

Chapter 13

Disk Mode

Disk mode lets you load and save files of objects between the K2vx and the outside world, through the K2vx's floppy drive or SCSI ports. The floppy drive accepts MS-DOS format double-sided double-density (DSDD—720K) and high-density (HD—1.4 M) floppy disks. It will also accept single-sided double-density disks (SSDD—360K) if they're formatted as double-sided. To save data from the K2vx to a floppy disk, the disk must be formatted (initialized) by the K2vx or by an IBM®-compatible computer running MS-DOS® software.

Most SCSI (Small Computer System Interface) devices will operate with the K2vx. The K2vx's SCSI ports require 25-pin SCSI connectors. The most common use for the SCSI ports is to connect a hard disk or chain of hard disks for storing samples and other objects. You can also connect a CD-ROM drive for reading files to the K2vx. The K2vx will treat a CD-ROM drive like any other SCSI device (except that you can't save files to it).

Disk Mode in the K2vx allows flexibility to organize disk files and their contents. Many powerful operations are included that can save a lot of time by allowing you to easily specify exactly what you want to load or save. Examples of this range from organizing related files into directories, to loading macro lists of files or selected objects from multiple SCSI drives, to setting up programs to automatically link with samples off of a CD-ROM.

Lastly, the K2vx offers MS-DOS file system compatibility, sample transfer using the standard audio file formats "AIFF" and "WAVE", and saving of song files as MIDI Type 0 format.

Disk Mode Page

To enter Disk mode, press the Disk button, and the Disk mode page will appear:

```

DiskMode      Samples:65535K  Memory:747K
Path = \DRUMS\
(Macro on)
CurrentDisk:SCSI 4      Startup:Off
                        Library:Off

Direct Access, 121MB
TAXMOR XL3-1001      1.07
<more>  Load  Save  Macro  Delete more>

```

As usual, the current mode is displayed on the top line. At the middle of this line, the amount of available sample RAM is shown (if you have added optional sample RAM; if you haven't, this part of the top line will be blank). To the right of the top line you see the amount of memory available for storing all other RAM objects. Straight from the box, this number should be around 240K. If you have added the optional P/RAM extension kit, the number will be about 1260K. The P/RAM kit enables you to store hundreds of additional programs, songs, and any other objects. If you do a lot of programming or sequencing, the P/RAM option is a good idea.

In the center of the page is a line indicating the currently selected device. Select different devices using any data entry method. You can select the floppy drive, or SCSI IDs 0–7. (If you connect an Apple Macintosh® personal computer, don't select SCSI ID 7, since that's the SCSI ID of the Mac, which can't be changed. All SCSI devices connected in a network must have different SCSI IDs in order for the network to function). When you want to communicate with any of the SCSI storage devices in your network, set the Current disk parameter to the value

that matches the SCSI ID of the device you want to address. That is, the K2vx will interact with the SCSI storage device whose SCSI ID matches the value of the Current Disk parameter. If you choose a value of Floppy, the K2vx will communicate with its floppy disk drive.

The manual for your SCSI disk should tell you its SCSI ID. Most newer SCSI disks show their SCSI IDs on their rear panels, and many have adjustable SCSI IDs.

The currently selected device will be read from or written to when you load, save, rename, or delete files. Use the soft buttons to start any of these operations. Refer to "Soft Buttons on the Disk Mode Page" on page 13-5 for complete information.

Formatting a Floppy Disk

Floppy disks must be formatted (initialized) to the Kurzweil format before they can be used with the K2vx. The K2vx uses the MS-DOS format, so floppies formatted on DOS computers should work with the K2vx. To format a disk on the K2vx, insert a floppy disk in the K2vx's floppy drive. Make sure the disk is unlocked (the sliding switch on the back of the disk is set so you can't see through the hole on that side of the disk).

Press the DISK button to enter Disk mode. Make sure the Current disk parameter says "Floppy," so you don't accidentally format any SCSI devices you might have connected! Press the soft button labeled Format. The K2vx will ask you if you want to format, and a pair of Yes/No soft buttons will appear. Press the **Yes** soft button, and three soft buttons will appear, letting you choose between formatting the floppy as a 720K (double-density) or 1.4M (high-density) disk. Press the appropriate soft button, or press the Cancel soft button.

If the floppy you inserted is double-density and you press the 1.4M soft button, the format procedure will fail. This is also true if the disk is high-density and you press the 720K soft button. If the floppy is single-sided (SSDD), you can press the 720K soft button, and the floppy will be formatted as a double-sided disk. (The only difference between single- and double-sided floppies is that both sides of a double-sided floppy have been tested by the manufacturer.)

Once you select between the 720K or 1.4M format, the K2vx will remind you that formatting will erase the floppy, and will give you two more chances to cancel the formatting procedure—we want to make sure you don't accidentally erase any disks. Press the **Yes** soft button to continue formatting. When formatting begins, the display will tell you that the disk is being formatted. You'll hear the disk drive turning, and the disk drive LED will light.

Formatting a 1.4M floppy disk takes just under three minutes, including the automatic verification. When formatting is finished, the display returns to the Disk mode page.

Connecting a SCSI Device

It's easy to connect SCSI devices to the K2vx's SCSI ports. Using a SCSI device will give you off-line storage, and can speed up your loading and saving operations considerably.

You'll need a SCSI cable with a 25-pin SCSI connector on the end to be connected to the K2vx. If your SCSI device does not have a 25-pin connector at one end, you can find SCSI cables like these at any personal computer store. Connect the 25-pin end of the cable to either of the K2vx's SCSI ports, and the other end to your SCSI device. Two SCSI ports are provided to enable you to chain SCSI devices together. Before you start connecting cables, however, please read the next section carefully. We've also included important information about SCSI in Chapter 8 of the *Reference Guide*.

SCSI Termination

Simply put, SCSI termination prevents the electrical signals used by SCSI devices from being reflected from unconnected SCSI ports, and possibly disrupting the data stream. The rule for SCSI termination is that the two SCSI devices on the ends of a chain of SCSI devices must be terminated, and all devices in between, however many, must be unterminated. Newer SCSI devices usually make it easy to enable or disable their termination settings. Older SCSI devices may require an external terminator to be installed. These are available at all personal computer stores. Make sure you get the right size for your device (25-pin or 50-pin).

It's impossible to describe all the possible configurations of SCSI devices, so we'll provide you with a few general guidelines that will cover the requirements for most SCSI systems. If you're chaining large numbers of SCSI devices together, you may have to do a little juggling, but chances are you'll already have some experience with SCSI termination.

First of all, it's *very* important that you terminate your SCSI system properly. Improper termination can result in lost data, can interfere with the operation of your SCSI devices, and over the long term, can damage them.

If your SCSI system includes an Apple Macintosh® personal computer, you'll need to be sure that it is internally terminated. This is because the Mac has only one SCSI port, and therefore is necessarily going to be at one end of your SCSI chain. If your Mac has an internal hard disk, you can go ahead and hook it up, since it's sure to be internally terminated. Some Macs without internal hard disks are *not* terminated, so this may cause a problem. If your Mac has no internal hard disk, and you're not sure whether it's internally terminated, you should call your local Apple dealer for confirmation.

If your SCSI system includes only the K2vx and an internally terminated Mac, you're all set. Just connect the Mac's SCSI port to either of the K2vx's SCSI ports, and you're ready to roll. The K2vx is internally terminated, so in this case, you've satisfied the requirement of terminating both ends of the chain.

If you have an internally terminated Mac, a K2vx and an external hard disk with *two* SCSI ports, setting up is also painless. Connect the Mac's SCSI port to one of the hard disk's SCSI ports, and the K2vx's SCSI port to the hard disk's other SCSI port. Make sure the hard disk is not terminated, since it's in the middle of the chain. In this configuration (with a terminated Mac at one end and the K2vx at the other), you can chain up to six hard disks between them. Make sure they're all unterminated, and don't forget to set each disk's SCSI ID to a different value.

If you have a Mac, a K2vx and an external hard disk with only *one* SCSI port, you'll need to have your K2vx serviced. The only way to connect a system of this sort is to have the K2vx in the middle (with one SCSI port connected to the Mac, and the other to the hard disk). In this case, you'll need to have your Kurzweil/Young Chang dealer or service center remove the K2vx's internal termination. This is very easy for a qualified technician to do, and shouldn't take more than a half hour or so.

If you install an internal hard disk in your K2vx, the technician who does the installation should remove the K2vx's termination as part of the installation. This will not necessarily enable you to connect the K2vx in the middle of a SCSI chain, however, since the internal hard disk has its own termination. With an internal hard disk, the K2vx should be installed at the end of a SCSI chain. The internal hard disk's terminator will provide the termination for itself and the K2vx. You can remove the termination from the internal hard disk as well, if you want to connect your K2vx in the middle of a SCSI chain. In this case, you must either connect it in the middle of a chain, or install an external terminator on the unused SCSI port if you connect it at the end of a chain.

In this case, the K2vx can be connected in the middle of a SCSI chain without causing problems. If you connect it at the end of a SCSI chain, however, you should install an external terminator on the unused SCSI port.

If you're planning to buy a SCSI hard disk to use with your K2vx, it's a good idea to buy one with two SCSI ports. Most new hard disks have two ports, and can be terminated or unterminated relatively easily. This gives you added flexibility, since you can install it at the end of a chain, leaving its termination in place, or in the middle of a chain, using both its SCSI ports, and removing its termination.

When your SCSI device is connected, you can select it with the Current disk parameter on the Disk mode page. Use any data entry method to select the SCSI ID that matches the SCSI ID of your SCSI device. If you're using the numeric keypad to select the device, enter "9" to select the floppy. Newer SCSI devices usually have an external switch for setting their IDs. Older units may not have these; check your device's owner's manual for its SCSI ID.

Using your K2vx in a SCSI System

SCSI IDs

All devices in a chain of SCSI devices must have different SCSI IDs, including the K2vx. The K2vx's SCSI ID is set at 6 by default, and can be changed on the RECV page in MIDI mode. If your SCSI system includes a Mac, be sure not to use SCSI ID 7 for any of your other devices, since the Mac's SCSI ID is 7, and can't be changed. Once you've made sure that all connected devices are set to different SCSI IDs, you should be able to select the devices, format them, and start loading and saving files.

Formatting a SCSI Device

The procedure for formatting hard disks is essentially the same as with floppy disks, once the SCSI device is selected with the CurrentDisk parameter. The K2vx will recognize the disk as a SCSI disk, and will warn you that formatting will erase the contents of the disk. Compared with personal computers, the K2vx's formatting time for SCSI disks is surprisingly short.

Directories

A directory is a file on the disk that lets you group other files together as you might separate documents using folders in a file cabinet. You can create directories on K2vx Format SCSI and floppy disks. You can even create directories within directories, which are also called *subdirectories*.

Directories are very useful for organizing your sample, song, and program files. The K2vx provides many operations for setting up and managing the directories on your disks and the files within them.

Path

The Path field shows the current directory on the current disk if it is a K2vx format disk. This field is displayed upon returning to the Disk Mode page after you have pressed one of the disk function soft buttons and viewed the file contents of a specific disk. The current directory is always set to the top-level or root directory when the K2vx is powered on, or when the CurrentDisk parameter is changed. After this, the current directory is remembered by the K2vx as the most recent directory accessed by the disk functions.

The root directory is displayed as

**Path = **

If you press the Load button and load in a file from a subdirectory called SOUNDS, the Path field will appear as

Path = \SOUNDS

The slash character is also used as a directory separator, as in the following Path:

Path = \NEWTUNE\SAMPLES\DOGS

This represents the subdirectory DOGS, that is located in the SAMPLES subdirectory of the NEWTUNE directory in the root directory. If the path is too long to fit on the top line of the screen, its display may be abbreviated. The maximum length of a path in the K2vx is 64 characters (including '\' characters).

Startup

The Startup parameter determines what disk drive will be used for loading the power-up macro file BOOT.MAC (See "Creating a Startup File," page 13-63). If this is set to None, then the K2vx will power-up in a normal fashion. If this is set to a SCSI device or Floppy, when the K2vx is next powered on it will look for the BOOT.MAC file in the root directory of the specified disk, and load in each of the entries in the macro specified within.

This feature provides a very flexible way to automatically configure your K2vx's memory contents whenever you turn the power on.

Library

This feature works in conjunction with the Macro feature to provide a way to distribute Macro files that load data from removable media without having to know in advance the SCSI ID of the removable drive. A Macro file stores its references to drives by drive ID (SCSI ID or Floppy), or by either a "Library" or "Unspecified" designation (see the section on Macro files, page 13-38.) Typically, you would set the Library parameter to be the same as the SCSI ID of your CD-ROM drive, if you were loading Macro files from a Floppy disk or another SCSI drive that referenced CD-ROM files containing samples or keymaps.

Use of the Library feature is described in more detail later on in this guide.

Disk Drive Information

Below the media type and disk capacity is displayed specific information about the current disk's manufacturer, model number and internal mechanism. The K2vx requests this information from a SCSI drive when the drive is selected by setting the Current Disk parameter. This information may be needed when determining if a given drive is compatible for SCSI operation with the K2vx.

Macro On Indicator

When the "**(Macro on)**" indication is displayed, it means that any load operation performed will be recorded in the K2vx's macro table. The Macro feature is described starting on page 13-38 in this guide.

Disk Mode Soft Buttons

Here is a brief description of each of disk mode's soft button:

Load - Loading files or loading files from the current disk into K2vx memory.

Save - Saving banks of objects, selected objects, or a macro as a K2vx file on the current disk.

Macro - Controls macro recording. Allows modifying entries from the current macro list.

Delete - Delete files from the current disk if it is a K2vx disk. See "Deleting files", below.

Rename- Change the file name of a file on a K2vx disk. See "Renaming Files", below.

Move - Change the location of a file from within one directory to another (on the same disk).

Util - Check the free space, find files, and view directory organization and sizes on the current disk.

NewDir- Create a new directory on K2vx disks.

Backup- Hierarchical file backup between drives.

Copy - Single file copy between drives.

Sleep - Send SCSI sleep command to the current disk. See "The Sleep Soft Button", below.

Format - Format the current disk as a K2vx disk.

The Sleep Soft Button

Many SCSI devices will "sleep" when they've been idle for a few minutes. In other words, the disk will stop spinning, in order to save power and reduce wear. The K2vx lets you tell your SCSI devices to sleep. Just press the Sleep soft button, and if your devices have this feature, they will sleep. This is particularly useful in a quiet studio situation.

Any Disk mode operation will "wake" the device again. The K2vx will ask you to wait while the device's disk starts spinning. As soon as the disk is spinning at full speed, the K2vx will execute the operation you selected. Some SCSI devices automatically sleep when they power up. (A device of this type usually provides a way to override this feature; check its manual). Any Disk mode operation will wake a disk in this case, as well.

File List Dialog

The file list dialog is displayed anytime you select a disk function to operate on one or more files on a drive (such as Load or Rename.) Here is a typical file list dialog, for the Load function:

```

Dir:\          Sel:0/3      Index: 1

File to load:  BASSOON .KRZ  3456K
                MHY25      (dir)
                PERC      .KRZ  101K
Total: 3557K
Select Root Parent Open OK Cancel
  
```

When you enter this dialog, the K2vx selects the current directory to display, and displays its files in an alphabetized scrolling list. If the current directory cannot be located (for instance, if a removable media disk has been changed), the default is to display the disk's root directory. The root directory will also be selected if the disk drive was just chosen by the CurrentDisk parameter on the Disk Mode page (remember that the current directory is always set to the top level when the CurrentDisk parameter is changed, or if the K2vx has just been powered on.)

The display for all drives (including floppy) shows the 3-character extension of all files in the directory (except directories themselves). Extensions are created when the file is saved by the K2vx. You cannot modify the extensions on the K2vx. This is because the K2vx uses the extensions to tell it what kind of data is stored within the file.

Directories created by the K2vx have up to 8-character names, with no extension. A directory can have an extension if it is created on an external computer (more on this later on).

Here is a list of extensions used by or accepted by the K2vx:

.AIF - Audio Interchange File Format (AIFF)
.KOS - Kurzweil K2vx operating system file
.KRZ - Kurzweil K2vx format file
.K25 - Kurzweil K2vx format file containing objects and/or sample data
.MAC - Kurzweil K2vx disk macro file
.MID - MIDI type 0 sequence file
.WAV - Microsoft RIFF WAVE format

When loading files, the K2vx will try to find out the type of file if the extension is not the same as is suggested above (with one exception, .MAC files.) The K2vx can create files with any of the above extensions.

