Dragon Professional User's Manual

Version 6.1 – February 13, 2011

Contents

List of Figures

Chapter 1

INTRODUCTION TO DRAGON

1.1   What is Dragon and What Can It Do?

Dragon/ips® is a software package for personal computers that displays and manipulates images, i.e. visual representations of data. Although **Dragon** can be used for a variety of different types of visual information, it is oriented primarily toward processing remotely-sensed images. The following section explains remote sensing image processing and discusses **Dragon**'s image processing capabilities from a remote sensing perspective.

1.1.1   Remote Sensing Image Processing

Remote sensing is gathering information about some object from a distance. In most common uses of remote sensing, the object is the earth, or some part of it, and the information gathered is measured levels of electromagnetic radiation reflected or emitted from the earth at various locations. A variety of sensing instruments can be used to measure and record this radiation, depending on its wavelength. The most commonly-used sensors are aircraft-borne cameras and scanning radiometers mounted on satellites orbiting the earth.

Remotely-sensed data are routinely used for investigation, management and monitoring in a wide variety of fields: meteorology, forestry, oceanography, exploration geology, hydrology and water resources engineering, agriculture, etc. Electromagnetic radiation levels can provide information on vegetation density and species, rock and soil types, water temperature and turbidity, settlement location and intensity, and many other important physical and social variables. Remote sensing makes it possible to view and evaluate conditions in a relatively large spatial area, at a relatively low cost compared to ground surveys.

Cameras and scanning radiometers measure the levels of radiation (visible light or infrared) distributed over some area of interest. The data gathered by these devices can be maintained and presented in either *digital* or *visual* form. In digital form, the measurements are represented by a set of numeric values, one number per *resolution element* or *pixel* (the smallest area from which a measure of radiation can be obtained.) The same measurements, however, can also be considered as a visual *image*. An image is a two-dimensional 'picture' of sensed radiation levels; positions on the image correspond (more or less) to positions on the earth, and the brightness of the image at any position is proportional to the amount of radiation measured at that point.

A visual image is the primary representation provided by a camera. For traditional cameras that use photographic film, special devices (called *scanners* or *frame grabbers*) are necessary to transform a photograph into digital form. For data gathered by digital cameras, or aerial or space-based radiometers, the digital form is usually primary. A variety of computer-based devices can be used to turn this digital data into a visual image. Figure 1.1 illustrates the basic idea behind remote sensing image acquisition and the relationship between visual and digital representations of a remotely sensed image.
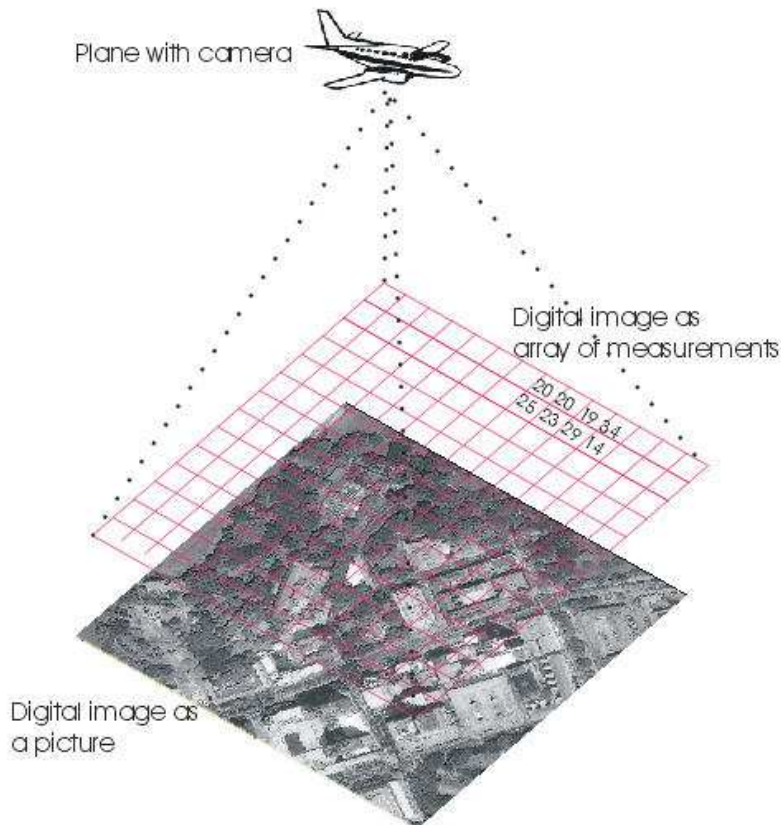
Figure 1.1: *Remote sensing images: digital and visual*

Both digital and visual image representations of remotely sensed data have certain advantages. Visual images are relatively easy for humans to analyze and interpret. The visual representation makes it possible for the human analyst to see spatial patterns in data and recognize objects or areas based on their appearance and spatial relationships. This sort of global relational information is difficult to extract from a set of numbers. However, there is information in a remotely-sensed image that is not directly available to the human perceiver. Distortions, sensor defects and other sources of noise mask some of the information in an image. Important patterns of information exist that can be detected only by combining data from several wavelengths of reflected radiation. Manipulations to minimize noise or reveal multi-dimensional patterns are difficult or impossible to perform optically or photographically on the visual form of the image. Such manipulations are relatively easy to perform on the digital form of the image data, using a computer-based digital image analysis system.

A *digital image analysis system* is a combination of computer hardware and software that can display, transform and combine image data so that the human analyst can extract as much information as possible. The system maintains and manipulates data in numeric (digital) form, but displays it in visual form. The flexibility and speed of computers makes it possible to try different image combinations, different color assignments, and different noise reduction techniques. The computer can also be used to search for multi-dimensional patterns not visible to the human eye.

1.1.2   Image Processing for Other Applications

Although **Dragon** is intended primarily for remote sensing work, it can be effectively applied to image analysis in other fields. Techniques used for medical image processing, for example, are essentially the same as those common in remote sensing. Only the vocabulary differs. **Dragon** will be less useful in domains such as graphics arts or computer-aided design. Although the system does include capabilities for overlaying and combining several images, it has only a few facilities for line drawing or definition of visual objects. Basically, **Dragon** can be applied in contexts where image information gathered from some outside source needs to be improved, modified, and combined with other image-like information.

1.1.3   Geographic Information Systems

Geographic information systems, or GIS, are closely related to remote sensing image analysis systems. Like a digital image analysis system, a GIS is a collection of hardware, software, and data used to study and draw conclusions about phenomena distributed in space, usually on the the surface of the earth. GIS systems can be used for applications like forestry, agriculture, disaster planning, and urban planning, just like remote sensing analysis systems. Like remote sensing results, GIS results can be displayed as map-like pictures.

The primary distinction between remote sensing and GIS analysis is that remote sensing works with data that represent more or less raw measurements of earth-related phenomena. GIS in contrast works with data from many sources, which has often been coded, interpreted, grouped, or combined with other information. GIS can handle data *layers* such as land use or water bodies, but is just as likely to deal with spatially distributed information about population, income, ecological sensitivity, and location of cases of bird flu, information that cannot be 'sensed' directly.

In fact, information derived from remote sensing data is frequently used in geographic information systems: forest or crop cover layers created by classifying satellite images, roads extracted by tracing out their locations with an aerial photo in the background, flood plain maps derived by examining images of flooded areas, slope maps derived from analysis of overlapping stereo images, and so on.

Like remote sensing images, GIS layers can be represented as grids of numbers, with each cell in the grid (each pixel) representing some location in space. **Dragon** has some capabilities for displaying and analyzing GIS data that is represented in this form (called a *raster representation*). Sometimes, however, it is more convenient to store GIS information in a *vector representation*, based on points, lines or regions

explicitly encoded by their geographic coordinates. At present, **Dragon** does not provide significant capabilities for analyzing GIS data in vector form. However, it does allow users to create vector layers with an image as a background (as in the roads example above). It is also possible to import vector data from other sources and display these data as an overlay with a remote sensing image as a background.

GIS is an important, complex, and fascinating topic. For further information, the reader should consult some of the references in the *Bibliography*.

1.1.4 Dragon Image Processing Capabilities

This section summarizes the image processing capabilities available with **Dragon**. It assumes that you have some familiarity with digital image processing techniques and terminology. For more information about these capabilities, you can consult the chapters on individual **Dragon** operations and the resources listed in the Bibliography.

Dragon has capabilities for displaying single-band images in color or gray-scale form. It also includes three-band color composite and image overlay capabilities. **Dragon** provides facilities for adding titles or labels to screen images and for displaying legends for classified images. The user can explicitly control contrast stretching and color assignment or allow **Dragon** to handle these processes automatically.

Most common enhancement operations are available in **Dragon**, including band sums, ratios and differences, edge enhancement, smoothing and arbitrary 3x3 filters. A few more unusual enhancement capabilities are also provided: symmetric differencing, relaxation, and masking. The results of each enhancement operation remain available so that multi-step computations such as vegetation indices can be accomplished efficiently. Images created by any **Dragon** operation or imported from outside sources can also serve as input to enhancement processes. **Dragon** also includes transformations such as principal components analysis as part of its image enhancement capabilities.

Dragon supports density slicing, supervised classification using boxcar (parallelepiped), minimum distance to mean, and maximum likelihood algorithms, and unsupervised clustering. Training samples are selected interactively and can be saved, examined, combined, and modified. Training area boundaries can also be saved and then applied to a new image. This allows identical areas to be used in two separate classifications (e.g. in multitemporal studies). The resulting image can be modified by combining or recoding classes. **Dragon** provides control over colors assigned to classes. Classified images can be visually overlaid on the original image or combined with the original using the masking option to assist in the interpretation of class contents.

