Using the Inferno System 3

Introduction

After you install the Inferno system, this chapter helps you get started using the Inferno system, including:

- Using the Inferno emulator in a host operating system (such as Windows 95)
- A walk-through of the sample applications included with the distribution.

Note: All examples are running the Inferno emulator under Windows 95.

Emu—The Inferno Emulator

When the Inferno operating system is running under a host operating system, it is in *emulation mode*. The Inferno emulator is called *emu*. The emulator supports the Inferno environment by creating a virtual machine for Inferno programs.

Starting and Stopping the Emulator

The *emu* command is installed in the <*inferno_bin*> directory for the host system that you are running. [The default location for the <*inferno_bin*> directory on Windows systems is \users\Inferno\Nt\386\bin. The default location for the <*inferno_bin*> directory on Unix systems is /usr/inferno/Solaris/sparc/bin. If you are not using the Solaris SPARC, you need to substitute the machine name for Solaris and the chip name for sparc.] Table 3-1 lists the various ways to start the emulator, depending on which host operating system you are using.

Note: For the Inferno system to reflect the correct local time: before starting the emulator, choose the appropriate time zone file from *<inferno_root>/locale* and copy it to *<inferno_root>/locale/timezone*.

Table 3-1 Starting the Inferno Emulator

Windows 95 or NT 4.0	Select **Emulator** from the **Start>>Programs>>Lucent Inferno** menu
Windows NT 3.51	Double-click on the **Emulator** icon in the **Lucent Inferno** folder of the Program Manager
Unix systems	Type **emu** from the appropriate <*inferno_bin*> directory

When you start the emulator in the interpreted mode, a window opens with a few lines of text and presents you with the system prompt, *<machinename>$*:

```
Inferno Release 2.0 Build 46 main (pid=-<pid> interp
Initialize Dis: /dis/sh.dis
<machinename>$
```

This **EMU** window is the Inferno control console.

To close the control console, you can use the *shutdown* command or press **Ctrl+C.** The *shutdown* command shuts down or reboots the emulator, depending on the option you specify. When you close the control console, you close all Inferno programs that have been started by this instance of the emulator.

To use the *shutdown* command to close the control console, enter

```
shutdown -h
```

Inferno Control Console

The control console is your interface to the Inferno system. You can type commands to configure your environment and start applications. The console displays standard output from programs, such as the output from *print* statements.

The console commands are Unix-like; for example, there are *cat*, *ls*, *mkdir*, *grep*, and *cd* commands. The commands are discussed in Chapter 6, *Inferno Command Line Utilities*, of this guide. For a complete discussion of the Inferno commands, see Chapter 5 of the *Inferno Reference Manual*.

Inferno File Namespace

The Inferno file system conforms to a set of conventions that promotes the flexibility of namespace manipulation. These conventions should be adhered to for the system to behave normally.

The default root of the Inferno file system is the directory known to the host operating system as */usr/inferno* on Unix systems and *\users\Inferno* on Window systems. This directory is referred to as <*inferno_root*>.

Below the root is the required directory structure that contains the various components of the Inferno system. This directory structure is created for you during installation.

When a program is invoked within the Inferno environment, the Inferno system looks for the executable file in the following order: 1) the explicitly named path, when given, 2) the current directory, and then 3) the <*inferno_root*>*/dis* directory. The Inferno namespace allows you to bind other directories to this */dis* directory, as you will see later.

The README file in Inferno describes the Inferno file system under <*inferno_root*>.

Applications

The applications shows the Inferno system in a variety of situations. An application can run anywhere in the network without knowing where resources are located. The application can choose to execute a process on the client or server, depending on where resources are located or it can use some other criteria.

These applications are are designed to show some of the capabilities of the Inferno system and the Limbo programming language and introduce an application programmer to them. Source code for these applications is found in *<inferno_root>/appl/wm* and you can adapt them for your use.

Window Manager—Desktop Environment

The Window Manager is a demonstration of a desktop environment. It shows the graphical user interface provided with the Inferno system. The Window Manager environment includes several applications that are distributed with the Inferno system, including:

- Charon Browser
- Mail Tool
- Text Editor and Shell
- Development and System Tools
- Media Players
- Graphics viewers

Starting Window Manager

You start the Window Manager from the **EMU** control console.

Note: If you have installed the Inferno system in a location other than the default <*inferno_root*> directory, you must specify the alternate or remote location using the *-r* option with the **emu** command.

Note: Some Window Manager template applications are designed to run at a screen size of 800x600. If your system supports a resolution of 800x600 or greater, you should include the *-g* option on the command line to specify a geometry greater than the 640x480 default. For more information about the command-line options for *emu*, see Chapter 4, *Inferno Host O/S Utilities* of the *Inferno Reference Manual*.

At the **EMU** control console prompt (<*machinename*>*$*), type the following lines. Press **Enter** or **Return** after each line:

```
<machinename>$ bind '#I' /net
<machinename>$ lib/cs
<machinename>$ wm/logon
```

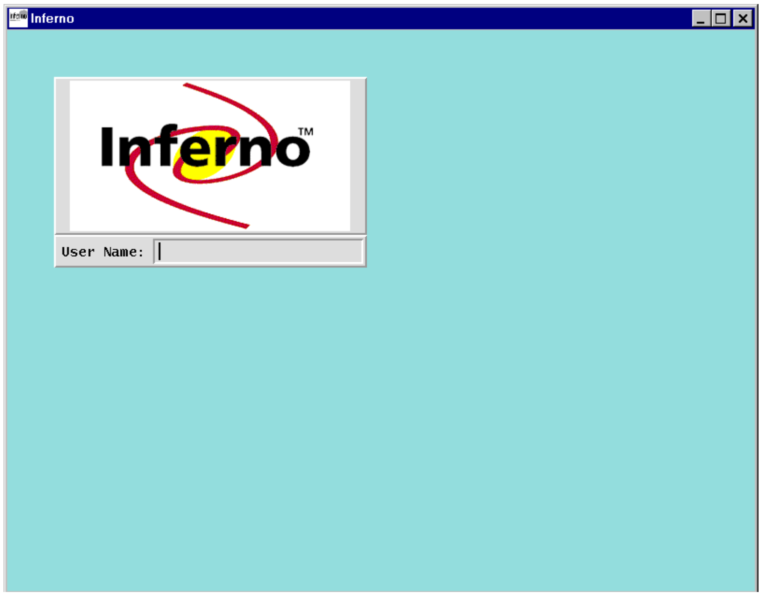
Note: The *bind* and *lib/cs* commands are necessary to use the sample browser and e-mail applications. These commands are discussed in the *Inferno Reference Manual*.

Note: The *lib/cs* and *wm/logon* commands can be run only once in an **EMU** window. To run then again, close the emulator instance and run the *emu* command again.

Note: For the sample applications use the **Enter** or **Return** key to move between fields.

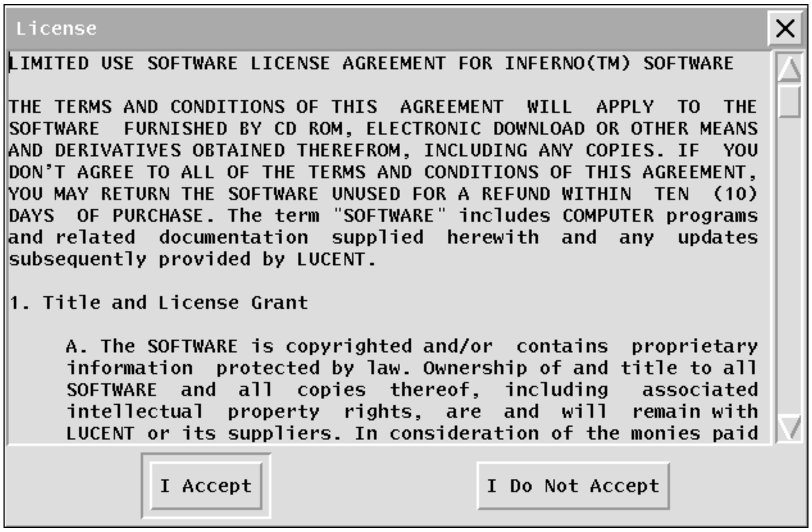
Figure 3-1 shows the Inferno Window Manager logon screen.

Figure 3-1 Inferno Window Manager Logon Screen



The default User Name is *Inferno*; type *Inferno* in the **User Name** field and press **Enter** or **Return** to begin a Window Manager session. You will see the Inferno license agreement as shown in Figure 3-2.

Figure 3-2 Inferno License Agreement Acceptance



LIMITED USE SOFTWARE LICENSE AGREEMENT FOR INFERNO(TM) SOFTWARE

THE TERMS AND CONDITIONS OF THIS AGREEMENT WILL APPLY TO THE SOFTWARE FURNISHED BY CD ROM, ELECTRONIC DOWNLOAD OR OTHER MEANS AND DERIVATIVES OBTAINED THEREFROM, INCLUDING ANY COPIES. IF YOU DON'T AGREE TO ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU MAY RETURN THE SOFTWARE UNUSED FOR A REFUND WITHIN TEN (10) DAYS OF PURCHASE. The term "SOFTWARE" includes COMPUTER programs and related documentation supplied herewith and any updates subsequently provided by LUCENT.

1. Title and License Grant

A. The SOFTWARE is copyrighted and/or contains proprietary information protected by law. Ownership of and title to all SOFTWARE and all copies thereof, including associated intellectual property rights, are and will remain with LUCENT or its suppliers. In consideration of the monies paid

Read the agreement and click on the **I Accept** button to begin using your Window Manager session. Otherwise, click on the **I Do Not Accept** button to decline using the Window Manager. If you decline to accept the agreement to use the Inferno software, remove the software from your system and return the software and related documentation to Lucent Technologies, or contact your system administrator.

Customizing the Window Manager Environment

There are three configuration files in a user's home directory that Window Manager uses when it is started: *namespace*, *plumbing* and *wmsetup*. The use of these files is similar to that of the *.profile* configuration files in Unix or *.ini* files in Windows. They are simple ASCII files that can be edited to provide a customized environment for each user. There is an empty *keyring* directory that is used for the security feature and an empty *config* directory that is used for the browser and address book applications. If the *config* directory is not present in a user's directory, when the browser or the address book is opened for the first time, the directory is created.

The *namespace* file contains the set of *mount* and *bind* commands that provides a specific view of network resources. Edit this file to provide transparent access to any available resources in the network. Initially and minimally, this file contains the following line:

```
bind -ia #C /
```

Note: Quotes are not needed around the *bind* command since the file is not run by the Inferno shell.

The *exec* statement in the wmsetup file can be used to customize your environment, also. For example, you can remove the -*task* option from the *exec* statement for the plumbing feature and the **Plumb** window will be opened each time you start the Window Manager. Or, you can use the *exec* statement to open a **Shell** window when you start Window Manager.

The wmsetup file

The *wmsetup* file contains *exec* statements for features that start automatically with the Inferno Window Manager and specifications for the Window Manager menu and submenus. This menu and submenu list augments the default options that are set in the */appl/wm/wm.b* file. The **Inferno** menu items that appear above the horizontal separator line and have submenus are set in the default *wmsetup* file, while **About** and the items below the separator line are set in the *wm.b* file. If **About** is the only item that appears above the separator line on the **Inferno** menu, the *wmsetup* file is not in the */usr* directory for this user.

Lines beginning with *exec* are commands that start automatically when you enter the *wm/logon* command and the Window Manager starts. For Inferno Release 2.0, there is one *exec* statement for the plumbing feature. The format of the statement is:

```
exec /dis/wm/plumb.dis
```

This *exec* command line must be included in the *wmsetup* file for each new Inferno user for the plumbing feature to work.

The format of the statements for the menu and submenu items in the *wmsetup* file is:

<Main Menu Label> : *<Menu Label>* : *<application path>*

The <*Main Menu Label*> field specifies the text that displays on the main menu. The <*Menu Label*> field specifies the text that displays on the cascaded submenu. The <*application path*> field is the full path from the <*inferno_root*> of the application that is invoked. For example:

```
Applications:Text Editor:/dis/wm/edit.dis
```

This line specifies that the **Text Editor** is on the **Applications** submenu and is implemented by */dis/wm/edit.dis*.

