

VM LABS

Puffin5

User Guide and Reference Manual

Version 5.0.2 (Beta)



A debugger for the NUON processor

PRELIMINARY

June 7, 2001



VM Labs, Inc.
520 San Antonio Road
Mountain View, California 94040

phone: 650 917 8050
fax: 650 917 8052

<i>Revision</i>	<i>Date</i>	<i>Author</i>	<i>Changes</i>
0.01	May 1, 2001	Christopher Heiny	Created document.
	May 8, 2001	Svetlana Tuchinsky	Contributed Quick Start tutorial.
	May 18, 2001	Christopher Heiny	More updates.
0.02	June 6, 2001	Christopher Heiny	Added info for 5.0.02 beta.

Copyright © 2001 by VM Labs, Inc., all rights reserved.

Confidential and Proprietary Information of VM Labs, Inc.

Merlin™, Merlin Media Architecture™, the  logo, NUON™ and the  logo are trademarks of VM Labs, Inc.

The information contained in this document is confidential and proprietary to VM Labs, Inc., and is provided pursuant to a non-disclosure agreement between VM Labs, Inc., and the recipient. It may not be distributed or copied in any form whatsoever without the express written permission of VM Labs.

The information in this document is preliminary and subject to change at any time. VM Labs reserves the right to make changes to any information described in this document.

Table of Contents

1-Introduction.....	6
1.1-About this release.....	6
1.2-History of Puffin5.....	6
1.3-About this document.....	7
1.3.1-Organization.....	7
1.3.1.1-Quick Start.....	7
1.3.1.2-User Guide.....	7
1.3.1.3-Reference Manual.....	8
1.3.2-Screenshots.....	8
1.3.3-Notation Conventions.....	8
1.3.3.1-Font conventions.....	8
1.3.3.2-Notes.....	9
1.3.3.3-Beta Gotchas.....	9
2-Puffin5 Quick Start.....	10
2.1-System Requirements.....	10
2.1.1-Hardware.....	10
2.1.2-Software.....	10
2.1.2.1-Libraries.....	10
2.1.2.2-Operating Systems.....	10
2.1.2.2.1-WindowsME.....	11
2.2-Obtaining Puffin5.....	11
2.3-Building your application.....	11
2.3.1-Building the example program.....	11
2.3.1.1-Building the room sample in Linux.....	11
2.3.1.2-Building the room sample Windows.....	12
2.4-Launching Puffin5.....	12
2.4.1-Setting up the MMP.....	13
2.5-Loading an object file	14
2.5.1-Loading the room demo.....	14
2.6-Loading a source file.....	15
2.6.1-Loading main.c from the room demo.....	16
2.7-Looking at the Source View.....	16
2.8-Breakpoints.....	16
2.8.1-Setting a breakpoint in main.c.....	17
2.9-Watching variables.....	17
2.9.1-Watching a variable in the room sample.....	18
2.9.2-Clearing a Watch.....	18
2.10-Running a program.....	19
2.10.1-Running the room sample.....	19
2.11-Exiting Puffin5.....	19
3-Puffin5 Controls.....	20
3.1-The Menu Bar.....	20
3.1.1-The File menu.....	20