Dragon also includes facilities for geometric correction and registration, and for point, line, and polygon operations. The geometric correction operations consist of interactive ground control point selection, calculation of image transformation equations, and resampling of the image. Point, line, and polygon operations include provisions for creating vector feature overlays, for extracting polygonal subimages ('cookie cutter' operations), and for measuring polygons and polylines.

Newer versions of **Dragon** have added facilities for basic raster GIS modeling. Capabilities include calculating slope and aspect images from digital elevation model

input data; identifying pixels that belong to a buffer within a certain distance of target pixels; and logically combining multiple raster layers based on a set of conditional rules.

Finally, **Dragon** offers a variety of utility functions including: image data value display, image subsetting, histograms, scatterplots, display and modification of identifying information in image file headers, import and export of images and vectors, and saving of images.

A variety of stand-alone programs are available for use with **Dragon**. These include programs for exchanging data with other remote sensing and GIS systems, and for exchanging both raster and vector data with other computers. In general, these are no longer needed; you can use the Import and Export operations within **Dragon**.

The basic capabilities of **Dragon** can be augmented by means of special-purpose routines created by the individual user. This possibility is discussed in more detail in the *Getting Images In and Out* chapter, and in the manuals supplied with the **Dragon** Programmer's Toolkit or the **libdtk** libraries.

1.2 Overview of This Manual

1.2.1 Dragon Documentation

The *User Manual* is the primary documentation for the **Dragon** software system. The complete and most up-to-date version of the *User Manual* is provided in machine readable form both as a Portable Document Format (PDF) file,and as a set of **.HTML** files. A reader program supplied with **Dragon** provides convenient access to the HTML version of the manual. You can also view the individual files using any web browser such as Firefox, Opera, or Internet Explorer.

In addition to the PDF and HTML versions of the *User Manual*, you may find a printed copy of some parts of the manual. Note that the printed copy is not the complete user manual and, in addition, may be less up-to-date. The printed version is provided only as a convenient introduction to the **Dragon** system.

Additional documentation in the form of Technical Notes, Release Notes or Installation Instructions may be distributed with the software from time to time, in either printed or computer-readable form. Check the **Dragon** distribution disk(s) for any files called **READxxx.ME** or **READxxx.HTML**. These will contain the most up-to-date information on using **Dragon**. Users should also check the **Dragon** website frequently for information about new releases and other events of interest.

1.2.2 The ManReader Application

Dragon includes an application specialized for reading the *User Manual*. This application, called the Manual Reader, is an enhanced HTML browser that knows about the structure of the **Dragon** documentation. The Manual Reader is used to provide context-sensitive operation-specific help within Dragon. However, it can also be used in a stand-alone mode to browse the manual as a whole.

You can run the Manual Reader either from the computer's **Start** menu (**Start⇒Dragon⇒User Manual**), from the command line, or from within

Dragon. To run the Manual Reader from the command line, switch to the **Dragon** system directory and type:

```
ManReader
```

To bring up the Manual Reader when you are running **Dragon**, choose the **Help⇒User Manual** menu item.

The Manual Reader has three tabbed panes. The leftmost tab, labeled **Contents**, displays a table of contents in a tree-structured form. Figure 1.2 shows the Manual Reader contents page. When you select a topic from the tree, the center tab, labeled **Description**, comes to the front, showing the text of the currently-selected topic. Buttons at the bottom of the application allow you to navigate to the next or previous topic in the document.
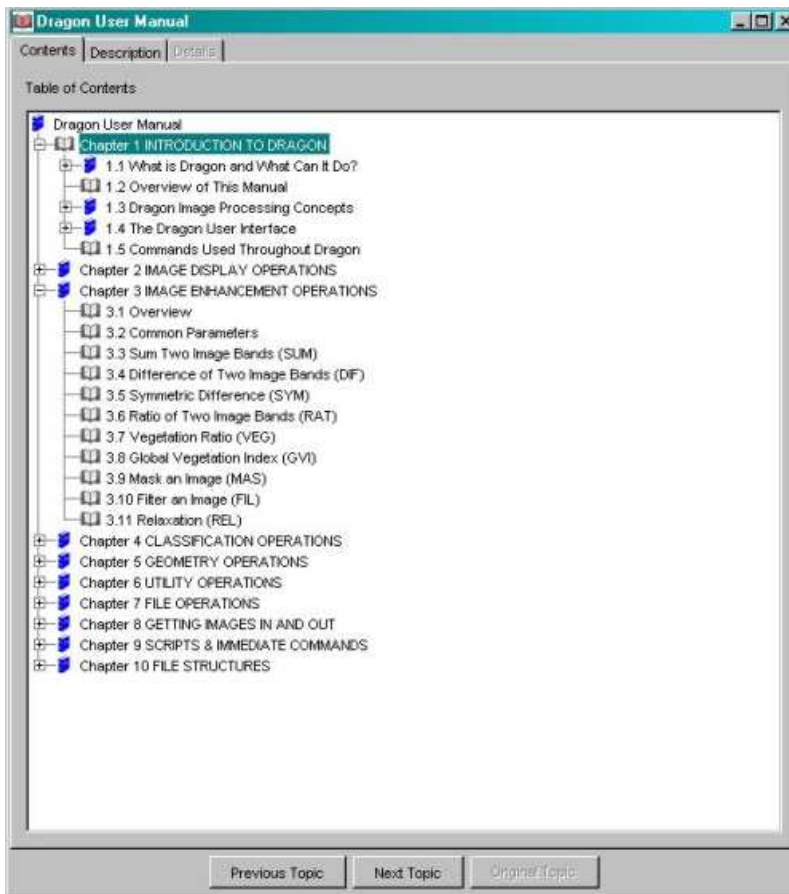

Figure 1.2: *Manual reader table of contents pane*

The third tab, labeled **Details**, is enabled only for sections of the manual that describe specific **Dragon** operations. In these cases, clicking on **Details** will display a table showing all the parameters associated with the operation, with their meanings, defaults, and the **Dragon** command language code (*parameter specifier*) for each one. If the topic displayed in the **Description** pane does not relate to a specific **Dragon** operation, the **Details** tab will be disabled. Figure 1.3 shows a sample **Details** page.

When the **Details** pane is displayed, the **Previous Topic** and **Next Topic** buttons are disabled. You must return to one of the other panes to select a new topic.

Copyright ©Goldin-Rudahl Systems, Inc
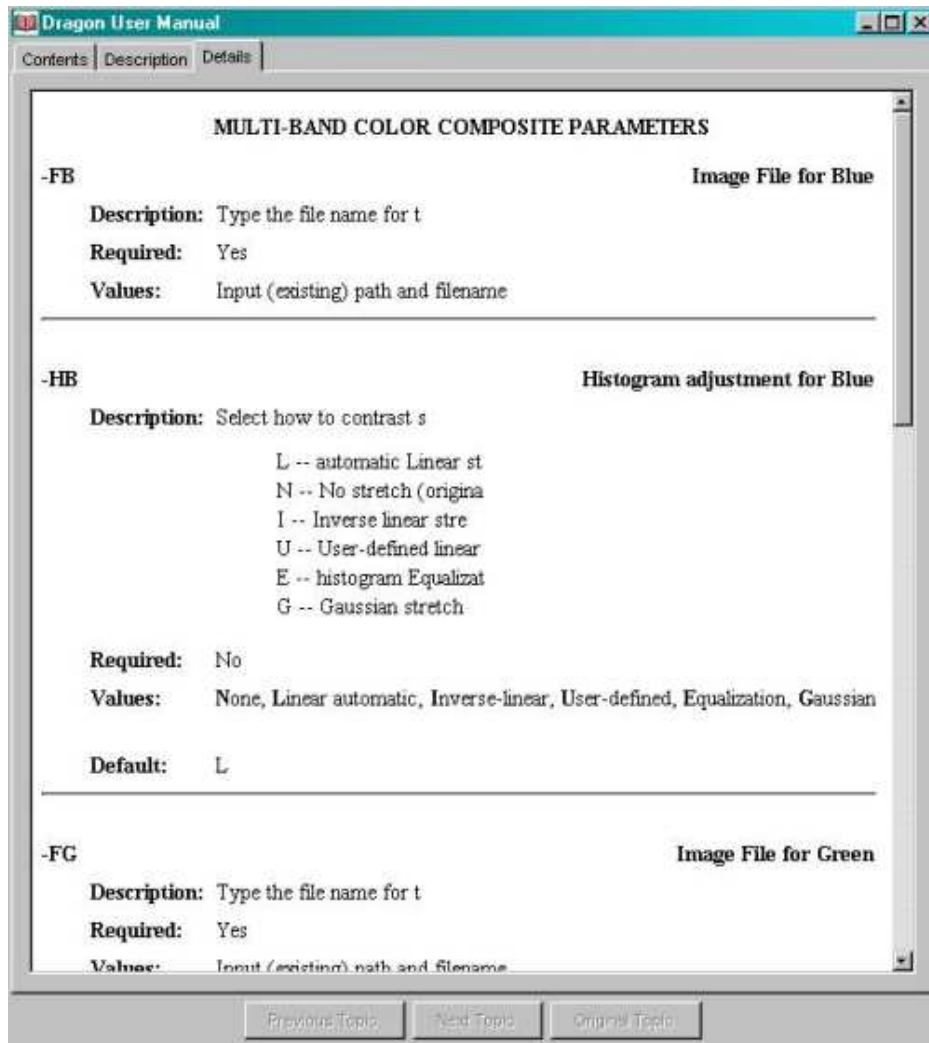
Figure 1.3: *Manual reader details pane*

As noted above, the Manual Reader can be accessed from inside **Dragon** by choosing **Help⇒User Manual**. If the Manual Reader is not already visible, it will be activated. Then, if **Dragon** is currently displaying the data entry screen (*response panel*) for a particular **Dragon** operation, the Manual Reader will immediately display the **Description** pane for that operation. Otherwise, the Manual Reader will simply display the first page of the first chapter.

