

# The KING manual

09/27/2023

Ian W. Davis Vincent B. Chen

### **Abstract**

KiNG (Kinemage, Next Generation) is an interactive system for three-dimensional vector graphics. It supports a set of graphics primitives that make it suitable for many types of graphs, plots, and other illustrations; although its first use was to display macromolecular structures for biophysical research. KiNG builds on MAGE, JAVAMAGE, and the “kinemage” (kinetic image) concept to deliver a full-featured Java application with a user-friendly interface and integrated editing features. It also operates as a Java applet to promote easy access to kinemages from a web browser.

# Contents

<b>1</b>	<b>Getting started</b>	<b>4</b>
1.1	What is a kinemage? . . . . .	4
1.2	Starting the KiNG program . . . . .	4
1.2.1	Download KiNG . . . . .	4
1.2.2	Make sure you have Java . . . . .	5
1.2.3	Launch KiNG . . . . .	5
1.3	Opening a kinemage . . . . .	6
1.4	Basic kinemage reading: Using KiNG to view a kinemage . . . .	6
1.4.1	Basic interaction . . . . .	6
1.4.1.1	Moving the kinemage around . . . . .	6
1.4.1.2	Identifying points . . . . .	6
1.4.1.3	Markers . . . . .	6
1.4.1.4	Turning groups on & off . . . . .	6
1.4.1.5	Re-centering . . . . .	7
1.4.1.6	Zooming in and out . . . . .	7
1.4.1.7	Clipping . . . . .	7
1.4.1.8	More modes of interaction . . . . .	7
1.4.2	Reading the kinemage text . . . . .	7
1.4.3	Using different viewpoints . . . . .	8
1.4.4	Other features . . . . .	8
<b>2</b>	<b>Input &amp; Output</b>	<b>9</b>
2.1	Opening a kinemage . . . . .	9
2.2	Working with multiple kinemages . . . . .	9
2.3	Appending a kinemage . . . . .	10
2.4	Saving kinemages . . . . .	10
2.5	Printing kinemages . . . . .	10
2.6	Exporting kinemages in other formats . . . . .	11
2.6.1	Exporting bitmapped images (JPEG, PNG) . . . . .	11
2.6.2	Exporting vector graphics (PDF, PostScript, <i>etc.</i> ) . . . .	11

<b>3</b>	<b>Advanced kinemage reading</b>	<b>12</b>
3.1	Finding points by name . . . . .	12
3.1.1	Whole word searches . . . . .	12
3.1.2	Substring searches . . . . .	12
3.1.3	Glob (simple regex) searches . . . . .	12
3.1.4	Perl5 regex searches . . . . .	13
3.1.5	Tips for faster searches . . . . .	13
3.2	Viewing animations . . . . .	13
3.3	Controlling presentation: The Display menu . . . . .	13
3.3.1	Stereo . . . . .	14
3.3.2	OpenGL . . . . .	14
3.4	Master buttons . . . . .	15
<b>4</b>	<b>Kinemage authoring: Using KiNG to edit a kinemage</b>	<b>16</b>
4.1	Merging kinemages together . . . . .	16
4.2	Editing text . . . . .	16
4.3	Editing views . . . . .	16
4.4	The hierarchy editor . . . . .	17
<b>5</b>	<b>The Tools</b>	<b>18</b>
5.1	Understanding the Tools and Plug-ins . . . . .	18
5.1.1	Extra measurements . . . . .	18
5.1.2	XYZ coordinates . . . . .	19
5.1.3	Pick objects . . . . .	19
5.2	Using KiNG with a multi-button mouse . . . . .	19
5.3	Navigate: Exploring in 3-D . . . . .	20
5.3.1	Z-rotation (“pinwheel”) . . . . .	20
5.3.2	Translation (“flatland”) . . . . .	20
5.3.3	Command reference . . . . .	20
5.4	Draw New: Reworking the kinemage . . . . .	21
5.5	Electron Density: Visualizing maps . . . . .	22
5.6	Update KiNG: Upgrading without the hassle . . . . .	23
<b>6</b>	<b>Configuring KiNG</b>	<b>24</b>
<b>7</b>	<b>Launching KiNG</b>	<b>25</b>
7.1	Running KiNG as an application . . . . .	25
7.2	Making KiNG an applet in a web page . . . . .	25
<b>8</b>	<b>Additional resources</b>	<b>28</b>
8.1	Included manuals . . . . .	28
8.2	Included kinemages . . . . .	28
8.3	External documentation . . . . .	28
8.4	Bug reporting and contact info . . . . .	28

<b>9</b>	<b>Copyright &amp; acknowledgments</b>	<b>29</b>
9.1	Copyright . . . . .	29
9.2	Licenses . . . . .	29
9.3	Acknowledgments . . . . .	30
9.4	Revision status . . . . .	30
<b>A</b>	<b>KiNG License</b>	<b>31</b>
A.1	What this license applies to . . . . .	31
A.2	The KiNG License . . . . .	31
A.3	Where this license came from . . . . .	32

# Chapter 1

## Getting started

### 1.1 What is a kinemage?

A kinemage is a “kinetic image,” an interactive, three-dimensional illustration. Because kinemages contain simple geometric objects — points, lines, spheres, *etc.* — they are suited to a great variety of subjects, from simple sketches to sophisticated plots to detailed schematics of 3-D objects. Their strength is in *selective* presentation of information — they were originally intended as an alternative to flat, static illustrations in scientific journals.

In fact, kinemages were first created as a way of visualizing macromolecular structures, particularly x-ray crystal structures of proteins. However, such a kinemage is a *drawing* of a molecule, albeit a 3-D drawing, and not the molecule itself. Thus, the kinemage lacks most of the information in a PDB file, and kinemage display program cannot read PDB files directly. Since their creation, kinemages have been extended to a variety of other fields, from the social sciences to ecology to education.

The kinemage concept is the brainchild of Drs. David and Jane Richardson at Duke University in Durham, North Carolina, USA. Their website is at <http://kinemage.biochem.duke.edu>.

### 1.2 Starting the KiNG program

If you’re reading this, you may already have KiNG running. In that case, feel free to skip this section. If not, we’ll guide you through getting KiNG up and running on your system.