3.1.1.1-Load Debug File.....	20
3.1.1.2-Define Search Paths.....	20
3.1.1.3-Exit.....	20
3.1.2-The Window menu.....	21
3.1.3-The Help menu.....	21
3.1.3.1-Puffin5 manual.....	21
3.1.3.2-About Puffin5.....	21
3.2-The Control Panel.....	21
3.2.1-RunAll and StopAll.....	22
3.2.2-The Window Control buttons.....	22
3.2.3-The Reset Button.....	23
3.3-Exit Confirmation Window.....	23
3.4-The Search Path Manager.....	24
3.4.1-The Search Path directory list.....	24
3.4.2-The Directory Name field and the Browse button.....	25
3.4.3-Add.....	25
3.4.4-Remove.....	25
3.4.5-Up.....	25
3.4.6-Down.....	25
3.4.7-Close.....	25
3.5-The Directory Browser.....	25
3.5.1-The Location Bar.....	26
3.5.2-The Directory List.....	26
3.5.3-The Directory Entry Field.....	26
3.5.4-The Control Buttons.....	26
4-The MPE Windows.....	27
4.1-The Menu Bar.....	27
4.1.1-The File Menu.....	28
4.1.1.1-Load object file.....	28
4.1.1.2-Use BIOS loader.....	28
4.1.2-The Debug Menu.....	29
4.1.2.1-Run, Stop, Step In, Step Over.....	29
4.1.2.2-Disassemble at.....	29
4.1.2.3-Set/Clear breakpoint at.....	30
4.1.2.4-Clear all breakpoints.....	30
4.1.2.5-Show breakpoints.....	30
4.1.3-The Font Menu.....	30
4.1.4-The Find Menu.....	30
4.1.5-The View Menu.....	31
4.1.5.1-Show registers/Show variables.....	31
4.1.5.1.1-Lock browser.....	31
4.1.5.2-Show file/Show disassembly.....	31
4.1.5.2.1-Lock source view.....	31
4.1.6-The Watch Menu.....	31
4.2-The Browser View.....	32
4.2.1-The Register Browser.....	32

4.2.2-The Variable Browser.....	32
4.3-The Source File List.....	32
4.3.1-Namespace Collisions.....	32
4.4-The Source View.....	32
4.4.1-Disassembly View.....	33
4.4.2-Text View.....	33
4.5-Breakpoints.....	33
5-Watch Window.....	34
5.1-Clearing a Watch.....	34
5.2-Changing the Format.....	34
5.3-The Watch Dialog.....	35
6-Breakpoint Window.....	36
7-Global Variables Window.....	37
8-Current Puffin5 Bug List.....	38

1- Introduction

1.1- About this release

This version of the Puffin5 User Manual corresponds to the 5.0.02 Beta release of Puffin5. Since Puffin5 is Beta-level software, please be aware of the following:

- ? not all functions are implemented
- ? not all implemented functions are fully implemented
- ? existing features, functions and actions may change between this version of Puffin5 and the final production release of Puffin5.

A summary of the known bugs in this version of Puffin5 may be found in Appendix A.

Due to missing features (in particular, the variable browser and local variable inspection) and odd behavior in certain cases (single step through C/C++ code), it is recommended that this release of Puffin5 be used mostly for assembly language debugging.

Note that this doesn't preclude using this release with C/C++ debugging. It's just that you'll probably find the available functions somewhat limited, eccentric or inaccurate when trying to use them with C/C++.

Bug reports, comments, criticisms, suggestions and donations of Dr Pepper are quite welcome. Please send them to the SDK Release Team (sdk-release@vmlabs.com). Be sure to include the following information:

- ? the OS you are running under (including version number, such as "Windows 2000", "RedHat Linux 7.1"),
- ? the revision/build information for the version of Puffin5 you are using¹
- ? the revision of the VM Labs SDK you are currently using,
- ? and **most** importantly a complete description of how to reproduce the problem, along with any relevant source files, make files, data files, scripts, and so on can be used to reproduce and/or test the scenario(s) you describe.

1.2- History of Puffin5

Puffin5 is the the fifth in a long line of puffins, being the successor to the Puffin2k debugger. Puffin2k was the successor to puffintk. puffintk was descended from puffinw, which in turn was begotten from puffin, the original text based debugger for NUON.

Puffin5 is intended to implement approximately the same feature set as Puffin2k, with the following provisions:

- ? not all Puffin2k features will be implemented;
- ? it is not necessary to implement the features in exactly the same way as Puffin2k did;
- ? all features that are implemented will work.

¹ This information can be found in the "About Puffin5" window. To view this window, go to the Puffin5 Controls window, and select "About Puffin5..." from the "Help" menu.