Note: You can customize the **Inferno** menu or submenus by modifying the **wmsetup** file. To add an application, specify the menu name, submenu name, and the location of the implementation file.

The Window Manager's interface is similar to common desktop computers' graphical interfaces. Therefore, getting around Window Manager should be familiar to anyone accustomed to interfaces like Windows 95 or X Windows.

The plumbing file

Plumbing messages have a fixed format. Plumbing rules in the *plumbing* configuration file specify how plumbing messages are to be routed.

When a new Inferno user is added, the plumbing file must be copied to the new user's directory for the plumbing feature to work. Additionally, the Plumb window must be opened before an application can use the plumbing feature.

Window Manager Features

Plumbing

The plumbing feature of the Window Manager enables message passing between applications running under the Inferno Window Manager. The small **Plumb** window that is positioned in the upper right corner of the Inferno Window Manager window displays a log of the plumbing messages. You can minimize this window, but you cannot close it.

Note: If the **Plumb** window has not been opened and the Brutus Text Editor application is selected, a message that will be written to the **EMU** control console. To avoid this message, open the **Plumb** window before you open the **Text Editor** or other plumbed application. Once the **Plumb** window has been opened, the message does not display.

The *exec* command in the *wmsetup* file automatically starts the plumbing feature when the Window Manager is started. The *exec* command line in that file is:

```
exec /dis/wm/plumb.dis
```

Plumbing messages have a fixed format and plumbing rules in the *plumbing* configuration file specify how plumbing messages are to be routed. The plumbing configuration file is a set of rules separated by blank lines. Each rule is a set of patterns followed by a set of actions. The rules are interpreted in sequence. The rules in the file are from the most specific to least specific.The first rule for which all patterns match is applied, and no further rules are examined. Comments begin with # and single quotes protect special characters. See the *plumbmsg—plumbing message module* in Chapter 15, *Limbo Miscellaneous Modules,* in the *Inferno Reference Manual* for more information.

The rule in the configuration file for *.m* files selected when using the Brutus text editor is:

```
kind is text
data matches '([a-zA-Z0-9]+\.m)(:[0-9]+)?'
data is file/module/$1
data is set/module/$0
plumb to edit
plumb start /dis/wm/brutus.dis $file
```

Plumbed programs

Figure 3-2 lists the Inferno programs that are plumbed.

Table 3-2 Plumbed Programs

wm/sh	The Shell sends messages, but doesn't receive messages
wm/brutus	The Text Editor, Brutus, receives file names and sends messages
wm/charon	The Charon browser receives URLs, but doesn't send them.
wm/view	The GIF video file viewer receives file names, but doesn't send them
wm/dir	Local receives directory names and sends file names
wm/deb	The Debugger sends messages

Plumbing example for *wm/brutus*:

To use the plumbing feature when you are using the **Text Editor**, click the right mouse button on a filename. The Window Manager opens a new **Brutus** window that displays the selected file. You can move this new window that covers the window in which you did the selection. You can continue to make selections in this way and each time a new **Brutus** window opens to display the newly selected file. Plumbing works for compiler error messages, also. Each time you select a file, you can see the plumbing message in the **Plumb** window. For example, with */appl/wm/edit.b* open and the filename *sys.m* is selected, the message

```
send “/module/sys.m” to edit
```

is displayed in the **Plumb** window.

If your selection does not match a filename, plumbing displays the message

```
don't know who message goes to
```

in the **Plumb** window.

Plumbing example for *wm/sh*:

When you are using the **Shell,** you can right click on a *.m* file in the same way that you do when already in **Brutus**. A new **Brutus** window opens and the file is displayed. You can right click on the name of a *.gif* file and, if the format is one recognized by the **GIF Viewer**, the image is displayed.

Plumbing example for *wm/charon*:

When you are using the **Shell** and you enter a web page address such as *http://www.lucent.com*, plumbing sends a message to the Charon browser, which opens the browser and loads the web page. Alternatively, you can right click on the web page address when it is in a file you are browsing using the **Shell**. The message in the Plumb window is:

```
start /dis/wm/charon.dis port web
```

Plumbing example for *wm/view*:

If you have a file *earth.gif* in the <*inferno_root*> directory, you can open a **Shell** window and enter

```
/earth.gif
```

after the $ prompt, plumbing sends a message to the **GIF Viewer** and the image is displayed if the GIF format is one that the **GIF Viewer** recognizes.

Alternatively, you can open **Local** from the **Inferno** menu and select a file and the image is displayed.

Plumbing example for *wm/dir*:

When you open **Local** and right click on a text file such as */services.txt*, a Brutus window opens and the file is displayed. If the file you select has a *.gif* suffix, the **GIF Viewer** displays the image if the GIF format is one that the **GIF Viewer** recognizes.

If you right click on a directory name when you are in a **Shell** window, plumbing sends a message and opens a **Local** window where the directory and the contents of the directory are displayed as icons.

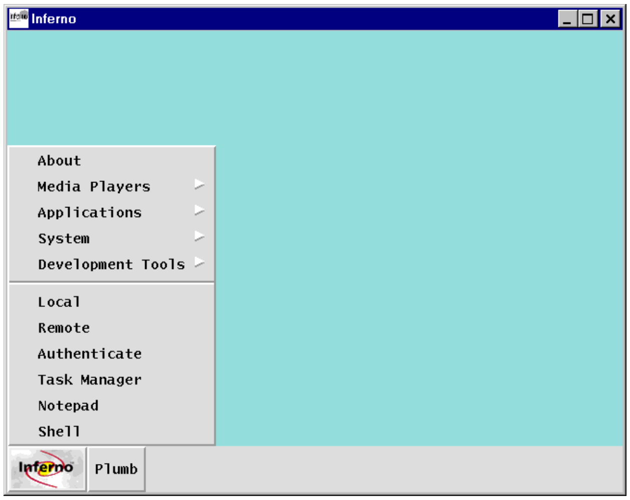
Plumbing example for *wm/deb*:

The plumbing feature is implemented differently for the Debugger. See the *Limbo Development Tools* chapter in the *Inferno Programmer's Guide*.

Task bar

After logging in, the Window Manager desktop is displayed. The task bar is displayed on the bottom of the screen. Click on the **Inferno** button on the lower left of the window and in the task bar to display the menu. Figure 3-3 shows this menu.

Figure 3-3 Window Manager Inferno Menu



From this menu, you can access the sample Window Manager applications.

Table 3-3 describes the options in the default **Inferno** menu.

Table 3-3 Default Inferno Menu Options

About	Displays information about the version of the Inferno system that you are running
Media Players	Displays a submenu with programs for viewing graphic files and playing video files
Applications	Displays a submenu with demo applications including a text editor for formatted text, an HTML browser, email, and an FTP file server

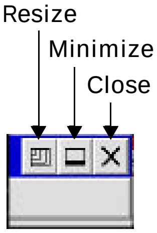
Table 3-3 Default Inferno Menu Options

System	Displays a menu with programs for showing memory usage and running tasks, debugging Limbo modules, and managing Dis modules
Development Tools	Displays a menu with a simple text editor, and programs for running tasks, debugging Limbo modules, and managing Dis modules.
Local	Shows the files and folders accessible to the local machine
Remote	Connects to a remote machine using *bind* and *mount* to add remote file system to namespace
Task Manager	Lists currently running modules
Notepad	Opens a simple text editor
Shell	Opens a command-line window similar to the EMU control console

Each of these options is described later in this chapter. The following section is a description of how to work in the Window Manager environment.

Window focus

Only one window element has the keyboard focus at any one time. In general, the first press of the mouse button in an entry, list, or text box gives it the focus. Moving the pointer to a menu gives it the focus. To bring a window to the front, click on the title bar. Usually, the window with focus has a blue title bar.

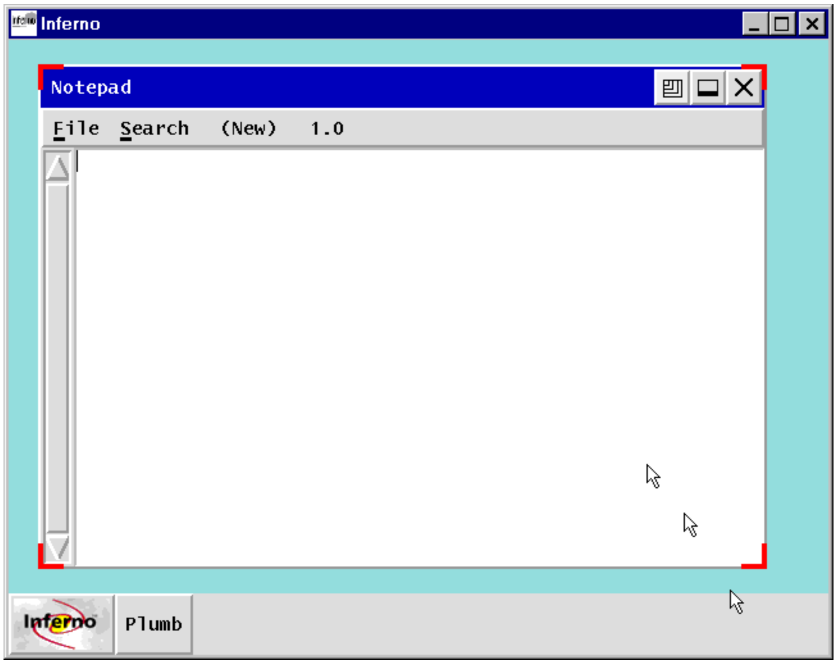


Resizing windows

You can resize a window to within the Window Manager borders.

To resize a window, click on the **Resize** button. Four corner brackets appear on the screen. You can ‘grab’ one of these corner brackets or click and drag inside the window border to resize the window. The corner brackets move as you move the mouse across the screen. Release the button to set the window to the new size. The window is divided into nine segments and you can click and drag from any point in a corner segment to change the size both horizontally and vertically. You can move one side by clicking on any point in the middle segment for that side. Figure 3-4 shows an example of enlarging a window.

Figure 3-4 Resizing a Window



Minimizing and restoring windows

Click on the **Minimize** button to reduce the window to an icon in the task bar. When a window is minimized, click on the icon in the task bar to restore it to the desktop.

Closing windows

Click on the **Close** button to close the window. This action shuts down the application running in that window.

Moving windows

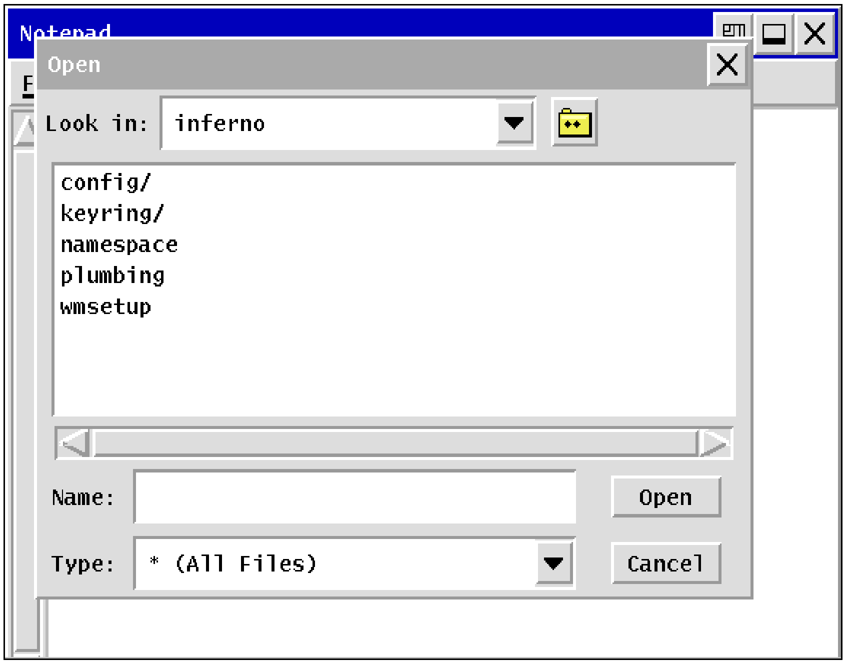
You can move an open window to any area within the Window Manager borders, including on top of the task bar.