This manual is designed to be accessed via the Manual Reader. This is the only way you can gain access in a context-sensitive manner to the detailed parameter information for each operation. However, the manual is actually provided as a set of HTML files, and it can be read using a standard Web browser such as Mozilla, Firefox, Opera or Internet Explorer. To access the document in this way, point your browser to the file **doc\Contents.html** within the **Dragon** system directory. This will bring up a simple list of links to each chapter.

1.2.3   Assumptions about the Reader

This manual assumes that you, the reader, know something about digital image processing principles, techniques and terminology. As an educational tool, **Dragon** is

intended to be used along with, but not to replace, a textbook on remote sensing image processing. The description of each operation explains briefly what the operation does, but it assumes you are familiar with the underlying concepts and assumptions. If you need further information on basic image analysis issues, we recommend that you consult references in the *Bibliography* for more in-depth treatment.

The manual does not assume you have had any previous hands-on experience with a digital image processing system. However, it does require that you have a basic understanding of how to use your computer and your operating system. You should know, for example:

- How to start up the computer and the operating system
- How to use the mouse
- How to insert, remove, and handle CDROMs
- What a file is; how to find out what files you have available; how to name, rename, copy, and delete files
- How to run programs
- What subdirectories and paths are and how you use them

You do not need to know anything about programming or programming languages to use **Dragon**.

1.2.4 Manual Organization

This manual is organized in three general parts. First, there is this introductory chapter, which provides an overview of **Dragon** and covers a number of important preliminary topics. In particular, this chapter describes some important operating concepts that you must understand to use **Dragon** effectively. It also discusses in detail the various ways in which you interact with **Dragon** to perform image processing tasks. These initial topics are essential. Regardless of your level of experience with computers and image processing, you should read this chapter first.

The second part of the manual consists of six chapters that describe in detail all the operations available in **Dragon** and how they are used. The six chapters correspond to the six main functional divisions in **Dragon**: display operations, enhancement operations, classification operations, geography operations, utility operations, and file handling operations. Each chapter begins with a general overview that summarizes the capabilities within that functional division. This is followed by separate sections for each operation. These individual sections describe what the operation does and make suggestions for its most effective use. If you are accessing the manual via the Manual Reader application, these chapters also provide access to a detailed table of all options and possible choices involved in using the operation, as discussed above.

You may use this second part of the manual in various ways, depending on your experience with digital image processing techniques and systems. If you are relatively inexperienced, you should probably read the chapter overview and operation descriptions first, to get a unified picture of what operations should be used for what purpose. Later you can go back and study the details of each operation. If you are fairly experienced with digital image processing systems, you can use the

Table of Contents to locate the operation you need, and consult that section directly for details on how to use it. The operations are described in this manual in the same sequence that **Dragon** presents them to you in the menus.

The third part of this manual consists of several chapters covering more miscellaneous topics. These include data transfer into and out of the system, and use of **Dragon**'s script capability which allows an ordinary text file to control the operation of **Dragon** without any user input. The script capability is particularly useful in an educational environment, for construction of demonstrations and partially-automated tutorials. If you are an instructor, a laboratory supervisor, or simply an advanced **Dragon** user, you should read this material. However, it is not essential for running the basic **Dragon** system.

The third part also includes a chapter that describes many of the file formats used in **Dragon**. You should consult this section if you need to write a program to convert data from some other image processing system for use with **Dragon**. Finally, there is a *Bibliography* which suggests some references for further information on remote sensing, image processing, and GIS. We also provide a comprehensive Index in the PDF version of the manual.

1.3 Dragon Image Processing Concepts

1.3.1 Images and Image Files

Dragon is a program that processes image data. For **Dragon**'s purposes, an image is a two-dimensional array of numbers, each number associated with a position in the array. Positions in this array are called *pixels* or *cells*. Each pixel or cell can be identified by a set of *coordinates*. In **Dragon**, the vertical coordinate is usually called the *line number*; the horizontal coordinate is called the *pixel number*. (Both the line number and pixel number coordinates start with a value of zero at the upper left corner of the image.) The maximum possible values of line and pixel number depend on the particular image. The pixel number at a particular point on an image is also sometimes referred to as the point's *X coordinate* while the line number is the point's *Y coordinate*. The pair of numbers is called the point's *image coordinates*.

Image data in the **Dragon** system are stored in *image files*. The image files contain the data values for each pixel, line by line. Image files are normal operating system files: you can copy them, rename them, delete them, etc.

Image files in **Dragon** can have any name up to 32 characters long. However, all image files must have the extension (filetype) **.img** (or **.IMG**) so that **Dragon** can recognize them. Normally, the name you give to an image file should tell you something about what the image data is and where it came from. For example, you might call a file holding aerial photography data from Bangkok **BangkokPhoto.img**.

One common type of image data that you will use with **Dragon** will be satellite image data gathered with scanning radiometers such as the Landsat TM instrument. These sensors usually measure reflected radiation in several different wavelengths, often called *bands* (or, in some books and image processing systems, *channels*). Thus, a complete set of satellite data for a particular scene will actually consist of several bands, all acquired simultaneously. Such data is called *multispectral*. Multispectral data are useful because some kinds of ground cover reflect very little

radiation in some wavelengths and a great deal in others. Thus, a particular ground cover type can be identified by its pattern of radiation levels across different bands (its *spectral signature*). Figure 1.4 illustrates the concept of multispectral image data.
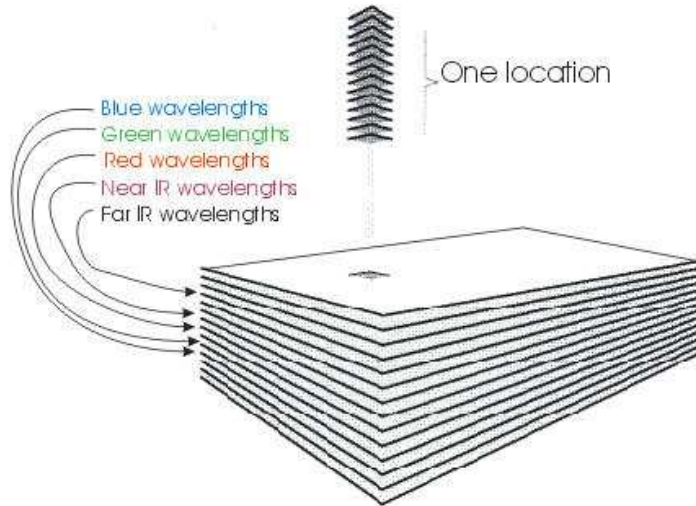
Figure 1.4: *Multispectral data: spatially registered, different wavelengths*

In each band of a multispectral scene, a particular pixel (identified by a specific set of image coordinates) represents the same location on the ground. Because of this property, all the bands of a multispectral scene are said to be *registered*. Registration is an important consideration in digital image processing. It normally does not make sense to combine two images if they are not registered, since you may be combining data from two different locations. Images of the same geographic area acquired at different times usually will not be registered. It is possible to perform a mathematical transformation on one image to register it to another.

Note

In some image processing systems, data from all the bands of one scene are stored together in the same file. In **Dragon**, each band of a scene is stored in a separate image file. You must explicitly decide which bands to include or exclude in any operation.

Since you frequently want to process several bands of the same scene together, it is important to know not only which scene but also which band the data in a file represents. We recommend that you use a standard method for naming your image files, such as reserving the last character in the file name for the band number. For files that represent the same scene, all other characters in the name would be the same. Thus, **Bangkok1.img**, **Bangkok2.img** and **Bangkok4.img** would be appropriate names for three bands of a multispectral image of Bangkok.

1.3.2   Image Header Information

In addition to the data values for each pixel, every **Dragon** image file contains a block of identifying information called the *header*. The information in this header is used to keep track of where image files came from, what they represent, and what processing they have received.

The different items of information stored in the image file header are called *header fields* or, sometimes, *metadata*. **Dragon** provides you with capabilities for examining header fields and for changing many of them. (See the chapter on Utility operations.) The most important header fields that you can list and/or modify are:

Identifying Fields

Number of Lines: How many lines are in the image represented by this file

Number of Pixels: how long each line is, in pixels

File Type: *I* if a normal or *radiometric* image file, *C* if a classified image *L* if a GIS layer or rasterized vector file.

Scene Identification: a text label that describes the primary location from which the data was recorded (should be the same for all bands of a given scene)

Subscene Identification: a second text label that identifies a subset of the primary location from which the data was recorded (should also be the same for all bands of a given scene)

Band Number: a label, usually but not necessarily a number, that identifies the band (if any) the image represents

Source: a three-character code identifying the **Dragon** operation that created the file

Comment: a phrase with any additional identifying information you wish to enter

Color File Name: Optional name of color scheme file to be used by default in displaying the image.

Statistical Fields

Mean value: arithmetic average of data in file, to nearest whole number

Maximum value: largest data value in file (range 0 to 65535)

Minimum value: smallest data value in file (range 0 to 65535)

Variance value: a measure of the 'spread' of values around the mean

Standard Deviation: another measure of spread

Histogram: table of frequencies counting how many pixels there are in the image that have each possible value

Special Fields

Names for each class: Labels for classes or specific data values in the image

Georeferencing information: Fields that define the relationship of the image coordinates to an external coordinate system.

Calibration information: Fields that define a mapping between image data values, which range from 0 to 65,535, to some external measured quantity, such as temperature or elevation.