The top line of the file list contains several items of information pertaining to the currently displayed directory contents. A typical information line looks like this:

```
Dir:..\HHS\ Sel:0/54 Index: 24
```

In the center of this line is an indicator of the number of files in the currently displayed directory. This number is grouped together with the number of selected files:

```
Sel:0/54
```

The first number means "No files selected" and the number after the slash means "Fifty-four files are in the directory". File selection is possible in several of the disk functions, more on this below. The total number of files also includes any subdirectories of the current directory, but not the files within the subdirectories.

On the left end of the top line of the file list page is the current directory, sometimes in an abbreviated form. If you are in the root directory, the display will read:

```
Dir: \ Sel:0/54 Index: 24
```

If you are in the directory \MONDAY, the display will read:

```
Dir:\MONDAY\ Sel:0/54 Index: 24
```

If you are in a directory that is more than one level down from the root directory, such as \FX2\GLASS\BREAKING, the display will read:

```
Dir:..\BREAKING\ Sel:0/54 Index: 24
```

The "..\" indicator tells you that you are more than one level down from the root directory.

The File Index

On the right end of the top line is the Index field. This tells the position of the highlighted file relative from the beginning of the file list. The first entry in a file list is index 1.

```

Dir:\                               Sel:2/23   Index:    3
      HXM      .KRZ*    122K
      CHIME     .KRZ     42K
File to load:DOORS .KRZ*   3456K
      JUNE27      (dir)
      LONGSMPS    (dir)
Total:21034K    FLUTE .KRZ     5K
Select  Root  Parent  Open  OK  Cancel

```

Typing a number on the K2vx numeric keypad will automatically scroll the display to the corresponding entry in the file list. Typing an out of range value such as 999 is a quick shortcut to get to the end of the file list.

In addition to remembering the current directory on the most recently used drive, the K2vx now also remembers the index within the file list for the current directory. For instance, if you were to hit cancel on the above screen, and then either choose a different disk function (such as Rename), or even if you exited Disk Mode, went to Program Mode, and then later returned to Disk Mode to load a file off of the same drive, the file index would still show up on 3 (DOORS.KRZ). This index is remembered until a new drive is selected by scrolling the Current Disk parameter on the Disk Mode page.

There are exceptions to this however. For example, when a file is written to the drive using the Save function, the index will subsequently be set to the file that was just saved (see Save function). The index can also be explicitly set using the List and Find utilities (see below).

If there are no files in the current directory, then the index is 0:

```

Dir:\                               Sel:0/0   Index:    0

File to load:

Total: 0K
Select  Root  Parent  Open  OK  Cancel

```

You will also see that there is a blank cursor bar, but no file names are displayed.

The maximum number of files that can be accessed within a single directory is 360. If you have more files than this amount in a single directory, then you will not be able to view the entries past index 360.

While in this dialog, pressing the CHAN/BANK buttons (to the left of the LCD display) will scroll the file list in either the forward or backward direction by "pages" of 5 entries. It is often easier to scroll the list this way when looking to see if a particular file is present in a directory.

Soft Buttons in the File List Dialog

Use the **Select** soft button for multiple file selection in the Load, Delete, and Move functions. In the display below, there are two files selected, as indicated by the asterisk ("*") after the .KRZ file extension. If OK were pressed for the following screen,


```

Dir:\                               Sel:2/23       Index:      3
                                AXM             .KRZ       122K
                                CHIME            .KRZ        42K
File to load:DOORS                .KRZ*       3456K
                                JUNE27           (dir)
                                LONGSMPS         (dir)
Total:21034K                      FLUTE          KRZ*        .5K
Select  Root  Parent  Open  OK  Cancel

```

the files DOORS.KRZ and FLUTE.KRZ would be loaded.

The **Select** button will toggle the selection, meaning that if you press **Select** on a given file, the asterisk will go on if it is currently off, and vice-versa. Selecting can only be done for files, not for directories. You can select as many files as you wish using the **Select** button. There is also a way to select all files at once, or clear all file selections at once, using a double-press of the cursor buttons:

Left/Right cursor double-press -> Select All Files

Up/Down cursor double-press -> Clear All Selections

Pressing either the Left or Right cursor individually performs a separate function for finding directories, described below.

You can only select multiple files from within a single directory. Changing directories clears any selections.

Once you have selected one or more files, press **OK** to perform the disk function (in this example, Load) on all files marked with an asterisk, whether or not they are visible on the screen. If there are no files marked with an asterisk, the function will operate only on the highlighted entry, as in earlier versions of the K2vx system software.

The **Root** soft button returns the listing to the top-level or root directory. If the display is already at the root directory (as indicated by the Dir: \field on the top line of the display) the only effect of pressing **Root** will be reset the file index to 1 if there are files in the directory.

The **Parent** soft button is used to go up one level in the directory hierarchy. If the display is already at the root directory, this button will have no effect.

The **Open** soft button performs a different operation depending on the disk function and the type of file that is currently indexed in the display. In all disk functions, pressing **Open** on a subdirectory (indicated by a "(dir)" after the file name) will open that directory, read in and display the file list.

When a directory is first opened for viewing, the index will be set to one (the first file in the list). You might notice that the K2vx remembers the index the previous directory you were in before you pressed **Open**, so that if you return to that directory by pressing **Parent**, you are put back where you were in the list. This index is only remembered for one level down, and therefore is useful when stepping through a list of subdirectories from a single directory level.

In the Load function, pressing **Open** for a standard .KRZ file will start the Load Object feature. This allows selected individual objects from the file to be loaded in to the K2vx. If **Open** is pressed on a Macro file (.MAC extension), then individual file entries within a macro file can be selected for loading. Both of these features are explained in great detail below.

For all other functions, if **Open** is pressed when censored to a .KRZ or a .MAC file, the object file or the macro file will be opened for viewing. For instance, pressing **Open** on a .KRZ file

while in the Delete function will display the objects within the file in a scrollable list, however no delete action will be possible on the individual objects.

Pressing the **OK** soft button will cause the K2vx to proceed with the selected function. After pressing OK, there may be further dialogs such as bank specification (for the Load function), confirmation (for Delete), or name entry (for Rename). One exception to this is in the Load function, if the file list is scrolled to a directory (dir) entry, then the OK button will enter the directory, the same as if the Open button is pressed on a directory entry.

The **Cancel** soft button exits the file list dialog, completing the disk function with or without any operation taking place. The K2vx returns to the Disk Mode page. Pressing the Exit button will do the same thing as Cancel.

Total

The total size of all the files in the directory is indicated at the bottom left of the file display above the soft buttons. This total represents only the disk space used by the files in the directory being viewed. The K2vx includes a free space utility that can be used find out how much space is being used on the current drive all of the files on the disk. Also, there is a List utility that can be used to calculate the size of all files within a selected directory sub-tree. These functions are described under "Disk Utilities" on page 13-55.

Quick Scrolling to Subdirectories

It is sometimes difficult to locate a subdirectory entry in the file list for the current directory, if there are many files in the current directory. To make this easier, individually pressing either the Left or Right cursor buttons will set the file index to the next or previous directory (respectively) in the current directory list. The index will wrap around the beginning or end of the list, so that repeated presses of either cursor button will cycle through all of the subdirectories. If you have many subdirectories, you can scroll through them all very quickly using this method.

For example, given the following file list display,

```

CYMBALS          (dir)
DOGS             .KRZ      122K
DOORS            .KRZ*    3456K
E4PRQG           .KRZ      10K
LONGSMPS         (dir)
LUTE             .KRZ*     .5K

```

Pressing the right cursor takes you two entries further to the next directory,

```

DOORS            .KRZ*    3456K
E4PRQG           .KRZ      10K
LONGSMPS         (dir)
LUTE             .KRZ*     .5K
MOON             .KRZ*    3456K
TRIANGLE         .KRZ      10K

```

or, pressing the left cursor takes you two entries back to the previous directory

```

ALTO          (dir)
ROOBAMS       .KRZ*      .5K
CYMBALS       (dir)
DOGS          .KRZ       122K
DOORS         .KRZ*      3456K
E4PROG        .KRZ       10K

```

Creating Directories

As stated above, you can create directories for organizing your K2vx files, whether you are using SCSI or Floppy disks. No reformatting is necessary to add directories to disks that were formatted with a K2vx.

Directories appear in the normal file list with the indicator "**(dir)**" to the right of the directory name.

There are two ways to create new directories.

First, the **NewDir** button can be pressed directly from the Disk Mode page, or second, a new directory can be created and entered in one step during the Save operation, by pressing **Save** then **NewDir**.

Creating a Directory with NewDir

When you press NewDir, you will first name the directory that you are about to make, for instance:

```
Directory name: THINGS_
```

```
Delete Insert >>End Choose OK Cancel
```

Pressing **>>End** will take the cursor to the last character in the name. The **Choose** button allows you to grab a file name from the current disk (this is discussed later, in the section titled "Saving Files".) Otherwise, the name will default to either "NEWFILE" after a power-up, or the name will be that of the most recent file saved or loaded. Once you choose a name to start with (or the default), you can edit the name using the left and right cursor buttons, the **Delete** and **Insert** soft buttons, and the **>>End** soft button.

After you have chosen the directory name and pressed OK, you have the choice of where (in what directory) to put the new directory you are creating.

```
Use current directory for THINGS?
(Path = \)
```

```
Change OK Cancel
```

Pressing OK will select the default path, which is the current directory. Pressing Change will allow you to view the disk, traversing its directories, until you find the one in which you want to create the new subdirectory, in this case "THINGS". (see Change Directory Dialog, below)

Creating a Directory with Save -> NewDir

Save->NewDir is used as a convenience when saving files to a directory. This is accomplished by pressing **Save** from the Disk Mode page and then pressing **NewDir** at the "Save what:" dialog. After you have created the directory in this way, the current directory is automatically switched to while you are still in the save dialog. See the section on saving files for more information.

When you have confirmed your choice of what directory to create a new sub-directory in, the K2vx will create the named directory on your drive, and print a brief message such as:

```
Created directory /THINGS
```

If the new directory function was created by pressing the NewDir button on the DiskMode page, the K2vx returns to the Disk Mode page. Otherwise, if NewDir was pressed from the Save dialog, the K2vx returns to the "Save what:" dialog, having set the current directory to be the one just created.

The Directory Selection Dialog

When making a new directory, as well as in many of the disk functions, you will be presented with the opportunity to change the current directory, or the default directory for a disk operation. A good example is the "Use current directory for THINGS? (Path = \)" screen above. If you press "Change" you will see a familiar file list dialog, with a few changes, through which you can select any directory on the disk. The display looks like this:

```
Dir:\                               Sel:0/23       Index: 7
                                MELLOTRN .KRZ      122K
                                N123AB   .KRZ       42K
Select directory: 00THINGS .KRZ    3456K
                                STRINGS   (dir)
                                T1        (dir)
Total:21034K   UNDULATE .KRZ      .5K
Root  Parent  Open          Current Exit
```

When you enter this dialog, you will be in whatever directory was displayed as the default. From here you can go into other directories by using the soft buttons Root, Parent, and Open. Notice that there is no "Select" button. This is because the purpose of this dialog is to choose a single directory as opposed to selecting multiple files. However, the Root, Parent, and Open buttons function exactly as described above (for the file list dialog). The Sel: parameter (on the top line) is useful as an indication of the number of entries in the directory you are looking at.

If you are scrolled to a directory entry, there is one additional soft button displayed, "SetDir". Notice the "Current" button moves over one button to the left:

```

Dir:\                               Sel:0/23       Index:    8
                                N123AB      .KRZ      42K
                                OCEANS      .KRZ     3456K
Select directory:STRINGS          (dir)
                                T1          (dir)
                                UNDULATE   .KRZ      .5K
                                ZORK       .KRZ      .5K
Total:21034K
Root  Parent  Open  Current  SetDir  Exit

```

There are two ways that you can select a directory in this dialog.

Pressing "Current" - This selects the directory you are currently in (whose file list you are viewing), as specified in the Dir: parameter on the top line of the display. For instance if you wished to select the directory STRINGS using the "Current" button, you would first press "Open" to display the contents of that directory, and then press "Current". If you instead wanted to choose the root directory, you would simply press "Current", since that is the directory you are viewing (notice the Dir: \ " at the top).

Pressing "SetDir" - This selects the directory you are scrolled to, such as STRINGS in the display above. This method is often quicker and more convenient than pressing "Open" followed by "Current," which does the same thing. The "SetDir" soft button is only present in the display when the scrollbar highlights a directory entry.

Disk Mode Functions

Now that you are familiar with the basics of creating directories and moving around in the K2vx file system, it is time to discuss some of the features provided in the disk functions themselves.

Loading Files

The **Load** button instructs the K2vx to copy a file, from the currently selected device, to the K2vx's RAM. Press the **Load** button, and a list of files stored in the currently selected device will appear. Scroll through the list of files with the Alpha Wheel or Plus/Minus buttons, then press **OK**—or press **Cancel** to return to the Disk mode page.

When you press **OK**, the Bank dialog will appear, as described in "Load Function Dialog" on page 13-18, and you'll be asked to select the memory bank to load the file into. Scroll through the list of banks with the Alpha Wheel or Plus/Minus buttons until the desired memory bank is highlighted, then press **OK**. Or press **Cancel** to back up a page and select another file to load. Once you have selected a bank to which to load, you will be asked to choose a method for loading. The method you choose determines how the objects in the file will be ordered when loaded into the bank.

Loading Individual Objects

Since files can contain over 3000 objects, it is often useful to load only a subset of the information contained in a K2vx file. Sometimes, this capability is necessary even to be able to load certain files, if the size of the file's samples or data is greater than the K2vx's internal RAM size (in case you haven't gotten those extra SIMMs or SIPP's yet).

You can select individual objects or groups of objects (i.e. Samples, Programs, Keymaps, Effects, Songs) for loading from within a single K2vx file.

The Load Object feature is accessible from within the Load File dialog. To activate it, scroll the file list until you have highlighted the file that you wish to load objects from:

```

Dir:\                               Sel:0/6       Index:    3
                                BASSDRMS .KRZ      426K
                                HIHATS  .KRZ      788K
File to load:SAXES                 .KRZ      2510K
                                TOMS    .KRZ       301K
                                TOMS1   .KRZ      1400K
Total: 5037K                      XCYMB  .KRZ      1012K
Select Root Parent Open OK Cancel

```

Press **Open** to begin the Load Object dialog. (Note: The file must have a .KRZ extension in order to load individual objects from it.) The K2vx then scans the file contents in order to present a list of all of the objects in the file. Sometimes this procedure can take a few moments, depending on how many objects are in the file. During this time, you will see the following display:

Reading file SAXES.KRZ [...]

```

Select Next Type Multi OK Cancel

```

The soft buttons in the above display do not become active until the process of scanning is finished. When this happens, the K2vx will display a list of the file's objects, in the exact order that they are stored in the file:

```

Func:LOAD                               Sel:0/5       Index:    1

Sample 200 Tenor Sax1                  G 2S      250K
Sample 201 Tenor Sax2                  C 3S      179K
Sample 203 Tenor Sax3                  F#3S      198K
Keymap 200 Tenor Sax                   176
Select Next Type Multi OK Cancel

```

The objects in the list are usually grouped by type (Sample, Program, Keymap, etc.) The list can be scrolled using the Alpha wheel or the up/down cursors. The CHAN/BANK buttons on the front panel can be used for fast scrolling. The list will jump by five entries at a time, moving the entry on the bottom line to the top line.

NOTE: When scrolling through large numbers of objects (more than 100), the K2vx may sometimes pause for a few seconds if it needs to get more information from the disk file. When this happens, some gyrating dots will briefly appear in place of the Index value on the top line of the screen.

Each line in the scrollable list represents one object, and displays the object's type, ID, name, and size. Samples have additional information, the sample's root key and a stereo sample indicator:

Object Type	ID	Name	Root Key and Stereo Indicator (samples only)	Size
SamPle	203	Tenor Sax3	F#3S	198K

The ID numbers are the same numbers that were used to reference the objects when the file was last saved by the K2vx. These numbers will usually be different after the objects are loaded in, depending upon the bank (i.e. 200...299) and mode that is specified for loading. There is more information on these modes below, in the section "Load Function Dialog."

The Size field is interpreted differently for samples and non-samples. For non-samples (i.e. Songs, Programs, etc.), it shows the number of bytes used by the object in the file, and hence the amount of program memory that the object will occupy in the K2vx. For samples, the size field shows the size of the all sample data associated with the object, and is displayed in kilobytes (K).