To move a window, click on the title bar *and hold* it while you drag the window to a new location.

The Open dialog box

The standard **Open** dialog box is displayed when you select **File>>Open** from an applications menu bar. Figure 3-5 shows the standard **Open** dialog box.

Figure 3-5 Open Dialog Box



Look in: inferno

config/
keyring/
namespace
plumbing
wmsetup

Name:

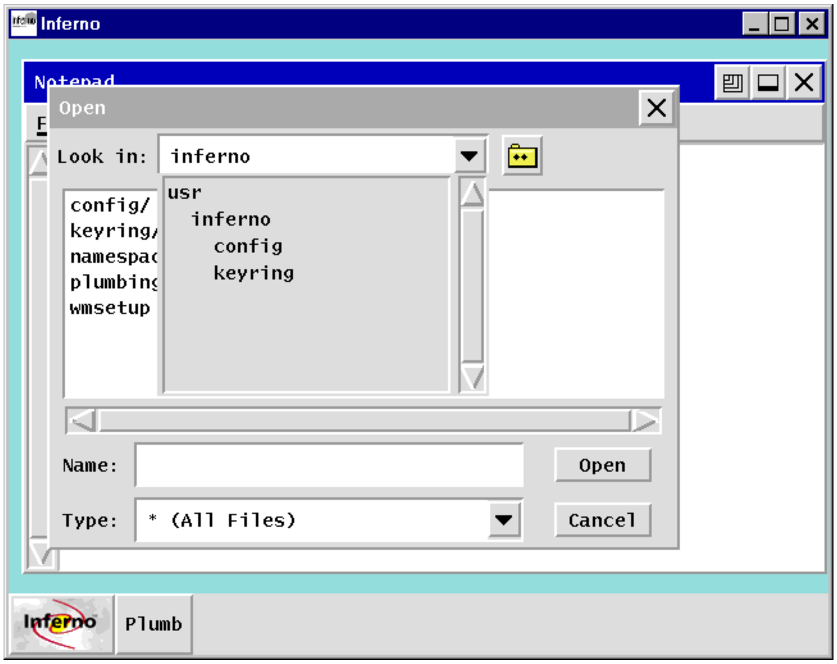
Type: * (All Files)

The **Open** dialog box displays the contents of the directory that is shown in the **Look in** box. In the screen sample above, *config* and *keyring* are followed by a slash indicating that each is a directory. The other listings are for files in the *inferno* directory. Double click on a filename to open the file. Alternatively, you can single click on the filename and click the **Open** button.

When you double click on a directory name, the directory entry will move to the **Look in** box.

To see the directories above and below the directory that is displayed in the **Look in** box, click on the arrow button to the right of the box. See Figure 3-6

Figure 3-6 Open Dialog Box - Directories



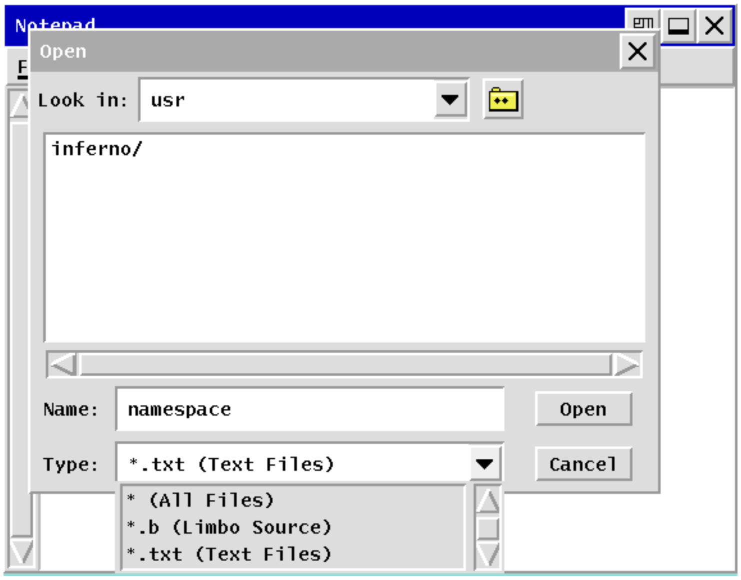
Three levels of directory names display in the above sample screen. The indentation of the list item shows the directory tree hierachy. The first level is the parent directory (*usr* in the above example); the second level (*inferno*) is the current directory; and the third level shows the subdirectories (*config* and *keyring*) of the current directory.

You can double click on a directory to reset the name in the **Look in** box.

To change the directory in the **Look in** box to the next higher or parent directory, click on the folder button to the right of the arrow button. The parent directory will move into the **Look in** box and the contents of this directory will be displayed in the screen area.

The default entry for the **Type** box is All Files. To see the other options, click on the arrow button to the right of the **Type** box. Figure 3-7 shows the alternative options. Double click on the option of your choice to replace the default entry.

Figure 3-7 Open Dialog Box - Type options



Look in: usr

inferno/

Name: namespace

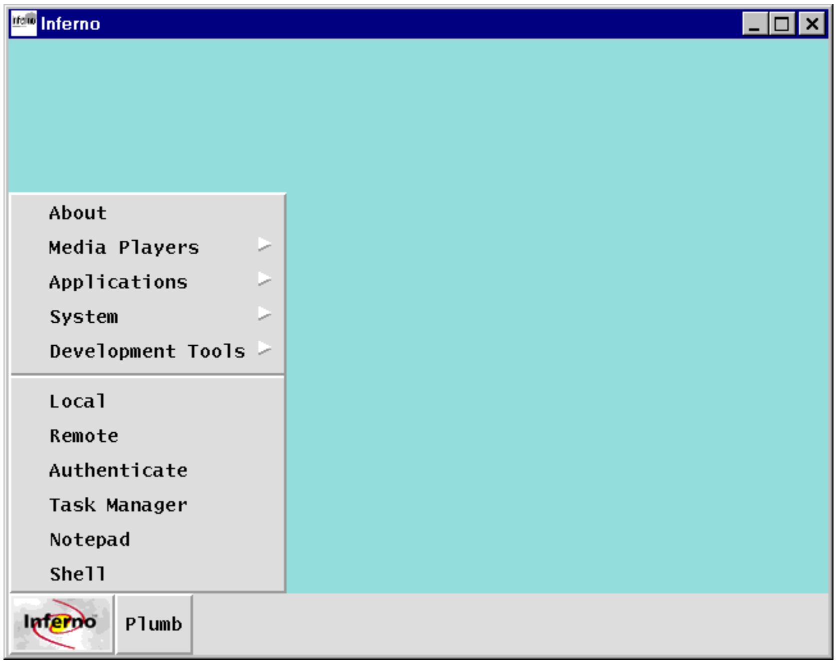
Type: *.txt (Text Files)

* (All Files)
*.b (Limbo Source)
*.txt (Text Files)

Inferno Menu

Click on the **Inferno** logo button in the lower left corner of the **Inferno** window to display the **Inferno** menu.

Figure 3-8 Inferno Menu

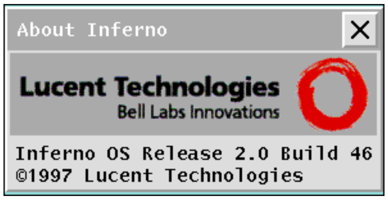


The applications listed on the menu are discussed in the sequence of the menu items.

About

The **About** dialog box displays the release and build number of the version of the Inferno system that you are running; see Figure 3-9.

Figure 3-9 About Inferno.

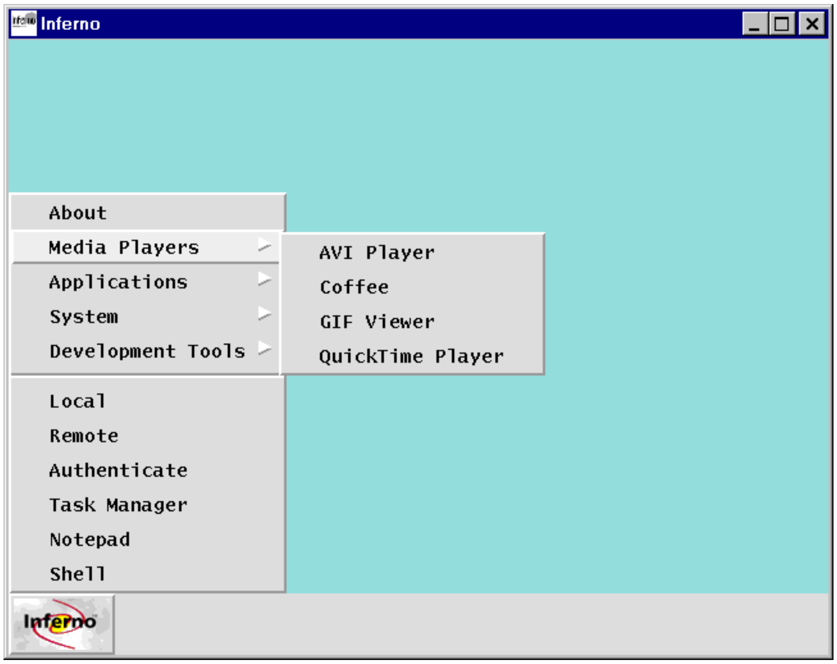


Inferno OS Release 2.0 Build 46
©1997 Lucent Technologies

Media Players

You can use the **Media Players** to view graphic files and play video files. Select **Media Players** on the **Inferno** menu to see the submenu shown in Figure 3-10.

Figure 3-10 Media Players Menu



The selections on the **Media Players** menu are listed in Table 3-4.

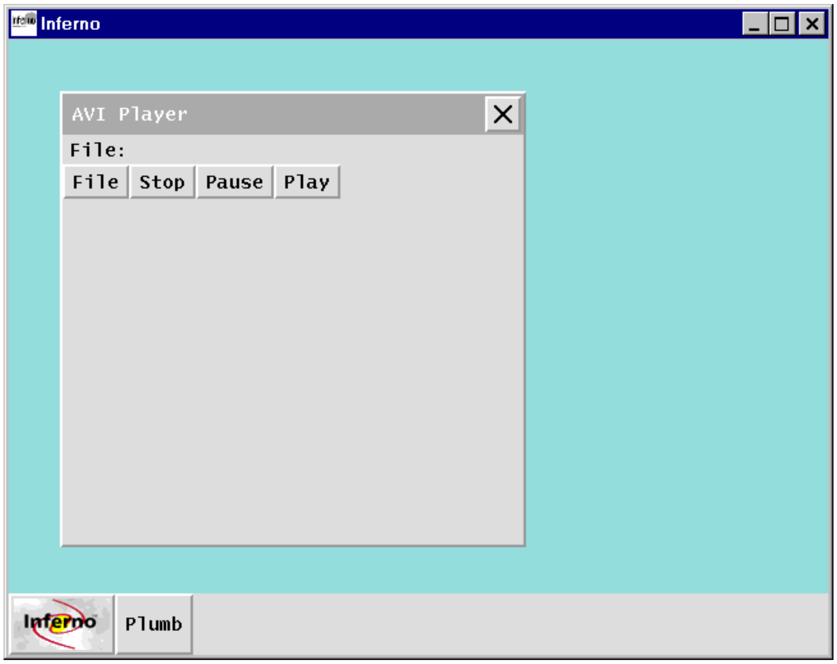
Table 3-4 Media Players

AVI Player	Plays .AVI files (RLE encoded only)
Coffee	Animated Inferno bitmap graphic files (*.bit*) using masks (*.mask*) to layer the images. Includes controls to vary the speed and direction of the animation.
GIF Viewer	Displays graphic files: ■ .GIF (87a/89a interlaced and non-interlaced) ■ .JPG and .JIF (JFIF 1.02 file exchange format) ■ Inferno bitmap .BIT ■ Inferno bitmap mask .MASK
QuickTime Player	Plays .QT files

AVI Player

When you select the AVI Player, the AVI Player screen is displayed:

