	The numerical value associated with the MILK_3H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 3. The sum of milk intake fractions over all milk sources must equal 1.
MILK_4I	The numerical value associated with the MILK_4I modifier identifies the fraction of milk intake that comes from individual (backyard) cows on feeding regime 4. The sum of milk intake fractions over all milk sources must equal 1.
MILK_4H	The numerical value associated with the MILK_4H modifier identifies the fraction of milk intake that comes from herd cows on feeding regime 4. The sum of milk intake fractions over all milk sources must equal 1.
MILK_GRO	The numerical value associated with the MILK_GRO modifier identifies the fraction of milk intake that comes from grocery stores. The sum of milk intake fractions over all milk sources must equal 1.
MILK_GOA	The numerical value associated with the MILK_GOA modifier identifies the fraction of milk intake that comes from a local goat. The sum of milk intake fractions over all milk sources must equal 1.
MILK_CRE	The numerical value associated with the MILK_CRE modifier identifies the fraction of milk intake that comes from a specific creamery. The creamery number must be provided with the CREAM_NU modifier. The sum of milk intake fractions over all milk sources must equal 1.

Modifier	Description
CREAM_NU	The numerical value associated with the CREAM_NU identifies the creamery number for intake of creamery milk identified with the MILK_CRE modifier.
VEGLOC	The numerical value associated with the VEGLOC modifier identifies the fraction of leafy vegetable consumption that is locally produced (a garden, for example). A value between 0 and 1 is required. Any vegetables consumed that are not locally produced are assumed to be commercially produced. A value of 0.2 for this entry means that 20% of vegetables consumed are locally produced and 80% are commercially produced.

The following example keyword identifies a reference diet where all milk comes from a local cow on feeding regime 1 and only locally produced vegetables are consumed.

```
DIETREF ID="RefDiet001" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, 0 to 6 months"
```

The consumption fractions for reference diets default to 0 for all milk consumption categories. The fraction of leafy vegetables from local sources defaults to 1. Thus, the following keyword has the same effect as the above keyword.

```
DIETREF ID="RefDiet001" milk_1i=1 TITLE="Child, 0 to 6 months"
```

The following set of DIETREF keywords identifies 14 reference diets for a range of age categories and life activities combined with rural lifestyle assumptions and local vegetable assumptions.

```
DIETREF ID="RefDiet001" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, 0 to 6 months"

DIETREF ID="RefDiet002" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, >6 months to 2 year"

DIETREF ID="RefDiet003" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, rural, >2 years to 7 years"

DIETREF ID="RefDiet004" milk_1i=0 milk_1h=1 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, urban, >2 years to 7 years"

DIETREF ID="RefDiet005" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, rural, >7 years to 12 years"

DIETREF ID="RefDiet006" milk_1i=0 milk_1h=1 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, urban, >7 years to 12 years"

DIETREF ID="RefDiet007" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, rural, >12 years to 17 years"
```

```
DIETREF ID="RefDiet008" milk_1i=0 milk_1h=1 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Child, urban, >12 years to 17 years"

DIETREF ID="RefDiet009" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Adult, rural"

DIETREF ID="RefDiet010" milk_1i=0 milk_1h=1 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Adult, urban"

DIETREF ID="RefDiet011" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Suckling child, 0 to 6 months"

DIETREF ID="RefDiet012" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Suckling child, >6 months to 2 year"

DIETREF ID="RefDiet013" milk_1i=1 milk_1h=0 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Adult, rural, pregnant/nursing female"

DIETREF ID="RefDiet014" milk_1i=0 milk_1h=1 milk_2i=0 milk_2h=0 milk_3i=0 milk_3h=0
   milk_4i=0 milk_4h=0 milk_gro=0 milk_cre=0 cream_nu=0 milk_goa=0 vegloc=1
   TITLE="Adult, urban, pregnant/nursing female"
```

9.7.8 DIETSCAL Keyword

The optional DIETSCAL keyword is used to provide scaling factors for reference diet intake rates that are specific to a calendar year. Multiple entries of this keyword may be used. The following is this keyword’s syntax:

```
DIETSCAL YEAR=N1 (f_milk=N2) (s_milk=N3) (l_veg=N4) (o_veg=N5) (fruit=N6)
   (grain=N7) (eggs=N8) (beef=N9) (poultry=N10)
```

The modifiers and associated data for this keyword are defined in Table 9.22. The modifiers on this keyword match with the food type tokens identified in Table 9.1. The YEAR modifier is required, but all other modifiers are optional. At least one modifier other than the YEAR modifier must be entered for this keyword to affect intake rates. The default value for all scaling factors is 1.

Table 9.22 Modifiers Associated with the DIETSCAL Keyword for CiderF

Modifier	**Description**
YEAR	The numerical value associated with the YEAR modifier specifies the year that the scaling factors are defined for.
f_milk	The numerical value associated with the f_milk modifier provides the scaling factor for the consumption of fresh milk products.
s_milk	The numerical value associated with the s_milk modifier provides the scaling factor for the consumption of stored milk products.

Modifier	Description
l_veg	The numerical value associated with the l_veg modifier provides the scaling factor for the consumption of leafy vegetable products.
o_veg	The numerical value associated with the o_veg modifier provides the scaling factor for the consumption of other vegetable products.
fruit	The numerical value associated with the fruit modifier provides the scaling factor for the consumption of fruit.
grain	The numerical value associated with the grain modifier provides the scaling factor for the consumption of grain products.
eggs	The numerical value associated with the eggs modifier provides the scaling factor for the consumption of eggs.
beef	The numerical value associated with the beef modifier provides the scaling factor for the consumption of beef products.
poultry	The numerical value associated with the poultry modifier provides the scaling factor for the consumption of poultry.

The purpose of this keyword is to allow easy modification of reference diets. For example, suppose that a late frost in the year 1956 greatly reduced the production of leafy vegetables. Thus, the consumption rate for leafy vegetables dropped to 40% of normal for one year and then rose to normal levels. The default value for all scaling factors is 1, thus only the l_veg entry needs to be specified. This change from reference diets could be accounted for with the following keyword:

```
DIETSCAL YEAR=1956 l_veg=0.4
```

For another example, suppose that the production of eggs doubled in 1966 from previous levels and the previous consumption rates had been limited by the supply. The following keywords double the reference diet consumption rates for eggs from 1966 through 1972.

```
DIETSCAL YEAR=1966 eggs=2.0
DIETSCAL YEAR=1967 eggs=2.0
DIETSCAL YEAR=1968 eggs=2.0
DIETSCAL YEAR=1969 eggs=2.0
DIETSCAL YEAR=1970 eggs=2.0
DIETSCAL YEAR=1971 eggs=2.0
DIETSCAL YEAR=1972 eggs=2.0
```

9.7.9 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword’s syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

9.7.10 FILE Keyword

The FILE keyword is used to enter the names of input files for files produced by the DESCARTES code. The following is this keyword's syntax:

```
FILE modifier "quote"
```

The file names are entered in quote strings. Path names up to 256 characters long (name length limitation in Windows 7) are supported. The modifiers associated with the FILE keyword are described in Table 9.23. More than one FILE keyword may be used.

Table 9.23 Modifiers Associated with the FILE Keyword for CiderF Factors Keywords

Modifier	Description
MED_AIRC	The quote string associated with the MED_AIRC modifier contains the file name for the air concentration data.
MED_AIRD	The quote string associated with the MED_AIRD modifier contains the file name for air deposition data.
MED_BEEF	The quote string associated with the MED_BEEF modifier contains the file name for beef data.
MED_EGGS	The quote string associated with the MED_EGGS modifier contains the file name for eggs data.
MED_FRTI	The quote string associated with the MED_FRTI modifier contains the file name for inner compartment fruit data.
MED_FRTO	The quote string associated with the MED_FRTO modifier contains the file name for outer compartment fruit data.
MED_GOAT	The quote string associated with the MED_GOAT modifier contains the file name for goat milk data.
MED_GRN	The quote string associated with the MED_GRN modifier contains the file name for grain data.
MED_LVIC	The quote string associated with the MED_LVIC modifier contains the file name for inner compartment commercial leafy vegetable data.
MED_LVIL	The quote string associated with the MED_LVIL modifier contains the file name for inner compartment local leafy vegetable data.
MED_LVOC	The quote string associated with the MED_LVOC modifier contains the file name for outer compartment commercial leafy vegetable data.
MED_LVOL	The quote string associated with the MED_LVOL modifier contains the file name for outer compartment local leafy vegetable data.
MED_MCRM	The quote string associated with the MED_MCRM modifier contains the file name for creamery milk data.

Modifier	Description
MED_MGRO	The quote string associated with the MED_MGRO modifier contains the file name for grocery milk data.
MED_MR1H	The quote string associated with the MED_MR1H modifier contains the file name for herd cow milk data for feeding regime 1.
MED_MR1I	The quote string associated with the MED_MR1I modifier contains the file name for individual (backyard) cow milk data for feeding regime 1.
MED_MR2H	(Optional Entry) The quote string associated with the MED_MR2H modifier contains the file name for herd cow milk data for feeding regime 2.
MED_MR2I	(Optional Entry) The quote string associated with the MED_MR2I modifier contains the file name for individual (backyard) cow milk data for feeding regime 2.
MED_MR3H	(Optional Entry) The quote string associated with the MED_MR3H modifier contains the file name for herd cow milk data for feeding regime 3.
MED_MR3I	(Optional Entry) The quote string associated with the MED_MR3I modifier contains the file name for individual (backyard) cow milk data for feeding regime 3.
MED_MR4H	(Optional Entry) The quote string associated with the MED_MR4H modifier contains the file name for herd cow milk data for feeding regime 4.
MED_MR4I	(Optional Entry) The quote string associated with the MED_MR4I modifier contains the file name for individual (backyard) cow milk data for feeding regime 4.
MED_OVG	The quote string associated with the MED_OVG modifier contains the file name for other vegetable data.
MED_POUL	The quote string associated with the MED_POUL modifier contains the file name for poultry data.
MED_RZ	The quote string associated with the MED_RZ modifier contains the file name for root zone soil data.
MED_USL	The quote string associated with the MED_USL modifier contains the file name for upper zone (surface) soil data.
USLR	The quote string associated with the USLR modifier contains the file name for common parameter data passed from DESCARTES to CiderF.

The following FILE keyword entries define all of the required data files for the FILE keyword.

```
FILE MED_AIRC "D:\Mayak\Best\ConcenBest.med"
FILE MED_AIRD "D:\Mayak\Best\DepositBest.med"
FILE MED_BEEF "D:\Mayak\Best\BeefBest.med"
FILE MED_EGGS "D:\Mayak\Best\EggsBest.med"
FILE MED_FRTI "D:\Mayak\Best\FruitinBest.med"
FILE MED_FRTO "D:\Mayak\Best\FruitotBest.med"
FILE MED_GOAT "D:\Mayak\Best\GoatBest.med"
FILE MED_GRN  "D:\Mayak\Best\GrainBest.med"
FILE MED_MGRO "D:\Mayak\Best\GroceryBest.med"
FILE MED_LVIL "D:\Mayak\Best\LvegInBest.med"
FILE MED_LVOL "D:\Mayak\Best\LvegOtBest.med"
FILE MED_LVIC "D:\Mayak\Best\LvegInComBest.med"
FILE MED_LVOC "D:\Mayak\Best\LvegOtComBest.med"
FILE MED_MCRM "D:\Mayak\Best\CreamBest.med"
FILE MED_MR1H "D:\Mayak\Best\Milk1HerdBest.med"
FILE MED_MR1I "D:\Mayak\Best\Milk1IndivBest.med"
FILE MED_OVG  "D:\Mayak\Best\OtherVegBest.med"
FILE MED_POUL "D:\Mayak\Best\PoultryBest.med"
FILE MED_RZ   "D:\Mayak\Best\SoilRootBest.med"
FILE MED_USL  "D:\Mayak\Best\SoilUpBest.med"
FILE USLR     "D:\Mayak\Best\PlantsBest.dat"
```

9.7.11 NODENUM Keyword

The NODENUM keyword is used to define the number of nodes that were used in the DESCARTES codes. The number of nodes is needed to correctly access the data in the binary media files written by DESCARTES. The following is this keyword’s syntax:

```
NODENUM N1
```

The number of nodes must be entered as an integer. The following keyword identifies that a total of 516 nodes were be used.

```
NODENUM 516
```

9.7.12 NUCLIDE Keyword

The NUCLIDE keyword is used to define the nuclide used in CiderF. The following is this keyword’s syntax:

```
NUCLIDE ID=“quote”  HALFLIFE=N1
```

The quote string associated with the ID modifier identifies the radionuclide of interest. For this application, the only valid entry is “I131”. The numerical value associated with the HALFLIFE modifier is the radioactive half-life of the nuclide in days. The following keyword identifies ^{131}I as the nuclide of interest and provides the 8.0207 day half-life.

```
NUCLIDE ID="I131" HALFLIFE=8.0207
```

9.7.13 SEASON Keyword

The data defining the fraction of time an individual spends outdoors and the reference diet definitions are indexed by the season of year. The SEASON keyword is used to define the range of days assigned to different seasons. The following is this keyword’s syntax:

```
SEASON [WINTER | SPRING | SUMMER | FALL] N1 N2
```

The days the seasons are active do not vary by realization. Four seasons are defined. These seasons are identified by one of the four modifiers WINTER, SPRING, SUMMER and FALL on the SEASON keyword. The numbers N1 and N2 denote the range of days (Julian days) that a specific season occurs. There is no default for the season definitions. Every day in the year must be assigned to a season.

Because the SEASON definitions apply to multiple years, the seasons must be defined for a 366 day year. At least one SEASON keyword is required for every season. The definition of a SEASON keyword will override the definition provided in earlier SEASON keywords if the day range overlaps.

The following set of SEASON keywords define a season for every day of the year:

```
SEASON WINTER 1   102
SEASON SPRING 103 163
SEASON SUMMER 164 297
SEASON FALL   298 330
SEASON WINTER 331 366
```

The following set of SEASON keywords defines the same seasons as the above keywords:

```
SEASON WINTER 1   366
SEASON SPRING 103 140
SEASON SPRING 130 163
SEASON SUMMER 164 297
SEASON FALL   298 330
```

9.7.14 STEP Keyword

The STEP keyword defines the output averaging interval for data in the media files written by the DESCARTES code. The STEP keyword used in CiderF must contain exactly the same information as the STEP keyword used in the DESCARTES code. The following is this keyword’s syntax:

```
STEP [ DAY | WEEK | MONTH ]
```

The following keyword entry should be used when the DESCARTES code outputs media concentrations on a monthly interval.

```
STEP MONTH
```

9.7.15 STOCHAST Keyword

Stochastic input variables are defined by the STOCHAST keyword. The following is this keyword’s syntax:

```
STOCHAST [ID=“Quote1”] [TITLE=“Quote2”] [UNITS=“quote3”] [DIST=“Quote4”]
   PARAM N1,…Nn (TRUNCATE U1 U2)
```

The function of the modifiers on the STOCHAST is explained in detail in Section 12.1 and that information is not repeated here. This description focuses on the how to define different values for the quote string associated with the ID modifier so the code can internally access the correct data. Many of the ID’s contain tokens defined in Table 9.1 along with other information. The quote strings associated with the ID modifier are highlighted in blue for presentation purposes.