Dragon recalculates image statistics after any operation that could change image values, and generally tries to maintain the header information in the computer memory in a correct state. **Dragon** uses the identifying fields in the header for checking logical consistency in multi-band operations, and the statistical fields for computing automatic contrast stretch parameters, density slicing boundaries, etc.

1.3.3 The Main Memory Image

Dragon's operation depends on the concept of a *main memory image*. All operations on images that produce a single band result are considered to take place in the main memory image area, and all results are left in the main memory image. Consequently the main memory image usually holds the results of the most recent computation. This is a central concept in the organization of **Dragon**.

The main memory image is very convenient because it lets you chain together a series of operations without having to write intermediate results out into files on your disk. Suppose, for example, you wanted to add together two image bands, and then subtract a third from the result. You would choose the operation for adding image bands and tell **Dragon** what image files held the data. **Dragon** would read the first set of data into the main memory image, then read and add successive lines from the second set of data. The result of the addition would be left in that memory image. Next you would choose the operation for subtracting one image from another. This time, when **Dragon** asked you what files held your data, you would select the current image in memory (referred to by the 'special filename' =**M**) for the first set of data. There would be no need to save the results of the addition in an image file and then read that data back into memory for the subtraction. Memory images make the sequence of operations quite efficient.

1.3.4 Coded Image Files

A standard image file is assumed by **Dragon** to be *radiometric data* derived from a sensor. The numbers which represent each image element are therefore physical measurements of the emitted or reflected radiation received by the sensor, and arithmetic operations (such as histogram normalization) can be reasonably applied to these measurements.

Some files, however, have the same structure as radiometric image files, but the numbers are codes or category labels rather than true physical measurements. They may carry order information but never magnitude information. For this reason, it does not make much sense to perform arithmetic operations on such coded images. **Dragon** refers to these files as *coded image* files, and processes them slightly differently than radiometric image files.

Structurally, a coded image is identical to any other image: a two dimensional array of numeric values, where each pixel value is associated with some position on the ground. **Dragon** will not stop you from performing arithmetic on coded files, but it will warn you that you are doing something unusual.

Coded image files can arise in several different ways. For example, a classified image results from applying a classification (categorization) procedure to one or more image bands. A classified image is distinguished by having a C in its header *File Type* field, and by having associated class names. Figure 1.5 shows an example of a classified image, including a legend that shows the association of each class with its display color.

When **Dragon** displays a classified image, it will not perform any default contrast stretch (since that would change the class values). Also, a classified image has its own associated color scheme, created when you save the image after performing a classification. This special color scheme, which can be modified using the **Utility**⇒**Colors** operation, is always used to display the classified image in color unless you explicitly request a different color scheme.
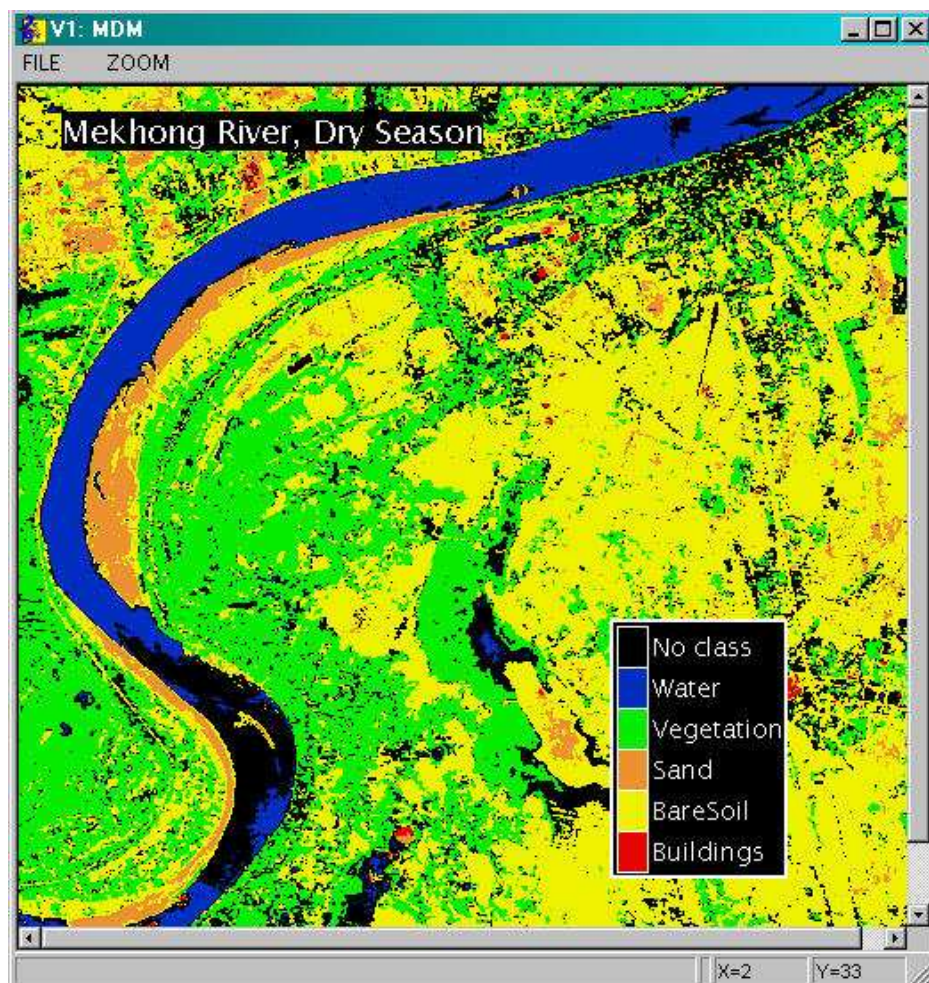

Figure 1.5: *A coded image, with legend*

A coded overlay image can also be generated using the **Geography**⇒**Vector** operations, and a coded layer file may be imported from a Geographic Information System (GIS) package. The **Geography**⇒**Aspect** and **Geography**⇒**Combine Layers** operations also produce coded layer files. Layer files are characterized by having an L (for *Layer*) in the header *File Type* field, and by including somewhat different ancillary information than a classified file.

Dragon uses the *File Type* field in the image header to determine whether an image is coded or not. If you want to perform some unusual operation such as a contrast

13

stretch on a classified image, you can do so by changing the value of the *File Type* from *C* to *I*. This will 'fool' **Dragon** into treating the file like a normal image. Likewise you can make a normal image look like a classified image to **Dragon** by changing its *File Type* to *C*. **Dragon** permits you to do almost any sort of bizarre processing, but tries to protect you from doing so by accident.

1.3.5   Mask Images: Processing Subregions of Images

Many of the **Dragon** analysis and classification operations can be limited to specified subregions of the image data. These regions, which are defined by a *mask image*, can be arbitrarily complex in shape and contain numerous disconnected regions. In general, you can think of the mask image being overlaid on the image data. Wherever the mask image is zero, the image data value is ignored and the resulting image will have zero value. Where the mask image is not zero, computation proceeds as usual. Figure 1.6 illustrates the concept of using a mask.

Because the mask data need not have the same dimensions (number of lines and pixels) as the image data, an arbitrary choice must be made about image data *outside* the mask image area. In most operations, pixels which are completely outside the mask image are treated as though the mask were **not** zero.

The **Classify⇒Recode** operation uses the mask image slightly differently. In this operation, pixel positions where the mask image is non-zero, as well as any which are outside the area of the mask image, will be recoded. Positions where the mask image is defined and is zero are left unchanged.

The **Utility⇒Histogram** operation also allows you to specify a mask image. If you do specify a mask, the histogram calculations include only the pixels where the corresponding mask image pixel is non-zero. Among its other uses, this function allows you to see a histogram of the raw data values of only those pixels which were classified as members of one particular class. For example, to see a histogram of the water pixels in your image, create a mask image using **Classify⇒Recode** based on your classified image with all non-water pixels recoded to zero. Then specify that mask as the mask image in **Utility⇒Histogram**.

All of the operations which permit use of a mask image (generally, most of the **Enhancement** and **Classification** operations) provide a field to specify a mask image file name. If this field is left blank, no mask will be applied.

1.3.6   Color Schemes and Color Files

A *color scheme*, in **Dragon** terminology, is an association between each of the 256 possible image data values (0 to 255) of an 8-bit image, and the color in which pixels with that value should be displayed. A color scheme may assign the same color to more than one data value. Frequently a contiguous range of values will be assigned to the same color, for example, all values between 16 and 31 might be assigned to light-green. You can, however, assign the same color to dissimilar image data values. This sort of color scheme is frequently used for special effects.

Note that you can not use a color file in combination with a 16-bit image file.

Color schemes are stored in *color files* until they are used.