For samples, the letter 'S' after the root key indicates a stereo sample.

Due to display space constraints, if the sample's root key happens to be in the lowest MIDI octave range (i.e. C -1 through B -1), it will be displayed in a truncated form. For instance, if a sample's root key was set to G#-1, the display would read:

SamPle	293	Tenor Sax	G#-	198K
--------	-----	-----------	-----	------

The status line at the top of the screen specifies the object function being performed,

Func:LUHU

the number of selected objects in the list followed by the total number of objects in the file,

Sel:0/5,

and the current list index:

Index:1

As with the file list, entering in a number from the numeric keypad will jump to the indexed entry, and typing in a large number like 9999 will go to the end of the list.

The soft buttons on this page are used for multiple selection of the objects in the list as well as for moving around the list when there are many items selected or listed. This same dialog is also used for many other functions in the K2vx, namely for saving selected objects to disk and for several object utility functions that are described later.

Here is a brief description of each button's function, followed by a detailed explanation of its operation.

Select - Select or deselect an object.
Next - Jump to the next selected object.
Type - Jump to the next object of a different type.
Multi - Go to the Multiple Object Selector page.
OK - Tell K2vx to proceed to load the selected objects.
Cancel - Exit back to the File List Dialog.

Press the **Select** button to choose the highlighted object for loading. An asterisk (*) is placed in between the object name and the object ID for any items that are selected. Deselect a selected

object by pressing **Select** again. The asterisk will disappear. The easiest way to choose objects for loading is to scroll the list and individually press **Select** on each object you want to load.

If you only want to select one object for loading, you need not select it with the **Select** button. Instead, pressing **OK** implicitly selects the highlighted object if there are no other objects selected. If there are objects selected, however, then the highlighted object will not be loaded unless it is selected.

This screen shows 3 samples selected for loading:

```

Func: LUHU      Sel: 3/20      Index:      4
Sample  201*Machine Snare      C 4S      100K
Sample  203*Big Bass Loop      C 4S      218K
Sample  304*Distorted Riff     C 4S      401K
Sample  305 EPiano1           D 2       24K
Sample  306 EPiano2           D 3       25K
Sample  313 EPiano3           D 4       15K
Select  Next  Type  Multi  OK  Cancel

```

The **Next** button will cause the index into the list to jump to the next selected object, forward in the list. When the end of the list is reached, the search will wrap around from the beginning. If you have more than one object selected, then if you repeatedly press **Next** you can easily cycle through all selected items. If there are no items selected, then this button doesn't do anything.

The **Type** button jumps to the next object of a different type from the one that is currently highlighted. This is a convenient way to find a particular type of object in the list. If you want to skip over the samples and the keymaps in an object list and jump right to the Programs, press **Type** about two or three times, stopping when you notice that a Program is highlighted.

Press the **Multi** button to enter the Multiple Object Selector (described on page 13-32.) This powerful utility can be used to control the selection or deselection of many objects, cross-referenced by object types and ranges of ID numbers.

When you are all done selecting objects to load, press **OK**. As stated above, if only one object is to be loaded, it is implicitly selected if it is the currently highlighted object and there are no other selected objects in the list.

Cancel returns to the file list dialog, highlighting the file you just opened. You can load the entire file after pressing **Cancel** by pressing **OK** when you return to the file list.

Shortcuts when Loading Objects

Select All/Deselect All

Selecting or deselecting all of the objects at once can be done with the same double-presses as described for the file list dialog, namely:

Left/Right cursor double-press -> Select All Objects

Up/Down cursor double-press -> Clear All Selections

If you want to load most but not all of the items from a file (for instance, if there happens to be a Master Table in the file that you don't want to load), it may be fastest to first select all objects using the Left/Right double-press, and then manually deselect any unwanted items.

Viewing the Name Table

The Name Table is an object that appears in files that were created using the "Save dependents by name" feature (described below, in the section on the "Save" function). This object contains a list of dependent objects needed by the other objects in the file at the time the file was saved. There is more information about this later on, however it is worth mentioning here that a highlighted Name Table object's contents can be viewed by pressing either one of the Left or Right cursor buttons.

Left cursor OR Right cursor -> View Name Table Contents

Loading Dependents of Selected Objects

When you press OK after selecting one or more objects, the K2vx may ask the question:

Load dependent objects?

Yes No

This question will be asked if any of the selected objects could possibly have dependents associated with them in the file. (Remember, dependents are those objects needed by other objects; Samples are dependents of Keymaps, Effects and Keymaps are dependents of Programs, and so on.) If this question is asked it does not necessarily mean that there really are dependents of the selected objects. The K2vx will not know whether there are dependent objects in the file until it begins to read in the selected objects, and determines what their dependents are.

Answering "Yes" to the question tells the K2vx to also load the dependents. You may wish to answer "No" if, for instance, you are simply loading in a Program or a Keymap as a template for use with other objects. You can also manually select only some of an object's dependents, and then answer "No" to "Load dependent objects?" to prevent other unwanted dependents from being loaded.

To summarize, it is not necessary to select any of the dependents of an object if you plan on loading all of the dependents in. As an example, for a file containing dozens of Programs, Keymaps, and Samples, you may choose to highlight a certain Program and press OK, and answer "Yes" to the "Load dependent objects?" question. The K2vx will do the rest, by only loading in the samples and keymaps that are needed by the selected program.

Similarly, if you were only to select certain keymaps from a file, and then answered "Yes" to "Load dependent objects", the K2vx would figure out exactly what samples need to be loaded in as dependents of the selected keymaps.

Auditioning Samples from a Disk File

Often when working with files that contain samples it is helpful to be able to hear what the samples sound like before loading in all or part of the file. It is possible to audition samples in the file, from within the Load Object dialog.

To audition a sample, first scroll to the sample that you wish to hear. Then, press either the Left or Right cursor. The K2vx will load in up to 1 second of the sample, or less if the total sample time is shorter than one second. The audition starts from the very beginning of the sample data

(note that if the first second of data is silence then you won't hear very much when the sample is auditioned.) If the loop points fall within the first second of the stored sample data, they will be loaded in as well. The K2vx display will "blink" after the completion of loading the sample audition data. When the sample segment has been loaded, it can then be played back at its root key as well as transposed up and down the keyboard.

Once a sample has been auditioned, it remains active across the keyboard until another sample is auditioned. The audition function ends when either OK or Cancel are pressed.

There must be sufficient sample RAM in the K2vx to load in one second of the sound for auditioning. This amount varies according to the sample rate of the sample, but for most samples this will be less than 100K bytes. If the auditioned sample does not play, check that there is enough free sample memory in the K2vx. It is also possible to see the following error if the K2vx object RAM is full or very near full:

Not enough memory to audition

Loading Objects from Floppy Disk Files

Individual objects can be loaded from K2vx floppy disk files, with the caveat that some samples in a multiple-floppy disk file cannot be auditioned because they reside on more than one disk. These samples will still appear in the object list, but their size will be in parentheses meaning they cannot be loaded (either explicitly or as dependent objects) and cannot be auditioned.

Load Function Dialog

Bank Status Indicator

After you have chosen what you wish to load, you are presented with a dialog allowing you to determine what bank will be used to load in the file's data.

This indicator takes the form of an asterisk after the bank. If an asterisk is present after the bank (i.e. 400...499*), it means that there are objects in the bank, whether they are RAM or ROM objects. ROM objects are in the 0's and 100's bank, unless you have an optional ROM block installed. In that case ROM objects are also stored in the 800's and/or 900's bank. If there is no asterisk on the line for a bank, it means the bank is empty.

In the following screen, there are user objects in the 200's and 400's bank, and possibly also in other banks that become visible when the selection is scrolled.

```
Load this file as: 200...299*
                   300...399
                   400...499*
                   500...599
                   OK Cancel
```

This indicator makes it easier to find an empty bank to use for loading, if needed.

Loading Methods

Once you have pressed **OK** to decide on what bank to use, you will see this screen if the bank is empty:

```

Load this file as: 200...299*
                   300...399
                   400...499*
                   500...599
                   600...699
                   Append Fill Cancel

```

You will see the following screen if the bank contains any objects (in RAM or ROM):

```

Load this file as: 200...299*
                   300...399
                   400...499*
                   500...599
                   600...699
                   700...799
OvFill Overwrt Merge Append Fill Cancel

```

The soft buttons control the mode for loading and renumbering of objects from the file. The meaning of these soft buttons is:

- OvFill** First deletes all RAM objects in the selected bank, and then loads in objects using consecutive numbering.
- Overwrt** First deletes all RAM objects in the selected bank, and then loads in objects using the object ID numbers stored in the file.
- Merge** Preserve the object ID numbers stored in the file for the objects to be loaded, overwrite objects already in memory if necessary.
- Append** Try to use the object ID numbers stored in the file for the objects to be loaded. If an ID number is already in use, increment the ID number until a free slot is found.
- Fill** Ignore the object ID numbers stored in the file. Try to use consecutive numbering from the beginning of the selected bank. If an ID number is already in use, increment the ID number until a free slot is found.
- Cancel** Cancel the mode selection, and go back to choosing a bank. Scrolling to a different bank value will have the same effect as Cancel.

Typically, you will just want to use the **Fill** method. **Append**, **Merge**, and **Overwrt** try to preserve the numbers stored with the objects in the file, but this should only really be necessary if you depend on program numbers or effect numbers to be at a certain MIDI program change number. **OvFill** is like **Fill** except the selected bank (or Everything) is cleared out before loading.

Overwrt and **OvFill** operate in different ways after a selected bank has been filled up for a given object type (for example, after you have loaded in more than 100 programs into a bank). **Overwrt** will begin continue to preserve the objectIDs stored in the file, and will individually overwrite objects in the bank following the just filled bank. **OvFill** will not overwrite any further objects past the end of the selected bank; it instead skips over object IDs that are in use. Because of this difference, it can sometimes be faster to load in a file using **OvFill** rather than **Overwrt**. However, this only applies if the objects to be loaded would extend past the end of a selected bank.

Note that when loading into a specific bank (as opposed to loading as "Everything"), the object ID's in the file are used as follows: The "bank" digit is ignored, and the remainder of the number is used when the K2vx re-banks the object ID into the bank that you specify. For instance, if you save Program 453 into a file, and load it back into the 300's bank, the K2vx will use the number "53" when deciding upon a new object ID. If the 300's bank was previously empty, and the load mode is Append, then the Program will end up with ID 353.

For loading as "Everything", the ID number for an object stored in a file is taken literally, and not re-banked (except if Fill or OvFill mode is chosen, in which case the K2vx will use ID numbers starting from 200.)

Here is an example that compares all of the above load modes.

EXAMPLE: Starting with the following objects already stored in the K2vx internal RAM:

```
Program 200 Existing Prog1
Program 204 Existing Prog2
Program 205 Existing Prog3
Program 210 Existing Prog4
Program 211 Existing Prog5
```

Suppose you were to load file containing the following objects into the 200's bank:

```
Program 405 Electric Piano
Program 406 Stereo EPiano
Program 409 Weird Piano
Program 410 Wild Funk Piano
```

Here is what these programs would end up with as ID numbers when loaded in, depending on the load mode:

<u>Name</u>	<u>Append</u>	<u>Fill</u>	<u>Merge</u>	<u>Overwrt</u>	<u>OvFill</u>
Electric Piano	206	201	205	205	200
Stereo EPiano	207	202	206	206	201
Weird Piano	209	203	209	209	202
Wild Funk Piano	212	206	210	210	203

In the Merge case, the existing programs 205 and 210 would have been deleted. In the Overwrite case, all of the existing programs in the 200's would have been deleted.

Multiple Selection of Files to Load

As stated previously, you can select multiple files for loading into the K2vx from within a single directory, in one operation. This is done from the file list dialog with the **Select** button.

After you have selected one or more files in this way, you will still choose a bank and mode to be used for the load process, just as with loading a single file. However, the dialog prompt will say "Load selected as":

```
Load selected as: 200...299*
                  300...399
                  400...499*
                  500...599
                  OK Cancel
```

If you selected any macro files (.MAC extension) from the directory, then once you have select the mode for loading, you will see the question:

```
Load macros as specified?
```

```
Yes No
```

The answer to this question instructs the K2vx that any macro files will have their macro entries loaded according to the bank and mode:

Yes specified in the macro entry.

No currently specified for this multiple file load. In other words, whatever you select for Bank and Mode will override the instructions for each entry in the macro.

At this point the files will begin to load. When all the files have been successfully loaded or the load process has been aborted, the K2vx returns to the Disk Mode page.

If there are any errors encountered during a multiple file load, such as running out of object RAM, you will be asked once if you wish to abort the load. In some cases, you may wish to continue loading. If you continue (and don't abort), the only way to abort will be to use a special procedure described in the next paragraph.

Aborting a Multiple File Load

There is a way to abort the process of loading multiple files. Aborting can only be done "in between" files that are being loaded, and not during the load of any one file (short of powering off or soft-resetting the machine by pressing +/-, 0, and CLR simultaneously, but this is not recommended!).

Aborting a multiple file load is done by pressing and holding down either of the Plus (+) or the Minus (-) buttons that are located just below the Alpha wheel. This should be done at least one-half second before you anticipate the current file to finish loading, or else the K2vx will not sense that you wish to abort the load.

You will see the following question after the current file being loaded is completed:

Abort the load?

Yes No

It may be a good idea to practice using this method of aborting a multiple file load, so that when the time comes that you accidentally select 100 files, you will remember how to abort the process. This same method (of holding the Plus or Minus buttons down) is also used to abort the Backup feature and the Macro file load feature.

If you run out of object or sample RAM, you will have one opportunity to abort the load as explained above. However, if you continue from that point you may end up seeing the same error message "Memory is full" for each file that you had selected. This can be a rather tedious process, however it is still possible to abort out of this by holding down the Plus or Minus button simultaneously while pressing "Yes" when you see the following question:

Abort this Partial load?

Yes No

More Load Function Enhancements

There are more features having to do with the Load function that are described later on in this guide (i.e. loading Macro files, loading AIFF files, etc.)

Saving Files

The Save button starts the process of saving from the K2vx to the currently selected device. When you press the **Save** soft button on the Disk Mode page you will see the bank dialog:

```
Save selection: 200...299
                300...399
                400...499
                500...599
Export Macro Object NewDir Ok Cancel
```

The **Macro** soft button will only be present if Macro recording is turned on. (See page 13-38 for more information on Macros.)

You can save an entire bank of objects, or by pressing the **Object** soft button, select individual objects to be saved. If you choose to save using the bank method, all RAM objects within that

bank will be saved. (You cannot save ROM objects. If you wish to save a ROM object, such as a program, you must first save it internally as a RAM program.) If any objects within the selected bank have dependent RAM objects that exist in a different bank, you will be asked if you want to save dependent objects. See page 13-26 for more on saving dependent objects.

Use one of the data entry methods to select a bank to be saved. If you press the **Cancel** soft button, you'll return to the Disk mode page. After you've selected the bank, press **OK**. The following page will appear:

Save as: **NEWFILE**

Delete **Insert** **>>End** **Choose** **OK** **Cancel**

You can now name the file according to the naming procedures outlined in Chapter 5. You can enter up to eight characters. When you've entered a name, press "OK" to save the file as shown in the display, or press Cancel to return to the file dialog. When the file is saved, the K2vx adds an extension (.KRZ) to the filename. This enables the K2vx to recognize it as a Kurzweil file when it examines the disk's directory.

Saving Master and Everything Files

Among your choices in the Bank dialog are Master files and Everything files. Master files consist primarily of the items on the Master mode page and the three MIDI mode pages. They also include information like marked pages, view settings, and MIDI channel and program assignment. In fact, saving Master files (or dumping them via SysEx) is a good way to configure your K2vx (or another K2vx) to your performance or sequencing needs. For example, you might save different Master files with every sequence you create using an external sequencer. Then, when you load the Master file, you would have all the correct programs assigned to the appropriate MIDI channels.

Everything files consist of the Master file parameters and every other RAM object. Saving an Everything file will literally save everything in RAM, including samples, into a single file. These can be quite large, so if you don't have a hard disk, be sure to have a few preformatted floppy disks handy.

Split Files

When you're saving memory banks, it's not uncommon to create files larger than 720K or even 1.4M—especially when you're saving RAM samples. If you're saving to floppy disks, the K2vx will create split files stored on multiple disks. A little advance preparation is necessary for this.