Information is entered in the factors keyword file for CiderF using STOCHAST keywords for the following groups of variables:

- Breathing rates (by age category)
- Fraction of time spent outdoors (by age category)
- Dose factors for internal exposure (by pathway, organ, sex and age categories)

- Dose factors for external exposure (by source and organ categories)
- Transfer data for prenatal individuals (by organ category)
- Transfer data for nursing individuals (by organ category)
- Indoor to outdoor air activity ratios
- Shielding factors
- Holdup times for foods (by food type and distribution category)
- Ratio of dry weight to wet weight (by food type)
- Food processing retention fractions (fruit and leafy vegetables)
- Reference diet consumption rates (by food type and reference diet)

9.7.15.1 Stochastic ID’s for Breathing Rates

The stochastic ID’s for breathing rate data are 9 characters in length. The first 7 characters are “BREATHE”, all in upper case. The next two characters are the age category index (see Section 9.7.1), where the lowest age category has index 1. Category indices less than 10 must use a leading zero. For example, age category 1 is denoted by “01”. As many keywords are required as there are breathing rate categories. Example stochastic keywords for six breathing rates are as follows:

```
STOCHAST ID="BREATHE01" DIST="constant" PARAM 1.62 UNITS="m^3/day"
    TITLE="Breathing rate for age: 0 to 6 months"
STOCHAST ID="BREATHE02" DIST="constant" PARAM 5.14 UNITS="m^3/day"
    TITLE="Breathing rate for age: >6 months to 2 year"
STOCHAST ID="BREATHE03" DIST="constant" PARAM 8.71 UNITS="m^3/day"
    TITLE="Breathing rate for age: >2 years to 7 years"
STOCHAST ID="BREATHE04" DIST="constant" PARAM 15.3 UNITS="m^3/day"
    TITLE="Breathing rate for age: >7 years to 12 years"
STOCHAST ID="BREATHE05" DIST="constant" PARAM 17.7 UNITS="m^3/day"
    TITLE="Breathing rate for age: >12 years to 17 years"
STOCHAST ID="BREATHE06" DIST="constant" PARAM 22   UNITS="m^3/day"
    TITLE="Breathing rate for age: >17 years"
```

9.7.15.2 Stochastic ID’s for Fraction of Time Spent Outdoors

The stochastic ID’s for the fraction of time spent outdoors are 26 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 7: The characters “FRACOUT”, all in upper case.
- Characters 8 to 9: The next two characters are the age category index (see Section 9.7.3). Category indices less than 10 use a leading zero. For example, age category 1 is denoted by “01”.
- Characters 10 to 14: One of the lifestyle type tokens from Table 9.1.
- Characters 15 to 20: One of the season tokens from Table 9.1, blank padded as needed.
- Characters 21 to 26: One of the gender tokens from Table 9.1, blank padded as needed.

A separate keyword entry is needed for every combination of age category, lifestyle, season and gender. If there are 6 age categories, then 48 distinct entries are required. A few example stochastic keywords for the fraction of time spent outdoors are as follows:

```
STOCHAST ID="FRACOUT01ruralwintermale  " DIST="Constant" PARAM 0    UNITS="none"
```

```
   TITLE="Fraction time outdoors for age 1 rural ruralwinter male
STOCHAST ID="FRACOUT03ruralfall  female" DIST="Constant" PARAM 0.21 UNITS="none"
   TITLE="Fraction time outdoors for age 3 rural fall   female"
STOCHAST ID="FRACOUT03urbanwintermale  " DIST="Constant" PARAM 0.05 UNITS="none"
   TITLE="Fraction time outdoors for age 3 urban winter male"
```

9.7.15.3 Stochastic ID’s for Dose Factors for Internal Exposure

The stochastic ID’s for dose factors for internal exposure are 29 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 8: The characters “INTERNAL”, all in upper case.
- Characters 9 to 11: One of the tokens for internal exposure types from Table 9.1, blank padded as needed.
- Characters 12 to 13: The age category index for internal dose factors (see Section 9.7.4). Category indices less than 10 use a leading zero. For example, age category 1 is denoted by “01”.
- Characters 14 to 23: One of the organ tokens from Table 9.1, blank padded as needed.
- Characters 24 to 29: One of the gender tokens from Table 9.1, blank padded as needed.

A separate keyword entry is needed for every combination of exposure type, age category, organ and gender. If there are 6 age categories, then 48 distinct entries are required. A few example stochastic keywords for the dose factors for internal exposure are as follows:

```
STOCHASTIC ID="INTERNALing01thyroid   male  " DIST="Constant" PARAM 1.100E+07
   UNITS="rad/Ci" TITLE="Dose factor for age 1 ing thyroid    male  "
STOCHASTIC ID="INTERNALing02thyroid   male  " DIST="Constant" PARAM 9.999E+06
   UNITS="rad/Ci" TITLE="Dose factor for age 2 ing thyroid    male  "
STOCHASTIC ID="INTERNALing03thyroid   male  " DIST="Constant" PARAM 6.101E+06
   UNITS="rad/Ci" TITLE="Dose factor for age 3 ing thyroid    male  "
STOCHASTIC ID="INTERNALing04thyroid   male  " DIST="Constant" PARAM 3.201E+06
   UNITS="rad/Ci" TITLE="Dose factor for age 4 ing thyroid    male  "
STOCHASTIC ID="INTERNALing05thyroid   male  " DIST="Constant" PARAM 2.001E+06
   UNITS="rad/Ci" TITLE="Dose factor for age 5 ing thyroid    male  "
STOCHASTIC ID="INTERNALinh05thyroid   female" DIST="Constant" PARAM 1.200E+06
   UNITS="rad/Ci" TITLE="Dose factor for age 5 inh thyroid    female"
STOCHASTIC ID="INTERNALinh06thyroid   female" DIST="Constant" PARAM 9.403E+05
   UNITS="rad/Ci" TITLE="Dose factor for age 6 inh thyroid    female"
STOCHASTIC ID="INTERNALinh01whole_bodymale  " DIST="Constant" PARAM 2.000E+05
   UNITS="rem/Ci" TITLE="Dose factor for age 1 inh whole_body male  "
```

9.7.15.4 Stochastic ID’s for Dose Factors for External Exposure

The stochastic ID’s for dose factors for internal exposure are 25 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 8: The characters “EXTERNAL”, all in upper case.
- Characters 9 to 15: One of the tokens for external exposure types from Table 9.1, blank padded as needed.
- Characters 16 to 25: One of the organ tokens from Table 9.1, optionally blank padded.

A separate keyword entry is needed for every combination of exposure type and organ, thus 6 distinct entries are required. Example stochastic keywords for the dose factors for external exposure are as follows:

```
STOCHASTIC ID="EXTERNALimm    thyroid   " DIST="Constant" PARAM 5680
     UNITS="(rad/day)/(ci/m^3)" TITLE="Dose factor EXTERNAL imm     thyroid   "

STOCHASTIC ID="EXTERNALu_soil thyroid   " DIST="Constant" PARAM 102
     UNITS="(rad/day)/(ci/m^2)" TITLE="Dose factor EXTERNAL u_soil  thyroid   "

STOCHASTIC ID="EXTERNALrz_soilthyroid   " DIST="Constant" PARAM 68.5
     UNITS="(rad/day)/(ci/m^2)" TITLE="Dose factor EXTERNAL rz_soil thyroid   "

STOCHASTIC ID="EXTERNALimm    whole_body" DIST="Constant" PARAM 5230
     UNITS="(rem/day)/(ci/m^3)" TITLE="Dose factor EXTERNAL imm     whole_body"

STOCHASTIC ID="EXTERNALu_soil whole_body" DIST="Constant" PARAM 93.5
     UNITS="(rem/day)/(ci/m^2)" TITLE="Dose factor EXTERNAL u_soil  whole_body"

STOCHASTIC ID="EXTERNALrz_soilwhole_body" DIST="Constant" PARAM 68.5
     UNITS="(rem/day)/(ci/m^2)" TITLE="Dose factor EXTERNAL rz_soil whole_body"
```

9.7.15.5 Stochastic ID's for Dose Factors for Prenatal Transfer

The stochastic ID's for dose factors from prenatal transfer from mother to fetus are 18 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 8: The characters "PRENATAL", all in upper case.
- Characters 9 to 18: One of the organ tokens from Table 9.1, blank padded as needed.

Only two keyword entries are needed, one for thyroid doses and one for whole body doses. Two example stochastic keywords for dose factors from prenatal transfer from mother to fetus are as follows:

```
STOCHAST ID="PRENATALthyroid   " DIST="Constant" PARAM 2.30E+6 UNITS="rad/Ci"
   TITLE="Transfer PRENATAL thyroid"

STOCHAST ID="PRENATALwhole_body" DIST="Constant" PARAM 2.30E+6 UNITS="rem/Ci"
   TITLE="Transfer PRENATAL whole_body"
```

9.7.15.6 Stochastic ID's for Dose Factors for Nursing Infant Transfer

The stochastic ID's for dose factors from transfer from mother to nursing infant are 17 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 7: The characters "NURSING", all in upper case.
- Characters 8 to 17: One of the organ tokens from Table 9.1, blank padded as needed.

Only two keyword entries are needed, one for thyroid doses and one for whole body doses. Two example stochastic keywords for dose factors from transfer from mother to nursing infant are as follows:

```
STOCHASTIC ID="NURSINGthyroid   " DIST="Constant" PARAM 2.40E+06 UNITS="rad/Ci"
   TITLE="Transfer NURSING thyroid   "

STOCHASTIC ID="NURSINGwhole_body" DIST="Constant" PARAM 7.00E+04 UNITS="rem/Ci"
   TITLE="Transfer NURSING whole_body"
```

9.7.15.7 Stochastic ID for Indoor to Outdoor Air Activity Ratio

The stochastic ID for the indoor to outdoor air activity ratio is the character string “RIO”. An example keyword for this variable is the following:

```
STOCHASTIC ID="RIO" DIST="Constant" PARAM 0.68 UNITS="none"
    TITLE="Indoor to outdoor ratio"
```

9.7.15.8 Stochastic ID for Indoor Shielding Factor

The stochastic ID for the indoor shielding factor is the character string “SHIELD”. An example keyword for this variable is the following:

```
STOCHASTIC ID="SHIELD" DIST="Constant" PARAM 0.5 UNITS="none"
    TITLE="Indoor to outdoor ratio"
```

9.7.15.9 Stochastic ID’s for Food Holdup Times

The stochastic ID’s for the holdup time (in days) between food production and human consumption are 25 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 6: The characters “HOLDUP”, all in upper case.
- Characters 7 to 14: One of the food type tokens from Table 9.1, blank padded as needed.
- Characters 15 to 25: One of the food distribution type tokens from Table 9.1.

A separate keyword entry is needed for every combination of food types and food distribution types, thus 18 distinct entries are required. A few example stochastic keywords for the food holdup times are as follows:

```
STOCHASTIC ID="HOLDUPbeef    local      " DIST="Constant" PARAM 14   UNITS="day"
    TITLE="Holdup time local beef"

STOCHASTIC ID="HOLDUPl_veg   local      " DIST="Constant" PARAM 3.5  UNITS="day"
    TITLE="Holdup time local l_veg"

STOCHASTIC ID="HOLDUPgrain   local      " DIST="Constant" PARAM 3.5  UNITS="day"
    TITLE="Holdup time local grain"

STOCHASTIC ID="HOLDUPeggs    distributed" DIST="Constant" PARAM 10.5 UNITS="day"
    TITLE="Holdup time distributed eggs"

STOCHASTIC ID="HOLDUPf_milk  distributed" DIST="Constant" PARAM 2.5  UNITS="day"
    TITLE="Holdup time distributed f_milk"

STOCHASTIC ID="HOLDUPs_milk  distributed" DIST="Constant" PARAM 37   UNITS="day"
    TITLE="Holdup time distributed s_milk"
```

9.7.15.10 Stochastic ID’s for Ratio of Dry to Wet Food Weights

The stochastic ID’s for the ratio of dry to wet food weights are 11 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 6: The characters “DRYWET”, all in upper case.
- Characters 7 to 11: One of the following four food type tokens from Table 9.1; (l_veg, o_veg, fruit and grain).

Only four separate keyword entries are needed for the ratio of dry to wet food weights. Examples for these four stochastic keywords are as follows:

```
STOCHAST ID="DRYWETl_veg" DIST="Constant" PARAM 0.07  UNITS="kg(dry)/kg(wet)"
   TITLE="Ratio (dry to wet weight) for l_veg"

STOCHAST ID="DRYWETo_veg" DIST="Constant" PARAM 0.15  UNITS="kg(dry)/kg(wet)"
   TITLE="Ratio (dry to wet weight) for o_veg"

STOCHAST ID="DRYWETfruit" DIST="Constant" PARAM 0.24  UNITS="kg(dry)/kg(wet)"
   TITLE="Ratio (dry to wet weight) for fruit"

STOCHAST ID="DRYWETgrain" DIST="Constant" PARAM 0.925 UNITS="kg(dry)/kg(wet)"
   TITLE="Ratio (dry to wet weight) for grain"
```

9.7.15.11 Stochastic ID's for Food Processing Retention Fractions

The stochastic ID's for food processing retention fractions are 11 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 6: The characters "RETAIN", all in upper case.
- Characters 7 to 11: One of the following two food type tokens from Table 9.1; (l_veg and fruit).

Only two separate keyword entries are needed for the food processing retention fractions. Examples for these two stochastic keywords are as follows:

```
STOCHASTIC ID="RETAINl_veg   " DIST="Constant" PARAM 0.45 UNITS="none"
   TITLE="Food processing retention fraction for l_veg"

STOCHASTIC ID="RETAINfruit   " DIST="Constant" PARAM 0.45 UNITS="none"
   TITLE="Food processing retention fraction for fruit"
```

9.7.15.12 Stochastic ID's for Reference Diet Consumption Amounts

The stochastic ID's for reference diet consumption amounts are up to 17 characters in length. The ID is constructed from the following pieces:

- Characters 1 to 10: One of the reference diet ID's defined by the DIETREF keyword (see Section 9.7.7).
- Characters 11 to 17: One of the food type tokens from Table 9.1.

A separate keyword entry is needed for every combination of food type and reference diets, thus a large number of entries may be required. If there are 14 reference diets, then 126 separate entries are required. A few example stochastic keywords for the reference diet consumption amounts are as follows:

```
STOCHASTIC ID="RefDiet005s_milk " DIST="uniform" PARAM 0.0675 0.0825  UNITS="L/day"
   TITLE="RefDiet005 s_milk  consumption" ! Child, rural, >7 years to 12 years

STOCHASTIC ID="RefDiet006f_milk " DIST="uniform" PARAM 0.3825 0.4675  UNITS="L/day"
   TITLE="RefDiet006 f_milk  consumption" ! Child, urban, >7 years to 12 years

STOCHASTIC ID="RefDiet007l_veg  " DIST="uniform" PARAM 0.063 0.077  UNITS="kg/day"
   TITLE="RefDiet007 l_veg   consumption" ! Child, rural, >12 years to 17 years

STOCHASTIC ID="RefDiet008o_veg  " DIST="uniform" PARAM 0.18 0.22  UNITS="kg/day"
   TITLE="RefDiet008 o_veg   consumption" ! Child, urban, >12 years to 17 years
```

```
STOCHASTIC ID="RefDiet009fruit  " DIST="uniform" PARAM 0.0109 0.0329
   UNITS="kg/day" TITLE="RefDiet009 fruit   consumption" ! Adult, rural
STOCHASTIC ID="RefDiet010grain  " DIST="uniform" PARAM 0.459 0.561  UNITS="kg/day"
   TITLE="RefDiet010 grain   consumption" ! Adult, urban
STOCHASTIC ID="RefDiet011eggs   " DIST="constant" PARAM 0  UNITS="kg/day"
   TITLE="RefDiet011 eggs    consumption" ! Suckling child, 0 to 6 months
STOCHASTIC ID="RefDiet013beef   " DIST="uniform" PARAM 0.1881 0.2299
   UNITS="kg/day" TITLE="RefDiet013 beef consumption"
   ! Adult, rural, pregnant/nursing female
STOCHASTIC ID="RefDiet014poultry" DIST="uniform" PARAM 0.0095 0.0115
   UNITS="kg/day" TITLE="RefDiet014 poultry consumption"
   ! Adult, urban, pregnant/nursing female
```

9.7.16 TIME Keyword

The TIME keyword defines the time span for data in the media files written by the DESCARTES code. The TIME keyword used in CiderF should contain exactly the same information as the TIME keyword used in the DESCARTES code. The following is this keyword’s syntax:

```
TIME N1 N2 N3 N4 N5 N6
```

The integer entries Nl, N2, and N3 contain the month number, day of month number and year for the desired start of the run. The integer entries N4, N5, and N6 contain the month number, day of month number and year for the desired end of the run. Years must be entered using four digits.

An example entry for this keyword is the following:

```
TIME 06:01:1948 12:31:1972
```

The following keyword entry cannot be used because the slash (/) is not a separator for the keyword decoding routines (see Section 11.4.3).

```
TIME 06/01/1948 12/31/1972
```

10.0 CiderView – User Instructions

The map mode of the CiderF program produces result files that contain doses at every node in the modeling domain (see Figure 1.2 for an example modeling domain). The CiderView program reads grid files produced by the CiderF code and writes dose contour plot files in KML format for use in Google Earth (Google 2013).