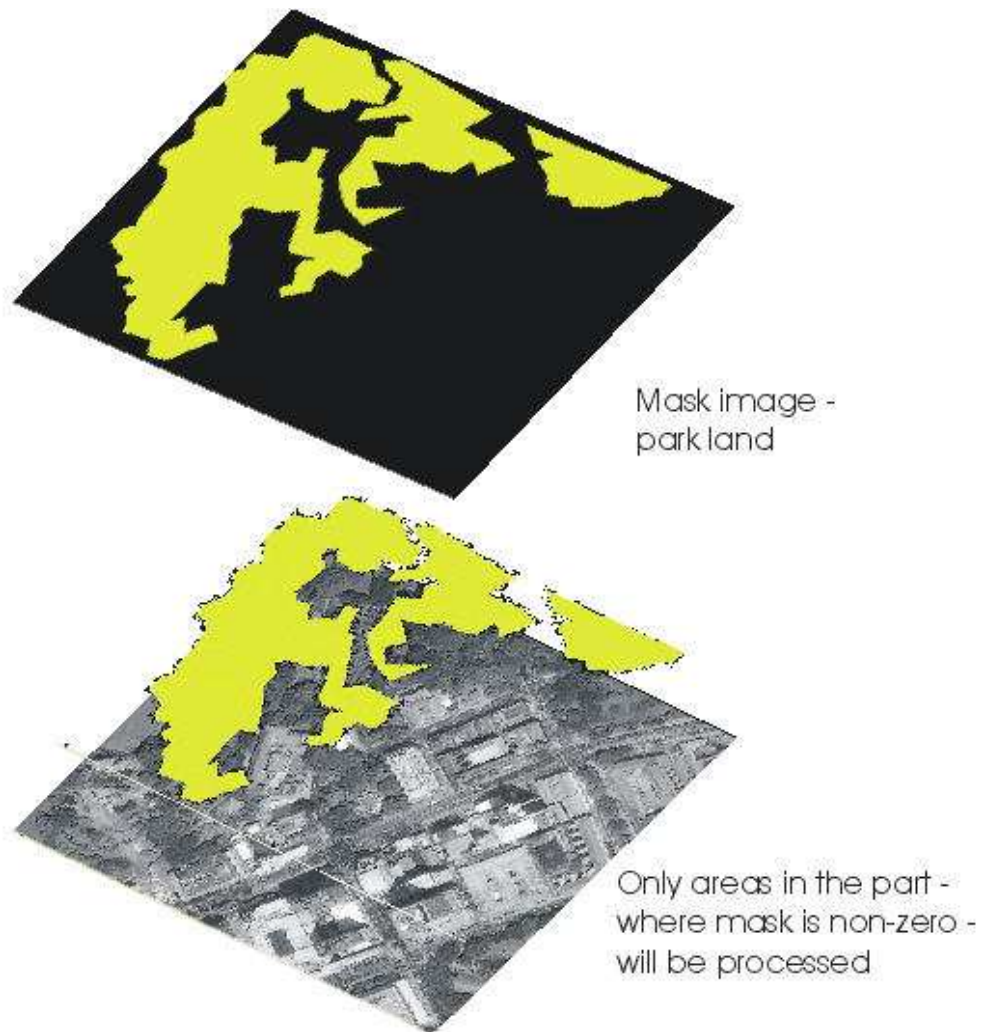Copyright ©Goldin-Rudahl Systems, Inc

Figure 1.6: *Mask images can restrict processing to relevant areas*

Color files can have any name up to thirty two characters long. The file extension (filetype) must be **.clf** (or **.CLF**). You should not create or modify any color files with names of the form **defxxxxx.clf**. These are standard color files that **Dragon** uses in various operations. Starting with **Dragon** Release 5.10, the standard color files are located in a subdirectory within the system directory, called **defcolor**.

Dragon includes an operation for defining and modifying color schemes and saving them in color files Thus, you can have control over the color assignments in your images. On the other hand, **Dragon** has a set of standard color schemes that often provide good results. You do not have to define your own color schemes unless you want to achieve some particular effect not possible with the standard schemes.

Using **Utility**⇒**Header**, you can store the name of a color file in the header of an image file. Then **Dragon** will use that color scheme, by default, whenever you display the image. Classified images always include the name of a color file in their headers, as discussed in above. Note that **Dragon** assumes that a color file recorded in an image file header is located in the same subdirectory as the image itself.

1.3.7 Other Dragon Files

There are several other kinds of data files used in **Dragon**. Information on the structure of the files listed here, and other files used by **Dragon**, is included the *File Formats* chapter.

First, there are *signature files*. Signature files store *training signature statistics*, values that are used to guide the classification process by defining typical values for each class. Signature files are created by the **Classify⇒Edit Signatures⇒Save** signature operation, in Classification. These files can have any file name (no embedded spaces), but their extension (filetype) must be **.sig**.

The second type of subsidiary file is the *polygon file*. Polygon files record the boundaries of training areas selected in the **Classify⇒Training** operations Polygon files can have any file name but must have the extension **.ply**.

Another kind of file used by **Dragon** is a *script file*. A script file is a text file of **Dragon** commands, created with any word processor or editor. **Dragon**'s script capability makes it possible to execute almost any series of commands automatically, without any user interaction. There are a few rules that must be followed, but for the most part, commands can be included in a script file in the same format as if you typed them on the **Dragon** command line. (See the section below on ***Dragon User Interface***.) Script files can have any name, but must have the extension **.spt** (or **.SPT**).

Dragon vector files, which have the extension **.vec**, are used to capture the coordinates and attributes of points, lines and polygons created in **Geography⇒Vector**. The captured data can be exported to other software packages which use vector data, or they can be read by **Geography⇒Vector** to become the basis for creating additional vectors, or to be transformed into a raster data layer.

A *recoding table file* may be used in combination with the **Classify⇒Recode** operation to provide an arbitrary remapping of image data values. Refer to the chapter on *File formats* for details.

Another ancillary file type is the *measurement data file*. Measurement data files are used to capture in detail the data which are displayed by the line profile and area histogram displays in the **Geography⇒Measure** operation. These output data files allow you to do statistical or graphical analyses on image data gathered from **Dragon**, using spreadsheet, statistics, data base, or presentation graphics software. The file format for the output data file is a generic comma-delimited form that is intended to be easy to import into other packages.

There are three types of files that **Dragon** uses or creates as part of the process of generating coefficients for geometric correction or registration. The most important are *coefficient files*, which have the standard file extension **.cof**. Coefficient files are used to record the regression coefficients calculated by **Geography⇒Calculate**. They serve as a primary input to the **Geography⇒Register** operation which actually performs the geometric correction/registration process.

Ground control point files (with the standard extension **.gcp**) record coordinates and user-defined labels for ground control points. GCP files can hold either image or map coordinates. They are created as an output of the **Geography⇒GCP** operation, and serve as input to **Geography⇒Calculate**.

Rule files are used to specify how to assign result values in the **Geography**⇒**Combine Layers** operation. Each *rule* in one of these files associates a particular output value with a set of input value ranges, across multiple bands. If the values of input bands all fall into the specified ranges, the designated output value is used in the result image. Rule files can be generated using the **Geography**⇒**Create Rules** operation.

Dragon also creates a number of temporary files during processing. In general, the detailed structure of these files is not important to the user and has not been included in this manual.

One special category of file which **Dragon** creates are those with any file extension including one or more dollar signs (**\$**). These files are intended to be temporary; that is, they are usually deleted automatically by **Dragon**. If you see any of these files in your directories when **Dragon** is not running, you may delete them without loss. On the other hand, you should not delete any such files while you are using **Dragon**; this might create serious problems.

1.3.8 Filenames and Special Filenames

Structure of a Filename

The complete name of a file, using the Windows or MS-DOS operating systems, has the appearance

```
C:\ppp\aa\tttt\hh\Nodename.ext
```

This name consists of four parts:

- The *drive letter*, shown as `C` in the example, is the disk drive where the file resides. The drive letter is followed by a colon (`:`).

- The *path*, shown as `\ppp\aa\tttt\hh\` in the example, consists of a list of directories and subdirectories separated by the backslash (`\`) character. This path specifies the location in the directory tree on the specified drive where the file is located.

- The *node name*, shown as `Nodename` in the example, is the name which actually identifies the desired file. Usually, when one refers to a file name, one is talking only about the node name portion. In Windows, node names can contain almost any character, including blanks. When you create or use files in **Dragon**, we strongly recommend that you avoid using special characters (including spaces) other than underscore or dash in your node names, and that you keep node names to 32 characters or fewer.

- The *extension* is the final part. Most of the standard extensions used in **Dragon** are limited to three characters, but this is not a requirement. The extension provides some additional identification of the particular file. However, by custom, the extension is used to distinguish particular types of files. For example, as discussed previously in this section, the file extensions **.img, .clf, .ply, .vec,** and **.spt** (among others) all have special meanings to **Dragon**.

Note

Unix and Linux have similar conventions for file names. However, in Unix/Linux, there is no drive letter, and a '/' is used instead of a '\' character in the path.

Selecting a File by Name

There are many times while using **Dragon**, when the user must specify a filename to **Dragon**. For almost all operations, **Dragon** already knows the drive, path, and extension portions of the name. Thus, you only need to enter the node name portion of the filename. You may, if you wish, provide the full *drive:path\nodename* form in order to access files which would not otherwise be available. Usually, you do not need to specify the file extension.

As discussed below, the **Dragon** menu system also allows you to browse through the files on your hard drives in order to find and select your desired file.

Special Filenames and the Status Window

Whenever **Dragon** permits you to specify the name of an image file to read, you are being asked which data the software should operate on. Frequently that data is stored in a file on your hard disk, or on your network. However, **Dragon** also permits you to operate on data already 'in memory'.

Dragon provides several *memory images* which you can use as input in some operations. You tell **Dragon** to use one of these as the source for your operations through *special filenames*. A special file name is a sequence of characters which would be illegal as a real file name. The most important ones are listed below.

- The **=M** (the 'M' stands for the English word *main*) memory image contains the results of the most recent image analysis operation. This means that all **Enhance** operations, and all **Classification** and **Geography** operations which produce an image result, place that result into **=M**. Because **=M** can be used as the (or an) input to most of these operations, it is very easy and fast in **Dragon** to chain a sequence of operations on the same data. In addition to the analysis operations, the **Display⇒1 Band**, **Display⇒Gray**, and **Display⇒Overlay** operations, when reading from a disk file, also load that file into **=M**.

- The **=R** (*Red*) memory image contains the image which was assigned to the red color plane during the most recent **Display⇒3 Band** operation. **=R** can be the input to most display and many analysis operations.

- The **=G** (*Green*) memory image contains the image which was assigned to the green color plane during the most recent **Display⇒3 Band** operation.