If the K2vx fills a floppy disk before it finishes saving a file, it will prompt you to insert a second disk in the floppy drive. This disk must be preformatted; the K2vx will not format disks in the middle of a save operation. When the second disk is inserted, the K2vx will continue to save. This process will be repeated until the entire file is saved. The K2vx marks these disks internally with a number that indicates the sequence in which they were saved.

When you're loading these split files, they must be inserted in the same sequence as they were saved. When the K2vx has loaded the contents of the first disk, it will prompt you to insert the second disk, and so on. Make sure to label your disks as soon as you save to them, so you'll keep them in the correct order.

Soft Buttons in the "Save Selection" Dialog

The meaning of the soft buttons in the "Save selection:" dialog is as follows:

Export Save a sample or a song in an exported file format (i.e. AIFF, WAVE, MIDI Type 0). This feature is described in the section "Using Standard File Formats."

Macro Save entries from the current macro table as a Macro file (.MAC). This soft button is only displayed if macro file recording is on. This is described later in the section "Saving Macro Files."

Object Save selected objects from the K2vx's RAM.

NewDir Create a new directory on the current disk, and return to this dialog afterwards. This is described previously in the section "Creating Directories."

OK Save all the objects from the highlighted bank (i.e. 200...299), and optionally also save dependent objects.

Cancel Exit from the Save function.

Export, **Macro**, and **NewDir** are all explained elsewhere in this guide. This section will describe the process of saving K2vx objects into K2vx format disk files.

Saving Individual Objects

You can select any group of objects in the K2vx's RAM for saving into a single file. To save individual objects, from the above dialog, press **Object**. The K2vx will display a scrollable list of all the objects in RAM, very similar to the display for the Load Object feature (described previously):

```

Func:SHVE      Sel:0/8      Index: 1

Sample 500 Lo Vocal      A 3S 250K
Sample 501 Hi Vocal      G 4S 179K
Keymap 500 VocalsMap      176
Program 500 Dry Vocals    270
Select Next Type Multi OK Cancel

```

The procedures for saving objects are essentially the same as the procedures described on page 13-13 for loading objects.

Shortcuts when Saving Objects

Select All/Deselect All

Selecting or deselecting all of the objects at once can be done with the following double-presses (two front-panel buttons simultaneously pressed):

Left/Right cursor double-press -> Select All Objects

Up/Down cursor double-press -> Clear All Selections

If you want to save most but not all of the items from a file (for instance, if there are some songs in RAM that you don't want to be saved in the file), it may be fastest to first select all objects using the Left/Right double-press, and then manually deselect any unwanted items.

Viewing Selected Objects

When there are lots of objects selected, but they are scattered in the objects list, it can be helpful to be able to view a list of only the currently selected objects. Such a list is displayed by double-pressing the CHAN/BANK buttons to the left of the K2vx's LCD display.

Double-press of CHAN/BANK buttons -> View Selected Objects

For instance, if there were 10 objects selected, and you pressed both CHAN/BANK buttons simultaneously, the K2vx would show a list similar to this:

```
View Selected Objects 10/134
Program 300 Biggest Kit 7124
Program 301 Repercussions 7124
Song 300 Drum Groove 7 12092
Song 301 Drum Groove 8 24700
Song 421 Nasty Funk 3122
Song 500 Beethoven 1024
OK
```

The top line shows 10 objects selected out of 134 that are in RAM. If the number of selected objects is larger than the 6 objects that fit on one screen (as in this example), the list can be scrolled to view all of the information.

Note that this feature is not available in the Load Object dialog.

Auditioning Objects in RAM

When deciding which individual objects to save, it can be difficult to know if you are selecting the correct ones. This is especially true if many objects have similar or identical names, or if the names of the objects are not descriptive enough to know what they are. The K2vx has a feature that allows auditioning of Samples, Keymaps, Programs, and Songs right from the Save Object dialog (as well as all of the other object utility dialogs that are discussed later.) To activate this feature, scroll to an object of an appropriate type to be auditioned, and press either the Left or Right cursor button. The screen will "blink", and the objects can now be heard as follows:

Samples play at their root key, as well as transposed across the keyboard. Stereo samples will play in stereo. Auditioning samples in this way is similar to listening to samples from the **MASTER->Sample** page. The samples are auditioned using a "hidden" program set up according to the parameters in **Program 199 Default Program**. This default program can be customized if needed by editing and saving a new program 199.

If you audition any sample objects, the last one that you audition will become the "preview" sample the next time you go to the **MASTER->Sample** page. This can be a quick way to edit the sample without having to edit a program and a keymap.

Keymaps are reproduced accurately, and are played according to the parameters in Program 199 Default Program. This default program in ROM is set up to have a 0% effects level (dry). Therefore, auditioning keymaps can be a very convenient way to hear them isolated from the effects.

Programs play exactly as they would if they were selected from the Program Mode page.

Songs start playing when either the Left or Right cursor button is pressed, and stop playing when either cursor is subsequently pressed. The most recent song that is

auditioned from this page become the Current Song (as seen on the **Song Mode** page).

Once auditioned, the above object types remain active on the keyboard until another object is auditioned, or **Cancel** is pressed. If a song is being auditioned, no other objects are auditioned until the song audition is stopped (by pressing one of the Left or Right cursor buttons).

Saving Dependent Objects

When you save a file, you may see a prompt as part of the Save dialog that asks you whether you want to save dependent objects. A dependent object is simply an object that's associated with another object. The dependent object can be stored in a different memory bank, for example, a RAM sample with ID 301 that's used in a program with ID 402, or in the same bank as the file being saved. Rather than forcing you to save dependent objects separately and to keep track of them yourself, the K2vx gives you the option of automatically saving the dependent objects as part of the file you save. When you load the file again, the dependent objects will be loaded along with the objects to which they're attached.

There are a few things to keep in mind regarding dependent objects. First, it's not uncommon for RAM samples to be dependent objects, and they can take up quite a bit of memory. If you save a RAM sample as a dependent object when you save a bank of programs to disk, you may create files which won't fit onto a 720K or 1.4M floppy disk. Since the K2vx can't format floppy disks in the middle of a save operation, you should have spare formatted disks ready to go before you start saving. See the section called "Split Files" on page 13-23.

While the K2vx makes it easy for you to keep track of your dependent objects, you need to keep aware of what happens with dependent objects when saving to disk and reloading. Consider this example. Suppose you create 30 new programs, each of which uses a keymap containing four different RAM samples. If you save these programs to a disk file, and save dependent objects with them, you've created a file containing 30 programs and 120 dependent RAM samples. So far, so good. Suppose you then load that file into the 300s bank. The K2vx will load the 30 programs into the 300s bank just fine, but it will be able to load (at most) only the first 100 dependent objects to the 300s bank (each memory bank can hold a maximum of 100 objects of a given type). The remaining 20 dependent objects will be loaded into the 400s bank. If there are no objects of the same type in the 400s bank, there's no problem. But if there are objects of the same type in the 400s bank, some or all of them will be replaced by the newly loaded dependent objects.

The easiest way to prevent this is to make sure that you don't create more than 100 dependent objects attached to the other objects in a given memory bank. The easiest way to do *this* is to avoid creating dependent objects when possible, by saving objects with IDs in the same memory bank as the objects to which they're related. For example, if you create a program that uses RAM samples, and you save the program with ID 201, resaving the RAM samples used by that program with IDs in the 200s will prevent dependent objects from being created for that program. If you do this, you'll minimize the number of dependent objects you create, and you'll be unlikely to force dependent objects to be loaded into a higher-numbered memory bank when you load files.

Once you have selected objects for saving (either individually as just described or by bank selection), the K2vx will determine if any of the items chosen to save have any dependent objects in RAM that were not chosen. For instance, if you select a program to be saved and nothing else (using the Save Object feature), the program may have dependent effects, keymaps, and samples that are in RAM. Dependent objects that are in ROM (i.e., ROM samples or keymaps) do not get saved to disk.

You will see the following dialog displayed if there are any dependent objects in RAM of any objects that were selected for saving:

Save dependent objects?

Names **Yes** **No**

Choosing **Yes** will cause any dependent objects to be saved in the file together with the selected objects. Choosing **No** means that unselected dependents will not be saved. The **Names** button creates a new kind of object to be stored in the file, called the Name Table.

The Name Table

The Name Table is a list of any dependent objects that were not explicitly selected for saving in the file. Each entry in the Name Table contains the object type, object ID, and the name of a dependent object.

A file's Name Table is used by the K2vx at only one time, when the file is loaded in. At that time, the K2vx will search for dependent objects that were not saved in the file originally. The search matches dependent objects by name with objects that are already in RAM, and links them to the "parent" object. The Name Table data is then discarded when the file load is finished. This search feature is referred to as **Relink-by-Name**.

Relink-by-name offers new and very efficient ways of working with K2vx objects and disk files. Careful use of this feature can save you many megabytes of disk storage, as well as a lot of time spent working on music and production instead of waiting for sample data to be resaved.

Relink-by-Name allows you to save objects and their dependent objects separately (in multiple files) and be able to link them up later on by loading the files in the correct order. This can be a very efficient way of working with the K2vx's many levels of dependent objects. The most common way in which Relink-by-Name speeds up development of sounds is when making small adjustments to a program that has as its dependents a large amount of sample data. you can separate the program and sample data, so that after changing a program parameter, only a file containing the program and a Name Table need be re-saved.

When loading in a file that contains a Name Table, the following rules should be observed in order for correct relinking to occur.

1. **Use unique names for dependent objects at every level.** As an example, this would mean that if you were going to be re-linking several samples from one file with a program and a keymap from another file, that each sample should have a different name. Otherwise, the dependent objects (the samples) will not get re-linked properly. This will create problems such as keymap ranges that don't play as they are supposed to.
2. **The dependents to be re-linked must already be loaded.** Otherwise they will not be found and re-linked when the file containing the parent objects is loaded in. This constraint on the order of file loading can be made easier to work with by using the Macro file feature (described later). You can construct a Macro file to automatically load in the dependents files and the parent files in the correct order, making sure that any files containing dependents are loaded first. An alternative to loading the files with a Macro would be to save the dependent and parent files in the same disk directory with similar file names such that they will appear consecutively in the alphabetized file list. Once you have done this, it is easy to select both files for loading in the correct order.

These rules may appear complicated at first, but they will seem natural once you have worked out a few examples with your own files.

The **Search Algorithm** used for relinking dependent objects to their parent objects during loading is as follows:

The search for a dependent object (whose name matches that of an entry in the name table) begins at the beginning of the bank that is specified for loading the parent file. All possible IDs are then consecutively searched. When the last ID of the 900's bank has been searched (typically 999), the search will wrap around to ID 1 up until the end of previous bank to the specified bank. The search stops once a dependent with a matching name has been found and relinked.

For example, if a file containing a one-layer program is loaded into the "400's" bank, and the file includes a Name Table that lists the layer's keymap by name, then the K2vx will begin to look through all possible keymap IDs starting at 400, until ID 999. The search then continues from ID 1, stopping at ID 399. If the search does not successfully find a match, the dependent will be unresolved, and in this example the program would have an "*object id* Not found" indication in its Keymap parameter, where the *object id* is the value that was stored in the file.

The search is done in this "circular" manner so as to allow you to direct which dependent objects get re-linked. This may be necessary if you end up with multiple copies of dependent objects with the same name; you can differentiate between them by loading the parent file into a specific bank that is the same bank or "before" the bank containing the objects you wish to re-link to. Note that this can only be taken so far, since it would be impossible for the K2vx to differentiate between objects with the same name within the **same** bank.

The re-linking process happens in the background, without any notification or error messages if items cannot be relinked.

Working with Relink-by-Name

Here are a couple of more in-depth examples that can show how Relink-by-Name works in a practical situation.

Consider that your K2vx's RAM contains the following one-layer program and also its dependent keymap and samples (the techniques used in this example could well apply no matter how many programs with any number of layers you want to save):

Program: **Program 317 Steinwave Piano**

Keymap: **Keymap 300 Steinwave Piano**

Samples: **Sample 300 StwaveG1 Sample 310 StwaveC7**

In this case you might wish to save the samples and the keymap in one file, and the Program in another file. So, from the Save Object dialog you could first select all the Samples 300-310, and Keymap 300, for saving into a file, let's say **STWAVE1.KRZ**.

You would then return to the Save Object dialog and save just Program 317 in a separate file in the same directory, let's say **STWAVE2.KRZ**.....only this time, you will be asked the "Save dependent objects" question pictured above. Answer this by pressing **Names**.

After saving, the file STWAVE2.KRZ will contain two objects in it, Program 317 and a Name Table. You can easily verify this by going to the Load function (or any other disk function) and

pressing "Open" on the file just saved (which should come up already highlighted). The display of objects for the file will look like this:

```
Func:LOADV Sel:0/2 Index: 1
```

```
Table 36 Names 334
Program 317 Steinwave Piano 274
```

```
Select Next Type Multi OK Cancel
```

The Name Table will always be the first object in the list. You can verify the exact contents of the name table by using the "View Name Table" shortcut (previously described in the Load Object section); make sure the Name Table object is highlighted, and **press either the Left or Right cursor button** on the front-panel (as if you were "auditioning" the Name Table object.) You would then see the following informational display:

```
Name Table Contents
Keymap 300 Steinwave Piano
Sample 300 StwaveG1
Sample 301 StwaveD2
Sample 302 StwaveB2
Sample 303 StwaveE3
Sample 304 StwaveB3
Sample 305 StwaveG4
```

```
OK
```

The **Name Table Contents** display shows what would have been saved in the file had you answered Yes to "Save dependent objects?" instead of answering by pressing Names. More importantly, it allows you to see what objects will need to be already loaded in to the K2vx before loading this file.

The object IDs shown in the table are the same numbers that those dependent objects used at the time this file was saved. (The ID numbers are necessary in order for Relink-by-Name to function, since they are the "link" between the higher level objects and the names of the dependents.)

An important thing to notice about this particular Name Table is that the Sample names are not needed by the K2vx for relinking purposes. In fact, the only information necessary for relinking the dependent objects of this file is the Keymap object. The reason for this is that when this file containing the program is loaded, all of these dependent objects should already have been loaded, and the Keymap should already be correctly linked to the Samples. Although the Samples' names are redundant from the K2vx's point of view, they are included for free, so to speak, and you may find them very helpful if you ever need to know exactly what the dependents of this file were intended to be.

The Name Table Contents List is scrollable if there is object data past the 7 lines that fit on the screen.

Now that the two files STWAVE1.KRZ and STWAVE2.KRZ have been created using the Name Table, they can be loaded back in and correctly relinked. The files can be loaded into any bank - they do not need to go back into the bank they were originally in - since the STWAVE2.KRZ file will search through all the banks to find the objects by name in order to relink them. In fact, if you were to immediately reload just the file containing the Program (STWAVE2.KRZ), into any bank, you would find that it was automatically relinked to the correct Keymap, since the Keymaps and samples are currently in memory.

Furthermore, you could edit the program and create more variations of it that reference the "Steinwave Piano" keymap, add ROM layers, and / or effects programs if desired, and resave all of the programs (and any effects) to the same or a new file (remember to press **Names** when you are asked "Save dependent objects?".) You never have to resave the file STWAVE1.KRZ that contains the keymap and samples, if all you have done is edited the programs or added more of them. This can be a tremendous time-saver.

If the Keymap and Sample files are found on a CD-ROM disk, then using Relink-by-Name is not only a time-saver, but a disk space saver as well. If you like the samples and keymaps from a CD-ROM file, there is no need to duplicate the sample data on your own writable hard drive. Instead, all you have to do is save a program file in the above manner, and then make sure the CD-ROM file is loaded first before you load the program file.

If you needed to add some sample data to the file (for instance, you want to add a root to the keymap or process and reloop a sample from the CD-ROM), you can do this by explicitly selecting the new sample data and the keymap for saving along with the program and the Name Table. Then, the new sample would not be listed in the Name Table (it would be in the same file as the Name Table), and the Keymap would be relinked to all of the samples by name instead of the program being relinked to the keymap (as before). What you put in the different files is up to you, and there is no limit to where you can break up the objects in one file or another. The main thing to be aware of are the two rules for Relink-by-name mentioned above:

1. **Always use unique names for like objects types.** (NOTE: In cases where duplicate names do exist in different banks, load the file that contains the name table into a bank equal to or preceding the bank in which the file with dependent objects was loaded.)
2. **Files containing dependent objects must be loaded first.**

As you will see later, you can create a **Macro file** that will automatically load both of the files in the correct order, no matter what drives they are on or what disk directories they are in. By using Macro files in this way, you can avoid having to explicitly load 2 files and remember the correct order each time.