Veteran Puffin2k users will recognize the last of these as a significant departure from their previous experiences.

Puffin5 is implemented almost entirely in C, using tcl/tk as the GUI widget manager.

The name “Puffin” exists for historical reasons. Jeff Minter, well known for his fondness for certain furry creatures, had named the Merlin assembler “LLAMA”, allegedly an acronym for Low Level Assembler for Merlin Architecture. He then went on to coin the name “ALPacA” for the (yet undeveloped) Abstract Level Packing Assembler. David Betz, maintainer both of XLISP and our debugger, thought this trend was getting out of hand and chose the name “Puffin” for the debugger, by way of contrast.

1.3- About this document

1.3.1- Organization

This document is organized into three major sections:

The User Guide chapters are not present in this revision.

- ? First is a Quick Start chapter, for those who are impatient and need to get started Right Now!
- ? The User Guide chapters chapters, which tell you how to perform typical debugging operations;
- ? The Reference Manual chapters comprise the last major section of the document, which detail the functions of specific Puffin5 components.

OK, there's also the usual collection of stuff at the end, like the latest bug report, an index, and so on and so forth.

1.3.1.1- Quick Start

Chapter 2 is the Quick Start info for Puffin5. This chapter is intended to answer the question:

How do I get started with Puffin5?

It tells you where to get Puffin5, what equipment and software you will require, how to start Puffin5, and how to accomplish some common debugging operations using Puffin5 (with examples based on the `sample/mgl/room` sample program).

1.3.1.2- User Guide

Chapters TBD through TBD comprise the Puffin5 User Guide. These chapters are intended to answer the question:

How do I do such-and-such an operation in Puffin5?

They are more detailed than the examples in the Quick Start, as well as covering more complex or more obscure operations.

These chapters are not present in this document revision, mainly because they haven't been written yet.

1.3.1.3- Reference Manual

The remainder of the main document body is a reference manual is organized around the major windows of the Puffin5 GUI. These chapters are intended to answer the question:

What does this button do?

There are specific chapters for each of the following:

- ? Puffin5 Controls
- ? MPE Windows
- ? Watch Window
- ? Breakpoint Window
- ? Global Variables Window

Dialogs relating to a specific window are covered in that window's chapter. For example, the dialog windows relating to setting variable watches are covered in the watch window chapter; the dialog windows relating to setting breakpoints are covered in the breakpoint window chapter.

Beta Gotcha: These chapters are not completely written at this time. They may be fragmentary or missing entirely.

1.3.2- Screenshots

The screenshots incorporated in this document were captured on a RedHat 7.0 Linux system running KDE 2.1.1. Although the window decorations and font sizes will vary depending on what version of which OS you are using, and your particular window manager/theme/scheme, the functionality and presentation of information will be consist between environments.

HOWEVER, since Puffin5 is still under development, the features/functions described in this manual may change with a future release. Please make sure you have the version of this manual corresponding to the release of Puffin5 you are using. This particular revision corresponds to Puffin5 version 5.0.02 Beta.

1.3.3- Notation Conventions

The following notational conventions are used throughout this document.

1.3.3.1- Font conventions

Narrative and descriptive text is presented in Times New Roman.

Filenames are presented in `Courier`, a fixed-pitch serif font. For example,

```
C:\VMLabs\bin\windows\  
~/.puffin5.cfg
```

Source code is presented in `Lucida Console`, a fixed-pitch sans-serif font. For example:

```
while ( i < max ) {  
    copy_line(i, from, to);  
    i++;  
}
```

Tty/terminal/shell/command prompt output is presented in *Courier*, with user input presented in **Courier Bold**. For example:

```
[cheiny@truck.vmlabs.com ~/proj]% cvs c
```

In all cases, values that are supplied by the user or that vary depending on context are presented in italics. For example:

```
VM Labs SDK release 0.8n of date  
x = hash(your-name-string);  
C:\>cd project-dir
```

1.3.3.2- Notes

The raccoon feeder feature is not implemented in version 5.0.02 Beta.