#### 1.2.1 Download KiNG

If you don’t already have a copy of KiNG, you can download it from <http://kinemage.biochem.duke.edu>. Even if you have a copy already, you might check periodically for new version that add features and fix bugs. If you run

on a Windows or Linux machine and aren't already knowledgeable about Java technology, we recommend downloading the package that includes a Java Virtual Machine in the bundle. (This isn't an issue for Macintosh owners.)

If you just want to run KiNG as an applet on MolProbity or some other web page, you can skip this step. You can always download a copy later if you want to run it off-line.

### 1.2.2 Make sure you have Java

KiNG is written in the Java programming language, and so requires that the Java libraries (version 1.5 or later) be present on your computer in order for it to run. Newer Apple computers (those running Mac OS X) come with Java. Most PCs do not, though recently several manufacturers have agreed to pre-install Java on their new computers. Look for the coffee-cup icon in your Windows task bar or in the Control Panel.

If you don't have Java, you may have downloaded a copy of KiNG that was bundled with its own installation of Java. If you ran the InstallAnywhere installation program, this is the copy you have; don't worry about downloading and installing another copy of Java.

If you don't have Java and you didn't download the bundled version of KiNG, point your web browser to <http://java.sun.com/getjava/index.html>. The software is free and available for Windows, Linux, and Solaris computers. We recommend getting the newest version that is available.

### 1.2.3 Launch KiNG

Launching KiNG is as simple as double clicking on the "KiNG" icon. If you didn't download a package with an installer, *etc.* you may need to click on `king.jar` instead.

For Unix types who want a shell script, we recommend something like this:

```
#!/bin/sh
exec java -Xms32m -Xmx512m -jar /home/ian/bin/jars/king.jar "$@"
```

The `-Xms` and `-Xmx` switches control the amount of memory initially allocated to the Java program and the maximum memory it can consume. The default maximum of 64 MB should be enough for almost any kinemage, but a few of the largest require more (hence the `-Xmx512m` recommendation).

If you just want to run KiNG as an applet from a web page, you can just visit that web page now. If you don't see the "Kinglet" logo after a minute or two, make sure your browser is properly configured to work with Java. How you do this depends on which browser you have; consult its documentation for details.



## 1.3 Opening a kinemage

If you already have a kinemage you want to look at, this step is simple: just choose **Open** from the **File** menu, and select the kinemage you'd like to view. If you don't have anything to look at right now, don't worry! Choose **Help | Built-in kinemages**, and select one of the provided files to explore. See 8 on page 28 for more places to get kinemages from.

## 1.4 Basic kinemage reading: Using KiNG to view a kinemage

### 1.4.1 Basic interaction

#### 1.4.1.1 Moving the kinemage around

A kinemage is a dynamic, 3-D illustration. The best way to take advantage of that is by rotating it and twisting it around. Just click the mouse near the center of the graphics window and slowly drag right or left, up or down. If you have multiple buttons on your mouse, use the left button or button 1. You can also use the left and right arrow keys for small rotations. If you get lost or disoriented, use the **Views** menu to get back on familiar ground.

#### 1.4.1.2 Identifying points

By clicking on points with the mouse (left button again), you'll be able to identify them. The label associated with each point will appear in the bottom right of the graphics area. Furthermore, the distance from this point to the last one you clicked will also be displayed.

#### 1.4.1.3 Markers

For keeping track of which point is selected, markers can be displayed. Two markers are displayed normally. The checkbox for markers is just below the graphics area, with the pickcenter checkbox.

#### 1.4.1.4 Turning groups on & off

Most kinemages have many different components to them, grouped together in logical ways. A protein might have one group for each subunit; or one group for mainchain, another for side chains, and yet another for hydrogens. Groups and subgroups are given names and can be made visible or hidden by the viewer — you might want to see only one subunit, or only the backbone (without sidechains and hydrogens cluttering up the view).

The groups and subgroups appear in the button pane, to the right of the graphics area. Each one has a button to turn it on or off. Groups that are on (visible) usually have a check mark or an X in there box; a blank button means

the group is hidden. Many kinemages start off with some information hidden, to avoid overwhelming the reader. Always check to see if there's more information available than is being displayed!

#### **1.4.1.5 Re-centering**

You can make any point in the kinemage be the center. The center point will be in the middle of the graphics area, and the kinemage will rotate about that point. There are several ways to set the center: you can hold the Shift key while you click the point, or use the right mouse button if you have one. You can also press the **Pick center** button below the graphics area and then click on the desired point.

Since re-centering brings a point to the center of the screen, using this repeatedly allows one to “walk” through a large structure.

#### **1.4.1.6 Zooming in and out**

KiNG allows you to zoom in to see small details in a kinemage. Use the **Zoom** slider, below the graphics area, to control how far you zoom in. You can click the mouse right or left of the knob for small movements or click the knob and drag it for larger ones. A small motion is usually all that's needed.

For easier access, one can hold Shift and drag (or drag using the right mouse button) in the graphics window. Dragging down zooms in; up zooms out. The up/down arrow keys and mouse wheel (Java 1.4 and later only) also control zooming.

#### **1.4.1.7 Clipping**

Objects that are too near the viewer, or too far away, are not displayed; otherwise, zooming in would result in a useless superposition. The depth of the clipping “slab” can be adjusted using the slider along the bottom, or by dragging side-to-side with the right mouse button (or with the Shift key held down).

#### **1.4.1.8 More modes of interaction**

For more ways to explore the kinemage and interact with it, be sure to read about the **Navigate** tool (see 5.3 on page 20).

### **1.4.2 Reading the kinemage text**

Exploring on your own is great, but the best guide to a kinemage is its author. Often, he or she includes a short text of a few paragraphs or pages that explains what the kinemage depicts. You can access this text using the **Show text** button, located just below the graphics area. An equivalent command is also available in the **Edit** menu.

### 1.4.3 Using different viewpoints

Most kinemages have one or more “views” that were selected by the author to highlight some significant feature. These views are accessible from the **Views** menu. Just click on one of the named views and the kinemage will “jump” to that view. You can then rotate it, zoom in or out, *etc.* to explore further. You can always return to the original view by selecting it again from the menu.