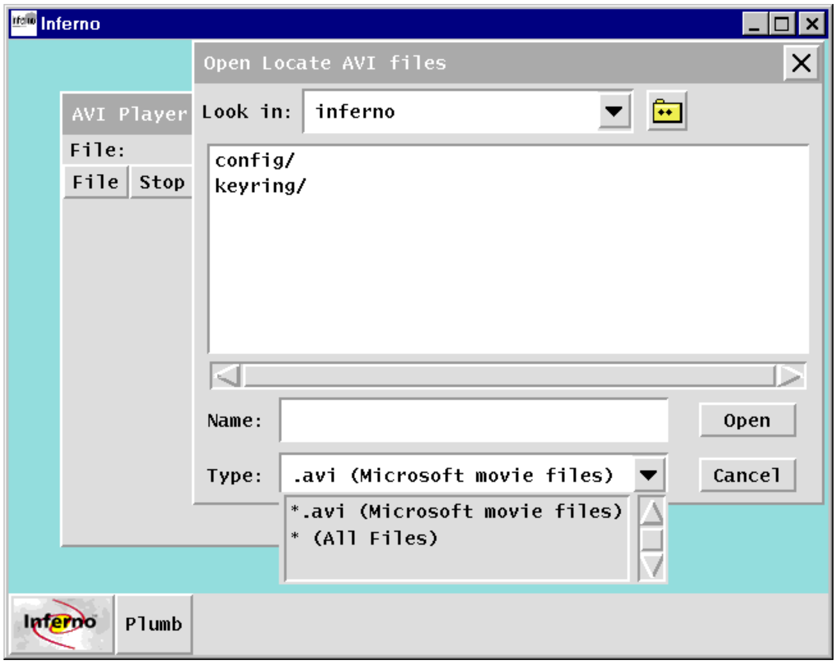
Figure 3-11 AVI Player Screen



The **Stop**, **Pause**, and **Play** buttons function as expected. When you select the **File** button, the **Open Locate AVI files** window is displayed.

Note: Only RLE encoded *.avi* files can be played with this version of **AVI Player**.

Figure 3-12 Locate AVI Player File

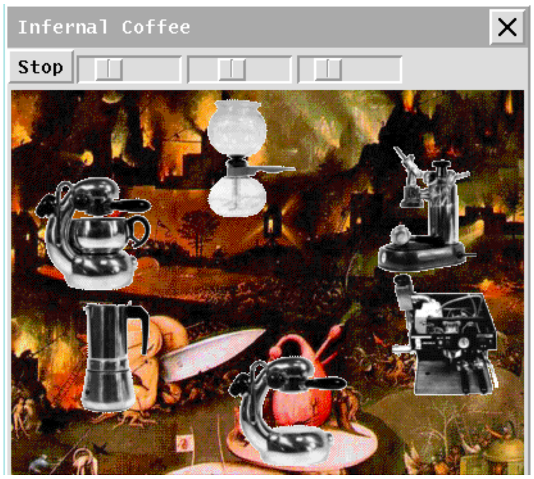


You can choose to see files with the *.avi* suffix only, or All Files. Select an RLE encoded AVI file to play.

Coffee

When you select the **Coffee** item on the **Media Players** menu, the **Infernal Coffee** animation is displayed.

Figure 3-13 Infernal Coffee

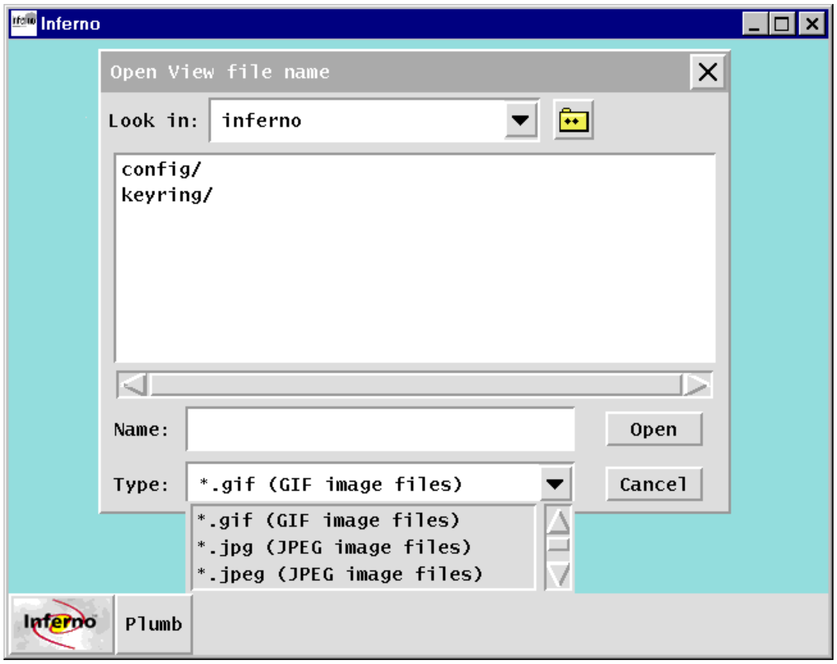


When you click the **Stop** button, it toggles to **Go** and stops the action. Move the slider controls to vary the speed, direction, and smoothness of the motion.

GIF Viewer

When you select **GIF Viewer** on the **Media Players** submenu, the **Open View** screen is displayed.

Figure 3-14 GIF Player screen



Look in: inferno

config/
keyring/

Name:

Type: *.gif (GIF image files)

The **Look in** and **Name** boxes and associated buttons act in the standard way for Inferno open dialog screens. The **Type** box allows you to select GIF or JPEG image formats.

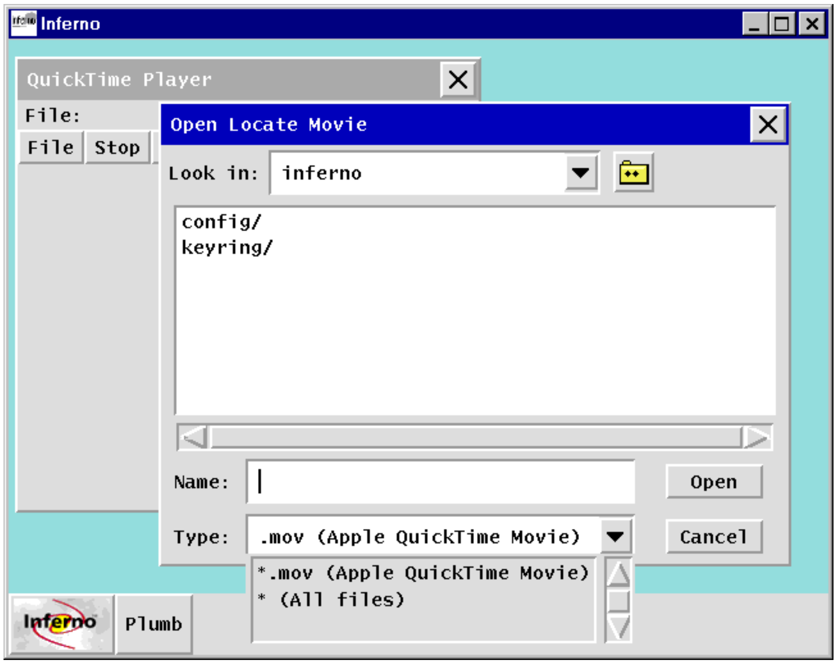
Note: When you close the **Open View** window without selecting a file, a small **View** box is left on the **Inferno** window and an “array bounds error” message for the View process is reported on the **EMU** console window. The minimize and close buttons on the **View** box show that additional features can be added, although they have not been added to this implementation. The **View** box does not minimize or close and will remain until you close the **Inferno** window.

QuickTime Player

When you choose the **QuickTime Player** item on the **Inferno** menu, the **QuickTime Player** window is displayed. The window is similar to the **AVI Player** window.

When you click on the **File** button, the **Open Locate Movie** window is displayed. Select a file in Apple QuickTime Movie format.

Figure 3-15 QuickTime Player screens and options



Inferno Audio Device

The Inferno Audio Device is not available through the **Inferno** menu, but you can set it up using the procedure discussed in this section.

You can use the Inferno Audio Device to:

- Use a microphone attached to your computer to hear the sound through your speakers or save the sound in a file
- Use a microphone attached to one computer to hear the sound through speakers on a machine across the network
- Play a sound file in .WAV or .SND format

The following examples use the audio device within the Inferno emulator.

Playing a sound file from the EMU control console

Enter the following commands on the **EMU** control console to hear a sound file named *soundfile*:

```
bind -a ‘#A’ /dev
cat soundfile > /dev/audio
```

The first command binds the audio device, #A, to the device directory, which is by default */dev*. The second command directs the sound file through the audio device, */dev/audio*.

The Inferno Audio Device uses the format, which can be .WAV or .SMD, and attributes specified in the *audioctl* file. This control file is in the directory where the audio device is bound. For more information, see *audio* - *interface to audio devices* in the *Inferno Devices* chapter of the *Inferno Reference Manua*l.

To view the current settings in the control file, enter the following command:

```
cat /dev/audioctl
```

The default control file is:

```
indev mic line
outdev spkr hdph line
enc pcm ulaw alaw
rate 8000 11025 22050 44100
bits 16 8
chans 2 1
left 100 0 100
right 100 0 100
buf 100 0 100
count 0
```

Each line represents an individual characteristic and all possible values. The second field in each line is the current setting.

To change a setting to match a particular sound file, use the *echo* command. For example, to change the sample rate, enter:

```
echo rate=11023 >/dev/audioctl
```

Playing a sound file from within Limbo programs

There are three Limbo files that demonstrate programming for the Inferno Audio Device: *play.b*, *talk.b*, and *talk across.b*. These files are available from the Inferno web site. Go to the URL *http://www.lucent.com/inferno* and select the *Technical Support* page. On that page is a link to the *Inferno Audio Device* page where there are links to the files. Save the three files. It is recommended that you save the files in *<inferno_root>/usr/inferno*.

To compile the *.b* files into executable *.dis* files, use the Limbo compiler. To compile *play.b*, enter:

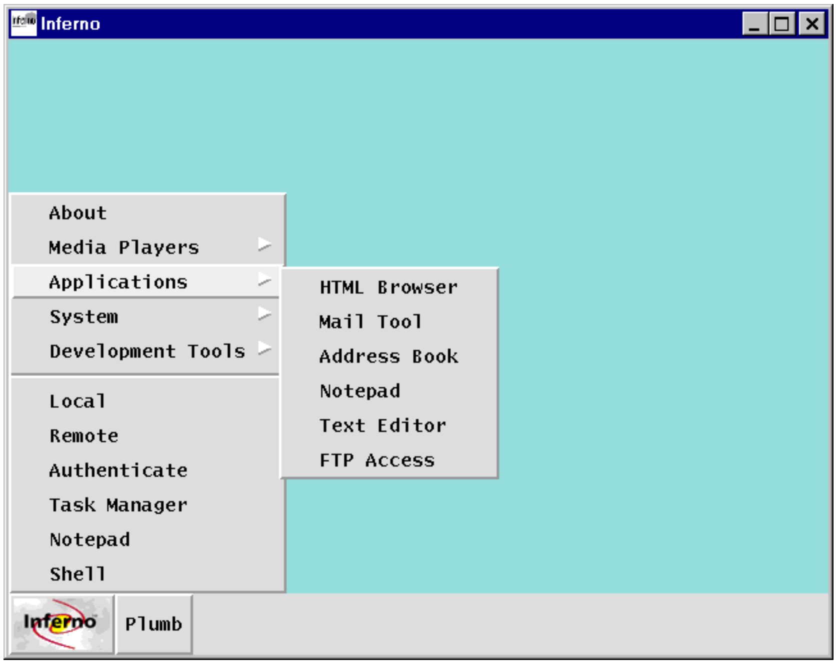
```
limbo play.b
```

You can experiment with these Limbo modules and recompile them to learn more about programming for the Inferno Audio Device.

Applications Menu

Applications is the next item on the **Inferno** menu and the submenu includes several sample applications. See the menu shown in Figure 3-16.