You can also use the **Multiple Object Selector** (described in the next section) to help in the process of identifying dependent objects and parent objects that you want to place into separate files. For instance you could easily select all dependent Keymaps and Samples of any group of programs, to create a "dependents" file. Then, you could quickly select the programs and any other objects that you wanted to be relinked later on, and save them in another file.

Here is another practical example using Songs (sequences). Suppose you have loaded several files into your machine, such that you now have all your favorite instruments on-board in the K2vx's RAM. Then, you make a bunch of songs using a combination of ROM programs and the RAM programs you loaded in.

The dependent object structure of the songs would look something like this:

Songs:	Song 400 Wild Jam ,Song 401 Memphis Groove
Programs:	Program 600 Drawbarz Program 245 FendJazzBass Program 231 Funky GTR Program 400 ObieWarble Pad Program 103 3 Layer Kit (from ROM)
Effects:	ROM effects
Keymaps, Samples:	Lots of them...

In this case you might want to save all of the songs in one file, and be able to automatically relink the dependent programs used by the song tracks. All of the programs are presumably already saved in separate files. The only file that needs to be created is one that contains all of

the Song objects, plus a Name Table. Once again, this is done by selecting the Songs from the Save Object dialog, and answering Names to "Save dependent objects?". The contents of this file can then be displayed by pressing Open (as was done for the previous example).

```

Func:LOAD    Sel:0/2    Index:    1

Table    36 Names    700
Song     400 Wild Jam    12114
Song     400 Memphis Groove    34002

Select  Next  Type  Multi  OK  Cancel

```

Also as shown in the previous example, you can display the contents of the Name Table:

```

Name Table Contents
Program 231 Funky Gtr
Program 245 FendJazzBass
Program 400 Obie Warble Pad
Program 600 Drawbarz
Keymap  220 Funk Guitar
Keymap  229 Jazz Bass

                                OK

```

Notice that the ROM program, "Program 103 3 Layer Kit", will not be listed in the Name Table. Any dependent objects that are in ROM do not need to be relinked by name. ROM objects are always directly referenced by their object ID number, since they don't get saved in any files.

Once the Song file has been saved, it can be loaded in at any time and correctly relinked, as long as the other files containing the necessary programs have already been loaded.

For this type of situation, where you may be working on songs always using a consistent set of programs, it is beneficial to make a Macro file that can be loaded in one step to direct all of the various program files to be loaded in. After that, any time you load in a song file containing a Name Table referencing these programs, the songs should get relinked to the correct programs.

If you happen to have multiple copies of the necessary programs already loaded into different banks, you can control which bank of programs will be linked to the songs by choosing a certain bank to load the song file into. The relinked programs will be the first set encountered according to the **Relink-by-Name search algorithm** defined above.

Not Loading the Name Table

There may be a time that you wish to load in objects from a file containing a Name Table, but you don't want the K2vx to relink any dependent objects according to the Name Table. This can be accomplished by "Opening" the file from the Load function, and selecting any desired objects from within the file, **except** the Name Table object. The selected objects will be loaded in to the bank you specify, however the Relink-by-Name mechanism will not function.

Relink-by-Name Processing Time

Normally, the time taken to relink several dependent objects using the name search will be insignificant, relative to the time it takes to load in the data from the file. However, if you are attempting to relink a very large amount of dependents by loading one file (i.e. say, 200 samples or so), there may be a noticeable wait while the K2vx searches its object database for the dependents. If this happens, it's best to be patient.

The Multiple Object Selector Page

The Multiple Object Selector is used for making multiple selections of objects from an object list according to combinations of selection modes such as object type, object ID, and dependencies. This page is available for use in the **Load Object** dialog, the **Save Object** dialog, and all of the object utilities that are described in the Master mode chapter (**Move**, **Copy**, **Name**, **Delete**, and **Dump Objects**.)

The Multiple Object Selector page is entered by pressing the **Multi** soft button when in the dialog pages for the above functions. This page contains parameters to define a group of objects cross-referenced by object type and object ID. Objects can be grouped according to the dependencies of one or more objects, for all of the object functions *except Load Object*. (As described previously, the Load Object function can not determine exact object dependency relationships until the objects are actually being loaded into the K2vx. The other object functions operate on lists of the objects currently stored in the K2vx's RAM, and because of this, dependent relationships are available to be used as selection criteria.)

This is what is displayed by the K2vx when the Multi button is first pressed:

```
Multiple Object Selector
Select :Type/Range
Type   :Sample
Bank   :200's
StartId: 200      EndId: 299

All    Type Toggle Clear Set Cancel
```

This is the screen from which you can specify a range of object ID numbers cross-referenced with object type. If you increment the **Select** parameter to the **Dependents** setting, you will see the following additional parameter layout (not available for Load Object). This is used for selecting dependent objects of either the object currently scrolled to (from the object list that you were just viewing before pressing Multi) or dependent objects of the selected objects (indicated with an asterisk in the object list):

```
Multiple Object Selector
Select :Dependents
Of      :Current Item
Specify:All

Current = Program 205 Viola Section

All    Type Toggle Clear Set Cancel
```

The "**Current =**" parameter is for information only and represents the currently highlighted item in the object list that you are working with.

Two other selection modes do not have any additional parameter associated with them. These are **Everything** and **Search String**.


```
Multiple Object Selector
Select :Everything
```

```
Multiple Object Selector
Select :SearchStrg
```

"Select: Everything" is one way to choose all objects in the list. The All button performs a similar function, as described below.

"Select:SearchStrg" allows you to enter a string to be searched for in all of the object names in the list. Any objects with names containing the string can be selected. (Like "Dependents", this mode is available for all functions except Load Object.)

Multiple Object Selector Soft Buttons

Here is a functional description of the soft buttons displayed on the Multiple Object Selector page. This is followed by an explanation of the parameter operation for all of the above object selection modes.

- All** Returns the Select parameter to **Type/Range**, if it was not already set that way. Sets the Type to **All Types** and the Bank to **All Banks**, and also the StartId to **0** and the EndId to **999**. This soft button is provided as a quick way to define the selection range as "Everything". Once the Type and Bank parameters have been set up in this manner, it is easy to change either parameter to further define the selection range (i.e. All Types of objects in the 400's bank, or Programs in All Banks)
- Type** Returns the Select parameter to **Type/Range**, if it was not already set that way. Sets the Bank to **All Banks**, and also the StartId to **0** and the EndId to **999**. The Type parameter is set so that it **matches the type of the currently indexed object from the object list**. For instance, if you were scrolled to a Setup object when you pressed Multi, pressing the Type soft button would set up the Type parameter to Setup. This is usually used to quickly select or deselect all objects of a particular type by scrolling to the first object of that type, and then pressing **Multi->Type->Set** or **Multi->Type->Clear**. If you don't want to include all banks in the selection range, it is easy to adjust the Bank or ID parameters to narrow the range.

The following three soft buttons will trigger the selection and deselection of items from the current object listing, according to the function of the particular button. Only objects that are included in the specified range will be selected or deselected. When this process is completed, the display returns to the object function dialog.

- Toggle** **For each of the objects in the specified range, "toggle" the selection status of the object.** In other words, if an object is already selected, then the object will be deselected (the asterisk will go away). If an object is not already selected, it will be selected (as indicated by an asterisk in the display between the object ID and object name.)

Toggle is very useful for situations where you want to select all objects in the list except those that meet certain qualifications. For instance, in the Delete Object utility (from the Master page, press **Object->Delete**), you may want to delete all objects that are not being used by a song that you are working on, in order to free up sample or object RAM in the K2vx. You could do this by first selecting the song and its dependents (with no other objects selected, highlight the song, press **Multi**, then select "Dependents" of "Current Item" and press **Set**) and then returning to the

Multiple Object Selector page, set the selection range to Everything (or press the **All** soft button), and press Toggle. This will return you to the object list display, with everything that is **not** in use by the song now selected for deletion. All you would have to do then is press OK and confirm the delete operation.

- Clear** **Clears the selection status of all objects within the specified range.** A basic example of this would be if you wanted to save a file containing everything in RAM except Programs. You could first select all the objects with the Left/Right cursor button double-press, and then go to the Multiple Object Selector. From there you could set the selection range to be Programs in All Banks, and press Clear. The K2vx would return to the object list display, with the only the desired objects selected for saving.
- Set** **For each of the objects in the specified range, select the object.** For Example, to select all Keymaps and Samples in the 300's bank for saving, you would first set the selection criteria to Type/Range, set the Type to Samples, set the Bank to 300's, and press Set. Then, you would press Multi again, and just change the Type parameter to Keymap, and press Set again. The desired range of objects would then be selected and ready to proceed with saving.
- Cancel** Exits from the Multiple Object Selector page, with no action, and returns to the object function, with the previously displayed object list.

Entering Selection Criteria in the Multiple Object Selector

This section describes the operation of the selection modes provided on the Multiple Object Selector page. These are accessed by scrolling the **Select:** parameter to different values, as pictured above.

Selection by Type/Range

This mode sets a specific range of object types cross referenced with a specific ID range.

Parameters	Possible Values	Function
Type	Sample, Keymap, Effect, Program, Setup, QABank, Vel-Map, PrsMap, IntTbl, Song, Table, All Types	Sets the desired object type. The value All Types will select all of the other possible types.
Bank	000's, 100's, 200's, 300's, 400's, 500's, 600's, 700's, 800's, 900's, All Banks	<p>Sets the desired bank. Changing this parameter causes the StartId and the EndId to be set to the limits of the chosen bank (i.e. setting this to 300's will set the StartId to 300 and the EndId to 399). Setting this parameter to the value All Banks will set the StartId to 0 and the EndId to 999.</p> <p>The actual range used for selections when Toggle, Set, or Clear is pressed is taken from the setting of the StartId and EndId parameters. For example, if you set the Bank to 200s and then change the StartID to 300 and the EndID to 399, the 300s bank will be selected, not the 200s.</p> <p>The Bank parameter is used as a quick way to set up the Id range for an entire bank, or all banks.</p>
StartID	0 - 999	Sets the specific starting ID of the selection range.
EndId	0 - 999	Sets the specific ending ID of the selection range.

It is possible to set the EndId before the StartId. If this is the case, no selection range will be enabled.

Selection by Dependents

This mode is used to select a group of objects that are dependents of other objects. This is not available in Load Object

Parameters	Possible Values	Function
Of	Current Item, Selected Items	<p>If set to Current Item, selection range is confined to those objects in the object list that are dependents of the currently indexed item (Current =), including the currently indexed item itself.</p> <p>If set to Selected Items, then the selection range includes any objects in the object list that are dependents of any currently selected objects. The currently indexed item that is displayed is ignored unless it is already explicitly selected (has an asterisk).</p>
Specify	All, All->Keymap, All->Program, Keymap->Sample, Samples Only	<p>This parameter is used to limit which dependent objects are included in the selection range for the appropriate objects included via the Of parameter. The normal setting is All, which means all dependents are included. The other settings are primarily useful when separating objects into different files for reloading later using Macros and Relink-by-Name.</p> <p>If set to All->Keymap, then the selection range will include any dependent objects down to the level of keymaps. That is, samples will be excluded from the selection range.</p> <p>If set to All->Program, then the selection range will include any dependent objects down to the level of programs and effects. Another way of saying this is that any Keymaps and Samples will be excluded from the selection range.</p> <p>Keymap->Sample cause the selection range to only include dependent objects that are keymaps or samples.</p> <p>Samples Only will just include sample dependents.</p>
Current =	Type, ID, and name of the currently indexed object	Displays the object that will be used if Current Items is the value of the Of parameter.

Everything

"Everything" will include all objects in the list. You may prefer to use the All button for this purpose.

Search String

This selection mode will ask for a search string to be entered, as soon as either the Toggle, Clear, or Set button is pressed. The range for the selection/deselection will be any objects whose names contain the search string, ignoring upper/lower case. Search String mode is not available in Load Object.

Working with the Multiple Object Selector

The Multiple Object Selector can be helpful to avoid a lot of button presses and quickly allow you select whatever group of items in the K2vx 's RAM that you want. It is available in all of the related object management functions.

You may notice that the cursor positions and parameter settings are remembered whenever you exit the Multi Selector dialog, even if you exit the dialog and choose a different different function. For instance, if you end up doing a lot of selecting of samples, or of dependents at various levels, the parameters will stay set up the way you left them as you move from function to function (i.e. from Copy to Delete to Save, etc.)

The "Select Dependents" mode is very useful not just for saving dependents, but also for splitting up groups of objects for placing in different files. By using the optional settings for the **Specify** parameter (**All->Keymap**, **All->Program**, **Samples Only** etc.), you can separate the group of objects that you want to save at any level of the object tree that is necessary.

Examples of possible operations using Multiple Object Selector:

- Select all the keymaps that are dependents of a block of programs.
- Select all the samples starting from ID number 398.
- Select all the objects that have "piano" in their object name.
- Select the programs, setups, and effects that are dependents of song 200.
- Select all of the keymaps and samples that are dependents of songs 400-410.

More Features of the Save Dialog

The Choose File Name Function

When entering in a file name for saving, there is a **Choose** soft button. This was briefly mentioned earlier in the section "Creating Directories." When Choose is pressed from the file naming dialog, the K2vx will access the current disk directory and display the following:

```
Dir:\          Sel:0/10      Index: 1

Choose file name: BOTTLE     .KRZ     48K
                  CLAU       .KRZ     1207K
                  REGGAE     (dir)
Total:664K      STICK       .KRZ     550K
                  Root      Parent    OK      Cancel
```

The function of this dialog is to grab the text of any file name on the current disk, and either use it as a starting point in the file naming dialog, or else use the chosen filename exactly. This helps when replacing files on the disk (where the name must exactly match the file being replaced), or adding files to the disk that have similar names or appended revision numbers. You can save time by not having to type the entire file name in from the K2vx's keypad.

The **Open** soft button is only visible in "Choose file name:" dialog when a subdirectory is highlighted.

Traversing directories from the Choose function does not change the current default directory.

Selecting the Directory to use for Saving a File

After you specify the file name when saving any file, you can then choose either the default current directory for saving the file in, or any other directory on the current disk:

```
Use current directory for BOTTLE.KRZ?  
(Path = \)
```

Change **OK** **Cancel**

Pressing OK will select the default path (the current directory), which in this display is the root directory, "\". Pressing Change will allow you to view the disk, traversing its directories, until you find the one in which you want to save the file. If you choose a different directory from the default, it will become the new default directory. For more information on selecting a directory, see "The Directory Selection Dialog", near the beginning of this chapter.

Saving Any File sets the File Index

After saving a file, you can go to any disk function (such as Load), and the just saved file will be automatically highlighted. This makes it easy to find a file that you have just saved, in case you want to delete it, add it to a Macro, move it to a different directory, open it (if it is a.KRZ or a .MAC file), etc.

Macros

The K2vx lets you create lists of disk files called Macros. The files can be located on any drive, whether SCSI based or the internal Floppy drive. Files from SCSI drives in Roland and Akai format can also appear in Macros. Ensoniq files are not currently supported by K2vx macros.

Macros are stored in a data object called a Macro Table, and these can exist in two forms:

1. A Macro Table object in the K2vx's non-volatile RAM.
2. A disk file, containing one Macro Table object. This disk file is called a Macro file, and it has a **.MAC** extension (visible in the directory listing.)

Macros are primarily used to configure a K2vx with sound and sequence data from several files, or selected objects within files. When a Macro file is loaded, every file that is listed in its Macro Table will be loaded in, according to the order of the entries in the Macro Table.

The Macro Page

There can only be one Macro Table in the K2vx's memory at any time. This object is created for the first time by turning on **Macro Record** mode, from the Macro page, which you reach by pressing the **Macro** soft button on the Disk mode page, as shown below.

```

DiskMode    Samples:12313K  Memory:102K
Path = \CYBER\

CurrentDisk:SCSI 2          Startup:Off
                          Library:Off
Direct Access, 84MB
Psyquest PS-427          XMC1.7
<more> Load Save Macro Delete more>

```

The following screen is what you will see if Macro recording is Off:

```

Func:MACRO    [ Off ]    Index: 0

Select Modify Load Record On Exit

```

The top line displays the disk function ,

```
Func:MACRO
```

the current Macro mode,

```
[ Off ]
```

and an index value into the Macro Table.

```
Index: 0
```

Macro Modes

The K2vx has 3 Macro modes -- Record, Pause, and Off. The meaning of each of these modes is:

Off There is no Macro object in the K2vx memory.

Record A Macro object exists, and the K2vx records any file load operations that occur.