Kinemages without any predefined views still get an automatic “overview” view. This is a useful way to return to a known starting point if you get lost in a complicated structure.

### 1.4.4 Other features

The kinemage format also allows for flip-book style animations, automatic searching for points based on their name (the label that appears in the lower left when the point is clicked), and much more. There is also a rich selection of features to control exactly how the graphics are rendered. See 3 on page 12 to learn more.

## Chapter 2

# Input & Output

### 2.1 Opening a kinemage

To open a new kinemage, go to the **File** menu and choose **Open**. Browse through your file system and find a kinemage file. These files typically end in `.kin`. The new version-control convention is to name modified files `.1.kin`, `.2.kin`, *etc.*, though older kinemages that have been edited may have revisions named `.kip`, `kip1`, `kip2`, *etc.* Click the button marked **OK** and KiNG will begin loading the file. You'll see a progress bar as KiNG loads the contents of the file, and then the first kinemage in that file will be displayed in the graphics area (where the logo used to be).

You can also drag and drop kinemages into the KiNG graphics window in order to open them, but this feature is not supported for Java versions prior to 1.4. On Mac OS X machines, you can also double-click `.kin` files in the Finder or drag kinemage files onto the KiNG Dock icon to launch KiNG and open the file.

Additionally, files can be specified on the command line, as in `king -s foo.kin bar.kin`, which will load all the kinemages in those two files upon startup. The `-s` (or `-single`) flag is required so that KiNG knows not to merge the two files together.

### 2.2 Working with multiple kinemages

If your file has more than one kinemage, you can choose which one you want to look at from the drop box on the right, just above the buttons. You can also use **File | Open** to load additional files; those kinemages will also appear in the drop box. When you're done, use **File | Close** to get rid of the current kinemage, or use **File | Close all** to get rid of all the open kinemages.

If you save your file when more than one kinemage is open, they will *all* be written out to that file, in the order they currently appear in. This is an easy

way to create multi-kinemage files, but be careful that you don't save *more* than you intended to!

## 2.3 Appending a kinemage

Sometimes you want to combine the contents of multiple kinemages. You might want to combine a graph of your latest sales data with some nicely formatted and labeled axes you prepared last month. Or maybe you want to combine a ribbon drawing of your favorite protein with a representation of the atomic detail at its active site. In either case, using **File** | **Append** will combine the contents of a new file with the contents of the currently displayed kinemage. Be careful — if the new file contains multiple kinemages, they'll *all* be merged into the current one!

This behavior can be achieved on the command line by specifying multiple files, as in `king -m foo.kin bar.kip1`, which will merge all the kinemages in those two files upon startup. The `-m` (or `-merge`) flag is optional; merging multiple files is the default behavior.

## 2.4 Saving kinemages

You can save kinemages using the **File** | **Save as** menu; these files are typically given the extension `.kin`. KiNG will try to help you by automatically appending a number to the filename so that your changes don't overwrite your original. The resulting kinemage file will be plain text that you can view and even edit by hand using any text editor (though of course you don't have to!). If you choose to do this and want use a word processor (like Microsoft Word), make sure that you load and save the file in plain text format.

Note that due to the security restrictions that web browsers place on Java applets, saving will never be possible when KiNG is running inside a web browser.

## 2.5 Printing kinemages

You can use **File** | **Print** to send the image in the graphics area to your printer. The image will fill as much of the page as possible without distorting it. The relative size of the elements will be exactly the same as on screen. Among the consequences of this are that an image printed from a small graphics window will have thicker lines when printed than will the image when printed from a larger graphics window. Don't worry about this too much; just remember printing is WYSIWYG (what-you-see-is-what-you-get). If it looks right on the screen, it should look right on the paper! Some images will fit better in landscape mode than portrait mode; this decision is left up to the user.

Printing should work even when KiNG is running in a web browser, although the user will generally be asked for permission first.

## 2.6 Exporting kinemages in other formats

Several facilities are provided that can help get data out of kinemage format and into other forms, either for making figures for publication or for further processing. Due to security restrictions placed on applets, the export features are unavailable when KiNG is running in a web browser.

### 2.6.1 Exporting bitmapped images (JPEG, PNG)

The **File | Export** menu allows one to save the current graphics image as a standard image file. Image export requires that you have Java 1.4 or newer; otherwise, an error message will appear. Several image formats are supported, including JPEG and PNG; the format can be selected from the export dialog box. These images are very useful for creating web pages, PowerPoint-style presentations, *etc.* For high-quality publication graphics, however, you'll want to use a vector-based format like PDF.

### 2.6.2 Exporting vector graphics (PDF, PostScript, *etc.*)

Rather than sending the image to the printer, it can be captured for inclusion in a publication or later manipulation in a graphics program like Adobe Illustrator. Note that there are several relevant options under the **Display** menu, particularly **White background** and **Monochrome**.

All platforms can generate PDF files using the **File | Export** menu. This is the simplest and best option for almost any use. "PDF" stands for Portable Document Format, the successor to PostScript as the standard for vector graphics. You can learn more about PDFs at <http://www.adobe.com>. It does require the (free) Apache PDFBox PDF library from <http://pdfbox.apache.org>, which is distributed with KiNG.

If for some reason you don't want to use PDF export, it is also often possible to hijack the print output into a file. For instance, with Macintosh OS X, both PostScript and PDF output are supported directly from the Print dialog. Under Linux, choosing **Print to file** from the Print dialog will generate a PostScript file, which can be converted many other formats using *e.g.* Ghostscript tools like **ps2pdf** (PDF) and **ps2epsi** (EPS, Encapsulated PostScript). Under Microsoft Windows, things don't seem to be as straightforward, but one possibility is to install the generic PostScript printer drivers from Adobe and then direct this "virtual printer" to **FILE** (rather than, *e.g.*, **LPT1**). Your mileage may vary; I'd be interested in hearing about better ways to do this from any Windows experts out there.