The automated dog walker is a new feature in version 5.0.02 Beta.

Features and functions that are not implemented are denoted by a red box in the left margin of the page, with the text "This feature is not implemented" (or phrasing to that effect). See the example to the left of this paragraph.

Features and functions that are new with the current release of Puffin5, or that have changed significantly in some way since the previous release of Puffin5, are denoted by a green box in the left margin of the page, with the text "This feature is new [or substantially updated, or whatever] with this release" (or phrasing to that effect). See the example to the left of this paragraph.

1.3.3.3- Beta Gotchas

Various malfunctions, misfires, hiccups and eccentricities with this release are highlighted as *Beta Gotchas*. Beta Gotchas are highlighted by a red surround, and are listed in the index at the end of this document. The first gotcha is right here:

Beta Gotcha: like Puffin5 itself, this manual is under development. At this point it's pretty heavy on the "Reference Manual" part, and very light on the "User Guide" – that is, it provides a map of the Puffin5 GUI and functionality (for example, this button does that thing), but is short on the "How To" sort of thing (for example, "How do I run my program?", "how do I set a breakpoint"). This will be remedied in future updates.

2- Puffin5 Quick Start

This chapter tells you how to get Puffin5 up and running, and how to perform some basic debug operations using Puffin5.

2.1- System Requirements

The following system requirements are subject to revision with formal releases of Puffin5.

2.1.1- Hardware

These are the suggested PC system parameters for using Puffin5.

Processor: Intel Pentium II, Pentium III, or Celeron. AMD K6-II, Athlon, Duron. Minimum 500 MhZ.

Memory: 64MB.

Disk space: 300MB (includes space for VM Labs SDK).

Network: 10baseT or 10/100baseT NIC

Note: Puffin5 will probably work with lower horsepower processors and less memory. These just represent the minimum systems we've tried it on.

2.1.2- Software

Puffin5 is shipped with the VM Labs SDK. The SDK should contain all software components not otherwise available in normal OS distributions that are required to run Puffin5.

2.1.2.1- Libraries

Puffin5 requires software installations for the following components:

pthreads - the pthreads packages is typically installed on all Linux systems. For Windows users, the necessary `pthreadVSE.dll` is included in the VM Labs SDK, in the same directory as Puffin5. You will need to ensure that your `PATH` variable includes this directory.

Tcl/Tk - Puffin5 requires Tcl/Tk version 8.3.3. If this version is not already installed on your machine, you can find it in the `/bundled` directory of the VM Labs SDK.²

2.1.2.2- Operating Systems

Puffin5 is supported on Windows 2000 and the following Linux distributions:

? RedHat 6.2, 7.0, 7.1

? Suse 7.0, 7.1

Current testing resources do not permit us to validate Puffin5 against a wider variety of platforms.

² WindowsME users, please see the note regarding Tcl/Tk on page 11.

2.1.2.2.1- WindowsME

WindowsME is not a currently supported OS for Puffin5. Problems encountered while running under WindowsME must also be reproducible under Windows2000.

You will need to obtain and install a more recent version of Tcl/Tk if you wish to run Puffin5 under WindowsME. Visit <http://www.scriptics.com/> and download version 8.4.a2. Note that this is a Beta version of Tcl/Tk, and may be a bit flaky. Unfortunately, important features of Puffin5 will not work on WindowsME with earlier versions of Tcl/Tk.

2.2- Obtaining Puffin5

Starting with VM Labs SDK 0.86 (May 10, 2001), Puffin5 is shipped as part of the VM Labs SDK - if you have the VM Labs SDK installed correctly on your machine, you should be able to access Puffin5 without difficulty. After ensuring that the correct software libraries are installed (see previous section), no additional setup should be required to begin using Puffin5.