- The **=B** (*Blue*) memory image contains the image which was assigned to the blue color plane during the most recent **Display⇒3 Band** operation.

- The special filename **=C** does not correspond to a single memory image, but rather to the collection **=R**, **=G**, and **=B**. **=C** (the 'C' stands for the English word *composite*) provides a convenient way to use a multi-band image as a reference or background for operations such as **Utility⇒Cursor** and

Geography⇒GCP. Note that it does not make any sense to specify **=C** as the input to an analysis operation.

In all cases, the memory images contain the true data values with no stretch applied. Thus, it is very fast and easy to experiment with viewing the effects of the different stretch algorithms supplied. For those operations which permit you to specify a composite image (**=C**), the composite image is displayed using the same stretches which were most recently used for **Display⇒3 Band**.

There may be times, however, when you want to change the image values (for example, by applying a histogram stretch). **Classify⇒Recode** provides some very powerful capabilities for doing this. The results of such a stretch can be saved as a new image using **File⇒Save**.

Not every operation or image file name field will accept every special filename. Generally the limitations are reasonable or obvious; however, the Help message for each field tells you which ones are permitted.

Of course, you also cannot specify a name of a special file if the corresponding memory image has no image loaded into it. To help you to remember which images are loaded, **Dragon** provides a *Status display* listing the memory files as well as other information. To see the Status window, press **<F6>** or click on **Status** radio button with your mouse. The Status window alternates with the History display. You can switch to the History display by pressing **<F7>** or selecting the **History** radio button.

1.3.9 How Dragon Finds a File

There are two types of files which **Dragon** needs to be able to locate:

- **Dragon** *system files*, a category which includes the executables, help files, configuration files, and various other kinds of files.

- *Data files*, including all image, script, signature, vector, polygon, and registration files, plus most color files.

The system files are almost never specified by name during normal use: the file names are built into **Dragon**. In order for **Dragon** to function properly, these files must all be in the currently-logged drive and directory while **Dragon** is executing. (If you do need to access or name one of the system files, you must include a path specifier consisting of '**.**').

Dragon also uses various kinds of data files. To assist you in organizing and accessing your hard disk, **Dragon** provides a facility for a *default image path* or *default data directory*. This is a drive and path specification where **Dragon** will automatically look for any input file which you may name.

For example, if the default data directory is **C:\images** and you type, in answer to some question requesting an image file name, a file name of **BangkokPhoto**, **Dragon** will look for the file

```
C:\images\BangkokPhoto.img
```

If the file is not actually located in the **C:\images** directory, your will receive an error message saying the file cannot be found. If, in fact, the file is on some other

directory (for example, on **C:\landsat**), you would have to specifically name the directory by typing

```
C:\landsat\BangkokPhoto.img
```

You must specify a value for the default data directory during the installation of the **Dragon** software. The default data directory can be changed while **Dragon** is executing by use of the **File⇒Preferences⇒Paths** command.

There is also a default location, called the *default file creation path*, where **Dragon** will create new data files. Like the default image path, the initial value for this path must be specified during the installation process, but can be changed via the **Preferences** submenu. Sometimes it is useful to have these two paths refer to the same location, so that you will automatically be able read files that you write, without changing the directory specification.

We have been discussing how to type a file name in order to identify a file for **Dragon**. In fact, a user rarely needs to type file names, since it is always possible to browse and select files graphically by using the **Dragon** file chooser. To bring up the file chooser, click on the down arrow button at the end of a file name data entry field, or press **<F3>**.

Several **Dragon** operations allow you to select multiple files in one step. The selected files can be typed, in which case they are added to a list when you press Enter, or they can be selected in a file chooser. Once the list of files has been created, the user can change their order or delete specific items using appropriately labeled buttons.

1.4   The Dragon User Interface

The term *user interface* refers to all aspects of an interactive computer program that involve the two-way communication between the program and the user. Unfortunately, users vary in their prior experience, knowledge, and ability. Thus, it is difficult to design a single interface suitable for all users.

1.4.1   Dragon Overview

Interfaces that offer multiple windows on the screen simultaneously, such as Microsoft Windows variants, Motif, Gnome and KDE, provide great flexibility for the user. Menus can be displayed in one window; images can be displayed in other windows; messages and text reports can be shown in transient windows which disappear when the user dismisses them. Current versions of **Dragon** take advantage of this flexibility providing a modern, fast, and flexible style of interaction.

The **Dragon** interface consists of the following components:

The Dragon menu client This is a window which presents a set of menus for selecting operations within **Dragon**. When a menu item is selected, typically the menu client will display a data entry form, called in **Dragon** terminology a *response panel*, where you can supply the parameters necessary to the operation. The menu client also provides a window for displaying image status information or command history, an optional command line, and several status areas at the bottom of the window. The status area on the left indicates whether **Dragon** is currently working on an operation or is ready for input. The larger status

area on the right displays information messages, such as the number of pixels truncated in the Filtering operation or changed in the Relaxation operation.

Help windows When your mouse enters the boundary of a menu item, or a control on a response panel, **Dragon** will wait for a short interval and then pop up a *Help window* near the control. The help window provides you with information on the meaning and use of the current control, in the currently selected natural language. To make the help window disappear, you can either move the mouse away from the control, or click in the control. You can change the delay between the time that your mouse enters a control area and the time that the help appears by editing the **HELPDELAY** value in the **dragon.cfg** configuration file.

Viewports All images are displayed in special *Viewport* windows. **Dragon** provides two viewports, labeled Viewport 0 and Viewport 1. Most image processing operations allow you to select which viewport to use for displaying the results. Viewports have their own menus, which control image specific operations such as zooming and printing images. A status line at the bottom of each viewport provides information such as the current cursor (mouse) position and brief instructions about the current operation.

InfoViewports *InfoViewport* windows are small sub-windows used for displaying ancillary graphics, such as histograms or scatterplots. InfoViewports are fixed size. They can be iconized or closed when not needed. Multiple InfoViewports can be visible simultaneously.

Report Windows A number of **Dragon** operations are available to allow you to view information about images, signatures and so on. This sort of information is displayed in a *report window*. The results tables created by classification operations are also shown in report windows. Like InfoViewports, report windows can be dismissed when you are finished with them. Most also offer the opportunity to save their contents as as HTML file.

Button Panels Some of the interactive operations in **Dragon**, such as **Classify⇒Training**, display a floating panel of push buttons which can be used to invoke particular actions. The buttons on these panels are enabled appropriately depending on the state of the interaction. These buttons always have function key equivalents to invoke their functionality.

Query Dialogs Some of the interactive operations in **Dragon** also display simple query dialogs. For instance, if you click on the **New Class** button during **Classify⇒Training**, **Dragon** will display a dialog where you can enter or select the next class to train.

Error and Message Boxes As is the case with most Windows applications, **Dragon** uses pop-up message boxes to communicate the existence of error or warning conditions. Usually these boxes have a single button, labeled **OK**, for you to dismiss them. Occasionally, they will require that you make a choice (e.g. **Exit** versus **Continue**).

Figure 1.7 shows many of the important aspects of the **Dragon** Menu Client.
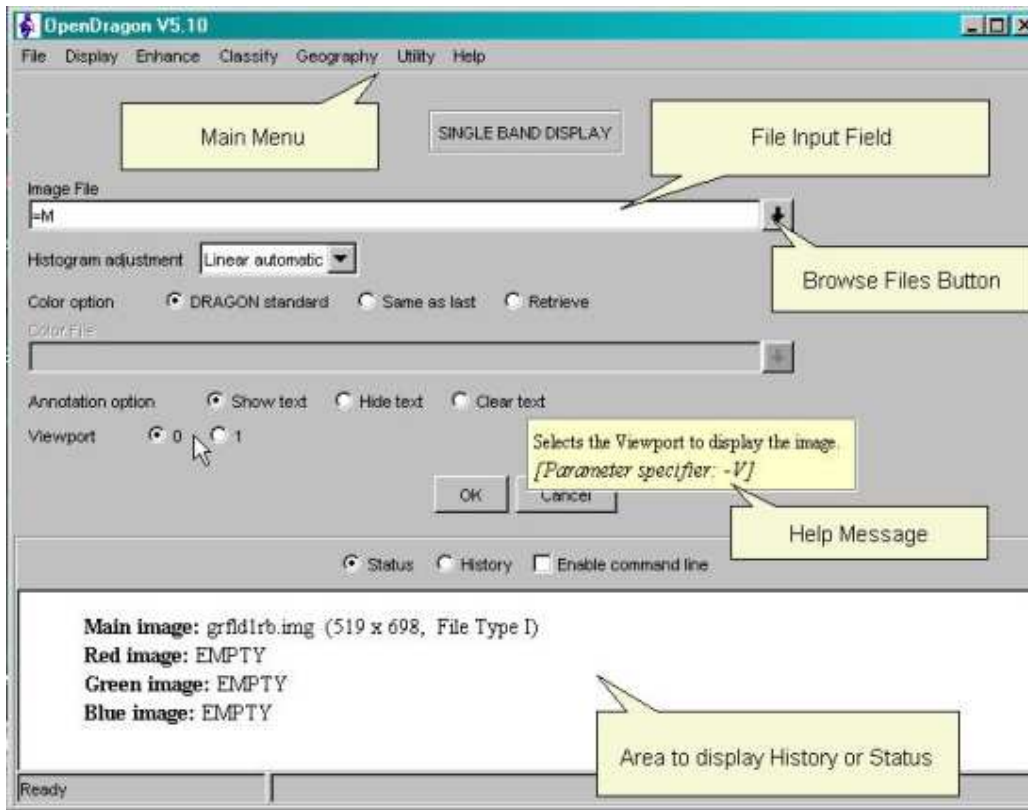
Figure 1.7: *User interface components*

1.4.2   Interacting with the Viewports

As explained in the last section, **Dragon** provides two special windows called viewports which are used to display images. Each viewport has its own menu bar and some specialized functionality, which is discussed in this section.