## Chapter 3

# Advanced kinemage reading

### 3.1 Finding points by name

Every point in a kinemage has an associated identifying label, which appears in the graphics area when the point is clicked on. You can find one or more points by searching through those labels for a word or phrase using the **Edit | Find point** dialog. When you press **Search**, KiNG will center the view on the first point that matches your criteria (you may need to zoom in/out to see it appropriately). **Edit | Find next** will then jump to the next matching point, until all matches are exhausted.

#### 3.1.1 Whole word searches

This is the easiest-to-use of the search options, and usually produces the desired results. You can enter one or more words or numbers in the search field (separated by spaces), and KiNG will find all the points that contain *all* of those words, in any order. A search for “val 17” would find “17 A val cb” but not “valine 117”.

#### 3.1.2 Substring searches

This works like whole word searches, but will even find points that have the search terms embedded in a larger word or number. In this case, a search for “val 17” *would* find “valine 117”. This often produces extraneous results but may occasionally be useful.

#### 3.1.3 Glob (simple regex) searches

This search is done with a simple “regular expression”, as is often used for specifying file names on the command line. A star (\*) matches any group of zero or more characters, and a question mark (?) matches any single character. Notice that the search is treated as a single term, rather than multiple words —

the pieces must be in the correct order. Thus a search for “val\*17” would find “val17”, “val 17”, and “valine 117”, among others; but would *not* find “17 val”.

### 3.1.4 Perl5 regex searches

This search is like the simple regex search but uses the full syntax of Perl 5 regular expressions, as implemented by the `gnu.regex` library. (I’m no expert, but it seems to be a very complete and official implementation. All but the most esoteric features are available.) Regular expressions are a very involved subject, and many books and on-line references are available for the interested reader.

### 3.1.5 Tips for faster searches

Remember that KiNG searches through *all* visible points to find ones matching your criteria. Turning off (*i.e.*, making invisible, hiding, unchecking) large groups that you’re not interested in will skip searching those groups. Since all the varieties of search are implemented internally with regular expressions, they are relatively computationally expensive. A search through a large kinemage may take several seconds.

## 3.2 Viewing animations

Kinemages often include animations. An animation in this sense is not a movie, but more like a flip-book, where similar images are aligned and displayed one at a time, in a series of discrete frames. If a kinemage contains animation(s), there will be forward and reverse buttons for the animation located below the checkboxes for groups and masters.

## 3.3 Controlling presentation: The Display menu

The **Display** menu offers several options for controlling how the kinemage is rendered. Of particular interest is the **Perspective** option. By default, KiNG renders objects with orthographic perspective; objects near the viewer are no larger than those far away. This is the preferred mode for viewing macromolecular structures and many 2-D graphs and figures. However, for geometrical objects and other cases where the eye expects perspective, enabling this option will result in a much more natural-looking image.

There are also several ways of controlling the coloring of the points. Each point may have its own color, and several “aspects” as well. For example, a map might be colored by the aspects of temperature, elevation, and annual rainfall; one could then choose which of these colorings to apply.

The first level of grouping above the individual point is the list; every point belongs to a list. Points that don’t have a color specified take their color from the list, but there are times when one might want to suppress the individual



colors and color points by the color of the list they belong to. This option is available as well.

### 3.3.1 Stereo

Upon request, KiNG will render the kinemage as a stereoscopic pair of images. The two views are very nearly the same, except that one is slightly rotated with respect to the other. If you can get one eye to look at each image, you can fool yourself into seeing additional depth information. Stereo can be either “cross-eye” or “wall-eye”; the former is somewhat easier to learn, but the later is more comfortable for long periods of time.

To see cross-eye stereo, cross your eyes and then look at the screen without refocusing. You should see 4 identical (though blurry) images. Relax your eyes slightly, and the center two will overlap. If you can focus on this phantom central image, you’ll be seeing in stereo. It may help bring the image into sharp focus to rotate the kinemage slightly using the mouse.

If you’ve got KiNG set for wall-eye stereo and your eyes set for cross-eye (or vice versa), you will see a stereoscopic image, but the parts that should be in back will be in front, and it will rotate the wrong direction when you move the mouse. Change one or the other to get a more reasonable image.

Incidentally, although KiNG doesn’t support multiple simultaneous views of the same kinemage, you can simulate a pair of orthogonal views by setting the stereo separation to 90 degrees in **Edit | Configure KiNG**. Of course, the views no longer forms a stereo image, but they’re useful in another way. Particularly applicable for “info-vis” kinemages, you can examine (for instance) the x-y plane and the y-z plane at the same time using this trick. It pairs well with the 90-degree rotations plugin.

### 3.3.2 OpenGL

KiNG is capable of using OpenGL rendering commands to take advantage of your graphics hardware and get (hopefully) accelerated graphics performance. You’ll need to have an up-to-date version of Java installed, along with the JOGL libraries (<https://jogamp.org/jogl/www/>). Make sure you get both `jogl.jar` and the native library or libraries (they end in `.dll`, `.so`, or `.jnilib`). *2008 update:* It is no longer necessary to download separate files; everything needed is included in the platform specific downloads at the JOGL site. The download includes directions for installing JOGL (which discourage dropping the JAR files in the extension directories as originally recommended). Essentially, the JOGL JAR and the GLUEGEN\_RT JAR need to be added to the classpath in order for JOGL to work. If you downloaded a platform-specific version of KiNG, the necessary libraries may already be included. *2016 update:* With the newer versions of Java, JOGL 1.1.1 doesn’t seem to be currently working. We’ve updated KiNG to use JOGL 2.3.2; the appropriate platform-specific JOGL jars are included with the platform-specific downloads of KiNG available on the kinemage website. *2021 update:* OpenGL may not work properly on Windows

if you have a high DPI monitor (such as on a Microsoft Surface or newer laptop). If you open up KiNG and the splash screen seems compressed into a corner, try the following fix: 1) Find out where your java.exe is installed (probably in Program Files\Java). 2) Right click on java.exe and open up Properties. 3) Go to the Compatibility tab. 4) Choose the “Change high DPI settings” button. 5) Check the box labeled “Override high DPI scaling behavior” and for “Scaling performed by:” select “System” in the drop down.

JOGL is still under development, as is this feature, and interacting so directly with the hardware is always risky, so it’s possible that OpenGL rendering may hang KiNG on your computer. You’ve been warned.