10.1 How the Code is Invoked

The CiderView code runs under the Windows operating system (Release 7). The code executes at a command prompt. A run of CiderF is initiated by entering the following command line:

```
CiderView Keyfilename1
```

For this command, CiderView is the name of the executable program and Keyfilename1 is the name of a keyword file. Both the name of the executable program and the keyword file may contain path information. If CiderView is invoked without entering the name of the keyword file, the code will prompt the user for the file name. The keyword file contains text control information describing the run. If CiderView cannot open the keyword file, then it will terminate execution after writing an error message to the standard output device.

10.2 File Definitions

A run of the CiderView code uses two or three input files and two or more output files.

10.2.1 Input Files

An example keyword file for CiderView is provided in Table 10.1. A run of CiderView using this file generates KML files for the median result for two dose cases calculated by CiderF. Most of the NODE keywords were omitted to reduce the size of this table. Text colored in red denotes comments ignored during code execution.

Table 10.1 Example Keyword File for CiderView

```
USER "Paul W. Eslinger"
! Required Files
FILE REPORT "Child5View.rpt"
FILE DOSE   "Child5Best.csv"
! Define the output grid and data interpolation information
KMLGRID LATMIN=55.075 LATMAX=55.975 LONMIN=59.95 LONMAX=62.15
  LATDEL=0.0065 MAXDIS=4.25
! Define which statistics are used for plots
KMLSTATS MEDIAN
! (Option) Include a blank region defined as a KML polygon
KMLBLANK FILE="D:\Mayak\Best\Maps\Child5\Mayak_Polygon.kml"
! (Option) Include a legend on the map
KMLLEGEND FILE="D:\Mayak\Best\Maps\Child5\WideRangeLegend.gif"
```

```
! Define color palette for Google Earth in hex (opacity,B,G,R)
KMLCOLOR Color_01="B00000FF"
  Color_02="B03399FF" Color_03="B000FFFF" Color_04="B0B4FFFF"
  Color_05="B000FF00" Color_06="B010FFA8" Color_07="B0FF0000"
  Color_08="B0FF9900" Color_09="B0FFFF00" Color_10="B0FFFFCC"
! Define contouring levels
KMLLEVEL 1000 500 100 50 10 1 0.1 0.01 0.001 0.0001 ABSOLUTE
! Choose a plot case
CASE ID="C5-49-Rur"
YEAR     1949
PERIOD  "total"
ORGAN   "thyroid"
FILEBASE "Best"
DOCNAME "Dose (rad) C5-49-Rur"
ENDCASE
! Choose a plot case
CASE ID="C5-50-Rur"
YEAR     1950
PERIOD  "total"
ORGAN   "thyroid"
FILEBASE "Best"
DOCNAME "Dose (rad) C5-50-Rur"
ENDCASE
! Number of realizations
REALIZAT 1
! Number of nodes used in CiderF
NODENUM 516
! Node definitions used in AirCombGrid, DESCARTES and CiderF
NODE NUMBER=1 LAT=55.1000 LON=60.0000
NODE NUMBER=2 LAT=55.1000 LON=60.1000
NODE NUMBER=3 LAT=55.1000 LON=60.2000
...
NODE NUMBER=514 LAT=55.9500 LON=61.9000
NODE NUMBER=515 LAT=55.9500 LON=62.0000
NODE NUMBER=516 LAT=55.9500 LON=62.1000
END
```

The CiderView code can only uses dose result files from the CiderF code that contain map mode cases. Excerpts from a file produced by CiderF are provided in Table 9.7 so another example is not provided in this section. Multiple map cases can be contained in the same file. The file is structured in comma separated variables format.

As an option, the CiderView code can also include an external polygon file that is in KML format. Use of this file will cause Google Earth to display the polygon on the dose map. Typically, the use of this optional file for dose reconstruction purposes is to highlight the source region.

Description of the format of the input polygon is beyond the scope of this document. The interested user is directed to on-line sources (Google 2012) for a full description of the possible tags in this file. However, an example file is provided in Table 10.2. The lines between the <coordinates> and </coordinates> tags contain the longitude, latitude, altitude triplets defining the vertices of a closed polygon. The user can modify the coordinates on these lines, and even add additional lines, to define a polygon of their choice. No embedded spaces are allowed in the data lines for the coordinates. The <color> tags can be used to change the color of the polygon. In this example, the polygon is black, with a slight amount of transparency.

Table 10.2 Example Blank Polygon File for CiderView

```
<Placemark>
  <Style id="conc0">
  <LineStyle>
    <color>AA000000</color>
  </LineStyle>
  <PolyStyle>
    <color>AA000000</color>
    <outline>0</outline>
  </PolyStyle>
  </Style>
  <name>Mayak</name>
  <styleUrl>#conc0</styleUrl>
  <Polygon>
    <extrude>1</extrude>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates>
          60.777,55.700,0
          60.791,55.690,0
          60.792,55.685,0
          60.800,55.6855,0
          60.809,55.687,0
          60.809,55.689,0
          60.814,55.693,0
          60.818,55.697,0
          60.815,55.705,0
          60.809,55.710,0
          60.799,55.712,0
          60.791,55.709,0
          60.776,55.706,0
          60.776,55.702,0
          60.777,55.700,0
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</Placemark>
```

10.2.2 Output Files

The CiderView program writes a report file and one or more KML files for viewing in Google Earth. A CiderView report file for a run that generated KML files for 3 cases is provided in Table 10.3. Separate KML files containing dose maps are generated for each case at three dose levels (5th percentile, median, 95th percentile).

Table 10.3 Example Report File Written by CiderView

```
                          CiderView  Version 1.01.002
                         Last Modified on 24 May 2013
            ----------------------------------------------------
            View Dose Data Produced by the CiderF Code on a Map
            ----------------------------------------------------

       Current Run ID = 20130524090812   User Name = Paul W. Eslinger
           System Date = 05-24-2013   System Time = 09:08:12.751


           The software used to generate this output is experimental
              and has not been formally tested or peer reviewed.


======================== Echo of the Problem Definition ========================

User:  Paul W. Eslinger

Input Keyword File Name   : "Child5View.kwd"
File Name for this file   : "Child5View.rpt"

Data ranges across nodes for case C5-49-Rur
    1.43116E+00 Minimum
    1.70346E+03 Maximum
Statistic = PCT05   KML file: Best_C5-49-Rur_thyroid_total_1949_PCT05.kml
Statistic = MEDIAN  KML file: Best_C5-49-Rur_thyroid_total_1949_MEDIAN.kml
Statistic = PCT95   KML file: Best_C5-49-Rur_thyroid_total_1949_PCT95.kml

Data ranges across nodes for case C5-50-Rur
    4.35104E-01 Minimum
    6.02916E+02 Maximum
Statistic = PCT05   KML file: Best_C5-50-Rur_thyroid_total_1950_PCT05.kml
Statistic = MEDIAN  KML file: Best_C5-50-Rur_thyroid_total_1950_MEDIAN.kml
Statistic = PCT95   KML file: Best_C5-50-Rur_thyroid_total_1950_PCT95.kml

Data ranges across nodes for case C5-51-Rur
    7.61943E-01 Minimum
    8.29578E+02 Maximum
Statistic = PCT05   KML file: Best_C5-51-Rur_thyroid_total_1951_PCT05.kml
Statistic = MEDIAN  KML file: Best_C5-51-Rur_thyroid_total_1951_MEDIAN.kml
Statistic = PCT95   KML file: Best_C5-51-Rur_thyroid_total_1951_PCT95.kml

Message originating in routine CiderView at 05/24/2013 on 09:08:58.692
Message: Normal Termination
```

A run of the CiderView program can generate a large number of KML files. The number of output files depend on selections for the CASE keyword (see Section 10.3.1) and the KMLSTATS keyword (see Section 10.3.14). Specific output KML file names are generated in CiderView based on the following rules. The file names are also provided in the report file (see Table 10.3). A KML file name is formed by concatenating the following pieces:

- The optional base file name,
- the case ID,
- the organ used in the dose calculation,
- the time period for the dose (annual or total),
- the year for the dose,
- the selected statistics level, and
- the extension ".kml".

An example plot generated by Google Earth from a CiderView-produced KML file is provided in Figure 10.1. This figure is included for illustration purposes only and does not represent real conditions. This figure shows the effect of the legend and polygon options. The polygon outline is the small irregularly-shaped dark-colored region near the center of the figure.

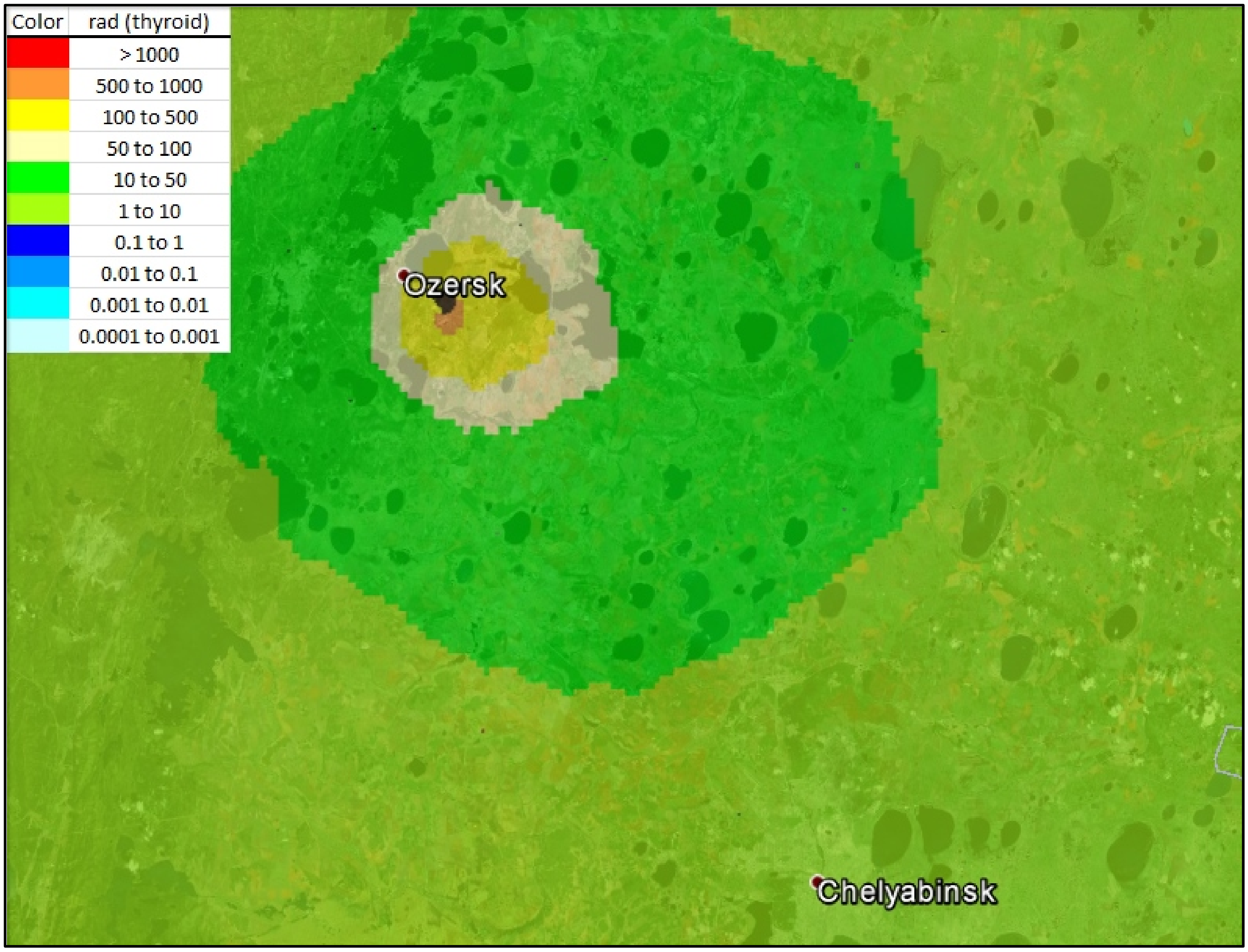


Figure 10.1 Example Plot from Google Earth using a KML File from the CiderView Code

10.3 Keywords

The keywords for the CiderView code can generally be entered in any order. However, the last keyword in the file must be the END keyword. All of the keywords used in the CiderView code are identified in alphabetical order in Table 10.4.

Table 10.4 Summary of Keywords Used in the CiderView Code

Keyword	Section	Purpose
CASE	10.3.1	The CASE keyword is used to select a map mode case computed by CiderF for plotting.
DOCNAME	10.3.2	The DOCNAME keyword is used to define the KML "document name" that will display in the navigation pane when the KML file is opened in Google Earth for a specific case.
END	10.3.3	The END keyword signifies the end of all keyword data.
ENDCASE	10.3.4	The ENDCASE keyword signifies the end of the definition for a dose case.
FILE	10.3.5	The FILE keyword is used to enter the names of input and output files.
FILEBASE	10.3.6	The optional FILEBASE keyword is used to provide a base portion of the output KML files for a specific case.
INTERPOL	10.3.7	The optional INTERPOL keyword is used to modify the default interpolation method used to map the dose data onto a regularly spaced grid for the contouring algorithm.
KMLBLANK	10.3.8	The optional KMLBLANK keyword is used to identify an input polygon file in KML format. Use of this file will cause Google Earth to display the polygon on the dose map.
KMLCOLOR	10.3.9	The optional KMLCOLOR keyword is used to modify the default colors for contours in a KML file.
KMLGRID	10.3.10	The KMLGRID keyword is used to define the grid used for outputting contours to the KML file.
KMLLEGEN	10.3.11	The optional KMLLEGEN keyword is used to identify the graphics file that will be displayed as a plot legend when the KML file is opened in Google Earth.
KMLLEVEL	10.3.12	The KMLLEVEL keyword is used to identify the contour levels in the output KML file.
KMLLOGO	10.3.13	The optional KMLLOGO keyword is used to identify a graphics file that will display in the upper right corner of the screen when the KML file is opened in Google Earth.
KMLSTATS	10.3.14	The KMLSTATS keyword is used to define the type of dose levels (by statistic) that are output to the KML contour files.

Keyword	Section	Purpose
NODE	10.3.15	Multiple NODE keywords are used to define the nodes that were used in the DESCARTES and CiderF codes.
NODENUM	10.3.16	The NODENUM keyword is used to define the total number of nodes that were used in the DESCARTES and CiderF codes.
ORGAN	10.3.17	The ORGAN keyword is used to select the type of dose result to plot.
PERIOD	10.3.18	The PERIOD keyword is used to select daily, annual or total doses from a result file written by the CiderF code.
REALIZAT	10.3.19	The REALIZAT keyword defines the number of realizations that were used in the CiderF runs that generated the data to be viewed.
USER	10.3.20	The USER keyword is used to identify the user of the program.
VERBOSE	10.3.21	The optional VERBOSE keyword can be used to direct run progress information to standard output.
YEAR	10.3.22	The YEAR keyword is used to select a single year within a dose case for output to a KML file.

10.3.1 CASE Keyword

The CASE keyword is used to select a map mode case computed by CiderF for plotting. The keywords associated with a CiderView case block are identified in Table 10.5. The first keyword in a case definition is the CASE keyword and the last keyword in a case definition is an ENDCASE keyword. The other keywords identified in Table 10.5 have no meaning outside a pair of CASE and ENDCASE keywords.

Table 10.5 Summary of Keywords Associated With a CASE Definition in CiderView

Keyword	Purpose
CASE	The CASE keyword begins the definition of a specific dose case.
DOCNAME	The DOCNNAME keyword is used to enter descriptive information in the output KML file.
FILEBASE	The optional FILEBASE keyword is used in constructing names for output KML files.
ORGAN	The ORGAN keyword is used to identify the organ for which doses are to be plotted. See Table 9.1 for the tokens used to define organ selection.
PERIOD	The PERIOD keyword is used to identify the time period for doses in the dose result file. The options are “annual” and “total” and they are case sensitive.
YEAR	The YEAR keyword is used to select the year for which doses are to be plotted.
ENDCASE	The ENDCASE keyword terminates the definition of a specific dose case.