Interaction with a viewport is the place where differences between **Dragon Academic** and **Dragon Professional** are most obvious.

Dragon Professional can process images of (almost) unlimited size. For the current generation of satellites, a *full scene* image can contain up to about 32,000 rows and columns of data. Processing this much data is not a problem for **Dragon**, but the result will have about 1000 times more data than can appear on a typical computer screen. This gives rise to two conflicting requirements: you want to see the full scene results, but you also want to see (and work with) the full resolution contained in your data. **Dragon Professional** lets you do either of these, but not both at the same time.

Each time you display a new image, whether from disk or as the result of some image processing, the entire image will be displayed but in a reduced resolution. We call this the *overview* image. The actual scaling factor, the ratio between the displayed data and the full image, is calculated based on your screen size, viewport size, and the full image size. This scaling factor is shown in the status line below the viewport.

In order to see more detail in part of the image, you can select a *region of interest* by positioning the mouse at one corner of the desired region. Press and hold the left button while dragging the mouse to the opposite corner of the desired region.

During this time, a white rectangle will outline the current region and the mouse coordinates will be displayed on the right size of the viewport status line. Normally, if the image is georeferenced the coordinates will the the actual UTM values.

After you select a region of interest, you can further refine your selection by choosing a new region within the current region using the mouse in the same way. Alternatively, you can return to the overview by clicking the *right* mouse button.

There may be times when you begin to select a region but change your mind. There are two ways to cancel a selection in process:

1. If the selection you make is too small (less than about 200 image rows x 200 image columns), **Dragon** will interpret that as a cancel. Thus, if you simply press and then immediately release the mouse button no region will be selected.

2. Alternatively, if you begin a region selection and then move the mouse outside of the viewport, that will also be considered to cancel the selection.

In addition, you can never select a region such that the display scale will be larger than 100%, which is one data element per screen pixel.

This region selection process works well when you simply want to view your processed images. However, there are several operations where your need to *interact* with the image by selecting particular points, possibly individual image data points, for classification training, image registration, etc. These additional requirements make the above process somewhat more complex.

The six (at present) interactive operations fall into two groups:

- Operations where the desired points **must** be chosen at maximum resolution. These operations are classification training, ground control selection, and **Utility⇒Cursor**. These operations are almost meaningless unless you are selecting individual data points. Therefore, you must select a region of interest before proceeding with the point selections. You can return to the overview and select a new region as often as you wish.

  The requirement that your region of interest must have full resolution does however, restrict you to selecting a region that will fit on your screen.

- The other interactive operations are the **Geography⇒Polygon** operations. These operations permit you to draw vectors either in the overview image or in the detailed 1:1 or 100% region image. Because mouse clicks in the overview image are used for drawing, you must use a specific command (**<F11>**) to begin the region selection.

The **Zoom** menu allows you to control the degree of magnification of the image during interactive operations. There are four items on this menu which are enabled only when an interactive region of interest has been selected and displayed:

- Magnify 3x
- Enlarge
- Restore
- Overview

The first two items enlarge the image, so that you can see more detail. To do a 3x zoom of the image, select the **Magnify 3x** item, then use the mouse to select

the location on the image that you want to be in the center of the zoomed image. To zoom to an arbitrary expansion factor (up to 10x), select **Enlarge**. **Dragon** displays a dialog for you to enter the desired zoom factor. After typing a number between 1.0 and 10.0 and clicking the **OK** button, click on the image to select the center of the zoom area.

Choose the third item, **Restore**, to redisplay the image region at full resolution (1:1 scale).

The fourth item reduces the image until it fits in the current size of the viewport. In contrast to the previous two menu options, **Overview** does not require any interaction. As soon as you select the menu item, the image will be shrunk to fit the viewport size and displayed.

After you zoom in, the viewport disables the two 'zoom' menu items listed above and enables the **Restore** option. Thus, you cannot do multiple successive zooms. If you enlarge the image to an arbitrary factor, and want to zoom in further, you need to restore the image to normal resolution first, and then re-execute the **Enlarge** option using a different magnification factor.

The viewports provide facilities for printing the current image, saving all or part of the image to a file, or copying all or part of the image to the clipboard. All these capabilities are accessible through the **File** menu on the viewport.

To print the currently-displayed image, choose the viewport **File⇒Print** menu item. **Dragon** will display a standard Windows print dialog, allowing you to select a printer where the image data should be sent. Obviously, your printer must have color capabilities if you want to correctly represent a color image. Almost any printer will do a reasonably good job with a gray-scale image.

To save the current image in TIFF format, a format that can be imported into most graphics and word-processing programs, choose **File⇒Capture**. The viewport will display a file selection dialog so that you can browse to the location where you want to save the image, and enter a filename. After you click on the **Ok** button on the file dialog, you decide what to save. If you want to save the entire viewport, press **<F1>**. Otherwise, place the mouse cursor on the image at the upper left corner of the area that you want to capture. Press the left mouse button, and drag the mouse to the lower right corner of the area you want to select. A flexible box will follow the mouse. When you release the mouse button, the box will disappear. The viewport will convert the image contents to TIFF format and save them in the designated file.

To capture all or part of the image to the Windows clipboard, choose **File⇒Copy**. Once again, you can copy the entire viewport to the clipboard by pressing the **<F1>**, or use the mouse cursor to choose a subarea. The viewport will copy the all the data or just the section included within the selection box to the clipboard. From there you can past the data into graphics programs or word processing documents.

To cancel a copy or capture operation, drag the mouse outside the viewport area.

At any time that there is an image displayed in a viewport, you can find out the X (pixel) and Y (line) coordinates of any point in that image. As you move the mouse cursor over the image, two fields at the bottom right of the viewport display the continuously updated coordinates of the cursor.

> **Note**
>
> You can control whether the mouse position display uses geographic
> coordinates (if available) or image coordinates, by editing the file
> **dragon.cfg**, in the **Dragon** installation directory, and modifying
> the value of the STATUSBAR3 field. You can also control the for-
> mat for displaying the image scale, by modifying the STATUSBAR2
> value.

On the other hand, the viewport only accepts point selections (clicks) at very specific
times in the processing sequence. If you try to select a point on the image when
the viewport is not accepting points, the viewport will beep and ignore the click.
You should note that some of the interactive operations, such as training and vector
definition, will not accept points until the button panel and any other subsidiary
controls are visible. This sometimes takes a few seconds. Also, some operations
require you to choose a detail region before you can select points.

1.4.3   Menu Mode

Dragon's user interface design accommodates varying levels of skill and experience
by providing two quite different modes of operation within the same program. For
users new to **Dragon**, to remote sensing image processing, or to computers, the
program offers the *Menu* mode of operation. **Dragon** menus gives detailed and
explicit information and instructions about each step needed to accomplish a task.
It prompts for each piece of information that you must supply and gives immediate
feedback if you make an error. Menu operations are discussed in the following
section.

The Menu mode offers considerable assistance to a new user. However, experienced
users may get impatient if they must go through prompts in order to accomplish an
operation. For these users, **Dragon** provides the *Command* mode. In Command
mode, the program does not prompt for specific information. You indicate exactly
what you want by typing in a command that identifies an image processing operation
plus command *parameters* that further specify the action: the source of data, the
color scheme to be used, etc.

In addition to the Menu and Command modes, you can also use **Dragon** in *Script
mode.* Script files hold **Dragon** commands, in exactly the same format you would
use in Command mode.

The **Dragon** menu can be presented in a variety of languages other than English.
You can change the language used by accessing **File⇒Preferences⇒Language**.
(The new language will be shown the next time you start **Dragon**.) When **Dragon**
is configured for a language other than English, all menus, prompts, messages, and
help will be displayed using that language, in the appropriate character set.

When working in Menu mode, **Dragon** uses *menus*, which are lists of items to select,
and *response panels*, which are data entry forms to fill out.

Dragon menus work like those in any other Windows application. Click on the
menu to drop it down and display its items. Use the mouse to click on the item you
want, or use the arrow keys to navigate to that item and hit the space bar to select
it.

Response panels act like standard Windows data entry forms, but have a few additional characteristics.

- Pressing the **<Enter>** key is equivalent to pressing **<Tab>**; it simply causes the focus to move to the next field. It does not activate the **Ok** button, as is the case in many programs. This allows you to cycle through the data fields multiple times before triggering the operation.

- Pressing the down-arrow key is equivalent to pressing **<Tab>**; it causes the focus to move to the next field. Pressing the up-arrow key is equivalent to **<Shift-Tab>**, moving backwards through the fields on the form.

- Pressing the **<F3>** key in a field where a filename is required will bring up a file dialog so you can browse to the file. You can accomplish the same thing with the mouse by clicking on the down-arrow graphic at the end of the field.

- Pressing the **<F3>** key in a field represented by a drop down list control (also called a 'combo box') causes the choice list to drop down.

- Pressing the **<Home>** key in a field that holds some data clears the field. Left and right arrow keys move the insertion cursor within the field.

- Pressing the **<End>** key in a field that holds some data moves the insertion cursor to the end of that data.

- In some cases, the information that you enter in one field will cause changes in other fields when you tab off the original field. The most common change is to enable or disable other fields.

- To indicate that you are finished entering parameters and want to actually execute the operation, click on **Ok** or press the **<F1>** key. To cancel the operation, click on **Cancel**, or press either the **<F10>** or **<Esc>** keys

The bottom section of the menu client is a multi-function text area. This section can be used to display image status information, or command history. You choose which category of information to display by clicking on the appropriate radio buttons, or by pressing function keys: **<F6>** for Status or **<F7>** for history.