Hopefully, a future version of Java will use OpenGL behind the scenes for all graphics operations, making KiNG much faster and making this feature obsolete. Until then, this is a work-around for large kinemages where performance is an issue.

### 3.4 Master buttons

Not all the buttons to the right of the graphics control groups and subgroups of points; some are “masters.” The master buttons sit below the others, separated by a spacer. When activated, a master turns on all the groups it controls; when deactivated, it turns them all off. The effects are only momentary; the group can be turned on or off directly regardless of the state of the master(s) that control it.

A single group can be controlled by multiple masters. Note that the “off” state of a master is dominant in this case: a master will always turn off the groups it controls, but will turn them on only if they are not also controlled by another master that is already off. Since the buttons of the individual groups, subgroups, and lists are sometimes hidden, one can generate a variety of patterns by turning the masters on and off in specific patterns.

## Chapter 4

# Kinemage authoring: Using KiNG to edit a kinemage

Kinemage files are stored as human-readable, plain-text files. They're also designed to be easily edited by human beings, using a text editor like Notepad (Windows), TextEdit (Mac), or vi (Unix). You can also use a word processor like Microsoft Word, but you have to be careful to save as plain text (.txt) rather than formatted text (.doc and others). A description of the kinemage file format is distributed with KiNG as a separate document from this manual.

### 4.1 Merging kinemages together

The most basic sort of editing is to combine two kinemages into one. For this, simply use the **File | Append** command (see 2.3 on page 10).

### 4.2 Editing text

The text of a kinemage is often critical for clarifying and amplifying its meaning. It's analogous to the caption that accompanies a printed figure. You can edit the kinemage text in the same window that you use for reading it (just press the **Show text** button).

### 4.3 Editing views

A simple editing tool is **Views | Save current view**, which records the current orientation and position of the kinemage as a new view in the **Views** menu. Creating a series of meaningful, clearly labeled views is a great way to make a kinemage more informative. You can delete, rename, and rearrange views using **Views | Edit saved views**.

## 4.4 The hierarchy editor

One of the goals of KiNG is to enable simple, visual editing of most parts of a kinemage. Many of the editing commands are activated using a tree-like display of the hierarchical kinemage structure, which will resemble the layout of buttons.

The structure of the kinemage can be rearranged by cutting, copying, and pasting elements; creating new elements and deleting unneeded ones; and re-ordering elements (using the **Up** and **Down** commands). Play with these commands, and their operation will soon become obvious. Elements can even be cut out of one kinemage and pasted into another!

Also, the properties of individual elements can be adjusted, which allows them to be renamed. What other properties are available depends on the level of the hierarchy (kinemage, group, subgroup, or list). Each element can also be toggled on and off, even if it doesn't have a button visible in the button pane.

Using the hierarchy editor effectively depends on a decent understanding of the kinemage format. If you're unfamiliar with kinemages, reading the kinemage format reference may be very helpful to you.

## Chapter 5

# The Tools

### 5.1 Understanding the Tools and Plug-ins

The **Tools** menu has a variety of tools and plug-ins for interacting with the kinemage. Many of these are highly domain-specific — tailored to doing structural biology, for example. Each tool “takes over” the mouse, assigning specific meanings to the various kinds of clicks and drags. Normally, the familiar **Navigate** tool is active, so mouse motions result in the sort of rotations, pans, and zooms that most users are accustomed to.

Plug-ins, on the other hand, just offer additional functionality that doesn’t change what the mouse does. Because of this, you can have several plug-ins active at once, but only one tool.

Some special-purpose tools and plugins have been developed by other people and ship separately from KiNG. Installing these additional tools and plugins for KiNG is easy, though. Just take all plugin files that end with **.jar** (*e.g.*, **chiropraxis.jar**, **pyking.jar**) and put them in the **plugins/** folder that’s part of your KiNG distribution. There will probably be a few JAR files in there already. Then just close KiNG and restart it to load the new tools and plugins. Note that these will *not* be available when running KiNG as an applet.

The tools menu also has some other options that are related to how mouse clicks and drags are interpreted, which are discussed below. The tools and plugins listed in the menu can be rearranged, grouped into submenus, or hidden entirely by using the **Tools | Customize** dialog box, which is also accessible from **Edit | Configure KiNG**.

#### 5.1.1 Extra measurements

In addition to tracking the distance between the last two points, clicking on points to identify them can track the angle between the last three points and the dihedral angle between the last four. When these measurements are active, the **Markers** option will display four markers instead of two.

### 5.1.2 XYZ coordinates

Likewise, KiNG can display the original, unrotated coordinates of the point. This is sometimes helpful when the kinemage is a graph or plot of some data, and the position has some absolute meaning. The display of this information can be enabled in the **Tools** menu.

### 5.1.3 Pick objects

Traditionally, one can only “pick” points in a kinemage. Enabling this option allows KiNG to recognize mouse clicks that hit the actual lines drawn between points and the surfaces of triangles. In each case, the result is equivalent to clicking on the point that determines the attributes (color, line width, etc.) of that graphics object.

## 5.2 Using KiNG with a multi-button mouse

In KiNG, even a one-button mouse is sufficient to access all of the available functions. However, it has also been designed to take full advantage of a three-button wheel mouse. The primary function of every tool is accessed by clicking or dragging with the left (or only) mouse button. This button is referred to as “Normal” in the command reference charts.

Other functions can be accessed either by holding down keyboard keys at the same time, or by using one of the other mouse buttons. Holding the Shift key is equivalent to using the right mouse button, and holding the Control key (often written Ctrl) is equivalent to using the middle mouse button. A few tools assign some function to a mouse action with both the Shift and Control keys held down. In this case, you can get the same effect when dragging by depressing both the left and right mouse button. Because of the difficulty of clicking two buttons at once, however, this shortcut does not apply to Shift+Ctrl+Click.

At the present time, the Shift and Control keys do not work properly in Java on Mac OS X. When using a one-button mouse with OS X, you should use the Command key (a.k.a. Open Apple) instead of Shift to get the same effect as the right mouse button, and you should use the Option key (a.k.a. Alt) instead of Control to get the same effect as the middle mouse button. Hopefully this will be fixed in OS X 10.4 / Java 1.5.