The following set of keywords illustrates the definition of a plot case using both CASE and ENDCASE keywords:

```
CASE ID="C5-49-Rur"
YEAR     1949
PERIOD  "total"
ORGAN   "thyroid"
FILEBASE "Best"
DOCNAME "Dose (rad) Rural in 1949"
ENDCASE
```

The following is the CASE keyword syntax:

```
CASE ID="quote"
```

The quote string associated with the ID modifier is up to 10 characters in length. It is used to select data from a CiderF output result file and it must match exactly with a case ID used in CiderF. The following keyword entry selects a case with ID "C5-49-Rur".

```
CASE ID="C5-49-Rur"
```

10.3.2 DOCNAME Keyword

The DOCNAME keyword is used to define the KML "document name" that will display in the navigation pane when the KML file is opened in Google Earth for a specific case. The DOCNAME keyword is only used in the context of a dose case (see Section 10.3.1). The following is this keyword's syntax:

```
DOCNAME "quote"
```

The document name is entered in a quote string. Up to 256 characters can be entered, but short names are recommended. The following keyword example identifies that the file contains doses for a specific dose case from CiderF.

```
DOCNAME "Dose (rad) C5-49-Rur"
```

10.3.3 END Keyword

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The following is this keyword's syntax:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

10.3.4 ENDCASE Keyword

The ENDCASE keyword signifies the end of the definition for a dose case. The following is this keyword's syntax:

```
ENDCASE
```

There are no modifiers or quote strings associated with the ENDCASE keyword. The CASE and ENDCASE keywords must be entered in pairs.

10.3.5 FILE Keyword

The FILE keyword is used to enter the names of input and output files. The following is this keyword's syntax:

```
FILE [modifier1 "quote1"] (modifier2 "quote2")
```

The file names are entered in quote strings. Path names up to 256 characters long are supported. The file name associated with a modifier must be entered before the next modifier is entered. At least one FILE keyword is required for every run of the code. The modifiers associated with the FILE keyword are described in Table 10.6.

Table 10.6 Modifiers Associated with the FILE Keyword for CiderView

Modifier	Description
REPORT	The file associated with the REPORT modifier contains limited information about the code run. This file also contains any information about errors encountered by the code. The file uses ASCII text formatting and is designed for human readability. This file is required.
DOSE	The file associated with the DOSE modifier contains doses computed by CiderF for one or more cases. This file is required.

The following two entries define the report file and the dose file produced by CiderF.

```
FILE REPORT “Child5Best.rpt”
FILE DOSE “Child5Best.csv”
```

10.3.6 FILEBASE Keyword

The optional FILEBASE keyword is used to provide a base portion of the output KML files for a specific case. The FILEBASE keyword is only used in the context of a dose case (see Section 10.3.1). The following is this keyword's syntax:

```
FILEBASE "quote"
```

The base portion of the file name is entered in a quote string. Up to 256 characters can be entered, but short entries are recommended because the entire file name is limited to 256 characters. Consider the use of the following keyword:

```
FILEBASE "BaseOfName_"
```

Suppose that an output file from CiderView is named “C5-50-Rur_thyroid_total_1950_MEDIAN.kml” when the FILEBASE keyword is not used. Use of the above keyword would change the output file name to “BaseOfName_C5-50-Rur_thyroid_total_1950_MEDIAN.kml**”.**

10.3.7 INTERPOL Keyword

The optional INTERPOL keyword is used to modify the default interpolation method used to map the dose data onto a regularly spaced grid for the contouring algorithm. The following is this keyword's syntax:

```
INTERPOL [NEAREST|DISTANCE] (IGNORE=N1)
```

The default method interpolation uses inverse distance weighting (squared distance) on only the four input data points closest to the contouring point. This method is selected by entering the modifier DISTANCE. As an alternative, a closest neighbor algorithm is also provided and it can be selected using the NEAREST modifier. No interpolation is done in the nearest neighbor algorithm. If a point on the contouring grid is nearly coincident with an input data point, then both algorithms return the data value rather than an interpolated value.

The numerical value associated with the optional modifier SCALE implements a further distance restriction. The numerical value contains a distance with units of kilometers. Data points nominally selected for inclusion in the interpolation that are further than N1 km from the point being interpolated are excluded from the interpolation. Small values can lead to situations where no data values are retained in the interpolation algorithm. An interpolated value of zero is returned in this situation. Large values can result in the nearest neighbor being a long distance from the interpolated point, especially if the contour grid is larger than the span of the input data. For the application using the grid shown in Figure 1.2, a value of about 7.5 km is recommended, and it is the default value in the code.

The INTERPOL keyword can be omitted if the default algorithm is desired. The following example keyword would implement the default algorithm:

```
INTERPOL DISTANCE IGNORE=7.5
```

10.3.8 KMLBLANK Keyword

The optional KMLBLANK keyword is used to identify an input polygon file in KML format. Use of this file will cause Google Earth to display the polygon on the dose map.

```
KMLBLANK “quote”
```

The quote string on this keyword contains the file name. The full pathname, including the drive letter, should be used. An example polygon file is provided in Table 10.2. An example use of this keyword is the following:

```
KMLBLANK “d:\mayak\best\polygon.html”
```

10.3.9 KMLCOLOR Keyword

The optional KMLCOLOR keyword is used to modify the default colors for contours in a KML file. Up to 10 contours may be used. The following is this keyword's syntax:

```
KMLCOLOR modifier1 “quote1” ... (modifier10 “quote10”)
```

The modifiers associated with the KMLCOLOR keyword are described in Table 10.7.

Table 10.7 Modifiers Associated with the KMLCOLOR Keyword for CiderView

Modifier	Description
COLOR_01	The quote string associated with the COLOR_01 modifier contains the information to set the opacity and color for contour level 1 (the largest values). The quote string must be 8 characters long and contain four hexadecimal numbers. The first value is the opacity of the contour. A value of “B3” is recommended. Smaller values will be less opaque and show more of the underlying map layer. The second value is the intensity of the color blue. The third value is the intensity of the color green. The fourth value is the intensity of the color red. An example for a red contour is “B30000FF”. An example for a green contour is “B300FF00”.
COLOR_02	The quote string associated with the COLOR_02 modifier contains the information to set the opacity and color for contour level 2.
COLOR_03	The quote string associated with the COLOR_03 modifier contains the information to set the opacity and color for contour level 3.
COLOR_04	The quote string associated with the COLOR_04 modifier contains the information to set the opacity and color for contour level 4.
COLOR_05	The quote string associated with the COLOR_05 modifier contains the information to set the opacity and color for contour level 5.
COLOR_06	The quote string associated with the COLOR_06 modifier contains the information to set the opacity and color for contour level 6.
COLOR_07	The quote string associated with the COLOR_07 modifier contains the information to set the opacity and color for contour level 7.
COLOR_08	The quote string associated with the COLOR_08 modifier contains the information to set the opacity and color for contour level 8.
COLOR_09	The quote string associated with the COLOR_09 modifier contains the information to set the opacity and color for contour level 9.
COLOR_10	The quote string associated with the COLOR_10 modifier contains the information to set the opacity and color for contour level 10 (smallest values).

The following keyword entry redefines all ten default contour colors.

```
KMLCOLOR Color_01="B30000FF" Color_02="B33399FF"
   Color_03="B300FFFF" Color_04="B3B4FFFF" Color_05="B300FF00"
   Color_06="B310FFA8" Color_07="B3FF0000" Color_08="B3FF9900"
   Color_09="B3FFFF00" Color_10="B3FFFFCC"
```

The following keyword entry redefines the default colors only for contour s 2 and 4.

```
KMLCOLOR Color_02="B33399FF" Color_04="B3B4FFFF
```

10.3.10 KMLGRID Keyword

The KMLGRID keyword is used to define the grid used for outputting contours to the KML file. The following is this keyword’s syntax:

```
KMLGRID LATMIN=N1 LATMAX=N2 LONMIN=N3 LONMAX=N4 LATDEL=N
```

The modifiers associated with the KMLGRID keyword are described in Table 10.8.

Table 10.8 Modifiers Associated with the KMLGRID Keyword for CiderView

Modifier	Description
LATMIN	The numerical value associated with the LATMIN modifier identifies the minimum latitude (decimal degrees) for the grid.
LATMAX	The numerical value associated with the LATMIN modifier identifies the maximum latitude (decimal degrees) for the grid.
LATDEL	The numerical value associated with the LATDEL modifier identifies the latitude spacing (decimal degrees) for the grid.
LONMIN	The numerical value associated with the LONMIN modifier identifies the minimum longitude (decimal degrees) for the grid.
LONMAX	The numerical value associated with the LONMIN modifier identifies the maximum longitude (decimal degrees) for the grid.

The grid used for determining contours in the output KML file for display in Google Earth is defined differently than the grid used by the AirCombGrid, DESCARTES and CiderF codes. It is a rectangular region specified by minimum and maximum latitude and longitude values. It has a latitude spacing defined by the LATDEL modifier. The longitude spacing is internally calculated so the east-west grid spacing (in km) is nearly equal to the north-south (latitude) grid spacing. An example KMLGRID keyword entry is the following:

```
KMLGRID LATMIN=53.5 LATMAX=57.5 LATDEL=0.025 LONMIN=57.0 LONMAX=65 LONDEL=0.025
```

10.3.11 KMLLEGEN Keyword

The optional KMLLEGEN keyword is used to identify the graphics file that will be displayed as a plot legend when the KML file is opened in Google Earth. The following is this keyword's syntax:

```
KMLLEGEN "filename"
```

The file name is entered in a quote string. Path names up to 256 characters long (name length limitation in Windows 7) are supported. The full pathname, including the drive letter, should be used.

Legends are not required, but they enhance the interpretation of the map. A legend is included in the KML file if this keyword is used to point to a file on the user's hard drive that contains the graphic legend in *.gif or *.jpg format. This legend file must be prepared using an external program. The following example keyword identifies a graphics file as a legend file, assuming that the file name is Conc_Contours.gif.

```
KMLLEGEN "H:\Mayak\Best\Conc_Contours.gif"
```

10.3.12 KMLLEVEL Keyword

The KMLLEVEL keyword is used to identify the contour levels in the output KML file. The following is this keyword's syntax:

```
KMLLEVEL N1 N2 (N3) ... (N10) [ABSOLUTE |RELATIVE]
```

This keyword defines the contouring levels in terms of specific values (modifier ABSOLUTE) or fractions of the maximum data value (modifier RELATIVE). Data values below the minimum contour level will not show on the contour plot. Data values at or above the highest contour level will show in the highest contour. At least one contour level must be entered. A maximum of ten contours can be defined.

An example KMLLEVEL keyword that chooses specific contour levels is the following:

```
KMLLEVEL 100 50 10 5 1 0.5 0.1 0.05 0.01 0.001 ABSOLUTE
```

The report file (see Table 10.3) contains information on the range of input data. An example KMLLEVEL keyword that uses nine contour levels based on a relative fraction of the maximum data value is the following:

```
KMLLEVEL 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 RELATIVE
```

10.3.13 KMLLOGO Keyword

The optional KMLLOGO keyword is used to identify a graphics file that will display in the upper right corner of the screen when the KML file is opened in Google Earth. This keyword has no effect and is not needed if a KML file is not written. The following is this keyword's syntax:

```
KMLLOGO "filename"
```

The file name is entered in a quote string. Path names up to 256 characters long (name length limitation in Windows 7) are supported. A path name, including the drive letter, must be used for the file name if the output KML file resides on a different drive than the drive where Google Earth is installed.

The following example keyword identifies a graphics file that will be displayed as a logo: A full pathname, including the drive letter, is recommended.

```
KMLLOGO "D:\Fortran\HEDR\AirCombGridView\Test\PNNL_Color_Logo_Vertical.jpg"
```

10.3.14 KMLSTATS Keyword

The KMLSTATS keyword is used to define the type of dose levels (by statistic) that are output to the KML contour files. The definition provided by this keyword applies to all cases defined in the CiderView keyword file. The following is this keyword's syntax:

```
KMLSTATS (MINIMUM) (PCT0)1 (PCT05) (PCT10) (PCT25) (MEDIAN) (PCT75) (PCT90) (PCT95)
   (PCT99) (MAXIMUM) (MEAN) (STDEV)
```

The modifiers associated with the KMLSTATS keyword are described in Table 10.9.

Table 10.9 Modifiers Associated with the KMLSTATS Keyword for CiderView

Modifier	Description
MINIMUM	Presence of the MINIMUM modifier will cause generation of a KML file containing contours based on the minimum dose across all realizations in the dose result file.
PCT01	Presence of the PCT01 modifier will cause generation of a KML file containing contours based on the 1st percentile of dose across all realizations in the dose result file.
PCT05	Presence of the PCT05 modifier will cause generation of a KML file containing contours based on the 5th percentile of dose across all realizations in the dose result file.
PCT10	Presence of the PCT10 modifier will cause generation of a KML file containing contours based on the 10th percentile of dose across all realizations in the dose result file.
PCT25	Presence of the PCT25 modifier will cause generation of a KML file containing contours based on the 25th percentile of dose across all realizations in the dose result file.
MEDIAN	Presence of the MEDIAN modifier will cause generation of a KML file containing contours based on the median dose across all realizations in the dose result file.
PCT75	Presence of the PCT75 modifier will cause generation of a KML file containing contours based on the 75th percentile of dose across all realizations in the dose result file.
PCT90	Presence of the PCT90 modifier will cause generation of a KML file containing contours based on the 90th percentile of dose across all realizations in the dose result file.
PCT95	Presence of the PCT95 modifier will cause generation of a KML file containing contours based on the 95th percentile of dose across all realizations in the dose result file.
PCT99	Presence of the PCT99 modifier will cause generation of a KML file containing contours based on the 99th percentile of dose across all realizations in the dose result file.
MAXIMUM	Presence of the MAXIMUM modifier will cause generation of a KML file containing contours based on the maximum dose across all realizations in the dose result file.
MEAN	Presence of the MEAN modifier will cause generation of a KML file containing contours based on the mean (average) dose across all realizations in the dose result file.
STDEV	Presence of the STDEV modifier will cause generation of a KML file containing contours based on the standard deviation of doses across all realizations in the dose result file.

The CiderF code can produce stochastic dose results over the entire dose domain. Plotting of dose contours requires selection of a single value at each point in the domain. Selection of single points is based on statistical descriptions of the data.

In the special case where only one realization is computed, the dose values associated with the STDEV modifier will be zero and all of the others will yield the single realization value. Therefore, the following keyword entry is recommended for single realization dose files:

```
KMLSTATS MEDIAN
```

Multiple contour plots can be generated for a single case by entering multiple modifiers on the KMLSTATS keyword. The following keyword entry will generate contour plots for the 5th percentile of doses, the median dose, and the 95th percentile of doses:

```
KMLSTATS MEDIAN PCT05 PCT95
```

10.3.15 NODE Keyword

The NODE keyword is used to define the nodes that were used in the DESCARTES and CiderF codes. The following is this keyword's syntax:

```
NODE NUMBER=N1 LAT=N2 LON=N3
```

The NODE keyword for CiderView has the same definition as the NODE keyword for AirCombGrid (see Section 6.3.7). The modifiers associated with the NODE keyword are described in Table 10.10.

Table 10.10 Modifiers Associated with the NODE Keyword for CiderView

Modifier	Description
LAT	The numerical value associated with the LAT modifier identifies the latitude (decimal degrees) for the node.
LON	The numerical value associated with the LON modifier identifies the longitude (decimal degrees) for the node.
NUMBER	The numerical value associated with the NUMBER modifier identifies the node number. The node number must be a unique integer in the range of 1 to the number of nodes identified with the NODENUM keyword.