Figure 3-16 Applications Menu



Charon Web Browser

See the link to the Charon Browser information on the Inferno Release 2.3 Alpha Version Web Page. Note that this Charon Browser is a completely new browser and is not an upgrade of the Inferno 1.0 Charon browser.

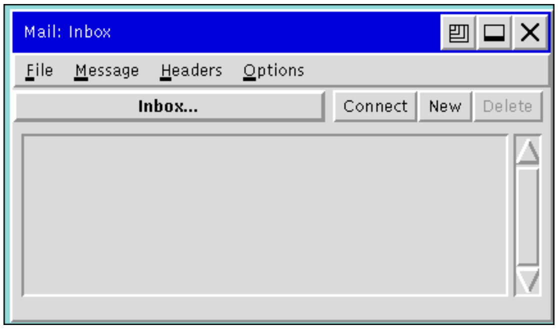
Mail Tool

The Inferno Mail Tool application is a sample implementation of a simple e-mail reader. It uses TCP/IP and the POP3 protocol to download your new mail from a remote mailbox. Composed messages are held in the Outbox until you connect to the server to deliver them.

You must have an open network connection and have started the Connection Server (*lib/cs*) before you can use the Mail Tool. To use POP3 services, you must have an account on the host machine.

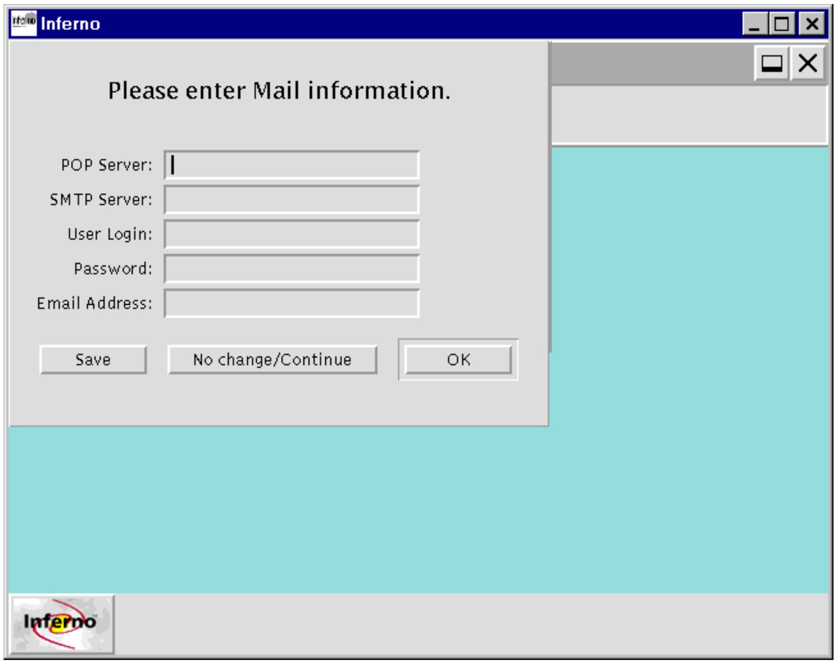
Select **Applications>>Mail Tool** from the **Inferno** menu to start the application. Figure 3-17 shows the initial **Mail Tool** window.

Figure 3-17 Mail Tool Window



You must enter the information for your mail connection. You can select the **Connect** button or use the **Connect to Server** item on the **File** submenu. The Mail Information dialog box will be displayed as shown in Figure 3-18.

Figure 3-18 Mail Tool Information Window



Please enter Mail information.

POP Server:

SMTP Server:

User Login:

Password:

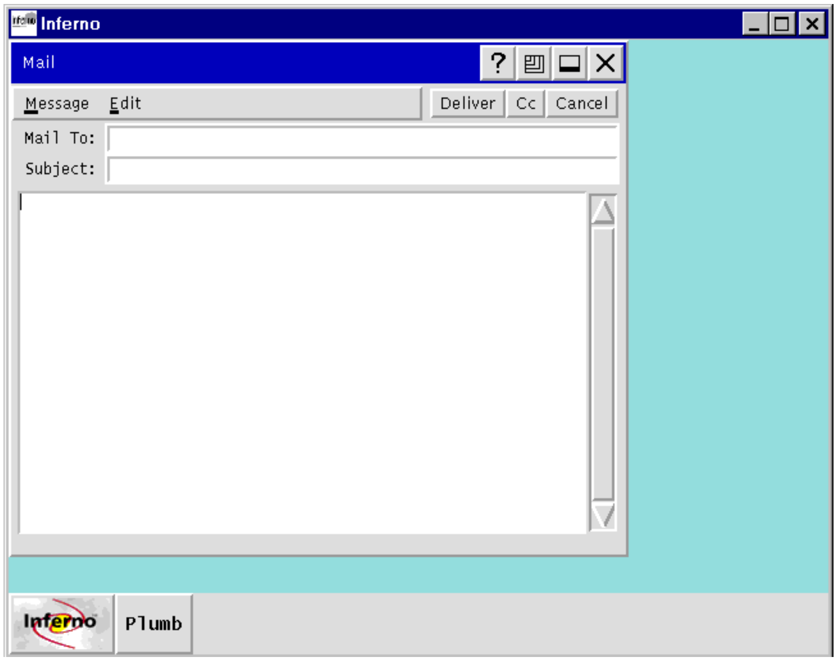
Email Address:

Use **Enter** to move to the next field. When you enter your password, the entry will be obscured. When you have entered the information, click on the Save button. The information is saved and you click on the OK button. Then select the OK button on the **Mail information** window to retrieve your mail messages from the server. The last 20 mail messages are identified in the Inbox.

You can retrieve the **Previous** or **Next** 20 mail headers by using these items in the **Headers** submenu.

When you select the **Compose** item on the **Message** menu, the **Compose** window opens as shown in Figure 3-19.

Figure 3-19 Mail Tool Compose Window.



The **Compose** window has two options: Message and Edit. You can add a CC line by using the **CC** toggle button on the menu bar or using the **Add CC...** submenu item under **Message.**

Status messages appear at the bottom of the window to help you with your mail handling. When your mail is being retrieved, Working appears in red on the right side of the bottom status bar. Other messages appear on the left side of the status bar. When you select Deliver, the message

```
Deliver message to Outbox and close Compose window
```

appears on the status line. To send mail, you must connect to the server.

Notepad

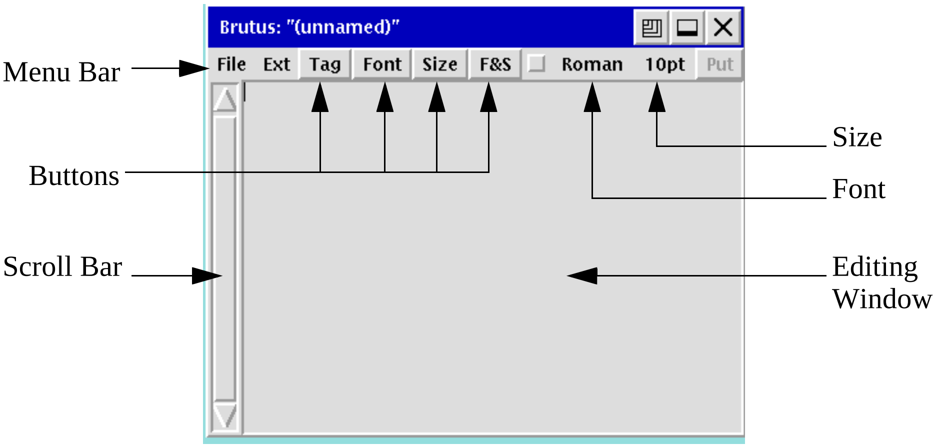
The **Notepad** is discussed later in this chapter under the **Inferno** menu item **Notepad**.

Text Editor—Brutus

Brutus is a sample text editor. For Inferno Release 2.0, the plumbing feature has been added to **Brutus**. Plumbing allows you to open related files by clicking with your right mouse button. Files created with **Brutus** can contain SGML (Standard Generalized Markup Language) tags.

Select **Applications>>Text Editor** from the **Inferno** menu to start **Brutus**. Figure 3-20 shows the parts of the Brutus window.

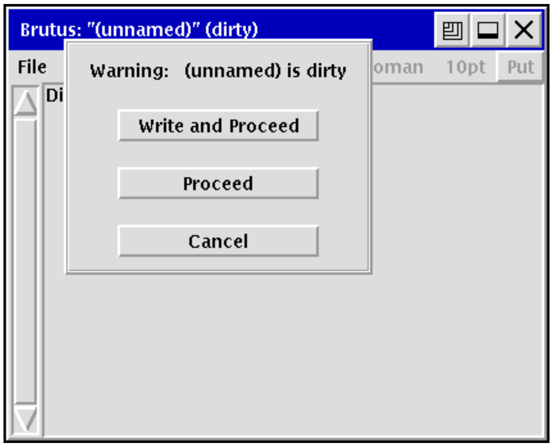
Figure 3-20 The Brutus Window



Brutus opens with an unnamed file and that is shown on the title bar as **"(unnamed)"**. When you enter data in the editing window, **(dirty)** follows the identification as unnamed or the pathname of the file. The **File and Ext** menu items and the **Tag, Font**, **Size,** and **F&S** button functions are discussed in more detail on the following pages.

If you select **File>>Open** to view another file or try to close the window, **Brutus** displays a warning message that the file you changed is dirty. You must choose to **Write and Proceed**, **Proceed**, or **Cancel**.

Figure 3-21 Brutus Dirty File Warning.



Brutus: "(unnamed)" (dirty)

Warning: (unnamed) is dirty

The **File** menu in **Brutus** contains standard file options. Press and hold the mouse button to view the options. Table 3-7 lists the options and their functions.

Table 3-5 Brutus File Menu

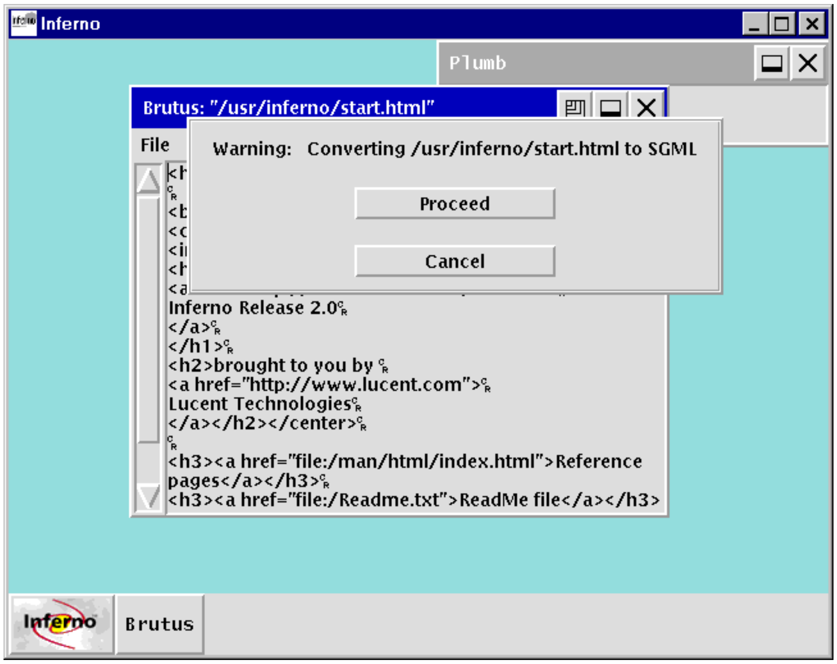
New	Open a file in a new window (may be an existing file or a new file).
Open	Open a file in the current window.
Name	Name the file in the current window.
Write	Write the file to disk.
Enable Fonts	Convert file to SGML. Menu item changes to Disable Fonts.
[Brutus]	Display the Brutus control window.
(unnamed) or *filename*	Display the window containing that file.

When you select **Enable Fonts** and the file has not already been converted to SGML, a warning box displays a warning that the file is being converted to SGML. You can choose to **Proceed** or **Cancel**. If you proceed, the file is converted and the Menu Bar buttons and the **Ext** menu item are enabled. If the file has already been converted to SGML, the buttons and the **Ext** item are enabled.