Previous versions of **Dragon** also used the multi-function text area to display help for whichever response panel field or menu item was currently under the mouse. This behavior has been replaced by the *Help Window* discussed above.

When the Status functionality is active, **Dragon** displays a summary of what image information corresponds to the special files **=M**, **=B**, **=G**, and **=R**. This includes the image file names, if the data have not been modified, and the image dimensions.

When the History functionality is active, **Dragon** displays a list of all the commands that have been executed, in the **Dragon** command language discussed in the next section. Even if you do not have History selected, the history information will continue to accumulate. Thus you can switch among the different information modes (*Status* and *History*), without losing any information.

In additional to the information mode radio buttons, the bar between the response panel area and information display area contains a check box labeled *Enable command line*. This check box displays or hides the command line; pressing the function key **<F2>** is equivalent. The operation and use of the command line is discussed in the next section.

1.4.4 Command-Line Mode

The Menu and Command modes of interaction look quite different. However, the logical structure of operations and the specifying information (parameters) each operation requires are almost exactly parallel in the two modes. You can move easily from one mode to the other, with a single key. Thus, you have great freedom in how much of each mode you use. As you learn more about **Dragon**, you can start to use the Command mode for some familiar operations while still using Menu mode for more unfamiliar ones. The choice is entirely yours.

In **Dragon** Command mode, you use the same menu choices, operations, parameters and parameters values as in Menu mode. You just express them somewhat differently.

All information that you type in Command mode is typed on the *command line.* To make the command line appear, click the check box labeled *Enable command line*, or press **<F2>** when a response panel is visible. The command line is a single line text entry area between the response panel area and the status/history area. It is labeled **CMD>>**.

When the command line is visible and active, **Dragon** still allows you to operate the menus. If you select a new operation from the menu, the command line will disappear and you will be returned to Menu mode. On the other hand, if there is a response panel displayed when you turn on the command line, the panel will remain on the screen, with all fields disabled. Even though the fields are disabled, you can still view the help messages for each field. This can assist you as you work to construct a **Dragon** command in the command line area.

Command mode input consists of four kinds of items:

- *Commands* are short sequences of characters that typically are abbreviations of menu labels. All the top-level menus have single letter commands. Other menu items have three or more letter commands. For example, **D** is the command to identify the **Display** operations menu, while **1BA** is the command for the single-band color display operation.

- *Parameter specifiers* (also called *p-specifiers*) are codes that identify parameters. All p-specifiers begin with a dash, followed by one or more letters or numbers. For example,

  -F (Image File parameter)

  -UBP (User-defined Upper Break Point parameter)

  The p-specifier code name is related in a more-or-less direct way to the name of the parameter as expressed in the response panel prompt string. Similar parameters used with different operations have the same or similar specifiers, to minimize confusion.

- *Parameter Values* are numbers or strings that indicate your selected value for an operation parameter. The permissible values for each parameter are the same in Command mode as in Menu mode. In Command mode, you simply type the parameter value immediately after the corresponding p-specifier on the command line, with at least one space between them, e.g.

      -F Bangkok2

27

If a parameter value includes spaces, e.g. in setting the Comment field in the **Dragon** header, you should enclose entire parameter value in quotation marks, e.g.

```
-CMT ''TM Band 2 of central Bangkok, uncorrected''
```

- *Execute Code*: In the Command mode, as in the Menu mode, **Dragon** does not execute the operation you have chosen until you are ready. An explicit Execute Code is necessary, since it may take more than one line of input to specify all parameters. To execute a command in Command mode, press **<F1>** or type the following at the end of the command (preceed by at least one blank):

    ```
    -.
    ```

(dash then period)

Because the Command mode uses an explicit Execute Code, you can change your mind about parameter values you have entered. Suppose you entered the following command string:

```
D 1BA -F Bangkok2 -H N
```

(Notice that spaces or carriage returns must separate each command, p-specifier or parameter value from the others items in the command string. If you use filenames or comments which include spaces, you must surround the entire filename or comment with quote marks.) This example requests that no histogram stretch be done. Since the Execute Code was not entered, **Dragon** does not immediately perform the display operation. Instead, it waits for more parameter input and/or the Execute Code. Suppose you change your mind and decide that you wanted the automatic linear stretch after all. You could type on the command line:

```
-H L -.
```

The newly-specified value would replace the one you typed in previously. The Execute Code would cause **Dragon** to start the actual processing for the display operation.

Note

Dragon allows you to use **<F1>** in the Command mode as an alternative to -. to provide consistency with the menu mode. The dash-dot Execute Code is the only way to start execution from inside a command script, however.

The examples above demonstrate how Command mode input strings are constructed. A full command requesting an operation will consist of the command for that operation, a set of p-specifiers, each followed by its value(s), and finally, the Execute Code. You can insert carriage returns anywhere in that sequence, except between a p-specifier and its corresponding value. P-specifiers must always be on the same line as their associated values, separated by at least one space.

The specific commands and p-specifiers associated with each **Dragon** operation and parameter are shown in the corresponding Help messages. This information is also documented in the 'Details' section associated with each operation in the Manual Reader.

Sometimes you will want to repeat a previously-entered command, perhaps with a few changes. Pressing the up-arrow or down-arrow keys allows you to sequence

through the entire command history. If you press up-arrow in the command line, **Dragon** displays in the command line the text of the previous command or partial command that was executed. If you press the down-arrow key, **Dragon** displays the next line from the history. When you get to the first or last lines in the history, continuing to press the arrow key will not change the text displayed in the command line.

Note that the history will contain the command-line equivalent of all the commands you have executed in this **Dragon** session, whether in Menu or Command mode. Thus, you can use the command constructed in response to a Menu-mode operation as the basis for a Command-mode operation (by recalling the earlier command into the command line as described above).

You may notice that when it constructs commands in response to Menu-mode interactions, **Dragon** places all parameter values in quotes. This is perfectly legal; you do not have to remove the quotation marks.

1.4.5 Script Mode

A **Dragon** *script* is a normal text file that you create with a text editor program. Script files hold **Dragon** commands, in exactly the same format you would use in Command mode. A script can contain a single command or a long series of commands. There are a few restrictions on what can be done in script files, but for the most part, if you can use Command mode, you can use scripts.

To actually execute a script, choose **File⇒Script⇒Run** and navigate to the desired script. Alternatively, you can type the *at-sign* character on the command line *(@)* followed by at least one space and then the name of the script file, e.g.:

```
@ MYCLASS
```

This starts **Dragon** reading and executing the commands in the file **MY-CLASS.spt**. (**Dragon** knows that script files have the extension **.spt** and looks for this script in the default data directory since you did not specify any drive or path.) **Dragon** continues executing commands in this script until it reaches the end of the file, then it returns you to normal Command mode.

You can also use the keyword *RUN* as an equivalent to @, for example:

```
RUN MYCLASS
```

Scripts can be used for many purposes: to automate a series of operations that are performed frequently, to create 'packaged' command sequences for limiting the typing required from novice users, and to create demonstrations and interactive tutorials.

There are several limitations of scripts of which you should be aware: First, you cannot 'nest' scripts. That is, you cannot use the *at-sign* inside a script, to access another script. Second, you must be sure to preface all file names with their full path, unless they are located in the default image path area. Finally, there are a few operations in **Dragon** which are so interactive that it would be meaningless to execute them in script mode. These operations cannot be included in a script file.

1.5 Commands Used Throughout Dragon

There are a few commands or control keys that can be used in all operations or sub-menus within **Dragon**. Some of these commands have been discussed before in other contexts, but all are included here for for reference.

1.5.1 Special Keys

Personal computer keyboards typically include a set of special keys that do not exist on a typewriter keyboard. **Dragon** uses many of these special keys to trigger frequent types of action or as synonyms for frequently used commands.

The most commonly-used special keys on IBM-type keyboards are the *function keys*. These keys are usually located either in a row above the normal alphanumeric keys, or to the left or right of the alphanumeric keys. They are usually labeled **<F>** or **<PF>** plus a number, e.g. **<F2>**, **<PF5>**.

IBM-type keyboards also include a keypad that contains other special keys. These keys include the **<arrow>** keys (up, down, left, and right), the **<Home>** key, **<PgUp>** (*Page Up*), and **<PgDn>** (*Page Down*) keys. Some of the keypad keys are also used for **Dragon** actions, as explained below.

1.5.2 Commands for Operation Control

Several control key sequences are used for special purposes when you have selected an operation and are entering parameter values. These control sequences are effective in either Menu or Command mode.

- Press **<F1>** or **<Ctrl-E>** to begin execution of the current operation. (Typing **-.** in Command mode has the same effect.) These key sequences are also used to accept the current values and dismiss a sub-panel.

- Press **<F2>** or **<Ctrl-N>** to switch from Menu to Command mode or vice-versa when a response panel is displayed.

- Press **<F3>** or click on the downward pointing triangle in a response panel field to get a list of possible values for that field, or to bring up a file selection dialog, for filename fields.

- Press **<F6>** to switch the multifunction text area in a response panel so that it displays the current system status information.

- Press **<F7>** to switch the multifunction text area in a response panel so that it displays history of recent commands and messages.

- Press **<F10>** or **<Esc>** to cancel the current operation and remove the current response panel from view. These key sequences are also used to cancel and dismiss a sub-panel. (Note that if you select a new operation from the menu while a response panel is displayed, this effectively cancels the previous operation as well as starting the new operation.)

All control keys (those that start with **<Ctrl->** in the text) are entered by pressing down the key labeled **<Ctrl>** and the corresponding letter key simultaneously.