Some tools also use the up & down arrow keys on the keyboard; these have the same effect as using the scroll wheel of a wheel mouse. Once again, these may be combined with the Shift and/or Control keys to give access to more functions. Note that the mouse wheel is never active when running in a Java 1.3.x virtual machine; Java 1.4 or later is needed to take advantage of this control.

## 5.3 Navigate: Exploring in 3-D

The Navigate tool encompasses all the usual modes of interacting with a kinemage. This is the tool that is active when KiNG is launched, and is probably the only tool many people will ever use. See 1.4.1 on page 6 for an introduction to using the Navigate tool to get around a kinemage. There are also a few more advanced modes of movement which are described below.

### 5.3.1 Z-rotation (“pinwheel”)

Clicking and dragging (with the left mouse button) near the top of the graphics area will cause the kinemage to spin around the Z axis like a pinwheel, rather than the more conventional sort of rotation. This is sometimes useful for putting something right side up.

### 5.3.2 Translation (“flatland”)

Holding Control while dragging (or dragging with the middle mouse button) allows one to slide the kinemage around in the plane of the screen. Dragging at the top of the screen, as for z-rotation, slides the kinemage in and out of the plane of the screen. (Note: enabling perspective in the **Display** menu is *very* helpful when translating in and out of the plane of the screen.) This sort of motion is sometimes called “panning” or “flatland,” and can be made the default mode of movement by selecting **Tools | Flatland**.

### 5.3.3 Command reference

Mouse clicks	
Normal	Mark and identify point; make measurement (pick)
With Shift	Center on selected point (pickcenter)
With Ctrl	-
With Shift+Ctrl	-

Mouse drags	
Normal	Rotate around X and Y axes; Z-rotate (pinwheel) near top of screen
With Shift	Adjust zoom (up/down); Adjust clipping (left/right)
With Ctrl	Translate in X-Y plane (flatland); Z-translate near top of screen
With Shift+Ctrl	Rotate around Y axis only

Mouse wheel / Up & Down arrow keys	
Normal	Adjust zoom
With Shift	Adjust clipping
With Ctrl	-
With Shift+Ctrl	-

## 5.4 Draw New: Reworking the kinemage

The Draw New tool provides a variety of simple tools for creating new objects in the kinemage or removing existing ones on a point-by-point basis.

**Edit list props** Picking a point opens the editing window for that point's list, where you can set color, width, radius, *etc.* You will need to activate the **Tools | Superpick** option to edit unpickable points.

**Edit point props** Picking a point opens the editing window for that point, where you can set color, position, point ID, *etc.* You will need to activate the **Tools | Superpick** option to edit unpickable points.

**Move points** Allows you to click on individual points and drag them around to new positions. Dragging them with the mouse moves them in the plane of the screen. Using the mouse wheel or up & down arrow keys moves them in and out (this is easier to see with Perspective enabled and Stereo on). A kinemage may have several "points" at the same location in 3-D. Under normal conditions, this tool will move them all together, so that lines stay connected and everything behaves just as you would expect. However, you can also use Ctrl+Drag to pick apart overlapped points and separate them from each other. You will need to activate the **Tools | Superpick** option to move unpickable points.

**Draw line segments** Simply draws a line between the first point picked and the second. Every pair of picks forms a new line. Lines can be shortened from their natural length (or lengthened, using negative numbers), which is sometimes useful for making illustrations.

**Draw dotted lines** Like drawing ordinary lines, except a string of dots is created instead of a single line. The number of dots is configurable. This can serve as a "template" for constructing more complex dot-dash patterns when used with line drawing and prune.

**Draw curved arc** Draws an arc from the second point picked to the third one, with the arc curving toward the first point picked. The user specifies how many degrees out of a circle will be used to create the arc, so small numbers give flat arcs and large numbers give exaggerated ones. The arc can also be shortened by a few degrees so it doesn't quite touch its endpoints, and a four-pronged arrowhead can be added.

**Draw balls** Places a small ball at each point picked with the mouse.

**Draw labels** Creates a label at the picked point, with its text taken from the picked point's ID. The label text can be changed using Edit Properties.

**Draw dots** Creates a dot at each point picked with the mouse.



**Draw triangles** Draws a triangle for each set of three points picked. Triangles can also be shrunk (values between 0 and 1) or grown (values greater than 1). A value of zero will make an infinitely small triangle; a value of one will give a triangle that just touches the three points.

**Prune one point** Removes the single point picked with the mouse. In the case of polylines and triangle strips, it will stop things from being drawn that depend on the removed point. That is, lines drawn to and from the point will be removed, and all triangles involving the point will be removed. Sometimes kinemages have multiple points that overlap in the same spatial location, so it may take multiple clicks to remove all the objects at a particular point in space.

**Punch a polyline** Removes all the line segments in a particular *polyline* (a series of lines drawn head to tail). This is sometimes useful with vectors and triangles; for other things, it behaves like prune.

**Auger a region** Removes all points that fall under the giant eraser. The eraser is drawn as a large circle that follows the mouse around the graphics area. Auger operations cannot be undone, so save your file first and click carefully.

**Spherical crop** Removes all points that fall more than a specified distance away from the picked point. Only visible points are cropped. Crop operations cannot be undone, so save your file first and click carefully.

**New subgroup** Kinemage elements created by each drawing option are placed into their own list, and all of those lists are placed in one subgroup under the “New group” group created by this tool. Creating a new subgroup starts a fresh list for each option above. This is a good way to separate out the different objects you’re drawing. New subgroup can’t be undone, but it’s pretty harmless.

**Undo** The Undo button undoes the last drawing action, with a few exceptions (mentioned above). Undo depth is unlimited, subject to memory constraints. Thus, for kinemages that aren’t too large you can often undo all the drawing operations you’ve performed since loading the kinemage.

## 5.5 Electron Density: Visualizing maps

This plug-in is for viewing electron density maps while exploring kinemages of crystal structures. (If that doesn’t mean anything to you, then you probably don’t need this plugin.)

Choose **Open** from the **Tools** menu in order to open a map file. Supported formats include X-PLOR (ASCII, not binary), O (DSN6 or Brix), and CCP4 (mode 2, the only one in common use).