Example: Open a Shell window and enter *wm/charon* to start the Charon **Inferno Web Browser**. Save the HTML source of the start page by using the **Save Source** item on the **File** submenu. The file is saved in your */usr* directory. For this example, the file is saved in */usr/inferno*.

When you open the file in Brutus, the buttons on the Menu Bar are not active. In the File submenu choose Enable Fonts. A warning box is displayed as shown in Figure 3-22.

Figure 3-22 Brutus Enable Fonts Warning Box



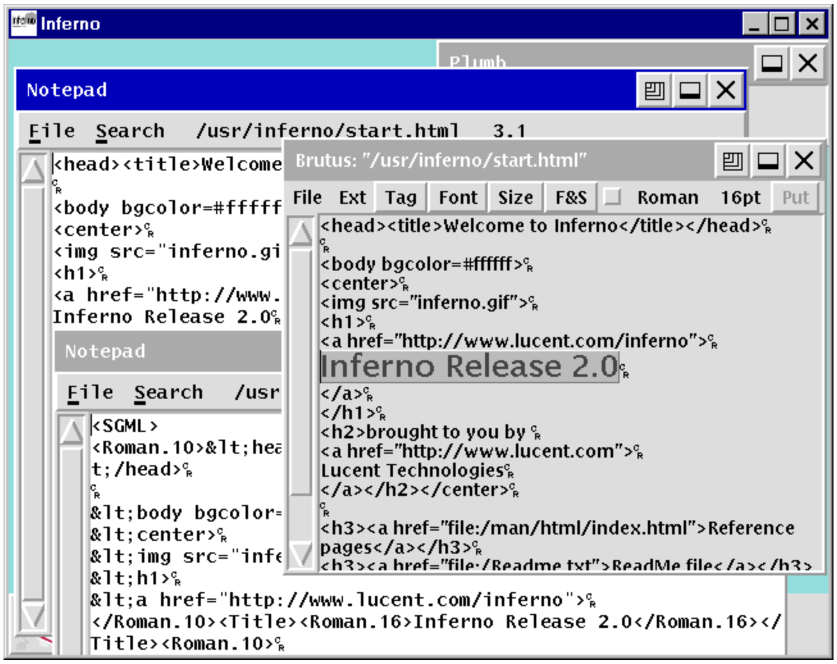
Warning: Converting /usr/inferno/start.html to SGML

```
Inferno Release 2.0
</a>
</h1>
<h2>brought to you by
<a href="http://www.lucent.com">
Lucent Technologies
</a></h2></center>

<h3><a href="file:/man/html/index.html">Reference
pages</a></h3>
<h3><a href="file:/Readme.txt">ReadMe file</a></h3>
```

When you click on **Proceed, Brutus** converts the file to SGML. The buttons on the Menu Bar and the **Ext** menu item are now active. You can write the file using the **Write on the File** submenu, open a **Notepad** window, and open the file. Then, in the **Brutus** window move the cursor to the line with *Inferno Release 2.0* and on the **Tag** check box list select **Title**. With the phrase still highlighted, select font size 16. Then use the **Put** button to write the file. Open another **Notepad** window and compare the SGML for the phrase. See Figure 3-23 where the three files are positioned so you can see these changes.

Figure 3-23 Brutus SGML Example


Inferno
Plumb
Notepad
File Search /usr/inferno/start.html 3.1
Brutus: "/usr/inferno/start.html"
File Ext Tag Font Size F&S □ Roman 16pt Put
Notepad
File Search /usr

```
<head><title>Welcome

<body bgcolor=#fffff
<center>
<img src="inferno.gi
<h1>
<a href="http://www.
Inferno Release 2.0
```

```
<head><title>Welcome to Inferno</title></head>

<body bgcolor=#ffffff>
<center>
<img src="inferno.gif">
<h1>
<a href="http://www.lucent.com/inferno">
Inferno Release 2.0
</a>
</h1>
<h2>brought to you by
<a href="http://www.lucent.com">
Lucent Technologies
</a></h2></center>

<h3><a href="file:/man/html/index.html">Reference
pages</a></h3>
<h3><a href="file:/Readme.txt">ReadMe file</a></h3>
```

```
<SGML>
<Roman.10>&lt;hea
t;/head>

&lt;body bgcolor=
&lt;center>
&lt;img src="infe
&lt;h1>
&lt;a href="http://www.lucent.com/inferno">
</Roman.10><Title><Roman.16>Inferno Release 2.0</Roman.16></
Title><Roman.10>
```

The **Ext** (Extension) menu in **Brutus** contains options that can be used to add elements to the file. Table 3-5 lists the options and their functions.

Table 3-6 Brutus Ext Menu

Add excerpt	Insert the contents of a file.
Add image	Insert an image in a file.
Add mod	Insert a module.
Add table	Insert a table.

You can use the **Add excerpt** extension to insert a text file in your SGML file. In the entry box on the **parameters for excerpt** dialog box enter the filename of the file that contains the information to be inserted.

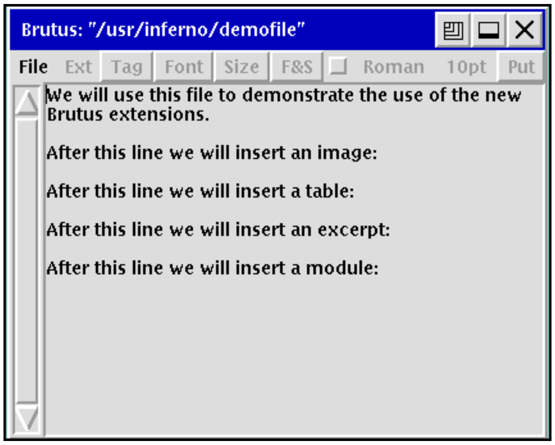
You can use the **Add image** extension to insert an image in the SGML file. In the entry box on the **parameters for image** dialog box enter the filename of the file that contains the image. The image must be in a file and be a *.bit* file or a format that is recognized by the GIF Viewer; see Table 3-4, Media Players, in this guide.

You can use the **Add mod** extension to insert a module in the SGML file. In the entry box on the **parameters for mod** dialog box enter the filename of the file that contains the module.

You can use the **Add table** extension to insert a table into an SGML file. In the entry box on the **parameters for table** dialog box enter the filename of the file that contains an HTML or SGML-coded table. The table will be inserted in the SGML file at the location of the cursor.

Example: Using **Notepad**, create and write a file *demofile* in */usr/inferno* or in your */usr/<username>* directory. Open a **Brutus** window (**Inferno>>Applications>>Text Editor**) and open *demofile* using **File>>Open**.

Figure 3-24 Brutus with Text File.



We will use this file to demonstrate the use of the new Brutus extensions.

After this line we will insert an image:

After this line we will insert a table:

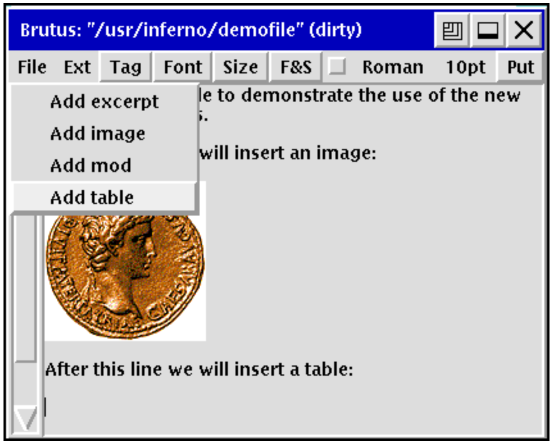
After this line we will insert an excerpt:

After this line we will insert a module:

Next, convert the file to SGML using **File>>Enable Fonts**. Now, **Ext** and the SGML buttons will show black type rather than the gray to indicate that they are active. Place your cursor in the file where you would like to insert an image. In the **Ext** submenu select **Add image**. A **parameters for image** dialog box will open. In the entry box, enter */icons/coin.bit*. The image is inserted in your file and it is displayed in the **Brutus** window.

You can insert a table in the same way. In the **Ext** submenu select **Add table**. Place your cursor in the file where you would like to insert a table. In the **Ext** submenu select **Add table** as shown in Figure 3-25.

Figure 3-25 Brutus Using SGML Extensions



le to demonstrate the use of the new
.

will insert an image:

After this line we will insert a table:

A **parameters for table** dialog box will open. In the entry box, enter the filename of a table in SGML format. The formatted table will be inserted in your file in the **Brutus** window.

Buttons

The buttons along the top of the editor window (to the right of the **File** and **Ext** menu items) are for formatting functions. Table 3-7 lists the buttons and their functions.

Table 3-7 Brutus Buttons

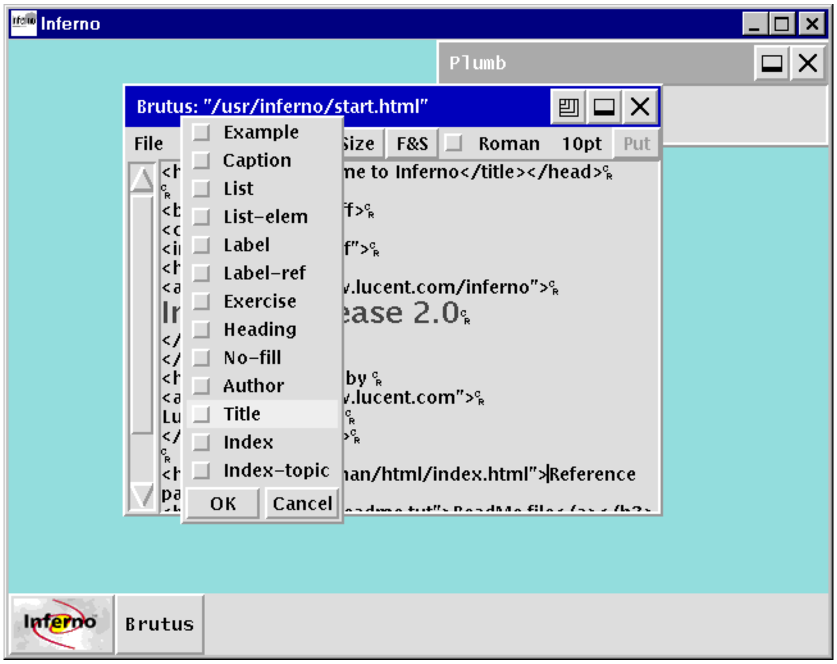
Tag	Apply a standard SGML tag to the selected text.
Font	Apply the specified font to the selected text.

Table 3-7 Brutus Buttons

Size	Apply the specified size to the selected text.
F&S	Apply the specified font and size to the selected text.
Put	Write the file to disk. (Only available when the file exists or has been written once and it is “dirty,” that is, it contains unsaved changes.)

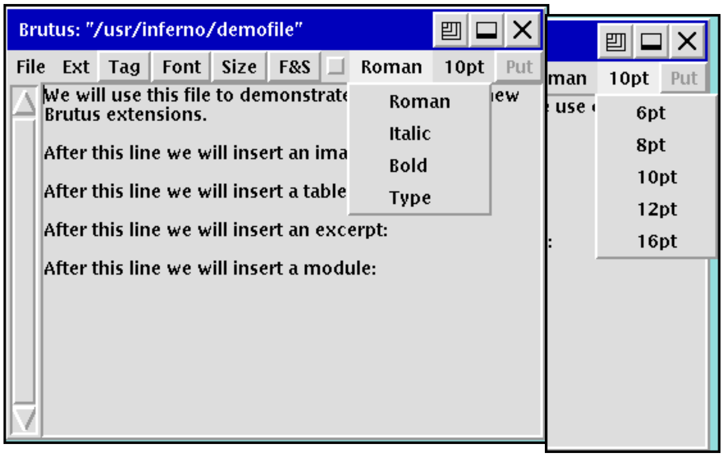
When you select the **Tag** button, a list of check boxes is displayed as shown in Figure 3-26.

Figure 3-26 Brutus SGML Tag List



You select font and size from pop-up menus.