A separate NODE keyword is used for defining each node. The following keywords illustrate the use of the NODE keyword in defining ten nodes.

```
NODE NUMBER=1 LAT=55.1000 LON=60.0000
NODE NUMBER=2 LAT=55.1000 LON=60.1000
NODE NUMBER=3 LAT=55.1000 LON=60.2000
NODE NUMBER=4 LAT=55.1000 LON=60.3000
NODE NUMBER=5 LAT=55.1000 LON=60.4000
NODE NUMBER=6 LAT=55.1000 LON=60.5000
NODE NUMBER=7 LAT=55.1000 LON=60.6000
NODE NUMBER=8 LAT=55.1000 LON=60.7000
NODE NUMBER=9 LAT=55.1000 LON=60.8000
NODE NUMBER=10 LAT=55.1000 LON=60.9000
```

10.3.16 NODENUM Keyword

The NODENUM keyword is used to define the total number of nodes that were used in the DESCARTES and CiderF codes. The following is this keyword's syntax:

```
NODENUM N1
```

The number of nodes must be entered as an integer. The following keyword identifies that a total of 516 nodes were be used.

```
NODENUM 516
```

10.3.17 ORGAN Keyword

The ORGAN keyword is used to select the type of dose result to plot. The ORGAN keyword is only used in the context of a dose case (see Section 10.3.1). The following is this keyword's syntax:

```
ORGAN “quote”
```

The quote string must contain one of the organ tokens identified in Table 9.1. An example keyword for plotting whole body doses is the following:

```
ORGAN “whole_body”
```

An example keyword for plotting thyroid doses is the following:

```
ORGAN “thyroid”
```

10.3.18 PERIOD Keyword

The PERIOD keyword is used to select daily, annual or total doses from a result file written by the CiderF code. The PERIOD keyword is only used in the context of a dose case (see Section 10.3.1). The following is this keyword's syntax:

```
PERIOD “quote”
```

The quote string entered on this keyword can take two possible values. The string “annual” selects annual doses calculated by the CiderF code. The string “total” selects total doses calculated by the CiderF code.

The following keyword selects total dose from the result file written by CiderF:

```
PERIOD “total”
```

The following keyword selects annual dose from the result file written by CiderF:

```
PERIOD “annual”
```

10.3.19 REALIZAT Keyword

The REALIZAT keyword defines the number of realizations that were used in the CiderF runs that generated the data to be viewed. The following is this keyword's syntax:

```
REALIZAT value1
```

The integer value1 has a minimum value of 1 and a maximum of the number of realizations used in the CiderF code to define the set of dose data. If the number of realizations is less than the number of realizations in the concentration data files, then only the first realizations will be used.

The following keyword record sets the number of realizations to 1:

```
REALIZAT 1
```

10.3.20 USER Keyword

The USER keyword is used to identify the user of the program. The user name will be written to output files. If the user name is not supplied, then the program will terminate after writing an error message. The following is this keyword’s syntax:

```
USER "quote"
```

The user name is entered in a quote string. User names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code:

```
USER "John Q. Public"
```

There are no modifiers associated with the USER keyword.

10.3.21 VERBOSE Keyword

The optional VERBOSE keyword can be used to direct run progress information to standard output. The following is this keyword’s syntax:

```
VERBOSE (MEDIA)
```

Presence of the VERBOSE keyword causes more run time information to be written to the standard output device than if the keyword is omitted. The additional information can sometimes be useful in estimating the completion time of a lengthy set of calculations. The following is an example use of the VERBOSE keyword:

```
VERBOSE
```

10.3.22 YEAR Keyword

The YEAR keyword is used to select a single year within a dose case for output to a KML file. The YEAR keyword is only used in the context of a dose case (see Section 10.3.1). The following is this keyword’s syntax:

```
YEAR N1
```

The year is entered as an integer. The following keyword example selects data for the year 1952 for output to the KML file.

```
YEAR 1952
```

11.0 Keyword Language Syntax

The general purpose of a keyword file is to provide flexible control to a computer program using human readable inputs. Each line of a keyword data file is parsed into numeric or character data. These data are interpreted to set up control information and define input parameters.

Every line of the input keyword file is considered a keyword record, a continuation record, or a comment record. Keyword records contain a keyword beginning in column 1. The keyword is used to determine the purpose of the subsequent data. Continuation records are used when a keyword record requires too much data to be placed on one line in the file. Comment lines are ignored by the reading software but they are useful for annotating the input file.

The information from each keyword record and subsequent continuation lines is moved into storage arrays. Data that can be deciphered as numeric values are placed in a numeric array. Other data are classified as “secondary keywords” (called “modifiers”), “quote strings” or even discarded. Secondary keywords are stored as character images in an array. All such keywords or modifiers read from the input file are changed to uppercase before being stored. Quote strings are text strings enclosed in double quotation marks. These are stored exactly as they are read from the input file. An input line can contain up to 2048 characters of information. Individual quote strings are limited to 512 characters in length.

11.1 Keyword Records

Keyword records start in column 1 with any letter from A to Z, in either uppercase or lowercase. The first eight characters of a keyword are converted to upper case and stored in a variable. Any additional characters in the keyword are discarded. For example, the keyword entries REALIZAT, REALIZATIO, and REALIZATIONS all are stored as the keyword REALIZAT. All subsequent lines of text that do not have an alphabetic character or comment character in column 1 are treated as continuation lines. The following is an example keyword record (where SAMPLEKEY starts in column 1):

```
SAMPLEKEY 2 0 500 1 100
```

The character sequence SAMPLEKE (first 8 characters) is the keyword. The numbers 2, 0, 500, 1, and 100 are numeric data.

11.2 Continuation Records

Continuation records start with any valid separator character (but not double quotation marks). These are treated as additional data to the previous keyword record. Valid separator characters are identified in Section 11.4.3. The combined data on a keyword line and on subsequent continuation lines are treated as a single block of information. All numeric values and character strings on those lines are used as input data relevant to the keyword of the keyword line. The two following keyword entries contain the same information:

```
SAMPLEKEY 2 0 500 1 100
SAMPLEKEY 2 0
      500 1 100
```

11.3 Comment Records

Any line with the character ! in column 1 will be treated as a comment record. These lines are ignored by the keyword reading software. The ! character also signifies an in-line comment if it is not in column 1. Any information on a keyword line that follows a ! will be ignored. The following example identifies a comment line:

```
!This entire line is a comment
```

The following keyword contains an in-line comment:

```
SAMPLEKEY 3 4.0 5.0 ! Trailing information is ignored by the software
```

A ! character entered within a quote string does not denote an in-line comment. In the following keyword example the quote string associated with the ID modifier will contain “SpecDiet01!” rather than “SpecDiet01”.

```
DIET ID="SpecDiet01!" BEGIN="1948-06-01" FINISH="1972-12-31"
```

11.4 Input Data Handling

Each line of a keyword file is parsed into tokens. Individual tokens are separated by one or more data separator characters. The tokens are then processed into one of the following five classes:

- Keywords – keywords are tokens beginning with an alphabetic character that start in column 1 of the input line
- Modifiers – modifiers are tokens beginning with a alphabetic character, but they do not start in column 1 of the input line
- Numerical data – numerical data are tokens that can be interpreted as numbers
- Quote strings – quote strings are tokens that start and end with the double quotation (“) symbol
- Ignored – tokens that do not fall into one of the above four categories are discarded.

11.4.1 Data Separators

Keywords, numeric data, and secondary keywords are separated by any of the following seven entries: space character, comma, equals sign, colon, semicolon, left parenthesis and right parenthesis. As an illustration of the use of separator characters, the following keyword records all contain and convey the same information:

```
SAMPLEKEY 3 4.5 5.6 6.7
SAMPLEKEY 3 (4.5,5.6,6.7)
SAMPLEKEY 3 ( 4.5=5.6(6.7 )
SAMPLEKEY 3:4.5  5.6;6.7
```

Double quotation marks are used differently than other separators. They indicate text strings that are stored without conversion.

11.4.2 Keyword Tokens

Tokens that start in column 1 with any letter from A to Z in either uppercase or lowercase are classified as keywords. The first eight characters of a keyword are converted to upper case and stored. Keywords fewer than eight characters long are left justified and blank-filled. Any trailing characters are discarded. For example, the following two keyword lines define the same REALIZAT keyword.

```
REALIZAT 250
REALIZATION 250
```

11.4.3 Modifier Tokens

Tokens beginning with any letter from A to Z in either uppercase or lowercase are classified as modifiers if the token does not begin in column 1 of the input line. The first eight characters of a modifier are converted to uppercase, where necessary, and stored in an array. Modifiers fewer than eight characters long are left justified and blank-filled.

11.4.4 Numerical Value Tokens

Any token containing data that can be processed without error using a FORTRAN read statement is classified as numerical data. The data can include a leading sign (+ or -), integer characters 0 through 9, a decimal point, and exponent indication (“E” or “e” or “D” or “d”) and an exponential sign (+ or -).

All numeric values are converted to double precision real numbers before being stored. Although double precision values are supported, generally the data are converted to integer or single precision real numbers for use in the specific application. In the context of the computer codes described in this document, the following data ranges are allowed:

- Double-precision real floating-point values ranging from 2.2250738585072013D-308 to 1.7976931348623158D+308.
- Single-precision real floating-point values ranging from 1.17549435E-38 to 3.40282347E+38.
- Signed integer values ranging from -2,147,483,648 to 2,147,483,647.

All the entries on the following keyword are valid numerical values:

```
SAMPLEKEY 3 -3 +3 4.5 5.6E-12 6.7E+21 9.33D-1 -21.0D0 6.666D23
```

11.4.5 Quote String Tokens

Tokens that are 512 characters or less in length and also start and end with the double quotation (“) symbol are classified as quote strings. A quote string contains text that is stored without modification. Quote strings typically are used for entering file names or text descriptions into a program. Quote strings must begin and end on a single line of the input file. The following is an example of quote string usage:

```
FILE “c:\apps\human\test.dat”
```

11.4.6 Data Association with Modifiers

If a numerical value or a quote string immediately follows a modifier, then that value or quote string is said to be associated with the modifier. The association is set if only valid data separators (see Section 11.4.1) are entered between the modifier and the value or quote string. The purpose of the association is to allow pairs of modifiers and numbers or quote strings on a keyword line to be grouped together conceptually. However, the association approach only applies to unique modifiers within the data for a specific keyword.

In the following example, the value 7.25 is associated with the modifier RATE, the quote string “SampleFile.docx” is associated with the modifier FILE, the value 3 is associated with the modifier TOP, and no data are associated with the modifier BOTTOM because the modifier BOTTOM is followed by the modifier WAS.

```
SAMPLEKEY RATE=7.25 FILE “SampleFile.docx” TOP:3 BOTTOM was 6
```

11.5 Description Conventions

A particular code may have a keyword with both required and optional modifiers. The convention for describing these options is as follows:

- Required modifiers can be enclosed in square brackets []. The brackets are for description only. They are not entered in the keyword file.
- Optional modifiers can be enclosed in parenthesis (). The parenthesis are for description only. They are not entered in the keyword file.
- When only one item out of a list of items is allowed, the items are separated by a vertical bar. The vertical bar is for description only. It is not entered in the keyword file.

The following example keyword has two required modifiers. Either TOP or BOTTOM can be entered, but not both. The USER modifier and the associated quote string are required.

```
SAMPLEKEY [TOP|BOTTOM] [USER=“Quote”]
```

The following two keywords are valid, given the above general description:

```
SAMPLEKEY TOP USER=“Ted”
SAMPLEKEY BOTTOM USER=“Bill”
```

The following example keyword has two required modifiers. One of the three modifiers BLUE, BLACK or RED must be entered. The EDGE modifier and the associated number are also required.

```
SAMPLEKEY [EDGE=N1] [BLUE|BLACK|RED]
```

The following two keyword entries are valid, given the above general description:

```
SAMPLEKEY EDGE=5 BLUE
SAMPLEKEY EDGE=1.243D17 RED
```

12.0 Stochastic Variable Generation

Many of the variables in CiderF and the associated utility codes can be defined as stochastic variables. The specific values for these variables are generated by a suite of statistical routines. The following major considerations apply to the generation process:

- Stratified sampling may be used when more than one realization is generated.
- Many distributions may be truncated between two limits that are specified as limits in the uniform domain on the interval 0 to 1.
- The user may specify a cumulative distribution function in the form of a table of values.
- Information about a stochastic variable is linked to a unique character ID. Access to all information about the variable is available through use of the variable ID.

This section describes stochastic variable generation for every code documented in this report except FROSTPUNO. The reader is referred to the DESCARTES User's Guide (Miley et al. 1994) for the description of stochastic variable generation in the FROSTPUNO code.

12.1 Keywords Defining Stochastic Variables

Stochastic variables are defined by entering data for the variable with the STOCHAST keyword. The following is this keyword's syntax:

```
STOCHAST [ID="Quote1"] [TITLE="Quote2"] [UNITS="quote3"] [DIST="Quote4"]
   PARAM N1,…Nn (TRUNCATE U1 U2)
```

The following information applies to the STOCHAST keyword:

- The quote string "Quote1" associated with the ID modifier must be a unique character string of up to 32 characters that will be used to identify this stochastic variable in subsequent uses. It is case sensitive and embedded spaces are significant. In many cases the ID is a combination of a variable name and other data so that it can be recreated internal to a code when stochastic data are needed.
- The quote string "Quote2" associated with the TITLE modifier is a required description for the stochastic variable up to 72 characters long. This information is used for output labeling purposes.
- The quote string "Quote3" associated with the UNITS modifier contains the units of the stochastic variable and can be up to 24 characters long. This information is used for output labeling purposes.
- The quote string "Quote4" associated with the DIST modifier must be the name of a statistical distribution. Supported distributions are defined in Table 12.1. Unlike most quote strings, the quote string associated with the DIST modifier is not case sensitive.
- A number of numerical values must follow the modifier PARAM. These numerical values identify the parameters in the statistical distribution. Different numbers of numerical values may be entered, depending on the choice of the statistical distribution.
- The additional modifier TRUNCATE can be used for a number of distribution types. If TRUNCATE is entered, it must be followed by two values in the interval 0 to 1, inclusive. The lower value must be less than the upper value. These two values specify the tail probabilities at which to impose range truncation for the distribution.

Table 12.1 Supported Statistical Distributions

DIST Value	Distribution	Truncate	Parameters Required
Constant	Constant	No	Single value
Uniform	Uniform	Yes	Lower limit, upper limit
Discrete	Discrete Uniform	No	Smallest integer, largest integer
Loguniform10	Loguniform (base 10)	Yes	Lower limit, upper limit
Loguniform	Loguniform (base e)	Yes	Lower limit, upper limit
Triangular	Triangular	Yes	Lower limit, mode, upper limit
Normal	Normal	Yes	Mean, standard deviation
Lognormal10	Lognormal (base 10)	Yes	Mean of logarithms, standard deviation of logarithms
Lognormal	Lognormal (base e)	Yes	Mean of logarithms, standard deviation of logarithms
User	User Defined	Yes	Number of pairs, data for pairs of values (Prob(X_i),X_i)
Beta	Beta	Yes	Alpha (exponent for x), beta (exponent for (1-x)), lower limit, upper limit
Weibull	Weibull	Yes	Scale, Exponent, Shift
Cauchy	Cauchy	Yes	Median, Scale parameter
Exponential	Exponential	Yes	Shift, Divisor
Gamma	Gamma	Yes	Exponent on X, Parameter in the exponential
Logistic	Logistic	Yes	Mean, Scale parameter
Binomial	Binomial	No	Probability of success
Extreme	Extreme value (Gumbel)	Yes	Shift, Divisor
Pareto	Pareto	Yes	Shift, Exponent
Power	Power Function	Yes	Limit, Exponent
Laplace	Laplace	Yes	Shift, Divisor
Poisson	Poisson	No	Parameter
Student	Student’ t	No	Degrees of freedom
F	F distribution	No	Numerator degrees of freedom, Denominator degrees of freedom
Chisquare	Chi Squared	No	Degrees of freedom

The following is an example STOCHAST keyword for a variable assigned a constant of 234.432:

```
STOCHAST ID=“Unique1” DIST=“CONSTANT” PARAM 234.432 UNITS=“m”
    TITLE=“Define a constant distribution”
```

The constant can take any value.

The following is an example STOCHAST keyword for a variable assigned a uniform distribution on -2 to 7:

```
STOCHAST ID=“Unique2” DIST=“UNIFORM” PARAM–2.0 7 UNITS=“m/s”
    TITLE=“Uniform distribution on –2 to 7”
```

The two limits can take any values as long as the second value is strictly greater than the first value. The following is an example stochastic keyword for a variable assigned a discrete uniform distribution on the integers 6 to 70:

```
STOCHAST ID=“Unique3” DIST=“discrete” PARAM 6 70 UNITS=“d”
    TITLE=“Discrete uniform distribution on 6 to 70”
```

The two limits must be integers where the second integer is strictly greater than the first integer.

The following is an example STOCHAST keyword for a variable assigned a loguniform (base 10) distribution on the interval 10^{-7} to 10^{-3}:

```
STOCHAST ID=“Unique4” DIST= “loguniform10” 1.0E-7 1.0E-3 UNITS=“kg/mL”
    TITLE =“Define a loguniform (base 10) variable on 0.0000001 to 0.001”
```

The two limits must both be greater than zero, and the second limit must be greater than the first limit.