Pause A Macro object exists, but the K2vx *does not* record any file load operations that occur.

Note that whenever the Macro mode is Off, there are two soft buttons labeled **Record** and **On**. Pressing On will enable Macro Record mode, and then will return to the Disk Mode page. As an alternative, pressing Record will also enable Macro Record mode, but the display will remain on the Macro page. Once you press Record, the soft buttons and the top line of the display will change. The screen will look like this:

```
Func:MACRO [Record]      Index:  0
```

```
Select Modify Load Pause Off Exit
```

The new Macro mode is displayed ([Record]), and the soft button that used to say Record now says **Pause**. The soft button that used to say On now says **Off**. Pressing Pause will cause the Macro mode to read [Pause] and the Record soft button will reappear. You can switch between Record and Pause by pressing this button repeatedly.

Whenever Macro Record mode is enabled, you will see the indicator "(Macro on)" near the top left of the display on the Disk Mode page:

```
DiskMode  Samples:10022K  Memory: 42K
Path = \
(Macro on)
CurrentDisk:Floppy      Startup:Off
                        Library:Off
```

```
<more Load Save Macro Delete more>
```

The Macro Table

When Macro Record mode is enabled after being in the Macro Off state, a new object called a Macro Table gets created in the K2vx's memory. (The Macro Table is sometimes also referred to as a *Macro list*.) In the Save Object function display list, the Macro Table would appear as:

```
Table      35 Macro      14
```

A Macro Table can only be deleted from memory by pressing the **Off** soft button, or by performing a hard-reset of the K2vx. Pressing the Off button will display the following question:

```
Reset macro?
```

```
Yes No
```

Pressing Yes will delete the Macro Table from memory, and then will return to the Disk Mode page. The Macro mode is set to Off, and the **(Macro on)** indicator is no longer displayed on the Disk Mode page.

Pressing No will return to the Macro page with no action taken. The "Reset macro?" question is displayed to allow you to change your mind about deleting the Macro Table, in case you have accidentally pressed the Off button.

When the Macro Table is first created it takes up a minimal size (14 bytes) in your non-volatile RAM. With each new entry that is added, the Macro Table will increase in size by approximately 40 to 100 bytes (or possibly more if the entry specifies an individual object list). When the Pause mode is entered, the Macro Table continues to remain in memory but does not change size since file operations are not being recorded. This is useful if you need to load some files into the K2vx but you don't want them to be entered into the Macro Table.

A Macro Table gets progressively larger as more file operations are recorded into it. Because of this, you will want to control when a Macro Table is stored in your K2vx's memory. If you are concerned about keeping enough free space for all of your other objects, you may want to practice the following strategy - when you want to record a Macro, enable Macro Record mode, and when you are finished (and possibly have saved some Macro files), turn Macro mode Off. This will reclaim the storage space taken up by the Macro Table.

If you are confident that you have enough K2vx object memory installed and that you can allow the Macro Table to increase in size as you work, then you may wish to leave Macro Record mode enabled all the time. This can be useful if you want to keep a running history of files you have recently loaded in. Both the Macro mode and the Macro Table are remembered between power-cycles of the K2vx via the battery-backed memory.

A Macro can hold as many entries as there is space for in your K2vx's non-volatile RAM.

How to Make a Macro File

This section will take you through creating, saving, and loading in a Macro file. A simple example will be used. Afterwards, you will be able to apply the example and create your own Macro files.

The first step in making a Macro file is to turn on Macro Record mode (from the Macro Page, press On), if you have not already done so.

Creating the Macro

Suppose you have the following four files on your disk (on SCSI ID 5, in the directory \ANALOG\) that contain analog style synthesizer patches, and you would like to have one Macro file that will load them all in:

```
Dir:\ANALOG\    Sel:4/4    Index:    1

File to load:
MULTIUDX .KRZ*    98K
NOISE .KRZ*    36K
RESONANT .KRZ*    109K
SYNAPSE .KRZ*    421K
Total:664K
Select Root Parent Open OK Cancel
```

Using multiple selection, you can select all four files, as shown. When you press OK you will see the "Load selected as:" dialog, but with the extra soft buttons **Macro** and **Insert**:

```

Load selected as: 200...299
                  300...399
                  400...499
                  500...599
Insert           Macro  OK  Cancel

```

The extra soft buttons are available only when a macro is being recorded. First, select the bank that you want, as usual. Pressing **OK** means to load all of the selected files into the K2vx, and add all of the files to the Macro list. If all you are doing is creating a Macro file, you may not want to load the files at this time. However, you still would like to enter the files into the Macro list. Pressing **Macro** does exactly that. It means *don't load the files, but do enter them into the Macro list*.

When you add entries to a Macro, they will be added to the end of the Macro list by default. **Insert** is for inserting file entries at any point in a Macro list. It is discussed a bit later, in the section on Macro editing.

Once you have pressed either Macro or OK, you will see the load modes soft button menu that contains all of the various load modes (OvFill, Overwrt, Merge, Append, and Fill), whether the bank is empty or not. The reason for this is that you might need to be able to specify a mode such as Overwrite even if at the current time, the selected bank is empty. Whatever you choose here for the load mode (as well as your choice of bank) will be saved in the Macro list. For the sake of this example, let's choose the 200's bank and Fill mode. As soon as you choose the mode, the K2vx will either load in the selected files or not, depending upon whether you pressed OK or Macro. In either case, they will be added to the Macro.

Saving the Macro File

You have now created a Macro. You can immediately verify that the selected files have been added to the Macro Table by looking at the Macro page (from the Disk Mode page, press Macro).

But right now, let's proceed with this example by saving the Macro file. From the Disk Mode page, press **Save->Macro**. You'll see the following screen:

```

Func:SAVE MACRO  Sel:0/4  Index: 1

5:\ANALOG\MULTIVOX.KRZ  200:F:
5:\ANALOG\NOISE.KRZ    200:F:
5:\ANALOG\RESONANT.KRZ 200:F:
5:\ANALOG\SYNAPSE.KRZ  200:F:
Select                All  OK  Exit

```

This is called the **Save Macro page**. The soft buttons on this page control which Macro list entries (*Macro entries*) will appear in the Macro Table that will be stored in the Macro file. You can do multiple selection of entries with the **Select** soft button. Selected entries will have an asterisk on the first character of the display line, such as this:

```

*5:\ANALOG\RESONANT.KRZ  200:F:

```

You can use the following double-presses to select and deselect all entries in the list:

Left/Right cursor double-press -> Select All Macro Entries

Up/Down cursor double-press -> Clear All Selections

The top line indicates how many total Macro entries are in the current Macro Table, and how many are selected.

Pressing **OK** will cause only the selected Macro entries to be saved in the file. If you press **OK** but have not selected any Macro entries, then the Macro file will only have one entry in it, that of the highlighted entry.

There isn't much use for a Macro file with only one entry in it, however it can be useful as a way to create a link to an often used file on a drive. For instance, you could create a Macro file called \PERC.MAC in the root directory on the disk that just loads in a file \MYSOUNDS\PERC\GOOD\PERC.KRZ. Then, you would have fast access to loading this file without having to go find it, yet the file would still be organized in a subdirectory.

If you know that you want to save all of the entries into the Macro file (as we do for this example,) just press **All**. The K2vx will go through the standard file saving dialog in which you choose a file name and select a directory to save the file in.

Let's save the file as \ANALOG\SYNTH.MAC. Macro files are automatically saved with the .MAC extension. You will see the following on the screen as the file is being saved:

Writing file SYNTH.MAC...

Table 35 Macro - 162 b

Loading the Macro File

So far, so good. We have created a Macro in memory and saved it to the disk, in the same directory as the files that are listed in the Macro.

While this example only loads files from within a single directory on a single drive, this is for the simplicity of the example. You can have files from any drive or directory in a Macro.

Now, let's go to the Load page and try to load the Macro file. This should load all of the files of the Macro one by one. When you return to the Load page, the file list will automatically highlight the Macro file that was just saved (as it would after any type of file that you save):

```
Dir:\ANALOG\      Sel:0/5      Index:      5
                RESONANT .KRZ      109K
                SYNAPSE .KRZ      421K
File to load:SYNTH .MAC      .5K
```

```
Total:664K
Select Root Parent Open OK Cancel
```

Press **OK** to load in SYNTH.MAC. Now the display reads:

```

Load this macro as: Specified
                   300...399
                   400...499
                   500...599
Insert             Macro  OK  Cancel

```

There are a couple of things to notice here. The first is a new choice in the bank list, **specified**. "Load this macro as specified" means load all the files in the Macro following the exact instructions for the bank and load mode for each file. All of the files were specified to be loaded into the 200's bank using Fill mode. If this is acceptable at the time you want to load the macro, you can just press OK. Otherwise, you can override the bank and mode settings for the entire Macro by choosing a different bank and mode before pressing OK (this is called *rebanking* the Macro.)

The other thing noticeable about the above display is that the Macro and Insert buttons are still available. This is because the Macro mode is still "Record". It is also perfectly acceptable to load a Macro file "into a Macro". If you do this, then **the K2vx will add all of the Macro file's entries into the in-memory Macro list**. Later on, you will see how loading a Macro file into the in-memory Macro list is a convenient way to edit a Macro file or combine elements of several Macro files into one Macro.

In this example, the in-memory Macro is still there - it didn't go away after saving the Macro file (this only happens when you turn Macro Off, as described above.) Since Macro Record is still enabled, loading in the SYNTH.MAC file will have the effect of adding a duplicate set of entries into the in-memory Macro.

Whatever method of loading you choose (i.e., specified in the Macro or overriding the Macro), the K2vx will proceed to locate each file in the Macro in the exact order in which the entries were listed when the Macro was saved. If the files were on different drives in your disk system, you would be able to observe your various drives as they were selected in turn and files were loaded from them.

If the K2vx cannot locate one of the files, you'll see a "Not Found" error message. If a drive cannot be accessed (for instance, if the SCSI ID stored with the Macro entries in this example was no longer the current SCSI ID of the drive), you will see the message "Problem mounting disk.", to which you must press **OK**. If "Confirm" is set to "On" on the Master Mode Page, loading will stop on the first error message, giving you a chance to cancel the operation or keep going. If you answer "yes" at this point, the operation will continue, even if the K2vx encounters subsequent errors. If you run into a lot of errors due to loading an out of date Macro file, the Macro process can be discontinued using a special procedure described later in the section "Aborting a Macro Load."

When the Macro is done loading, you will see the screen:

```
Macro SYNTH.MAC completed...
```

The K2vx will then return to the Disk Mode page. You should now be able to go to Program Mode, or the Save Objects display, and verify that all of the objects from the files of the Macro are now in the K2vx's memory.

Macro Entries

Each file load operation that is recorded into the Macro Table is called a Macro entry. While Macro Record is enabled, you have the option of either loading files into the K2vx or just adding files to the Macro without actually loading them.

Each Macro entry stores information about how a disk file should be loaded. Each entry is displayed as a single item in a scrollable list on the Macro page, with various fields indicating the parameters of the entry. This picture shows how the Macro page might look once 4 files have been entered into the table:

```

Func:MACRO [Record]      Index: 1

3:\DRUMS\REALKITS.KRZ    200:F:Obj
3:\BASSES\WALKING.KRZ    200:F:
3:\KEYS\CHROMA12.KRZ     200:F:
F:\SONG42.KRZ            200:F:
Select  Modify  Load  Pause  Off  Exit

```

The following chart describes the information stored in a Macro entry, and also tells how to the understand the settings of a displayed Macro entry:

Disk drive ID	<p>This represents what drive the file is to be loaded from. There are 10 possible values:</p> <p>The numbers 0-7 represent SCSI 0 through SCSI 7.</p> <p>The letter 'F' represents the Floppy drive.</p> <p>The letter 'U' means Unspecified drive. (see below)</p> <p>The letter 'L' means the Library drive. (see below)</p>
Directory path/file name	<p>This is the directory path and the file name of the file on the disk to be loaded by this Macro entry. This is displayed to the right of the Disk drive ID, following a colon, using as many characters as needed that will fit in the display (up to 28). The internal Macro Table can store any path length, however.</p>
Bank	<p>The bank that the file should be loaded into. This will have a value from 0 through 900 (by 100s), or the letter 'E' for Everything. This field is the first field to the right of the path/file name in the display.</p>

Mode	<p>The mode specified for loading in the file. The following one-letter codes are shown in the display:</p> <p>'O' means Overwrite mode (Overwrt)</p> <p>'V' means Overwrite/Fill mode (OvFill)</p> <p>'M' means Merge mode</p> <p>'A' means Append mode</p> <p>'F' means Fill mode</p> <p>This field is to the right of the bank field, after a colon.</p>
Object indicator	<p>If a list of objects will be loaded from the file, the Macro entry display line will have an "Obj" indicator as the last field on the right. (see above, for the file \DRUMS\REALKITS.KRZ)</p>
Object list	<p>If the Obj indicator is present for a Macro entry, it means a list of individual object types and IDs are stored in the Macro entry for the particular file. When the Macro is loaded in, only these individual objects will be loaded from this file.</p>
Disk Format type	<p>If the disk is a third-party SCSI format disk such as Akai or Roland, a format indicator will appear on the Macro entry display line in the last field on the right. The following codes are used:</p> <p>'Aka' means Akai format</p> <p>'Rol' means Roland format</p> <p>Ensoniq format disks are not currently supported in Macros.</p>

Using the Bank and Mode Fields

The bank and mode fields will only be used if a Macro file is loaded "as specified". (This is one of the options when loading in a .MAC file) This means that each file listed in the Macro will be loaded in exactly as the fields of the Macro entry are specified. These fields can optionally be overridden in the Load dialog, and a new bank and mode can be chosen for the entire Macro.

It is not always important what the bank and mode are set to in a Macro entry, since the Macro can really be loaded anywhere by overriding the bank and mode in the Load dialog. These fields are more important when you want to automatically configure the banks of your K2vx the same way each time you load a Macro, with different files being explicitly loaded into different banks.

Viewing the Object List for a Macro Entry

If a Macro entry contains an object list, it can be examined by scrolling the Macro list display until the item with the **Obj** indicator is highlighted, and then by pressing either the Left or Right cursor button on the front panel. You will see a display that looks like this:

```
Macro Object List (load dependents)
Program 210
Program 211
Program 212
Program 213
```

OK

Left or Right cursor button -> Displays Macro entry object list

The "Macro Object List" display, a scrollable list, shows what objects are to be loaded from the file specified in the currently indexed Macro entry. You will not see the names of the objects in this display, because they are not stored in the Macro Table. The objects are referenced only by object type and object ID. The "(load dependents)" indicator means that the Macro process should also load all dependents of the objects in this list.

If you need to know the names of objects in a Macro entry object list, it is possible to begin a disk function (such as Load), find the file specified in the Macro entry, press Open to display the file's objects, and look up what the objects are, using the information in this display.

Unspecified Disk Drive ID

Before the time a Macro file is saved to disk, you can specify that a file's location will not be tied to a specific SCSI ID number. Instead, the file should be loaded from the same drive that contains the Macro file itself. This is called the "Unspecified" drive ID, since the actual ID is not known until the time of loading the Macro.

The Unspecified drive ID will typically be used by people who distribute K2vx sound files and Macro files on removable disk media such as **CD-ROMs** or **Magneto-Optical Cartridges**. This feature makes it possible to avoid specifying the exact SCSI ID used in a Macro file that is to be loaded on someone else's disk drive, since there is no way to predict the SCSI ID of the drive in advance.

Another use would be, if you are consistently working with Macro files that load mostly files from the same drive, then if you set the Macro's drive IDs to Unspecified you won't have to edit your Macro files if you happen to change the drive's SCSI ID.

Setting Macro entries to have an Unspecified drive ID is done from the Modify Macro page (press **DISK->Macro->Modify**), described later in this section.

The Library Disk

If a Macro entry is set to the "Library" drive ID, it means that the file to be loaded should be found on the drive at the SCSI ID set by the **Library parameter** on the Disk Mode page. This designation is similar in purpose to the Unspecified drive ID, because it is a way to avoid needing to hard-wire the SCSI ID in advance. The Library drive ID is intended to be used in Macro files that reside on removable media such as a Floppy disk, whose Macro entries are supposed to load in particular files on a specific sample CD.

The main purpose of this feature is so that Macro files can be distributed on a floppy disks containing programs that link up with sample files from CD-ROMs. Formerly, a user would have to copy the CD-ROM file to their own hard drive if they wanted to make any new programs that use a CD-ROM file's sample and keymap data.