Figure 3-27 Brutus Font and Size Pop-up Menus


Brutus: "/usr/inferno/demofile"

We will use this file to demonstrate
Brutus extensions.

After this line we will insert an ima

After this line we will insert a table

After this line we will insert an excerpt:

After this line we will insert a module:

The font options are **Roman**, **Italic**, **Bold,** and **Type**. The font is changed for the word identified by the cursor or the area that is highlighted. Once the font has been reset, you can apply that font to an area by highlighting the area and clicking on the **Font** or **F&S** button.

You select the font size from the pop-up menu. The options are **6pt** (point)**, 8pt**, **10pt**, **12pt** and **16pt**. The size is applied in the same way as the font is applied except you use the **Size** or **F&S** button.

Mouse buttons

For Release 2.0, the cut and paste functions that were available on the right mouse button are now on button two. They are not available when you use a two button mouse.

Brutus Plumbing

Brutus is a plumbed program.To use the plumbing feature when you are using the **Text Editor**, click the right mouse button on a filename. The Window Manager opens a new **Brutus** window that displays the selected file. You can move this new window that covers the window in which you did the selection. You can continue to make selections in this way and each time a new **Brutus** window opens to display the newly selected file. Plumbing works for compiler error messages, also. Each time you select a file, you can see the plumbing message in the **Plumb** window. For example, with */appl/wm/edit.b* open and the filename *sys.m* is selected, the message

```
send “/module/sys.m” to edit
```

is displayed in the **Plumb** window.

If your selection does not match a filename, plumbing displays the message

```
don’t know who message goes to
```

in the **Plumb** window.

The **Plumb** window is minimized when the Window Manager is started. It must be opened before the plumbing feature will work. If you open Brutus before the **Plumb** window is opened, a message is displayed on the **EMU** window:

```
Brutus: can’t read /chan/plumb.edit: fd out of range or not
          open
```

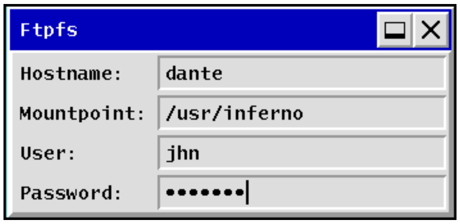
If you see this message and you want to use the plumbing feature, you need to close the **Brutus** window, open the **Plumb** window, and then open a **Brutus** window for the plumbing feature to work. Plumbing will not work in **Brutus** windows that were opened before the **Plumb** window was opened.

FTP Access

Note: The source code for the FTP Access application is not supplied with the evaluation copy of the Inferno system.

When you select the **FTP Access** item from the Inferno **Applications** menu, the **Ftpfs** dialog box is displayed on the screen.

Figure 3-28 Ftpfs Dialog Box



Hostname: dante

Mountpoint: /usr/inferno

User: jhn

Password: •••••••

Enter the name of the remote machine in the **Hostname** box and enter the location to mount the remote machine’s file system. By default the location for mount and bind operations of remote file systems using FTP is in /n/ftp.

Note: Use the **Enter** or **Return** key to move between fields.

Enter the user name and password in the dialog box. Then press **Enter** or **Return** to make the connection.

The remote file system is mounted at the mountpoint directory. The Directory window is displayed, showing the files and directories mounted from the remote machine. The files from the remote machine are accessible as if they were physically located on the local machine. The Inferno system is using the File Transfer Protocol to access the files across the network on the remote machine.

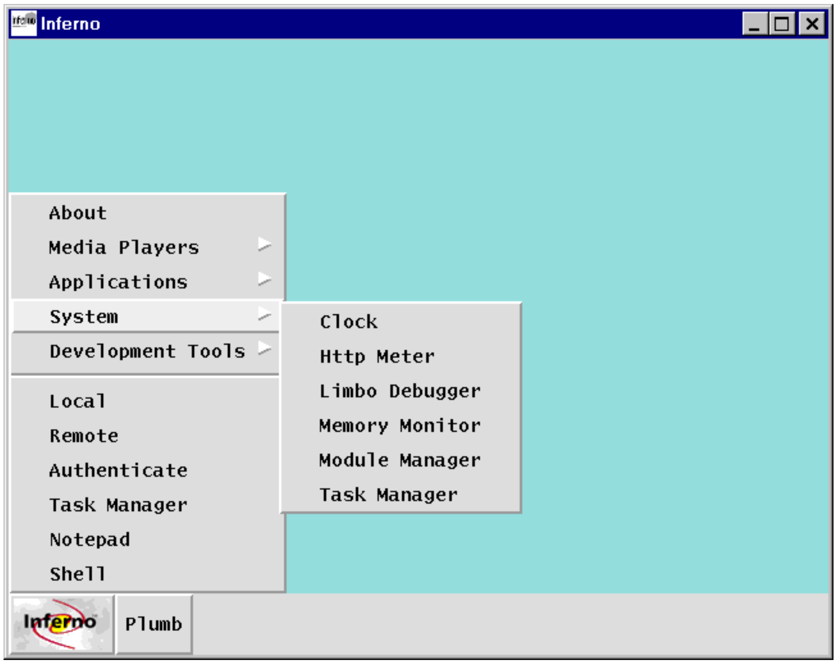
Ftpfs works with firewalls, too. Prepend **proxy!** to the hostname you enter. In the /services/cs/db file add *$PROXY* with the name of the proxy. For example:

```
$PROXY=nj001.firewall.corp.com
```

System

The **System** menu includes sample applications for a clock, debugging Limbo programs, viewing memory usage, managing running tasks (modules), and the **Task Manager** for viewing Dis modules.

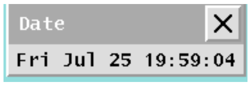
Figure 3-29 System Menu



The **Task Manager** is described later in this chapter under the **Tasks** menu item. The **Limbo Debugger** and the **Module Manager** are described in the chapter on *Limbo Development Tools* in the *Inferno Programmer's Guide.*

When you click on the **Clock** menu item, a digital clock is displayed that also includes the day of the week, the month and the day of the month.

Figure 3-30 System Clock



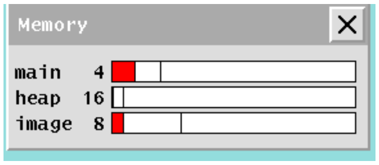
Date ☒

Fri Jul 25 19:59:04

To remove the **Date** box, click on the **X** button in the upper left corner.

When you click on the **Memory** menu item, memory usage for main, heap, and image memory is shown.

Figure 3-31 System Memory



Memory ☒

main 4

heap 16

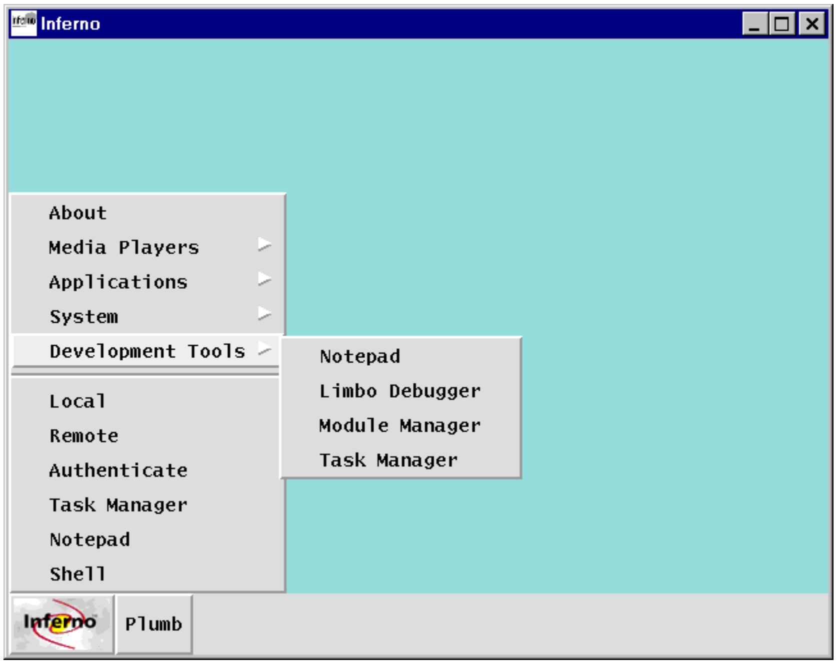
image 8

Memory usage is reflected for each type of memory. The file */dev/memory* is the source of the information. The solid bars show how much memory is being used at the current time, while the line to the right of the solid bar is the maximum amount of memory used during this instance of the Inferno system. The number listed before each bar is the maximum memory allocated for that type of memory.

Development Tools

Select **Development Tools** from the **Inferno** menu to see the choices on this menu.

Figure 3-32 Inferno Development Tools Menu

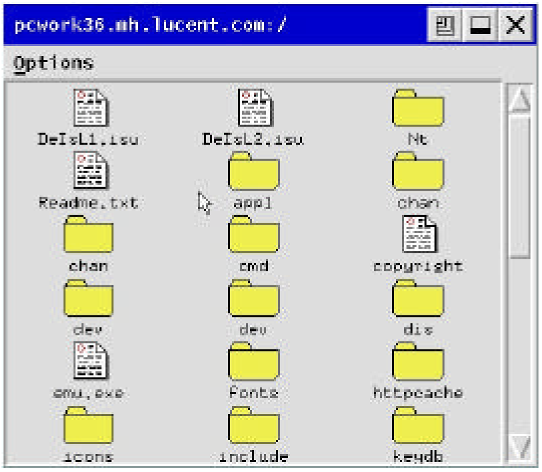


The **Notepad** and **Task Manager** are discussed later in this chapter under the **Inferno** menu items **Notepad** and **Task Manager**, respectively. The **Limbo Debugger** and the **Module Manager** are described in the chapter on *Limbo Development Tools* in the *Inferno Programmer's Guide.*

Local

Select **Local** from the **Inferno** menu to display the files and folders (directories) visible to the local machine. File systems that have been mounted from remote machines are listed here, too. Figure 3-33 shows the window at <*inferno_root*>.

Figure 3-33 Display Local Files and Folders



When you select **Options**, the options are listed as shown in Figure 3-34.

Figure 3-34 Display Local Files Options

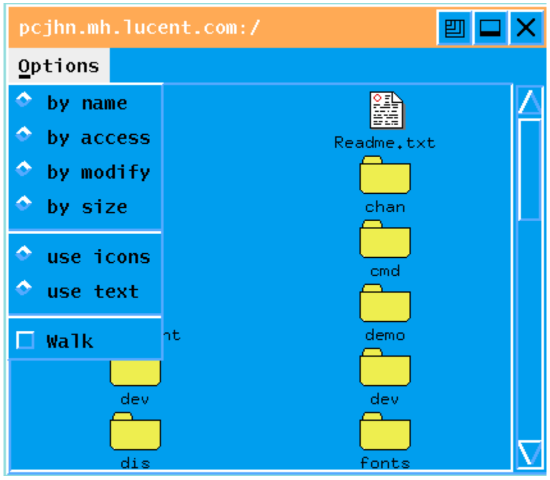


Table 3-8 lists the Local options.