The following is an example STOCHAST keyword for a variable assigned a loguniform (base e) distribution on the interval 10^3 to 10^6:

```
STOCHAST ID=“Unique4” DIST= “loguniform” 1.0E3 1.0E6 UNITS=“w/sq m”
    TITLE =“Define a loguniform (base e) variable on 1000 to 1000000”
```

The two limits must both be greater than zero, and the second limit must be greater than the first limit.

The following is an example STOCHAST keyword for a variable assigned a triangular distribution with a minimum of 2, a mode of 3, and a maximum of 7:

```
STOCHAST ID=“Unique6” DIST=“triangular” PARAM 2 3 7 UNITS=“km-m/s”
    TITLE=“Triangular distribution on (2,3,7)”
```

The three values that define the triangular must all be different, and they must be entered in increasing order.

The following keyword would define a different stochastic variable from the one just entered because the identification string (Quote1) is case sensitive:

```
STOCHAST ID=“UniQue6” DIST=“triangular” PARAM 2 3 7 UNITS=“km-m/s”
    TITLE=“Triangular distribution on (2,3,7)”
```

The following keyword entry would define a lognormal (base 10) distribution where the mean and standard deviation (of the logarithms) are –2.0 and 0.5:

```
STOCHASTIC ID=“Unique8” DIST=“lognormal10” –2 0.5 UNITS= “J/K”
    TITLE “Lognormal (base 10) variable”
```

The mean value can be any number, but the standard deviation must be greater than zero.

The following keyword entry illustrates the use of the user-defined distribution (distribution type USER). This example entry uses seven pairs of values. The first pair of numbers uses a probability of 0 to define the lower limit of the distribution at 8.4E-7. The last pair of numbers uses a probability of 1 to define the upper limit of the distribution at 1.73E-6. The other values are associated with the probability levels of .025, .167, .5, .833, and .975. The probability data and distribution percentiles must be entered in strictly increasing order.

```
STOCHASTIC ID=“Sr90Con” DIST= “user” UNITS=“Mol”
   TITLE= “An example use of the user defined distribution”
   PARAM = 7
   0         8.40E-7
   2.50E-02  9.20E-7
   1.67E-01  1.06E-6
   5.00E-01  1.21E-6
   8.33E-01  1.37E-6
   9.75E-01  1.58E-6
   1         1.73E-6
```

The first pair of numbers always uses a probability of 0 to define the lower limit of the distribution. The last pair of numbers always uses a probability of 1 to define the upper limit of the distribution. The intervening pairs define probability levels and the associated data values. The probabilities must be entered in strictly increasing order.

The following keyword entry would define a beta distribution with parameters 1.1 and 2.1 on the interval (0,1):

```
STOCHASTIC ID=“Uniq” DIST= “Beta” PARAM=1.1 2.1 0.0 1.0 UNITS=“m”
   TITLE=“Beta (1.1,2.1) on the interval 0,1”
```

Let the first parameter be denoted by α and the second parameter be denoted by β. The mean of the beta distribution would be α /(α +β) if the limits were 0 and 1. Both α and β must be greater than zero. The lower limit must be less than the upper limit.

Examples of other distribution keywords are as follows:

```
STOCHAST ID="w1" DIST="Weibull" PARAM=2.0 3.0 5.0  units="yr" TITLE="Weibull"

STOCHAST ID="C2" DIST="Cauchy" PARAM=0.0 1.0 units="yr" TITLE="Cauchy"

STOCHAST ID="E3" DIST="Exponential" PARAM=10.0 5.0 units="yr" TITLE="Exponential"

STOCHAST ID="G4" DIST="Gamma"  PARAM=2.0 3.0 5.0 units="yr” TITLE="Gamma"

STOCHAST ID="44L" DIST="Logistic" PARAM=-1.0 3.0 units="yr" TITLE="Logistic"

STOCHAST ID="Bin5" DIST="Binomial" PARAM=0.25  units="yr" TITLE="Binomial"

STOCHAST ID="xtr7” DIST="Extreme" PARAM=4.0 3.0 units="yr" TITLE="Extreme value"

STOCHAST ID="P8" DIST="Pareto"  PARAM=2.0 3.0  units="yr" TITLE="Pareto variable"

STOCHAST ID="Po9" DIST="Power"  PARAM=2.0 3.0  units="yr" TITLE="Power function"

STOCHAST ID="Lap" DIST="Laplace"  PARAM=2.55 3.33 units="none" TITLE="Laplace"

STOCHAST ID="Lap-3" DIST="Laplace" PARAM=0 1 units="none"
   TRUNCATE 0.05 0.95 TITLE="Truncated Laplace variable"

STOCHAST ID="Poi-1" DIST="Poisson" PARAM=4 units="none" TITLE="Poisson (mean 4)"

STOCHAST ID="Stud-2" DIST="Student" PARAM=2 units="none" TITLE="Student 2 df"
STOCHAST ID="S-Big" DIST="Student" PARAM=5000 units="none" TITLE="Student 5000 df"
STOCHAST ID="F12,8" DIST="F" PARAM=12.0 8.0 units="None" TITLE="F(12,8)"
STOCHAST ID="ChiSq-6" DIST="ChiSquare" PARAM=6 units="none" TITLE="Chi Square 6 df"
```

12.2 Probability Concepts

Generally, random variables can be classified as continuous or discrete. In this context, the term *continuous* indicates that the random variable is defined over a continuum of values. Similarly, the term *discrete* indicates that the random variable is defined over a discrete set of values. The discrete set of values can be small, such as the integers 0 and 1, or it can be infinite. The Poisson distribution is an example of a discrete distribution that is defined on an infinite number of discrete points (all nonnegative integers).

The distribution of a continuous random variable X is completely described by its probability density function, f(x). The interpretation of the probability density function is that the area under f(x), for an interval a<x<b, equals the probability that the random variable, X, will fall in the interval (a,b), denoted P(a<X<b). One cannot make the statement P(X=t) for continuous random variables, because the area under the probability density function is zero at any given point. Two axioms of probability theory (Mood et al. 1974), p. 22, are that the probability of any event is between zero and one, and the integral of the probability density function over the entire support (the interval [L,U]) of X equals 1.

The integral of the probability density function from the lower bound L to some value x less than the upper bound U represents the probability that X will be observed in the interval (L,x). This integral operation defines the cumulative distribution function for the random variable X. The cumulative distribution function is denoted by F(x) (the capital F for the cumulative distribution function corresponds to the lowercase f for the probability density function) and mathematically is represented by the following:

$$F(x)=\int_{L}^{U} f(s)ds$$

Similarly, the distribution of a discrete random variable X is completely defined by its probability function, f(x). The cumulative distribution function, F(X), is defined in terms of a sum rather than an integral:

$$F(x)=\Pr(X \leq x)=\sum_{i} f(x_i)I\left(x_i \leq x\right)$$

where $I(x_i \leq x)$ is the indicator funtion that takes the value 1 when the argument is satisfied and takes the value 0 otherwise.

12.2.1 Generation Using the Probability Integral Transform Method

Generation of a random variable from a given distribution typically involves the use of information either about f or F. There are two philosophical approaches to generating random numbers: exact methods and approximate methods. The algorithms embedded in the codes described in this document employ exact methods. Exact methods can be further categorized into probability integral transform methods and functional methods. The probability integral transform method is used.

In the probability integral transform method, the random variable of interest is expressed as a function of a U(0,1) random variable, where U(0,1) denotes the continuous random variable ranging uniformly over the interval (0,1). It can be shown that any cumulative distribution function evaluated at a random value X (instead of being evaluated at a known value x) is distributed uniformly over the interval (0,1) (Mood et al. 1974), p. 202. Therefore, given a realization u of the U(0,1) random variable and a selected statistical distribution, one can set u=F(x) and solve to obtain $x=F^{-1}(u)$. The value x thus obtained is a random realization from the selected statistical distribution.

The inverse cumulative distribution function, $F^{-1}(u)$, is single-valued for continuous random variables if x is in the interval (L,U). In principle, one can obtain an exact solution for x given any specific cumulative distribution function and value u. There are some distributions, such as the normal, beta and gamma distributions, for which no closed-form analytical expression for F^{-1} exists, and hence approximation methods must be used.

12.2.2 Truncated Distributions

There are some situations where a user wants to sample values from a classical statistical distribution but they also wish to exclude any values above or below specific levels. For any random variable with continuous probability density function f(x) and cumulative distribution function F(x), the probability density function, under truncation to the interval (c,d), is $f_T(x) = f(x)/[F(d)-F(c)]$ and it is 0 outside the interval (c,d). The divisor [F(d)-F(c)] ensures that $f_T(x)$ integrates to unity.

The probability integral transform method supports efficient sampling from truncated distributions for continuous random variables. Suppose that a random variable originally defined on the interval (L,U) is truncated to the interval (c,d), where $L\leq c<d\leq U$. Define $u_c=F(c)$ and $u_d=F(d)$. Sampling from the truncated distribution occurs in three steps.

- Generate a value u from the uniform (0,1) distribution.
- Transform u to z using the expression $z=(u_d-u_c)u+u_c$.
- Obtain the random value for x using the equation $x=F^{-1}(z)$. The inverse cumulative distribution function in this expression is from the original distribution, not the truncated distribution.

12.2.3 Stratified Sampling

The primary purpose of stratified sampling is to achieve more evenly spaced (in a probability sense) samples from the distribution of a random variable than would result from randomly sampling over the whole range of the distribution. Stratified sampling can result in more efficient estimation of simulation results for a variety of estimators than when using simple random sampling (Iman and Conover 1982).

Stratified sampling can easily be implemented when generating random deviates using the probability integral transform method. This is accomplished by dividing the interval (0,1) of the uniform distribution into subintervals, or strata, and sampling a specified number of times within each stratum, each time obtaining the corresponding value of x. Within these codes, the strata intervals are assigned equal probability, and exactly one value is sampled within each stratum.

Two steps are used to obtain a stratified sample s_i, (i = 1, 2, ..., N) for the uniformly distributed variable S. First, generate values for s_i using the following equation:

$$s_i = \frac{(n-1)+u_i}{N}$$

where u_i is a uniformly distributed number between 0 and 1 generated using simple random sampling. These values satisfy the relationship $0<s_i<s_{i+1}<1$. In the second step, the s_i are reshuffled to a random order. This reshuffling can be achieved by generating a new sequence of uniformly distributed random numbers, a_i, (i = 1, 2, ..., N) using simple random sampling. The set of values (s_i) are then reordered so they have the same rank order as the corresponding a_i. The sorting method uses a variation of the Quicksort algorithm (Hoare 1961).

12.3 Generation Algorithms

The statistical distributions for use on STOCHAST keywords are summarized in Table 12.1. The following paragraphs describe the generation algorithms.

12.3.1 Algorithms for the Uniform Distribution

Probability Distribution Function: The probability density function for a uniform random variable on the interval (a,b) takes the form:

$$f(x) = \frac{1}{b-a}$$

for $a\underline{<}x\underline{<}b$ and is zero otherwise.

Cumulative Distribution Function: The cumulative distribution function for a uniform random variable on the interval (a,b) takes the form:

$$F(x) = \left\{ \begin{array}{cc} 0 & x < a \\ \left[\dfrac{x-a}{b-a}\right] & a \le x \le b \\ 1 & x > b \end{array} \right\}$$

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a uniform random variable on the interval (a,b) takes the form:

$$F^{-1}(u) = a + u(b-a)$$

Expected Value: The expected value of a uniform random variable on the interval (a,b) is (a+b)/2.

Variance: The variance of a uniform random variable on the interval (a,b) is $(b-a)^2/12$.

Median: The median of a uniform random variable on the interval (a,b) is (a+b)/2.

Generation Algorithm: Algorithms that generate truly random uniform numbers do not exist, although many algorithms generate pseudo-random deviates (hereafter loosely referred to as random numbers). Linear congruential methods (Park and Miller 1988) are commonly used for random number generation and this code uses a linear congruential method. The linear congruential generator generates random integers using an algorithm of the form $S_i = (A \times S_{i-1} + C) \bmod(M)$, where S_i is the i^{th} generated random integer, A and C are constants, M is the modulus of the generated integers, and mod denotes the remainder function. These integers are converted to approximate uniform (0,1) numbers by the division $U_i = S_i / M$.

The period of a sequence U_i of generated deviates is the minimal value k such that $U_i=U_{i+k}$ (this occurs independent of i for linear congruential generators). It can be shown that the period of any congruential generator does not exceed M. Therefore, if one is generating a large number of uniform random values, it is desirable that M be large. The performance of each congruential generator (each choice of A, C, and M) can thus be examined with respect to criteria proceeding from the four considerations given above.

The current implementation uses a linear congruential generator with A=16807, C=0, and M=2147483647. These choices yield a sequence (U_i) that is implementable in Fortran on a 32-bit computer without machine language coding, is sufficiently independent on an element-by-element basis, possesses a long period (cycle), and has a reasonable degree of coverage over all hypercubes of dimension less than k (Fishman and Moore 1986). The generation algorithm uses two steps;

$$S_i = A \times S_{i-1} \text{ Mod(M)}$$
$$U_i = S_i / M$$

Any value, x, generated from the uniform (a,b) distribution makes use of a value, u, from the U(0,1) distribution. The value u is first generated, and then x is evaluated as $x=F^{-1}(u)$.

12.3.2 Algorithms for the Discrete Uniform Distribution

Probability Function: The probability function of a discrete uniform random variable on integers in the interval from a to b is the following:

$$f(x) = \frac{1}{b-a+1}$$

This function is 0 for all integers less than a or greater than b.

Cumulative Distribution Function: The cumulative distribution function of a discrete uniform random variable on integers in the interval from a to b is the following:

$$F(x) = \left\{ \begin{array}{cc} 0 & x < a \\ \dfrac{1}{(b-a+1)} \sum_{i=a}^{b} I(a \leq i \leq x) & a \leq x \leq b \\ 1 & x > b \end{array} \right\}$$

Where $I(a \leq i \leq x)$ is the indicator funtion that takes the value 1 when the argument is satisfied and takes the value 0 otherwise.

Inverse Cumulative Distribution Function: The inverse cumulative distribution function of a discrete uniform random variable on integers in the interval from a to b is the following:

$$F^{-1}(u) = a + \mathrm{int}[u(b-a+1)]$$

where the int(·) function returns the integer portion of its argument.

Generation Algorithm: A random value from the discrete uniform random variable on integers in the interval from a to b is obtained by generating a value from the uniform distribution and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value of a discrete uniform random variable on integers in the interval from a to b is the following:

$$E(X) = \frac{b(b+1) - a(a+1)}{2(b-a+1)}$$

Variance: The variance of a discrete uniform random variable on integers in the interval from a to b is the following:

$$V(X) = \frac{x(x+1)(2x+1) - (a-1)a(2a+1)}{6(b-a+1)} - \left[\frac{b(b+1) - a(a+1)}{2(b-a+1)} \right]^2$$

Median: The median of a discrete uniform random variable on integers in the interval from a to b is the integer closest to (a+b)/2.

12.3.3 Algorithms for the Loguniform Distribution

Probability Density Function: The probability density function for a loguniform random variable of base b is the following:

$$f(x) = \frac{I(b^c < x < b^d)}{x(d-c)\ln(b)}$$

for $-\infty < c < d < \infty$, where I is an indicator function (0 if false, 1 if true), b is the logarithm base (either 10 or the natural constant e), and ln(b) denotes the natural logarithm of b.

Cumulative Distribution Function: The cumulative distribution function for a loguniform random variable is the following:

$$F(x) = \left\{ \begin{matrix} 0 & x < b^c \\ \left[\dfrac{\ln(x) - c\ln(b)}{(d-c)\ln(b)} \right] & x \in [b^c, b^d] \\ 1 & x > b^d \end{matrix} \right\}$$

Inverse Cumulative Distribution Function: The inverse of the cumulative distribution function for a loguniform random variable is the following:

$$F^{-1}(u) = b^u$$

Generation Algorithm: The generation algorithm first generates a value, u*, from the uniform (c,d) distribution and then evaluates $F^{-1}(u^*)$.