There are a lot of possibilities in the K2vx to come up with new programs for using with many available sample CDs. Now it is possible to distribute programs or sequences based on a floppy to other K2vx users who already have a certain sample CD in their collection. They

should be able to load the correct sequences of files just by loading a Macro, without those nasty SCSI ID conflicts.

A macro of this type would first load the CD-ROM files containing samples and keymaps (off the "Library" disk), and then load in files from the Floppy disk. The program files from the floppy should correctly link up to the pre-loaded keymaps and samples via the Relink-by-Name mechanism.

In order for the files in the Macro with a Library drive ID to be found, the user of the Macro must have already set up their Library parameter to be the SCSI ID of their CD-ROM drive.

In the following picture, notice that the Library parameter has been set to SCSI 3.

```

DiskMode  Samples:10022K  Memory: 42K
Path = \
(Macro on)
CurrentDisk:Floppy          Startup:Off
                             Library:SCSI 3

<more>  Load  Save  Macro  Delete more>

```

Assuming SCSI 3 is the user's CD-ROM drive, the user is now prepared to load a Macro file from on a floppy disk that contains the following entries:

```

L:\PIANO4MB.KRZ          200:F:Obj
F:\PNOPROGS.KRZ          200:F:

```

When the above Macro entries are loaded, the file \PIANO4MB.KRZ will be looked for on the drive with address SCSI 3, since this is what the Library parameter on the Disk Mode page has been set to. If the correct CD-ROM has been placed in the drive, the K2vx will load the file, and then continue and load \PNOPROGS.KRZ from the floppy disk.

If the K2vx executes the above Macro, but the Library parameter has not been set (is set to Off) an error message

Library has not been set

will be displayed when the Macro process attempts to load \PIANO4MB.KRZ.

The above example is a typical case where the Relink-by-Name feature is used to have one file containing programs depend on another file containing samples and keymaps, yet the files are on different drives. The CD-ROM in the example might just as well have been an Akai or Roland format, and the file from the floppy would contain the latest custom K2vx programs making good use of the imported data.

The Library parameter is remembered by the K2vx across power-cycles, via the battery backed memory. This means that if you use this feature to load Macros you only have to set it up once.

Setting Macro entries to have a Library drive ID is done from the Modify Macro page (press **DISK->Macro->Modify**), described later in this section.

Although the Unspecified and Library drive IDs are meant to be used with distributable media such as CD-ROMs and floppy disks, these features will work with any supported disk drives.

Loading Selected Entries from a Macro File

It is possible to examine the contents of a Macro file from any disk function page - the same way you would open a .KRZ file to check out what objects are stored in it - by highlighting the .MAC file and pressing **Open**:

```
Dir:\ANALOG\      Sel:0/5      Index:    5
                RESONANT .KRZ    109K
                SYNAPSE  .KRZ    421K
File to load:SYNTH  .MAC      .5K
```

```
Total:664K
Select Root Parent Open OK Cancel
```

The K2vx will need to read the Macro file into a temporary area of internal memory, which means there needs to be enough free RAM to accommodate it. When Open is pressed from the Load function, a dialog similar to the Macro page and the Save Macro page will be entered, called the **Load Macro page**:

```
Func:LOAD MACRO  Sel:2/4      Index:    1

5:\ANALOG\MULTIVOX.KRZ      200:F:
*5:\ANALOG\NOISE.KRZ       200:F:
5:\ANALOG\RESONANT.KRZ     200:F:
*5:\ANALOG\SYNAPSE.KRZ     200:F:
Select Check              All OK Exit
```

From the Load Macro page, you can select one or more individual Macro entries for loading, instead of having to load the entire Macro. This is done using the **Select** soft button, identical to the method of saving Macro entries. In fact, this dialog operates identically to the Save Macro dialog, with one exception, the **Check** soft button.

The Check button will cross-check all of the Macro entries in this opened Macro file against the currently in-memory Macro Table, if there is one. Any entries in the opened Macro file that are not in the in-memory Macro Table will be selected when you press the Check button. The selected Macro entries can then be loaded in by pressing OK. This can be helpful to avoid loading in duplicate files if you...

- a) use Macro Record to keep a running history of files that you have already loaded into the K2vx, and
- b) have a lot of Macro files that load in a similar lists of files.

If the Open button is pressed from a disk function other than Load, you will see the View Macro page:

```

Func:VIEW MACRO   Sel:0/4   Index:   1

5:\ANALOG\MULTIVOX.KRZ   200:F:
5:\ANALOG\NOISE.KRZ     200:F:
5:\ANALOG\RESONANT.KRZ  200:F:
5:\ANALOG\SYNAPSE.KRZ   200:F:
                                Exit

```

The only function of this dialog is to view the Macro list entries stored in a Macro file. This feature is useful when, for instance, you are about to delete a Macro file and want to know what information is contained in the file before you remove it from the drive.

Editing Macros

The in-memory Macro Table can be edited from the Macro page. You can select one or more Macro entries and do any of the following operations to them:

1. Change the Drive ID.
2. Change the Bank and Mode settings.
3. Delete the selected Macro entries.

To edit a Macro file already saved on your disk, it is necessary to first load the Macro file into the in-memory Macro. If you recall, this is done by the following procedure:

Making sure that Macro Record mode is enabled, go to the Load function, highlight the Macro file you wish to edit, and then either select certain entries from the Macro file (by pressing Open to get to the Load Macro page), or just press OK to load the entire Macro file. When you see the following display, press the Macro soft button, so that the K2vx will not try to load the files listed in the Macro.

```

Load this macro as:Specified
                    300...399
                    400...499
                    500...599
Insert             Macro  OK  Cancel

```

If you want, you can rebank the Macro on the way in by scrolling the bank list to something other than "specified". Similarly, if you override the load mode, it will be reflected in the in-memory Macro Table as well.

To edit entries from the Macro Table, return to the Macro page. For this example we will edit all the entries at once (like other similar dialogs, if you only are concerned with one list entry, it does not need to be explicitly selected with the Select soft button.) You can use the following double-presses to select and deselect all entries in the list:

Left/Right cursor double-press -> Select All Macro Entries
Up/Down cursor double-press -> Clear All Selections

With all the entries selected, our display looks like this:

```

Func:MACRO [Record]      Index: 1

*4:\NEWMIX\TRASHX12.KRZ  200:F:
*4:\POTS\TEHPOT.KRZ      300:F:
*4:\PANS\FRYING.KRZ       400:F:
*4:\KITCHEN\SINK.KRZ      500:F:Obj
Select Modify Load Pause Off Exit

```

Two soft buttons have yet to be explained. They are **Modify** and **Load**.

Press the **Modify** soft button to change any items mentioned at the top of this section. You will see the following display:

```

Modify Macro Entries

Modify:Drive
Drive :SCSI 4

4 entries selected.

Delete OK Cancel

```

This shows the screen that allows you to change the SCSI drive for the selected Macro entries. If you increment the **Modify:** parameter, the display switches to let you modify Bank and Mode information:

```

Modify Macro Entries

Modify:Bank/Mode
Bank :200's      Mode:Fill

4 entries selected.

Delete OK Cancel

```

The initial settings of the parameters on these screens are always taken from the lowest indexed Macro entry that is selected on the Macro page. In addition, every time you return to the "Modify Macro Entries" page, both the **Modify** parameter and the cursored parameter field will be the same.

Pressing **OK** will set all of the selected Macro entries to have drive ID or Bank and Mode settings according to the parameters set up on this page. The display will return to the Macro page with the same entries still selected. Any modifications to the parameters will be visibly apparent. Selecting multiple entries for editing allows you to change those entries in a uniform way. In our example above, you could change the Macro so that all the files were loaded into a single bank, instead of the separate banks they had previously been loaded to.

Press **Delete** to remove the Macro entries from the Macro list. You will see the display:

Delete macro entries?

Yes**No**

If you answer **Yes**, the display returns to the Macro page and all of the previously selected entries will be gone from the list. If you answer **No**, the display will return to the Modify Macro Entries dialog.

Pressing **Cancel** in the Modify Macro Entries dialog will return to the Macro page with everything that was selected still selected, but with no parameter changes made to any Macro entries.

Here are the parameter values for Modify Macro Entries:

Parameter	Values
Modify	Drive, Bank/Mode
Drive	Floppy, SCSI 0, SCSI 1, SCSI 2, SCSI 3, SCSI 4, SCSI 5, SCSI 6, SCSI 7, Unspecified, Library
Bank	000's, 100's, 200's, 300's, 400's, 500's, 600's, 700's, 800's, 900's, Everything
Mode	Append, Merge, Fill, Overwrite, OvFill

Once you have made the necessary changes to the in-memory Macro, you can go to the Save Macro page to write selected (or all) entries to a new disk file (or replace over the original Macro file you may have loaded in for editing.)

Executing the In-Memory Macro

You can load any group of files listed in the in-memory Macro. This is done using the **Load** soft button on the Macro page:

```

Func: MHCRO [Record] Index: 1

1:\STRINGS.KRZ 200:F:
*1:\PIANO.KRZ 300:F:
1:\DRUMS.KRZ 400:F:
*1:\NOISE.AIF 500:F:
Select Modify Load Pause Off Exit

```

Pressing the **Load** button gives you the choice of loading either **all of the files** in the Macro list or loading **only the files that are selected**:

Load selected items or all items?

All Selected Cancel

If you don't have any items explicitly selected (with an asterisk), the message you see when you press load is slightly different:

Load current item or all items?

This word "current" means the highlighted entry in the display will be loaded.

The files that have been selected for loading will be loaded in their respective order in the Macro list, using the bank and mode parameters that are specified in the list. In the above screen, if you were to load selected items, first \PIANO.KRZ would be loaded in the 300's bank, and then \NOISE.AIF would be loaded in the 500's bank.

Macro Insert

It is possible to insert new Macro entries into the middle of the Macro Table if necessary. This is done by pressing the **Insert** soft button at the "Load this file as:" prompt, when loading in a file (if Macro Record is enabled):

```
Load this file as: 200...299
                  300...399
                  400...499*
                  500...599*
Insert          Macro OK Cancel
```

When you press Insert, you will see a dialog displaying the current in-memory Macro list:

```
Set Macro Insert Point      Index: 2
5:\ARCEILE.KRZ              200:F:
5:\METALIC.KRZ              200:F:
5:\STEREOZX.KRZ             300:F:Obj
5:\STRINGS\DBLBASS.KRZ      400:F:
5:\STRINGS\CELLOS.KRZ       400:F:
                           OK Cancel
```

Scroll the Macro list until the entry before which you want to insert is highlighted. In the above display, any new macro entries added by this load operation will be inserted in the Macro list just before the entry for METALIC.KRZ.

Press **OK** to enable the insert point. Press **Cancel** to disable the insert point.

The display will return to the "Load this file as:" dialog. If a Macro insert point has been set, an indicator will appear at the top left of the screen:

(Macro insert)

```

Load this file as: 200...299
                   300...399
                   400...499*
                   500...599*
Insert             Macro  OK  Cancel

```

The insert point can be disabled before loading, by pressing Insert again and pressing Cancel from within the "Set Macro Insert Point" dialog. The "(Macro insert)" indicator will disappear. Similarly, the insert point can be changed before loading by pressing Insert again (the screen will highlight the current insert point,) scrolling to a different insert point, and pressing OK.

Saving and Loading a Macro Table in a .KRZ file

Macro Table objects can be explicitly saved or loaded (without being "executed") using Save Individual Object and Load Individual Object. If for some reason you wanted to save a Macro Table that was being worked on "in-progress", and then be able to load it back in later on to be worked on some more, you would use this method. The Macro Table can be selected for saving and loading just like any other object. When you load in a Macro Table using Load Individual Object, it will overwrite any Macro Table already in memory. Once it is loaded in, you may have to go to the Macro page and enable Macro Record mode to continue to record further load operations into the Macro.

The "Save Everything" feature of the Save dialog will not include the Macro Table. This is done to prevent inadvertent distribution of what would most likely be a meaningless Macro Table to other users.

Aborting a Macro Load

There is a way to abort the process of loading a Macro file. Aborting can only be done "in between" files that are being loaded, and not during the load of any one file (short of powering off or soft-resetting the machine with [simultaneously pressing +/- 0 CLR]...but this is not recommended!).

Aborting a Macro load is done by pressing and holding down either of the Plus (+) or the Minus (-) buttons that are located just below the Alpha wheel. This should be done at least one-half second before you anticipate the current file to finish loading, or else the K2vx will not sense that you wish to abort the load.

You will see the following question after the current file being loaded is completed:

Abort the macro?

Yes No

It may be a good idea to practice using this method of aborting a macro file load, so that when the time comes that you load in an out of date Macro with 25 entries all at the wrong SCSI ID. you will remember how to abort the process. This same method (of holding the Plus or Minus buttons down) is also used to abort the Backup feature and the multiple file load feature.

If you run out of object or sample RAM, you will have one opportunity to abort the Macro as explained above. However, if you continue from that point you may end up seeing the same error message "Memory is full" for each file that you had selected. This can be a rather tedious process, however it is still possible to abort out of this by holding down the Plus or Minus button simultaneously while pressing "Yes" when you see the following question:

Abort this Partial load?

Yes No

If the Macro that you abort was loaded by multiple selection together with other files (that may or may not be Macro files), you will have to "abort twice", once to get out of the current Macro file load, and a second time to get out of the multiple file load process.

If you are aborting a Macro because the Disk Drive ID is incorrectly specified (as evidenced by lots of "Problem mounting disk" errors) you will need to hold down either the Plus or the Minus button simultaneously while pressing OK to satisfy the error prompt. The screen may "blink" while holding down the repeating Plus or Minus button, but as soon as you have pressed OK you will see the "Abort the macro?" question.

Disk Utilities

The Disk Utility functions provide certain necessary information about disks and their files and directories. These functions are useful when you want to know how your disks are organized and how much disk space you have available. They also help you to locate files and directories.

To access the Disk Utilities page, press the **Util** button from the Disk Mode page. The Disk Utilities page looks like this:

DiskUtil: SCSI 2

Select utility function:

Find List Free Done

The functions on the Disk Utilities page are used for finding out information about the Current Disk. The Current Disk is always indicated on the top line of this page. If you want to use the utility functions for a different disk, you must first set it to be the Current Disk on the Disk Mode page.

Here is a brief description of each soft button:

- Find** Find files utility. Allows you to search for files that match a certain character string in their file names.
- List** Lists an expanded directory tree from any level of the hierarchy. This function can be used to determine the total size of files within any tree of subdirectory. It is also helpful for finding files on the disk.
- Free** Calculates the available space on the disk.
- Done** Exits back the Disk Mode page.

Find Files

The Find files utility first prompts you to enter a character string to be searched for:

```
Search string: RAT_
```

```
Delete Insert >>End Choose OK Cancel
```

You can use the **Choose** button to grab the text of a file name from the current disk , as described previously.

If you press **OK**, the K2vx will begin to search the disk for any files or directories that contain the search string in their names. The search starts in the root directory and searches the entire disk. When a matching file or directory is found, you'll see one of the following:

```
Found file:
```

```
\BABYTOYS\RATTLE.KRZ
```

```
FindNext Go To Cancel
```

```
Found directory:
```

```
\CRATES
```

```
FindNext Go To Cancel
```

If the search string is found anywhere within a file name it will be matched. The search algorithm independently checks both the file name and the extension. For instance, if you wanted to find any Macro file on the disk, you could enter in "MAC" for the search string. This would find any Macro files as well as any files or directories that have "MAC" in their filename.

When a match is found, there are three choices displayed:

FindNext Continue searching the disk for another file or directory that matches the search string.

Go To Exit to the Disk Mode page, setting the current directory and file index of the K2vx to the location of the found file or directory.

The next disk function you choose will display the current directory with the found file already highlighted. If a directory was found, then the first file in the directory list will be highlighted.

Cancel Exit to the Disk Mode page.

When the search has checked all of the items on the disk, the K2vx displays this screen:

Search completed.

OK

If no matching files were found, you will also see

(No files found)

The K2vx will remember the last search string that you entered. This makes it easy to repeat the same search. If you press **Util->Find** again, the "Search string:" dialog will contain the previously used string.