Each map opens a new control window, which allows adjustment of its extent, color, and level. The segment of map that is visible at any given time is centered in the current view; when the view changes, the portion that is visible will automatically be updated.

Any dynamic segment of map can be "captured" as part of a static kinemage; just use the **Export** button to make the currently visible map segment part of the current kinemage. When you've finished with a map, be sure to **Discard** it to free up the memory it consumed.

If you don't have an electron density map for the structure you're interested in, you can try looking at the Uppsala Electron Density Server (<http://fsrv1.bmc.uu.se/eds/>). They offer standard maps for many of the structures deposited in the Protein Data Bank.

Maps can't be loaded from the local filesystem when KiNG is running as an applet due to security restrictions. However, you can use the **XMAP** or **OMAP** parameter along with the **<PARAM>** tag to load a (single) map from a URL. For example:

```
<PARAM name="omap" value="http://example.com/electron/density/my-protein.dsn6.gz">
```

You can also use the **EDMAPBASE** and **EDMAPLIST** parameters to allow the user to select and load a map. **EDMAPBASE** specifies a path to the maps relative to the current document base URL. **EDMAPLIST** is a space-separated list of filenames that can be loaded from that path. Users can also enter and modify URLs directly, though obviously this isn't very convenient.

## 5.6 Update KiNG: Upgrading without the hassle

The Update KiNG plugin allows you to quickly and easily update KiNG to the latest version published on the Kinemage website ([kinemage.biochem.duke.edu](http://kinemage.biochem.duke.edu)). After you confirm your intentions, KiNG will begin downloading the new version. If the download proceeds too slowly, you can cancel with no changes to KiNG. Once complete, the downloaded version will be unpacked in the same place as your current installed version of KiNG, and the existing version will be overwritten. You can then quit KiNG and restart to take advantage of new features and capabilities.

The Update plugin has been tested on a variety of platforms under a variety of conditions, but there is always a chance that it won't work quite right on *your* computer. The chance is greater if you created a non-standard custom installation from the JAR files rather than using the pre-packaged platform-specific installer. If KiNG fails to start after the update, you may have to remove it and then download and install a clean copy by hand.

There are certain things the automatic updater *can't* update, like the bundled binary programs (**probe**, *etc.*) or the displayed version number of the Mac application bundle. For this reason, you may want to periodically remove the existing installation completely and reinstall from scratch.

## Chapter 6

# Configuring KiNG

By choosing **Edit | Configure KiNG**, one can adjust several aspects of KiNG's behavior and appearance, such as font sizes, stereo settings, and whether or not certain warnings are issued. The configuration dialog also allows for reorganizing the **Tools** menu to suit your own work habits. Preferences may be saved to disk, where they are stored in a simple text file named `.king_prefs` in the user's home directory. There may be some settings accessible only through this file; that is, they cannot be edited from the graphical interface. Nonetheless, pressing **Reset to defaults** will restore them to their standard values.

Note that due to security restrictions, configurations cannot be saved when KiNG is running as an applet. However, the web page creator may provide a custom configuration file in the same format as the one saved by KiNG the application, thereby altering the default behavior for all visitors to the website. If you create a preferences file named `MyKingPrefs.txt` and place it in the same directory as `king.jar` and your HTML file, add something like this to the **APPLET** tag of your webpage:

```
<param name="king_prefs" value="MyKingPrefs.txt">
```

## Chapter 7

# Launching KiNG

KiNG runs both as a stand-alone application, and as an applet that can be included in web pages. In any case, KiNG requires a Java Virtual Machine (JVM), version 1.8 or higher. You can get Java from <http://java.sun.com>.

### 7.1 Running KiNG as an application

How you launch the KiNG application will depend on what operating system (Windows, Mac OS X, Linux, *etc.*) you're using. With Windows and the Mac, you can usually just double-click on the `king.jar` file to start KiNG.

On all operating systems, you can type

```
java -jar king.jar [-merge | -single] [file1 file2 ...]
```

All the files listed on the command line will be merged together into one kinemage, unless the `-s` flag is included, in which case all the files will be opened as separate kinemages.

As of KiNG 2.16, the files KiNG can read in from the command line include kinemages, PDB, mmCIF, and electron density maps. For the structure files, QuickKin is used to generate quick-browsing ribbon+stick representations of the structures.

### 7.2 Making KiNG an applet in a web page

KiNG can also be included as a web page applet. The following HTML code is a simple example that uses the `<APPLET>` tag, which will work with browsers using a recent JVM. Older browsers may need `<OBJECT>` or `<EMBED>` tags; more information on this is available from Sun Microsystems, Inc.

```
<applet code="king/Kinglet.class" archive="king.jar" width=300 height=150>
```

```
<param name="kinSource" value="some_file_name.kin">
</applet>
```

Just put all the jar files, including king.jar, in the same directory as the HTML file. Users who visit the page will see a 300x150 KiNG logo in the web page, along with a floating KiNG window that looks very much like the stand-alone application. However, due to security restrictions imposed by the browser environment, some functions (*e.g.*, saving files) will be disabled.

Note that the `code` property really must have the value `king/Kinglet.class`, even though *king.jar* is not in a sub-directory called *king/*. This is the most common problem that people have when trying to run KiNG as an applet. If you did place `king.jar` somewhere else, that path should precede the jar file name in the `archive` property instead.

If you want to include additional jar files for plugins and tools, you need to list those in the `archive` field, after `king.jar`, separated by commas. For example:

```
<applet code="king/Kinglet.class" archive="king.jar,chiropraxis.jar,extratools.jar">
```

It is possible to run KiNG in two other applet modes to achieve a slightly different effect. First, the large KiNG logo can be replaced with a small one that serves as a launcher button. Thus, loading the main window is deferred until the user clicks the button, at which time the main window appears and the kinemage is loaded. Below is sample HTML for this effect. Note the smaller width and height, to account for the small size of the button.

```
<applet code="king/Kinglet.class" archive="king.jar" width=60 height=21>
<param name="mode" value="launcher">
<param name="kinSource" value="some_file_name.kin">
</applet>
```

If you would like a custom image for the launcher, specify its URL with the `launcherImage` parameter. You may need to change the applet width/height to match. If no image is specified, the default one will be used.