Expected Value: The expected value of a loguniform random variable of base b is the following:

$$E(X) = \frac{b^d - b^c}{(d-c)\ln(b)}$$

Variance: The variance of a loguniform random variable of base b is the following:

$$V(X) = \frac{b^{2d} - b^{2c}}{2(d-c)\ln(b)} - \left[\frac{b^d - b^c}{(d-c)\ln(b)} \right]^2$$

Median: The median of a loguniform random variable of base b is $b^{(c+d)/2}$.

12.3.4 Algorithms for the Triangular Distribution

Probability Distribution Function: The probability density function for a triangular random variable has the following form:

$$f(x) = \left\{ \begin{matrix} 2(x-a)/[(b-a)(c-a)] & \text{for} \quad a < x \le b \\ 2(c-x)/[(c-b)(c-a)] & \text{for} \quad b \le x < c \end{matrix} \right\}$$

and takes the value 0 elsewhere.

Cumulative Distribution Function: The cumulative distribution function for a triangular random variable has the following form:

$$F(x) = \begin{cases} 0 & x \le a \\ \dfrac{(x-a)^2}{(c-a)(b-a)} & a < x \le b \\ 1 - \dfrac{(x-c)^2}{(c-a)(c-b)} & b < x \le c \\ 1 & x \ge c \end{cases}$$

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a triangular random variable has the following form:

$$F^{-1}(u) = \begin{cases} a + \sqrt{u(c-a)(b-a)} & \text{for} \quad 0 \le u \le (b-a)/(c-a) \\ c - \sqrt{(1-u)(c-a)(c-b)} & \text{for} \quad (b-a)/(c-a) \le u \le 1 \end{cases}$$

Generation Algorithm: A random value from the triangular distribution is obtained by generating a value from the uniform distribution and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value of a random variable with the triangular distribution is (a+b+c)/3.

Variance: The variance of a random variable with the triangular distribution is

$$V(X) = \left[\frac{a^4c - a^4b + ab^4 - ac^4 - b^4c + bc^4}{6(b-a)(c-a)(c-b)} \right] - \left[\frac{a+b+c}{3} \right]^2$$

Median: The median of a random variable with the triangular distribution depends on the location of the mode b. The median is:

$$M(X) = \begin{cases} a + \sqrt{(b-a)(c-a)/2} & \text{for } b \ge (a+c)/2 \\ c - \sqrt{(c-b)(c-a)/2} & \text{for } b \le (a+c)/2 \end{cases}$$

12.3.5 Algorithms for the Normal Distribution

Probability Distribution Function: The probability density function for a normal random variable with mean μ and variance σ^2, denoted as $N(\mu,\sigma^2)$, is the following:

$$f(x) = \frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}$$

for real x, real μ, and σ>0.

Cumulative Distribution Function: There is no closed form analytic expression for the cumulative distribution function of a normal random variable. Algorithm AS 66 (Hill 1985) can be used to generate an accurate numerical approximation.

Inverse Cumulative Distribution Function: There is no closed form analytic expression for the inverse cumulative distribution function of a normal random variable. Algorithm AS 111 (Beasley and Springer 1985) can be used to generate an accurate numerical approximation.

Generation Algorithm: A value from a $N(\mu,\sigma^2)$ random variable, x, is obtained by generating a value from a normal(0,1) random variable, z, using Algorithm AS 111 (Beasley and Springer 1985). This value is further modified using the transformation $x=\mu+\sigma z$.

Expected Value: The expected value of a normal random variable is μ.

Variance: The variance of a normal random variable is σ^2.

Median: The median of a normal random variable is μ.

12.3.6 Algorithms for the Lognormal Distribution

The logarithm of a random variable that is lognormally distributed is a normal $N(\mu,\sigma^2)$ random variable (thus the name lognormal).

Probability Density Function: The probability density function of a lognormal random variable is the following:

$$f(x) = \frac{A}{x\sigma\sqrt{2\pi}} e^{-[\log(x)-\mu]^2/2\sigma^2}$$

for x>0 and σ>0. Because this distribution is available in both base 10 and natural logarithm base form, the constant A is $1/\log_e 10$ for base 10 and 1 for the natural logarithm base. The logarithm log(x) is also evaluated in terms of the chosen base.

Cumulative Distribution Function: There is no closed form analytic expression for the cumulative distribution function a lognormal random variable.

Inverse Cumulative Distribution Function: There is no closed form analytic expression for the inverse cumulative distribution function a lognormal random variable.

Generation Algorithm: A value, x, for a lognormal random variable is generated using a two-step process. First, a value, y, is generated from the $N(\mu,\sigma^2)$ distribution. This generated value is then used in the expression $x=b^y$, where the base b is either 10 or the natural constant e, as desired.

Expected Value: The expected value of a lognormal random variable (base e) is the following:

$$E(X) = e^{\left(\mu+0.5\sigma^2\right)}$$

Variance: The variance of a lognormal random variable (base e) is the following:

$$V(X) = e^{\sigma^2}\left(e^{\sigma^2} - 1\right)e^{2\mu}$$

Median: The median of a lognormal random variable (base e) is e^{μ}.

12.3.7 Algorithms for the User-Defined Distribution

Cumulative Distribution Function: The user may implement any continuous statistical distribution by supplying a table of data pairs of the form [F(x),x]. In essence, the user provides discrete evaluations of the cumulative distribution function for a set of x values.

Generation Algorithm: The generation algorithm for the user-defined statistical distribution linearly interpolates between the user supplied points to solve $F^{-1}(u)$. This approach implicitly assumes that the user-supplied data defines a continuous random variable rather than a discrete random variable.

12.3.8 Algorithms for the Beta Distribution

Probability Density Function: The probability density function for a beta random variable is the following:

$$f(x) = \frac{x^{p-1}(1-x)^{q-1}}{B(p,q)}$$

for p>0, q>0, and 0<x<1. This variable can be transformed to the interval (a,b), and the resulting probability density function takes the form:

$$f(x) = \frac{(b-a)^{-(p+q+1)}(y-a)^{p-1}(b-y)^{q-1}}{B(p,q)}$$

for p>0, q>0, and a<y<b. The second expression for the probability density function can be obtained from the first by the change of variable y=(b–a)x+a.

Cumulative Distribution Function: There is no closed form analytic expression for the cumulative distribution function of a beta random variable. The numerical approximation specified in Algorithm AS 63 (Majumdar and Bhattacharjee 1985a) is used. This algorithm is based on the reduction method first published by (Soper 1921).

Inverse Cumulative Distribution Function: There is no closed form analytic expression for the inverse cumulative distribution function of a beta random variable. The numerical approximation specified in Algorithm AS 64/AS 109 (Majumdar and Bhattacharjee 1985b) is used.

Generation Algorithm: A random value from the beta distribution is obtained by generating a value from the uniform distribution and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value of a Beta random variable is the following:

$$E(X) = a + (b-a)\frac{p}{p+q}$$

Variance: The variance of a Beta random variable is the following:

$$V(X)=\frac{pq(b-a)^2}{(p+q+1)(p+q)^2}$$

Median: No analytically tractable expression is available for the median of a Beta random variable. For both $p\geq 1$ and $q\geq 1$, the median for a Beta can be approximated by the following expression:

$$\text{med}(X)\approx a+\left(b\text{-}a\right)\left(\frac{p\text{-}\,{}^{1}\!/_{3}}{p\text{+}q\text{-}\,{}^{2}\!/_{3}}\right)$$

12.3.9 Algorithms for the Weibull Distribution

Probability Density Function: The probability density function for a Weibull random variable is the following:

$$f(x)=\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^k}$$

for $x\geq 0$, $k>0$ and $\lambda>0$. It is 0 for $x<0$. This distribution can be shifted by a constant, c, and takes the form

$$f(x)=\frac{k}{\lambda}\left(\frac{(x\text{-}c)}{\lambda}\right)^{k-1}e^{-((x-c)/\lambda)^k}$$

for $x\geq c$, $k>0$ and $\lambda>0$. It is 0 for $x<c$.

Cumulative Distribution Function: The cumulative distribution function for for a Weibull random variable is the following:

$$F(x)=1\text{-}e^{-((x-c)/\lambda)^k}$$

for $x\geq c$ and 0 for $x<0$.

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a Weibull random variable takes the following form:

$$F^{-1}(u)=c+\left(-\frac{\ln(1\text{-}u)}{\lambda}\right)^{1/k}$$

Generation Algorithm: A random value from the shifted Weibull distribution is obtained by generating a value from the uniform distribution, u, and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value for a shifted Weibull random variable can be expressed in terms of the gamma function:

$$\mathrm{E(X)}=\lambda\Gamma\left(1+\frac{1}{\mathrm{k}}\right)+\mathrm{c}$$

Variance: The variance for a shifted Weibull random variable can be expressed in terms of the gamma function:

$$\mathrm{var(X)}=\lambda^2\left[\Gamma\left(1+\frac{2}{\mathrm{k}}\right)-\left\{\Gamma\left(1+\frac{1}{\mathrm{k}}\right)\right\}^2\right]$$

Median: The median for a shifted Weibull random variable can be evaluated from the expression:

$$\mathrm{med(X)}=\lambda\left(\ln(2)\right)^{1/\mathrm{k}}+\mathrm{c}$$

12.3.10 Algorithms for the Cauchy Distribution

Probability Density Function: The canonical probability density function for the Cauchy random variable is

$$\mathrm{f(x)}=\frac{1}{\pi\left(1+\mathrm{x}^2\right)}$$

for all real values of x. This density function can be shifted by a value, c, and scaled by a value, δ, and the revised density function takes the following form:

$$\mathrm{f(x)}=\frac{1}{\pi\delta\left(1+\left(\frac{\mathrm{x-c}}{\delta}\right)^2\right)}$$

Cumulative Distribution Function: The cumulative distribution function for a Cauchy random variable is the following:

$$\mathrm{F(x)}=\frac{1}{\pi}\arctan\left(\frac{\mathrm{x-c}}{\delta}\right)+\frac{1}{2}$$

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a Cauchy random variable is the following:

$$\mathrm{F}^{-1}(\mathrm{u})=\delta\tan\left(\pi(\mathrm{u}-0.5)\right)+\mathrm{c}$$

Generation Algorithm: A random value from the Cauchy distribution is obtained by generating a value from the uniform distribution and then solving the equation x = $F^{-1}(u)$.

Expected Value: The expected value for a shifted and scaled Cauchy random variable does not exist.

Variance: The variance for a shifted and scaled Cauchy random variable does not exist.

Median: The median of a shifted and scaled Cauchy random variable is c.

12.3.11 Algorithms for the Exponential Distribution

Probability Distribution Function: The probability density function for an exponential random variable is

$$f(x)=\frac{1}{\lambda}e^{-\left(\frac{x}{\lambda}\right)}$$

for x>0 and zero otherwise. This density function can be shifted so that the density is nonzero for x>c, and the revised density function takes the following form:

$$f(x)=\frac{1}{\lambda}e^{-\left(\frac{x-c}{\lambda}\right)}$$

Cumulative Distribution Function: The cumulative distribution function for an exponential random variable is the following:

$$F(x)=\left\{\begin{matrix} 0 & x<c \\ 1-e^{-\left(\frac{x-c}{\lambda}\right)} & x \geq c \end{matrix}\right\}$$

Inverse Cumulative Distribution Function: The cumulative distribution function for an exponential random variable is the following:

$$F^{-1}(u)=c-\lambda\ln(1-u)$$

Generation Algorithm: A random value from the exponential distribution is obtained by generating a value from the uniform distribution and then solving the equation x = $F^{-1}(u)$.

Expected Value: The expected value of a shifted exponential random variable is $c+\lambda$.

Variance: The variance of a shifted exponential random variable is λ^2.

Median: The median of a shifted exponential random variable is $c+\lambda\log_e(2)$.

12.3.12 Algorithms for the Gamma Distribution

Probability Distribution Function: The probability density function for the gamma distribution is

$$f(x) = \frac{1}{\theta} \frac{1}{\Gamma(k)} x^{k-1} e^{-\left(\frac{x}{\theta}\right)}$$

for x>0, shape parameter k>0 and scale parameter θ>0. It is zero otherwise.

Cumulative Distribution Function: No closed form expression exists for the cumulative distribution function of a gamma random variable. A numerical approximation is available as Algorithm AS32 (Bhattacharjee 1970).

Inverse Cumulative Distribution Function: No closed form expression exists for the inverse cumulative distribution function of a gamma random variable. The inverse distribution function is evaluated using a bisection method and the cumulative distribution function.

Generation Algorithm: A random value from the gamma distribution is obtained by generating a value from the uniform distribution and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value of a gamma distribution is kθ.

Variance: The variance of a gamma distribution is $k\theta^2$.

Median: There is no simple expression for the median of a gamma distribution. However, the mode of the distribution is (k-1)θ for k>1.

12.3.13 Algorithms for the Logistic Distribution

Probability Density Function: The probability density function for the logistic distribution is

$$\mathrm{f(x)} = \frac{e^{-\left(\frac{x-\mu}{\sigma}\right)}}{\sigma\left(1+e^{-\left(\frac{x-\mu}{\sigma}\right)}\right)^2}$$

for real x, real μ, and shape parameter σ>0.

Cumulative Distribution Function: The cumulative distribution function for a logistic random variable is the following:

$$\mathrm{F(x)} = \frac{1}{1+e^{-\left(\frac{x-\mu}{\sigma}\right)}}$$

for real x, real μ, and shape parameter σ>0.

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a logistic random variable is the following:

$$F^{-1}(u)=\mu-\sigma\ln\left(\frac{1}{u}-1\right)$$

Generation Algorithm: A random value from the logistic distribution is obtained by generating a value from the uniform distribution and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value of a logistic random variable is μ.

Variance: The variance of logistic random variable is $(1/3)\sigma^2\pi^2$.

Median: The median value of a logistic random variable is μ.

12.3.14 Algorithms for the Binomial Distribution

Probability Function: The probability function for a binomial random variable is

$$f(k;n,p)=\Pr(X=k)=\frac{n!}{k!(n-k)!}p^k(1-p)^{n-k}$$

for 0<p<1 and integer n>0 and integer k.

Cumulative Distribution Function: The cumulative distribution function for a binomial random variable is the following:

$$F(x;n,p)=\Pr(X\leq x)=\sum_{i=0}^{\lfloor x\rfloor}\binom{n}{i}p^i(1-p)^{n-i}$$

Where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x.

Generation Algorithm: A random value from the binomial distribution is obtained by generating a value from the uniform distribution and then solving the following equation:

$$x=F^{-1}(u)=\left\{\begin{matrix}1 & \text{for } u\leq p\\ 0 & \text{for } u>p\end{matrix}\right\}$$

Expected Value: The expected value of a binomial random variable is np.

Variance: The variance of a binomial random variable is np(1-p).

Median: The median value of a binomial random variable is the integer closest to np.

12.3.15 Algorithms for the Extreme Value (Gumbel) Distribution

Probability Density Function: The probability distribution function for an extreme value (Gumbel) random variable is the following:

$$f(x)=\frac{e^{\left(-(x-\mu)/\beta\right)}}{\beta}e^{-e^{\left(-(x-\mu)/\beta\right)}}$$

for β>0.

Cumulative Distribution Function: The cumulative distribution function for an extreme value (Gumbel) random variable is the following:

$$F(x)=e^{-e^{\left(-(x-\mu)/\beta\right)}}$$

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for an extreme value (Gumbel) random variable is the following:

$$F^{-1}(u)=\beta\ln\left(-\ln\left(u\right)\right)+\mu$$

Generation Algorithm: A random value from the extreme value distribution is obtained by generating a value from the uniform distribution and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value of an extreme value random variable is $\mu+\beta\gamma$ where γ is the Euler-Mascheroni constant ≈0. 5772156.

Variance: The variance of an extreme value random variable is $\beta^2\pi^2/6$.

Median: The median of an extreme value random variable is μ-β ln[ln(2)].

12.3.16 Algorithms for the Pareto Distribution

Probability Density Function: The probability density function for a Pareto random variable is the following:

$$f(x)=\begin{Bmatrix} 0 & x<c \\ \alpha\dfrac{c^{\alpha}}{x^{\alpha+1}} & x\geq c \end{Bmatrix}$$

Cumulative Distribution Function: The cumulative distribution function for a Pareto random variable is the following:

$$F(x)=\begin{Bmatrix} 0 & x<c \\ 1-\left(\dfrac{c}{x}\right)^{\alpha} & x\geq c \end{Bmatrix}$$

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a Pareto random variable is the following:

$$F^{-1}(u)=c\left(1-u\right)^{-1/\alpha}$$

Generation Algorithm: A random value from the Pareto distribution is obtained by generating a value from the uniform distribution and then solving the equation x = F^{-1}(u).