Table 3-8 Local Display Options

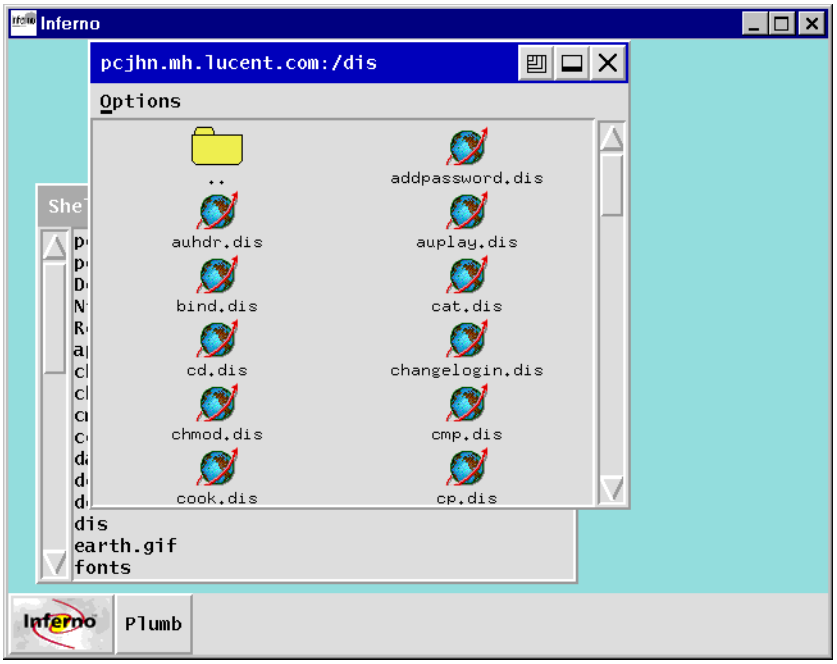
by name	Sort by file name; alphabetical, descending (a-z)
by access	Sort by last access date; decending (most recent first)
by modify	Sort by last modified date; decending (most recent first)
by size	Sort by file size; descending (largest first)
use icons	Display as icons (see Figure 3-13)
use text	Display as text; lists file type, file size, date/time and file name
Walk	Browse folders using a single window that changes as you open each folder By default, a new window opens for each folder

The display **Local** files module, *wm/dir*, is plumbed. It receives directory names and sends file names.

When you open **Local** and right click on a text file such as */services.txt*, a Brutus window opens and the file is displayed. If the file you select has a *.gif* suffix, the **GIF Viewer** displays the image if the GIF format is one that the **GIF Viewer** recognizes.

If you right click on a directory name when you are in a **Shell** window, plumbing sends a message and opens a **Local** window where the directory and the contents of the directory are displayed as icons.

Figure 3-35 Local Receives Plumbing Messages

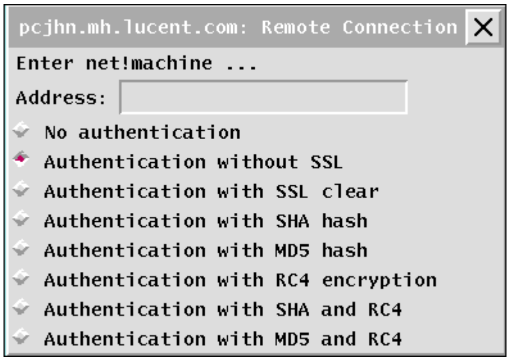


Remote

Select **Remote** from the **Inferno** menu to connect to a remote machine. The remote machine must be an Inferno server (running *lib/srv*).

When you attempt to connect to a remote machine, you must specify the address of the remote machine and a connection method. Figure 3-36 shows the Remote Connect dialog box.

Figure 3-36 Connect to a Remote Machine



pcjhn.mh.lucent.com: Remote Connection

Enter net!machine ...

Address:

○ No authentication
● Authentication without SSL
○ Authentication with SSL clear
○ Authentication with SHA hash
○ Authentication with MD5 hash
○ Authentication with RC4 encryption
○ Authentication with SHA and RC4
○ Authentication with MD5 and RC4

The machine name must be in the *net!machine* format. For example, *tcp!<remote.machine>*.

Table 3-9 lists the connection options and the differences between them.

Table 3-9 Remote Connection Security Options

No authentication	No security checks are made.
Authentication without SSL	Uses digital signatures to verify the authenticity of the machines connecting to each other. The Secure Sockets Layer (SSL) is not used.
Authentication with SSL clear	Uses digital signatures to verify the authenticity of the machines connecting to each other. The Secure Sockets Layer is used to provide encryption using an asymmetric or public key.
Authentication with SHA hash	Authenticates and uses message digesting with the Secure Hash Algorithm (SHA) to verify the authenticity of each message sent and received
Authentication with MD5 hash	Authenticates and uses message digesting with the Message Digest Algorithm #5 (MD5) to verify the authenticity of each message sent and received.
Authentication with RC4 encryption	Authenticates and uses RC4 symmetric or private key encryption to encrypt each message
Authentication with SHA and RC4	Authenticates, uses RC4 private key encryption, and uses the Secure Hash Algorithm to verify messages sent and received.
Authentication with MD5 and RC4	Authenticates, uses RC4 private key encryption, and uses MD5 hashing to verify messages sent and received.

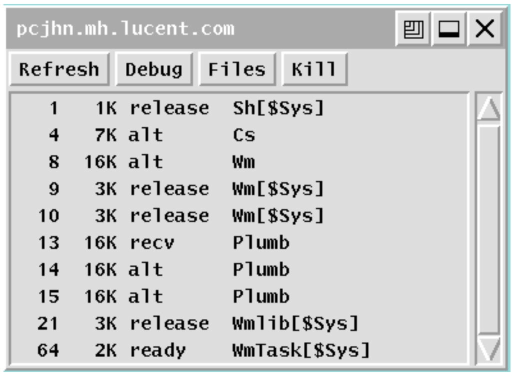
When authentication is complete, a window is displayed that is similar to the one displayed when you select **Local** from the **Inferno** menu.

Task Manager

The **Task Manager** program shows which Inferno tasks, or modules, are currently in memory. This can be used to invoke the debugger for a loaded module or to kill a module that has stopped responding to the system.

Select **Task Manager** from the **Inferno** menu to start the Task Manager program.

Figure 3-37 Task Manager Window



```
  1   1K release  Sh[$Sys]
  4   7K alt      Cs
  8  16K alt      Wm
  9   3K release  Wm[$Sys]
 10   3K release  Wm[$Sys]
 13  16K recv     Plumb
 14  16K alt      Plumb
 15  16K alt      Plumb
 21   3K release  Wmlib[$Sys]
 64   2K ready    WmTask[$Sys]
```

Figure 3-37 shows a typical task list, including:

- Shell module (*Sh*) that was loaded when you started the emulator
- Connection Server (*Cs*) that was loaded when you started the module by typing *lib/cs* on the **EMU** control console
- Window Manager modules (*Wm*) that were loaded when you started Window Manager by typing *wm/logon*

- The Plumbing module *(Plumb)* that is started with the Window Manager
- The library module *(WmLib)* that was started with the WIndow Manager since most programs reference it
- Tasks module (*WmTask*) that was loaded when you selected **Task Manager**

If you load or unload modules after you started the **Task Manager**, clicking on the **Refresh** button refreshes the display with the current status information.

You can select a task and invoke the Limbo Debugger by clicking on the **Debug** button. For more information about the Limbo Debugger, see the chapter on *Limbo Development Tools* in the *Inferno Programmer's Guide*.

You can select a task and click on the **Files** button to display the resources associated with the task. The **Notepad** is opened for the display.

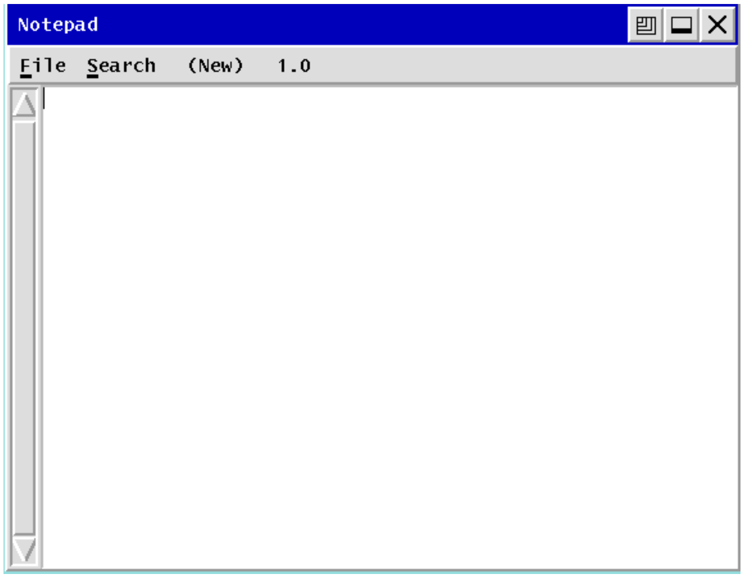
You can kill a task by highlighting the task and clicking on the **Kill** button. This is useful to close a module that has stopped responding.

Note: You should be careful not to kill a module that may be in use by the system, such as the Shell, the Connection Server, or any of the Window Manager modules. Doing this can cause the Inferno system to halt abruptly.

Notepad

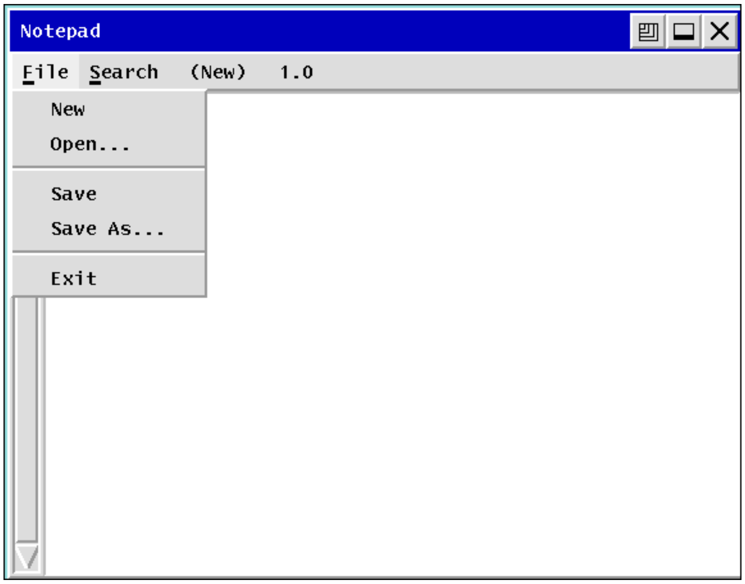
You can use the Notepad to create and edit plain text (ASCII) files. Select **Notepad** from the **Inferno** menu to start the Notepad application. Figure 3-38 shows the **Notepad** window.

Figure 3-38 Notepad Window



Click on the **File** menu to display the options available. See Figure 3-39.

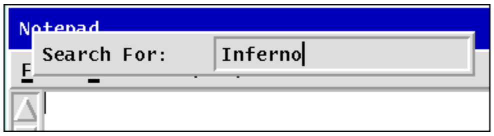
Figure 3-39 Notepad Window - File Options



The **File** options are the usual **New**, **Open**, **Save**, **Save As...**, and **Exit**. When you click on **Save** for a new file or **Save As...**, a dialog box appears for you to enter the filename. When the file is named, the filename replaces the **(New)** entry on the options bar. When you open a file, the full pathname of the file appears in the menu bar. The number to the right of the filename or **(New)** entry is the line and character placement of the cursor in the format <*line-number*>.<*character-position*>.

The **Search** options are **Search** and **Search For...**. When you select **Search**, the search is for the next iteration of the search string that was previously entered. When you select the **Search For...** option, a dialog box is displayed for you to enter your search string. See Figure 3-40.

Figure 3-40 Notepad Window - Search For... Dialog Box



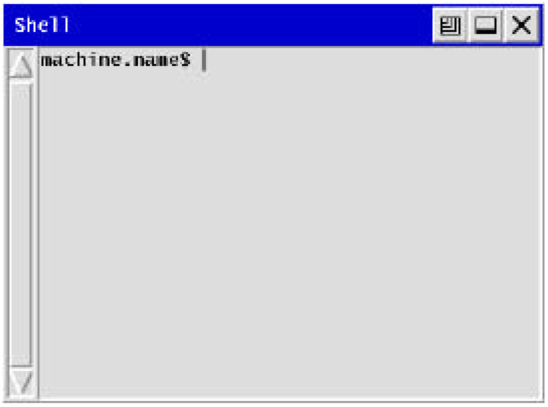
Search For: Inferno

Press **Enter** or **Return** to start the search.

Shell

The Window Manager **Shell** is a command-line interface similar to the **EMU** control console. Select **Inferno>>Shell** to start the **Shell.** Figure 3-41 shows the Shell window.

Figure 3-41 Shell Window



```
machine.name$
```

You type Inferno commands in the **Shell** as you do in the control console. See the *Inferno Command Line Utilities* chapter in this guide for information about the commands available with the Inferno system.