Second, the applet can be embedded in the web page (rather than floating in its own window), by placing this line inside the `<applet>` tags instead:

```
<param name="mode" value="flat">
```

One also probably wants to specify a larger width and/or height, possibly as a fraction of the page size (*e.g.* 95%). Be warned that using KiNG this way may not work well in some browsers; test your pages thoroughly.

Normally, a KiNG applet displays only the kinemage specified in its `kinSource` parameter. However, you can also use the `kinfileBase` and `kinfileList` parameters to allow the user to open and/or append additional kinemages from

a pre-specified set. **kinfileBase** specifies a path to the maps relative to the current document base URL. **kinfileList** is a space-separated list of filenames that can be loaded from that path. Users can also enter and modify URLs directly, though obviously this isn't very convenient.

The KiNG applet also supports a **kinfileSaveHandler** parameter, which is the URL of a CGI script that can accept HTTP POST kinemage output. On a properly configured webserver, this allows users of the KiNG applet to save their modified kinemages directly to the server! However, this is definitely a custom programming job on the server side and so is not an option for most "normal" uses of the KiNG applet.

Some Tools and Plugins may define their own <PARAM> tags, which will be described in their documentation.

As of KiNG version 2.16, the KiNG applet can read PDB or CIF files as well as kinemages. In order to supply a PDB or CIF file for KiNG, instead of the **kinSource** parameter, use **pdbSource** or **pdbFile**. This feature uses QuickKin to generate a quick ribbons and stick representation of the input file.

## Chapter 8

# Additional resources

### 8.1 Included manuals

In addition to this User's Guide, KiNG may be accompanied by a guide to the kinemage file format. Kinemage files are simple text files, so that anyone can create kinemage illustrations by hand. This is especially important to fields where no automated tools for kinemage creation yet exist.

### 8.2 Included kinemages

Under the **Help** menu, KiNG has several built-in kinemages that demonstrate various capabilities of the graphics system. The palette is particularly useful to kinemage authors when they are designing color schemes.

### 8.3 External documentation

The best resource for learning more about kinemages is the Richardson lab website, <http://kinemage.biochem.duke.edu>. Many other resources exist; hopefully, they will be documented here in the future.

### 8.4 Bug reporting and contact info

Ian Davis, the original author of KiNG, has graduated. KiNG is currently being maintained by Vincent Chen. *Please*, do not hesitate to report bugs and problems with KiNG — your input can only improve the program. Reach us at <mailto:rlabduke@gmail.com?subject=KiNGissues/questions>

## Chapter 9

# Copyright & acknowledgments

### 9.1 Copyright

The KiNG program and all its associated original resources and documentation are copyright (C) 2002-2021 by Ian W. Davis & Vincent B. Chen. KiNG makes use of the gnu.regex library, which is copyright (C) by Wes Biggs, and of icons copyright (C) by Sun Microsystems provided for use with Java programs. It is also capable of using the Apache PDFBox library for generating PDF files, and the JOGL library for OpenGL rendering.

### 9.2 Licenses

KiNG is free software available under the terms of its own BSD-style license (see A on page 31). The source code and new versions can be downloaded from the Richardson lab website, <http://kinemage.biochem.duke.edu>. The gnu.regex library is covered by the GNU Lesser General Public License (LGPL). In accordance with that license, the full source of the gnu.regex library is available from its website at <http://www.cacas.org/java/gnu/regex/>. The Apache PDFBox library is licensed under the Apache License 2.0. PDFBox is available from <https://pdfbox.apache.org/>. JOGL is available under a BSD-style license from <https://jogamp.org/jogl/www/>. Many of the icons are the property of Sun Microsystems and have their own license; they are available from <http://developer.java.sun.com/developer/techDocs/hi/repository/>.

All license documents are provided in the downloaded KiNG bundles and are found in the `doc/` directory. Additionally, the KiNG license appears in the Appendix (see A on page 31).



## 9.3 Acknowledgments

I owe an enormous debt of gratitude to many, many people for making this program possible. First and foremost, I thank David and Jane Richardson, who are my bosses and mentors, and who developed the kinemage concept, the file format, and most of the features found in KiNG — my contribution was only the Java implementation. Second, I thank Vincent Chen for his ongoing contributions to KiNG; he has developed many useful plugins and has made significant contributions to the core. I am also grateful to the other members of the Richardson lab, who have supported and encouraged me in developing KiNG, providing invaluable insight into how to improve the program. Of course, this work would not have been possible without support from the National Institutes of Health and the Howard Hughes Medical Institute.

KiNG is free software built on free software. I can't hope to acknowledge all the authors of all the software that has been used to create KiNG, but I can mention a few of the most prominent: Sun Microsystems, the creators of the Java language and libraries; the L<sup>A</sup>T<sub>E</sub>X document processor, used for writing the documentation; jEdit, a fantastic programmer's text editor written entirely in Java; and the entire Linux movement, for providing a real alternative to commercial operating systems. I have also taken advantage of a number of excellent free Java libraries, which are described above.

## 9.4 Revision status

This manual was last updated 05 Jan 2021 by VBC for KiNG version 2.24.

# Appendix A

## KiNG License

### A.1 What this license applies to

*This license applies to everything in this project that is the original work of the copyright holder(s) and contributor(s), including but not limited to the source code, binary executables, artwork, and documentation. This distribution includes certain third-party libraries and resources which are not covered by this license, and which are governed by their own license terms. See documentation for more details.*

### A.2 The KiNG License

Copyright (C) 2002-2021 Ian W. Davis & Vincent B. Chen. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

*"This product includes software developed in the Richardson lab at Duke University (<http://kinemage.biochem.duke.edu>)."*

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The name of the copyright holder, the names of the contributors, and the names "KiNG", "Kinemage, Next Generation", and "Mage" must not be

used to endorse or promote products derived from this software without prior written permission.

5. Products derived from this software may not be called "KiNG", nor may "KiNG" appear in their name, without prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **A.3 Where this license came from**

This license is based on the BSD and Apache licenses. See <http://www.opensource.org/> for more details on these and other licenses.