Expected Value: The expected value of a Pareto random variable doesn't exist if $\alpha \leq 1$. For $\alpha>1$ the mean is

$$E(X)=c\left(\frac{\alpha}{\alpha-1}\right)$$

Variance: The variance of a Pareto variable doesn't exist if $\alpha \leq 2$. For $\alpha>2$ the variance is

$$Var(X)=\frac{c^2\alpha}{\left(\alpha-1\right)^2\left(\alpha-2\right)}$$

Median: The median of a Pareto random variable is

$$med(X)=c\sqrt[\alpha]{2}$$

12.3.17 Algorithms for the Power Function Distribution

Probability Density Function: The probability density function for a power function random variable is the following:

$$f(x)=\alpha^{\beta-1}\beta x^{\beta-1}$$

for o<x<α, where $\alpha>0$ is the scale parameter and β is the shape parameter.

Cumulative Distribution Function: The cumulative distribution function for a power function random variable is the following:

$$F(x)=\left(\frac{x}{\alpha}\right)^{\beta}$$

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a power function random variable is the following:

$$F^{-1}(u)=\alpha\, u^{1/\beta}$$

Generation Algorithm: A random value from the power function distribution is obtained by generating a value from the uniform distribution and then solving the equation x = F^{-1}(u).

Expected Value: The expected value of a power function random variable is the following:

$$E(X)=\frac{\alpha\beta}{\beta+1}$$

Variance: The variance of a power function random variable is the following:

$$Var(X)=\frac{\alpha^2\beta}{(\beta+2)(\beta+1)^2}$$

Median: The median of a power function random variable is the following:

$$med(X)=\alpha\left(\frac{1}{2}\right)^{1/\beta}$$

12.3.18 Algorithms for the Laplace Distribution

Probability Distribution Function: The probability density function for a Laplace random variable is

$$f(x)=\frac{1}{2b}\left\{\begin{matrix} e^{-\left(\frac{\mu-x}{\lambda}\right)} & \text{if } x<\mu \\ e^{-\left(\frac{x-\mu}{\lambda}\right)} & \text{if } x\geq\mu \end{matrix}\right\}$$

Cumulative Distribution Function: The cumulative distribution function for a Laplace random variable is the following:

$$F(x)=\left\{\begin{matrix} \frac{1}{2}e^{-\left(\frac{\mu-x}{\lambda}\right)} & \text{if } x<\mu \\ 1-\frac{1}{2}e^{-\left(\frac{x-\mu}{\lambda}\right)} & \text{if } x\geq\mu \end{matrix}\right\}$$

Inverse Cumulative Distribution Function: The inverse cumulative distribution function for a Laplace random variable is the following:

$$F^{-1}(u)=\left\{\begin{matrix} \mu+\lambda\ln(2u) & \text{if } u<0.5 \\ \mu-\lambda\ln(2-2u) & \text{if } u\geq 0.5 \end{matrix}\right\}$$

Generation Algorithm: A random value from the Laplace distribution is obtained by generating a value from the uniform distribution and then solving the equation $x = F^{-1}(u)$.

Expected Value: The expected value of a Laplace random variable is μ.

Variance: The variance of a Laplace random variable is $s\lambda^2$.

Median: The median of a Laplace random variable is μ.

12.3.19 Algorithms for the Poisson Distribution

Probability Function: The probability function for a Poisson random variable is

$$f(k)=\Pr(X=k)=\frac{\lambda^{k}e^{-\lambda}}{k!}$$

for integer $k\geq 0$ and $\lambda>0$.

Cumulative Distribution Function: The cumulative distribution function for a Poisson random variable is the following:

$$F(x)=\Pr(X\leq x)=e^{-\lambda}\sum_{i=0}^{\lfloor k\rfloor}\frac{\lambda^{i}}{i!}$$

Where $\lfloor k\rfloor$ denotes the greatest integer less than or equal to k.

Generation Algorithm: The generation algorithm starts by generating a value, u, from the uniform distribution. If $\lambda\leq 500$, then a value from the Poisson distribution is generated as the first integer, k, where $F(k)\geq u$. If $\lambda>500$, then a normal approximation $N(\mu=\lambda,\sigma^2=\lambda)$ is used. The generated value is the normal value rounded to the nearest integer.

Expected Value: The expected value of a Poisson random variable is λ.

Variance: The variance of a Poisson random variable is λ.

Median: The median of a Poisson random variable is approximately equal (Choi 1994) to $\left\lfloor \lambda+\frac{1}{3}-\frac{0.02}{\lambda}\right\rfloor$. In this situation, $\lfloor k\rfloor$ denotes the greatest integer less than or equal to k.

12.3.20 Algorithms for the F Distribution

Probability Density Function: The probability density function for an F random variable is the following:

$$f(x,d_1,d_2)=\frac{1}{B\left(\frac{d_1}{2},\frac{d_2}{2}\right)}\left(\frac{d_1}{d_2}\right)^{\frac{d_1}{2}}x^{\frac{d_1}{2}-1}\left(1+x\left(\frac{d_1}{d_2}\right)\right)^{-\frac{d_1+d_2}{2}}$$

for $x\geq 0$, $d_1>0$ and $d_2>0$.

Cumulative Distribution Function: The cumulative distribution function for an F random variable is the following:

$$F(x;d_1,d_2)=IB_{\frac{d_1x}{d_1x+d_2}}\left(\frac{d_1}{2},\frac{d_2}{2}\right)$$

Where IB is the regularized incomplete beta function.

Inverse Cumulative Distribution Function: No closed form analytical expression exists for the inverse cumulative distribution function for an F function random variable.

Generation Algorithm: Because the cumulative distribution function for the F distribution can be written as a special case of the beta distribution, values from the F distribution are obtained by generating a value from the uniform distribution and then solving the equation w = F^{-1}(u), where the F^{-1}(u) denotes the beta inverse cumulative distribution function. If $u\geq0.95$, then $\alpha=0.5d_2$ and $\beta=0.5d_1$ and $x = d_2(1-w)/(d_1w)$; otherwise with $\alpha=0.5d_1$ and $\beta=0.5d_2$ and $x=d_2w/(d_1(1-w))$.

Expected Value: The expected value of an F random variable is the following:

$$E(X)=\frac{d_2}{d_2-2}$$

Variance: The variance of an F random variable is the following:

$$Var(X)=\frac{2d_2^2\left(d_1+d_2+2\right)}{d_1\left(d_2-2\right)^2\left(d_2-4\right)}$$

Mode: The mode of an F random variable is the following:

$$mode(X)=\frac{d_1-2}{d_1}\frac{d_2}{d_2+2}$$

12.3.21 Algorithms for the Chi-Squared Distribution

Probability Density Function: The probability density function for a Chi-squared random variable is the following:

$$f(x,k)=\frac{x^{\frac{k}{2}-1}e^{-\frac{x}{2}}}{2^{\frac{k}{2}}\Gamma\left(\frac{k}{2}\right)}$$

for $x\geq0$, and $k\geq1$.

Cumulative Distribution Function: The cumulative distribution function for a Chi-squared variable is the following:

$$F(x;k)=\frac{I\Gamma\left(\frac{k}{2},\frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}$$

Where IΓ is the incomplete gamma function and Γ is the gamma function.

Inverse Cumulative Distribution Function: No closed form analytical expression exists for the inverse cumulative distribution function for a Chi-squared random variable.

Generation Algorithm: Because the cumulative Chi-squared distribution can be written as a special case of the gamma distribution, values from the Chi-squared distribution are obtained by generating a value from the uniform distribution and then solving the equation $w = F^{-1}(u)$, where the $F^{-1}(u)$ denotes the gamma inverse cumulative distribution function with parameter $\theta=1/2$ and where k/2 is used instead of the nominal k in the gamma distribution.

Expected Value: The expected value of a Chi-squared function random variable is k.

Variance: The variance of a Chi-squared random variable is 2k.

Median: The median of a Chi-squared random variable is approximately the following:

$$\text{median}(X) \approx k\left(1-\frac{2}{9k}\right)^{3}$$

13.0 References

Beasley, JD, and SG Springer. 1985. "Algorithm AS111, the Percentage Points of the Normal Distribution." In *Applied Statistics Algorithms*, eds. P Griffiths and ID Hill. Ellis Horwood Limited, Chichester, U.K.

Bhattacharjee, GP. 1970. "Algorithm AS32: The Incomplete Gamma Integral." *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 19(3):285-287.

Choi, KP. 1994. "On the Medians of Gamma Distributions and an Equation of Ramanujan." *Proceedings of the American Mathematical Society* 121(1):245-251, DOI:http://dx.doi.org/10.1090/S0002-9939-1994-1195477-8.

Deonigi, DE, DM Anderson, and GL Wilfert. 1994. *Commercial Milk Distribution Profiles and Production Locations*, PNWD-2218 HEDR, Battelle, Pacific Northwest Laboratories, Richland, Washington.

Draxler, RR, B Stunder, G Rolph, A Stein, and A Taylor. 2012. *HYSPLIT4 User's Guide*, Air Resources Laboratory, National Oceanic and Atmospheric Administration (NOAA), Silver Spring, MD. http://www.arl.noaa.gov/documents/reports/hysplit_user_guide.pdf

Eslinger, PW, KS Lessor, and SJ Ouderkirk. 1994. *User Instructions for the CIDER Dose Code*, PNWD-2252 HEDR, Battelle, Pacific Northwest Laboratories, Richland, Washington.

ESRI. 2013. *ArcGIS - Mapping and Understanding Our World*. Accessed on May 16, 2013, at http://www.esri.com/software/arcgis

Farris, WT, BA Napier, PW Eslinger, TA Ikenberry, DB Shipler, and CJ Simpson. 1994. *Atmospheric Pathway Dosimetry Report, 1944–1992*, PNWD-2228 HEDR, Battelle, Pacific Northwest Division, Richland, Washington.

Fishman, G, and L Moore, III. 1986. "An Exhaustive Analysis of Multiplicative Congruential Random Number Generators with Modulus 2^{31} - 1." *SIAM Journal on Scientific and Statistical Computing* 7(1):24-45, DOI:http://dx.doi.org/10.1137/0907002.

GDAS. 2012. *Global Data Assimilation System Archive*. Air Resources Laboratory, National Oceanic and Atmospheric Administration. Accessed on April 2, 2012, at ftp://arlftp.arlhq.noaa.gov/pub/archives/gdas1/

Google. 2012. *Keyhole Markup Language*. Accessed on May 20, 2013, at https://developers.google.com/kml/documentation/kmlreference (last updated January 2, 2013).

Google. 2013. *Google Earth*. Accessed on May 16, 2013, at http://www.google.com/intl/en/earth/index.html

Hill, ID. 1985. "Algorithm as 66, the Normal Integral." In *Applied Statistics Algorithms*, eds. P Griffiths and ID Hill. Chichester, U.K., Chichester, U.K.

Hoare, CAR. 1961. "Algorithm 63: Partition." *Commun. ACM* 4(7):321, DOI:http://dx.doi.org/10.1145/366622.366642.

HYSPLIT. 2013. *Air Resources Laboratory HYSPLIT Model*. NOAA, Air Resources Laboratory. Accessed on February 20, 2013, at http://www.arl.noaa.gov/documents/Summaries/HYSPLIT_FINAL.pdf (last updated November 2011).

Iman, RL, and WJ Conover. 1982. "A Distribution-Free Approach to Inducing Rank Correlations among Input Variables." *Communications in Statistics - Simulation and Computation* B11(3):311-334, DOI:http://dx.doi.org/10.1080/03610918208812265.

Majumdar, KL, and GP Bhattacharjee. 1985a. "Algorithm AS63: The Incomplete Beta Integral." In *Applied Statistics Algorithms,* eds. P Griffiths and ID Hill. Ellis Horwood Limited, Chichester, U.K.

Majumdar, KL, and GP Bhattacharjee. 1985b. "Algorithm AS64/AS109 Inverse of the Incomplete Beta Function Ratio." In *Applied Statistics Algorithms,* eds. P Griffiths and ID Hill. Ellis Horwood Limited, Chichester, U.K.

Marsh, TL, DM Anderson, WT Farris, TA Ikenberry, BA Napier, and GL Wilfert. 1992. *Commercial Production and Distribution of Fresh Fruits and Vegetables: A Scoping Study on the Importance of Produce Pathways to Dose*, PNWD-2022 HEDR, Battelle, Pacific Northwest Laboratories, Richland, Washington.

Miley, TB, PW Eslinger, WE Nichols, KS Lessor, and SJ Ouderkirk. 1994. *User Instructions for the DESCARTES Environmental Accumulation Code*, PNWD-2251 HEDR, Battelle, Pacific Northwest Laboratories, Richland, Washington.

Mokrov, YG, LR Anspaugh, and BA Napier. 2004. *Reconstruction of Dose to the Residents of Ozersk from Operation of the Mayak Production Association: 1948-2002 (Final Report on the Feasibility Study for Project 1.4)*, US-Russian Joint Coordinating Committee on Radiation Effects Research Project 1.4, Ozersk, Russia.

Mokrov, YG, VZ Martyushov, PM Stukalov, IA Ivanov, and ES Levunina. 2007. *Milestone 8: Production and Delivery of Food to Ozersk*, US-Russian Joint Coordinating Committee on Radiation Effects Research Project 1.4, Ozersk, Russia.

Mood, AM, FA Graybill, and DC Boes. 1974. *Introduction to the Theory of Statistics*. Third ed., Mcgraw-Hill Series in Probability and Statistics, McGraw-Hill, Inc., New York, New York, ISBN:0-07-042864-6.

Napier, BA, LR Anspaugh, YG Mokrov, and SI Rovny. 2008. *Milestone 4: Evaluation of Chemical Forms of ^{131}I Involved in Atmoispheric Transport*, US-Russian Joint Coordinating Committee on Radiation Effects Research Project 1.4, Ozersk, Russia.

NGA. 2013. *Office of Geomatics: World Geodetic System 1984 (WGS 84)*. National Geospatial-Intelligence Agency. Accessed on May 21, 2013, at http://earth-info.nga.mil/GandG/wgs84/ (last updated April 29, 2013).

NOAA. 2013. *Gridded Meteorological Data Archives*. NOAA Air Resources Laboratory (ARL). Accessed on May 15, 2013, at http://ready.arl.noaa.gov/archives.php (last updated April 8, 2013).

Park, SK, and KW Miller. 1988. "Random Number Generators: Good Ones Are Hard to Find." *Commun. ACM* 31(10):1192-1201, DOI:http://dx.doi.org/10.1145/63039.63042.

Ramsdell Jr., JV, CA Simonen, and KW Burk. 1994. *Regional Atmospheric Transport Code for Hanford Emission Tracking (RATCHET)*, PNWD-2224 HEDR, Battelle, Pacific Northwest Laboratories, Richland, Washington.

Rovny, SI, YG Mokrov, PM Stukalov, DA Beregich, II Teplyakov, LR Anspaugh, and BA Napier. 2009. *Milestone 12, Part 1: Methods for Calculating Thyroid Doses to the Residents of Ozersk Due to ^{131}I Releases from the Stacks of the Mayak Production Association*, US-Russian Joint Coordinating Committee on Radiation Effects Research Project 1.4, Ozersk, Russia.

sbatch. 2013. *sbatch - Submit a Batch Script to SLURM*. Accessed on May 16, 2013, at https://computing.llnl.gov/linux/slurm/sbatch.html (last updated February 28, 2012).

Snyder, SF, WT Farris, BA Napier, TA Ikenberry, and RO Gilbert. 1994. *Parameters Used in the Environmental Pathway and Radiological Dose Modules (DESCARTES, CIDER and CRD Codes) of the Hanford Environmental Dose Reconstruction Integrated Codes (HEDRIC).* PNWD-2023 HEDR, Rev. 1, Battelle, Pacific Northwest Laboratories, Richland, Washington.

Soper, HE. 1921. *The Numerical Evaluation of the Incomplete B-Function: Or of the Integral X (1-X) Dx for Ranges of X between 0 and 1*. Tracts for Computers No. 7, Cambridge University Press, London.

Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)
www.pnl.gov