List

The List utility allows you to view directories on a disk with the expanded contents of all sub-directories. Each line is indented according to its directory level, so that you can get sense of how your directory tree is organized.

```
Dir:\          [ 1968K] Index: 1

ELEPHANT.KRZ 148K
BSOUNDS\
  TRUMPETS\
    JSBACH.KRZ 712K
Root Up Down Go To Exit
```

The **Dir:** parameter shows the directory that is being listed. The size value displayed on the top line of the display represents the total size of the directory subtree that is currently being viewed. The **Root**, **Up**, and **Down** soft buttons control going from one directory to another in the hierarchy:

Root	Display the disk from the top level, meaning that all of the files on the disk will appear in the scrollable list.
Down	Set the display to the level of the highlighted file or directory. For example, scrolling to JSBACH.KRZ in the above list, and pressing Down , would focus the list on the contents of the TRUMPETS directory.
Up	Set the display up one directory level.
Go To	Exit to the Disk Mode page, setting the current directory and file index to the location of the highlighted file or directory.
Cancel	Exit back to the Disk Mode page. The current directory is unchanged.

The files are listed in the order that they appear on the disk, unalphabetized. The traversal order of the directories is the same one that is also used for the Backup function.

Free

The Free utility calculates the amount of free space on the current disk and displays the result on the Disk Utilities page. This may take a small amount of time to calculate, depending on the drive.

```
Computing free space on SCSI 2
```

```
Please wait...
```

```

DiskUtil: SCSI 2      (DOS)
Used:23%   Free:94814K   Total:124396K

Select utility function:

```

```

Find List Free Done

```

The parameters tell you the following:

Used The percentage of the disk that is taken up by the existing information stored on it.

Free The amount of disk space available for new files.

Total The size of the disk. This will be the size of the usable partition if MS-DOS format.

For SCSI drives, if the current disk was formatted on a PC or a Mac in MS-DOS format and contains at least one primary partition, you will see the "(DOS)" indicator on the top line. Using this format is described in the section on "MS-DOS compatibility".

For floppy drives, the density (HD or DD) will be displayed. All K2vx floppy disks are MS-DOS format compatible.

Moving Files Between Directories

Files and directories can be moved from one directory to another on a given disk. You can either choose multiple files to move using the Select soft button, or just move the single highlighted file or directory (if no other files are selected). As you would probably expect, moving a directory also moves all the files within the directory. To use this function, press the **Move** soft button from the Disk Mode page. Then choose the file or files that you want to move:

```
Dir:\          Sel:0/15   Index:    1

File to move:ATOMTOM .KRZ    98K
              BLOWFISH.KRZ   36K
              COLORS  .KRZ   109K
Total:6846K    DRUMHITS      (dir)
Select  Root  Parent  Open  OK  Cancel
```

Press **OK** when you have made your selection. Press **Cancel** to return to the Disk Mode page.

The K2vx remembers the most recent destination directory that a file was moved to. If the current directory is different from the most recent destination directory, you will see the question:

```
Use default directory for ATOMTOM.KRZ?
(Path = \DRUMHITS\)
```

```
Change  OK  Cancel
```

Press **OK** to use the default.

Press **Change** if you want to select a different destination from the default shown. The K2vx will then display a directory selection dialog (described early in this chapter), and you can select the move destination directory from there:

```
Dir:\          Sel:0/15   Index:    4
              BLOWFISH.KRZ   36K
              COLORS  .KRZ   109K
Select dest dir:DRUMHITS      (dir)
              EARTHLNG.KRZ   144K
              HANDCLAP.KRZ   645K
Total:6846K    INDUSTRY      (dir)
Root  Parent  Open  Current  SetDir  Exit
```

The move operation begins when you press either **Current** or **SetDir**.

If the default destination directory is the same as the source directory, the K2vx will skip the "Use default directory?" question and instead go right to the "Select dest dir:" dialog.

A good way to organize files into subdirectories, with a disk drive that was being used with previous versions of K2vx software, is by using the Move command. First, create the subdirectories you need, using the **NewDir** function. Then, use multiple file selection to select the files to be grouped into a particular subdirectory. The files can be moved to their new location in one operation.

For each file that is moved, you will see a confirmation message:

```
\ATOMTOM.KRZ moved to
\DRUMHITS\ATOMTOM.KRZ
```

Note the following:

- You can select multiple files for moving within a directory. However, you cannot move files from more than one directory at a time. If you select any files and then switch to another directory, the files you had chosen will be deselected.
- If you are moving a directory, you cannot move it in to a subdirectory of itself.
- If the source and destination directories are the same, the file will not be moved, and an error message such as the one below is displayed. This would happen if you pressed **Current** above.
- The same message will be displayed if there is a file in the destination directory with the same name as the file to be moved.

```
File \ATOMTOM.KRZ not moved.
```

Renaming Files

Press the **Rename** soft button (from Disk Mode) to rename files or directories without loading them. When you press **Rename**, the K2vx will prompt you to select the file to be changed, by showing you a list of the files found on the current disk.

When you've selected the file to be renamed, press **OK**, and the K2vx will ask you to enter the new file name. When you've done this, press **OK**, and the file name will be changed.

```
Dir:\          Sel:0/1      Index: 1
```

```
File to rename:FILE9876 .KRZ 2048K
```

```
Total:2048K
  Root  Parent  Open  OK  Cancel
```

This function can only be used to change the 8-character file name and not the extension. When you press **Rename**, the File List dialog is displayed and you can navigate through the directories to choose the file or directory you wish to rename. Unlike the other disk functions that use the File List dialog, you will not see the **Select** soft button. This is because you can only rename one file at a time. Therefore you simply choose the file you want and press **OK**. The K2vx will then ask you to enter a new name. Once you've done this, press **OK** again, and the file name will be changed.

Deleting Files and Directories

Press the **Delete** soft button (from Disk Mode) to delete files and directories. The Delete function supports multiple selection of files for deletion. Select the file(s) and/or directory(s) to be deleted, and press **OK** (or **Cancel** to abort). Be careful! You don't get a second chance to change your mind once you've pressed **OK**. Once a file is deleted, it's gone. Remember the fundamental directive of computer users: Save early, save often; make backups.

```

Dir:\          Sel:2/4      Index:  2

          BLUES          (dir)
File to delete:MOTOR      .KRZ*    98K
          QUACKS      .KRZ      344K
          ZAPPER      .KRZ*    802K
Total:1244K
Select  Root  Parent  Open  OK  Cancel

```

A directory must be empty in order to delete it (i.e. it must not contain any files or subdirectories.)

Backup and Copy Functions

File Backup

To access the Backup function from the Disk Mode page, first make sure that the Current Disk is set to be the drive that you want to make a backup of. Next, press the **Backup** soft button:

```

Dir:\          Sel:0/15     Index:  1

Set backup dir:ANIMALS .KRZ  1097K
          BREAKAGE      (dir)
          LOWINST       (dir)
Total:9040K          PLANKTON (dir)
Root  Parent  Open  Current  Exit

```

On this screen, choose the directory tree that you want to backup by selecting a directory (see Selecting Directories, near the beginning of this chapter.) Backup allows you to copy all of the files within a directory tree from one disk to another. All of the files within the directory that you choose, plus all of its subdirectories and files within them will be copied to the new drive. If you want to backup the entire disk, then make sure the current directory is the root directory (as in the picture), and press **Current**.

Next, you will see a dialog for choosing the destination disk:

```

Destination disk:SCSI 0
                  SCSI 1
                  SCSI 2
                  SCSI 3
                  [OK] [Cancel]

```

Select the disk you wish to transfer files to. It must be a different disk than the current disk.

Next, you can select a directory on the destination disk that will receive the transferred files.

```

Use default directory on SCSI 0?
(Path = \)

```

```

[Change] [OK] [Cancel]

```

The default is always the root directory on the destination disk. Press **OK** to select the default. To select a different directory, press **Change**.

Next, select the **Backup mode** when you see the following question:

```

RePlace or increment mode?

```

```

[Help] [RePlace] [Increment] [Cancel]

```

Replace Any files to be transferred that already exist in the destination directory will be replaced (overwritten.)

Increment Any files to be transferred that already exist in the destination directory will be skipped (not transferred.)

Help Displays a reminder about the meaning of Replace and Increment modes.

After you select the Backup mode, you will see a confirmation screen with all of your selections so far:

```

Press OK to start backup:SCSI 4->SCSI 0
Mode =RePlace
Source=\
Dest  =\

```

```

[SetFile] [OK] [Cancel]

```

OK Begin the Backup function according to the parameters on this page.

Cancel Exit to Disk Mode.

SetFile Set Backup starting file. This is mainly used when you are backing up files from a larger SCSI drive to several smaller removable media disks. Once one of the removable destination media is full, note the last file that was successfully copied. Enter the Backup function again, setting up the source and destination path parameters exactly as before. But this time use the **SetFile** button to start from the file after the last file copied.

You will notice that the file list displayed when SetFile is pressed is not in alphabetical order, but in the actual order the files occur in the directory on the disk:

```

Dir:\          Sel:0/15      Index: 1

Set start file:ZYXWVUTS.KRZ 418K
                PLANKTON (dir)
Total:9040K      ANIMALS .KRZ 1097K
                VEGGIES .KRZ 2801K
  Root  Parent  UPen  OK  Cancel

```

Find the last file copied from the previous partial backup. Set the list index to one entry past the location of this file. If this file is already the last entry in the file list, the "next" file to continue the multi-part backup would be the file or directory that comes after the file list entry for the currently viewed directory. To find this file or directory, you will need to look in the Parent directory.

A large SCSI drive backed up in this way can be restored by individually backing up the removable media in the reverse direction.

Backup and Copy will transfer files much faster if you have cleared your object RAM first using the Delete Everything command (save any objects in RAM to disk first.)

When using the Backup function to transfer files from SCSI to Floppy, directories are ignored (i.e. not created on the floppy.) This "flattens" out any directory structure so that multi-disk files can be created on the floppy disks.

File Copy

The Copy function lets you copy a single file from a directory on one disk to a specified directory on the destination disk.

Creating a Startup File

You can create a Macro file that will automatically be loaded in when you power-on your K2vx. This file, called the **Startup file**, or **Boot Macro**, can be on a disk drive at any SCSI ID or else on the Floppy disk. See the section on Macros for background information.

The steps needed to create a Startup file are:

First, create a Macro file called **BOOT.MAC** in the root directory of the SCSI or Floppy disk that you will use as the Startup disk. Specify in the Macro the exact ordering of files that you would like to have loaded into the K2vx when powered on. When you save the Macro file, just name the file "BOOT", and the K2vx will add the ".MAC" extension.

Second, set the **Startup** parameter on the Disk Mode page to be the drive ID of the Startup disk. So, if your BOOT.MAC file was on a floppy disk, set the Startup parameter to **Floppy** and make sure that you have the correct floppy in the drive when you next turn the K2vx on.

When the K2vx is powered on, it will display the following message (after the introductory VAST logo):

About to load startup file...

Cancel

The K2vx looks for a file BOOT.MAC in the **root directory** on the drive specified by the Startup parameter. If the file is not found, or the drive cannot be accessed, you will get an error message. The Startup load can be bypassed in the first few seconds after the K2vx is turned on, by pressing the **Cancel** button that is displayed on the screen.

If BOOT.MAC is found by the K2vx, it will begin to load the Macro file as if you had loaded it explicitly from the Load function in Disk Mode. When the Macro has completed, you will see the following:

Macro BOOT.MAC completed...

The K2vx will go directly to Program Mode afterwards.

Deleting Banks in a Startup File

You may want the Startup file to clear out one or all banks in the K2vx before loading files. This could help overcome the problem of having "silent" copies of programs in your RAM that depend on samples that are no longer there (because they disappeared the last time the K2vx's power was turned off.) The following trick will allow a Macro entry to essentially function as a Delete Bank or Delete Everything command:

Create a file somewhere on (preferably) your Startup disk, by saving an empty bank from the K2vx. Call the file NULL.KRZ. Now, insert this file at the beginning of a boot Macro you are creating - load the file, specify the bank you want to delete in the Startup file (choose 'Everything' to clear out all of the object RAM), and specify **Overwrite** for the load mode. Make sure you press **Macro** and not **OK**, so that the overwrite doesn't take place until you use the Startup file.

Here is what that Macro entry for this file might look like on the Macro page, if you were doing a "Delete Everything"

3: \NULL.KRZ

E:O:

The "E:O" stands for "Load as Everything, using Overwrite mode".

MS-DOS File System Compatibility

The K2vx is compatible with fixed and removable disk drives that use the MS-DOS hard disk and floppy disk formats. If you want to use this feature, **you must first format the disk media on a computer** such as a PC compatible or a Mac running appropriate MS-DOS conversion software.

The MS-DOS hard disk format is structured so that the disk can be split up into multiple partitions. **The K2vx can only use the first partition that it finds on the disk.** Therefore, it is usually best to format the media with only one partition taking up all usable space on the disk. Working from the K2vx front-panel with an MS-DOS formatted drive will appear the same as working with a drive that has been formatted with the K2vx's own Format function (on the Disk Mode page.) The Free utility (Disk Mode->Util->Free) can be used to identify whether a disk is DOS format or standard K2vx format. If the "DOS" indicator is displayed, it means the K2vx has determined that the disk is a DOS format hard drive with at least one primary DOS partition. Floppy disks will not display the DOS message.

Some advantages of working with an MS-DOS compatible drive format over the standard K2vx format are:

- easier sharing of K2vx files with other users over computer communications lines,
- being able to use graphical file management interfaces for organizing files and directories,
- being able to back up K2vx data using a PC compatible or Mac with commercially available software, and
- easier transfer of data using standard file formats such as AIFF, WAVE, and MIDI Type 0, for importing and exporting samples and sequences.

File Name Compatibility

DOS format does not support space characters in file names. The K2vx, though, allows spaces to be used within file names. If you plan to transfer files between the K2vx and a DOS compatible computer, it is recommended that you use only file names without space characters in them. Otherwise, a computer may have trouble identifying the files.

Importing and Exporting Data using Standard File Formats

The K2vx supports three common data interchange file formats, Apple Interchange File Format (AIFF), Microsoft RIFF WAVE, and MIDI Type 0. The first two are used to transfer sample data, and the latter is used for sequences.

The K2vx can recognize these file types automatically on loading, regardless of the file extension. You can load these files as you would any standard K2vx file, and also as part of a Macro file load. The most recent sample file loaded will become the "preview" sample, which means you can quickly access it for playing or editing on the Sample page (**MASTER->Sample**). Similarly, the most recent MIDI Type 0 file loaded in will become the current song on the Song Mode page.

You can save files in these formats on the **Export** page. This page is accessible from the Disk Mode page by pressing **Save->Export**.

```

Save AIFF file          Index: 1

Sample 200 NewSample    C 4S 100K
Sample 201 Babaganoush  C 4S 444K
Sample 202 YaYaYa       C 4S 190K
Sample 203 GregorianChant C 4S 6515K
AIFF    WAVE    MIDI    OK    Cancel

```

The Export page allows you to save one sample or song object per file. Choose the format you wish to save in, and press the corresponding soft button. For AIFF and WAVE, only sample objects are listed. For MIDI Type 0, only song objects are listed. Scroll to the object that you wish to save, and press **OK**. The dialog will proceed the same as if you were saving a K2vx file. You will be prompted for a file name, and will have the option to select a different default directory to save the file in.

The K2vx will automatically place a standard extension on the file when it is created on the disk. These extensions are sometimes necessary when transferring files to external programs that can recognize the file format based on the extension. They also help you to recognize what files are of a what format when looking at a directory listing. You can use the Find Files utility (Disk Mode->Util->Find) to search for files that match a certain extension. The standard extensions used on the files are:

```

.AIF    AIFF
.WAV    WAVE
.MID    MIDI Type 0

```

The first time you enter the **Export** page after powering on (or after a soft-reset of the K2vx), the format defaults to AIFF. After that, the K2vx remembers the most recent format that you used. For example if you save a MIDI Type 0 file, and then go back to the Song Mode page to record more sequences, the next time you return to the Export page, the file format will still be set to "MIDI", and all of the song objects will be listed. You can audition the samples and songs the same way as you would on the Save Object or Object Utility pages (by pressing the Left or the Right cursor buttons on the front-panel.).

AIFF and AIFF-C Files

The K2vx can read 8 or 16 bit AIFF files, mono or stereo. The sample rate, sustain loop, loop mode, base note, sample name, and sample detuning are supported. **AIFF-C** files that do not use compression can also be read by the K2vx. The K2vx will save 16-bit AIFF files, either mono or stereo.

WAVE Files

The K2vx can read 8 or 16 bit WAVE files, mono or stereo. WAVE files do not support looping. The K2vx will save 16-bit WAVE files, either mono or stereo.

MIDI Type 0 Files

Use this format to transfer a song to and from an external sequencer. Tempo information is supported. Time Signature is not supported. System Exclusive data is not supported.