2.3- Building your application

No special tools are required to build your application for use with Puffin5.

In order to ensure that line number and symbol definition information is included in your object file for use by Puffin5, you will need to compile your C/C++ programs with the `-g` switch. See the command lines in the following section for examples. This information is included by default by Llama.

2.3.1- Building the example program

The examples presented in the chapter are based on the room mgl sample. This is in the `sample/mgl/room` directory of the VM Labs SDK directory. You will need to build this sample to execute the examples shown in the following pages.

2.3.1.1- Building the room sample in Linux

Open your favorite command prompt, and type:

```
cp -pr $VMLABS/sample/mgl/room room
cd room
gmake
```

The result should look something like this:

```
[chris@truck chris]$ cp -pr $VMLABS/sample/mgl/room .
[chris@truck chris]$ cd room
[chris@truck room]$ gmake
mgcc -c -g -I/home/chris/vmlabs-local/include -I/usr/local/vmlabs/include -Wall -O2 -mrom
-mpe3 -o main.o main.c
main.c: In function `main':
main.c:102: warning: implicit declaration of function `DebugWS'
llama -I/home/chris/vmlabs-local/include -I/usr/local/vmlabs/include -fcoff -nologo -c -o
room.o room.s
mgcc -c -g -I/home/chris/vmlabs-local/include -I/usr/local/vmlabs/include -Wall -O2 -mrom
-mpe3 -o drawroom.o drawroom.c
mgcc -L/home/chris/vmlabs-local/lib -L/usr/local/vmlabs/lib -mrom -mpe3 main.o room.o
drawroom.o -lmgl -ljpeg -lmml2d -lmutil -lm -o room.cof
[chris@truck room]$
```

Sample 1 - Building the room sample in Linux.

2.3.1.2- Building the room sample Windows

Open a DOS Prompt window (Windows 98) or a Command Prompt window (Windows 2000).

Type the following commands:

```
cd %VMLABS%\sample\mgl\room
gmake
```

You should see something like this:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.
C:\>cd %VMLABS%\sample\mgl\room
C:\VMLabs\sample\mgl\room>gmake
mgcc -c -g -IC:/SDK-Local/include -IC:\VMLABS/include -Wall -O2 -mrom -mpe3 -o main.o
main.c
main.c: In function `main':
main.c:102: warning: implicit declaration of function `DebugWS'
llama -IC:/SDK-Local/include -IC:\VMLABS/include -fcoff -nologo -c -o room.o room.s
mgcc -c -g -IC:/SDK-Local/include -IC:\VMLABS/include -Wall -O2 -mrom -mpe3 -o drawroom.o
drawroom.c
mgcc -LC:/SDK-Local/lib -LC:VMLABS/lib -mrom -mpe3 main.o room.o drawroom.o -lmgl -ljpeg
-lmml2d -lmutil -lm -o room.cof
C:\VMLabs\sample\mgl\room>
```

Sample 2 - Building the room sample under Windows.

2.4- Launching Puffin5

Type `puffin5` at a command prompt.

```
[chris@truck.vmlabs.com room ] puffin5
NUON IP address is 10.1.47.168
Opening connection to 10.1.47.168... Checking stub version.
Success.
This is a test error
Creating object for MPE 0.
Creating object for MPE 1.
Creating object for MPE 2.
Creating object for MPE 3.
Creating watch window...
Created watch window successfully.
Creating watch dialog window.
Created watch dialog successfully.
Creating globalC window.
  Created main window .globalC
  Frames... done.
Done with globalC window.
%
```

Sample 3 - Launching Puffin5 (Linux shown, Windows will be similar)

You should get splash screen with a picture of a puffin and version information about Puffin5, followed by a bunch of windows popping up. Puffin5 initialization is complete when the "bird head" disappears, leaving you with a stack of windows with the Puffin5 Controls window on top.

Beta Gotcha: Linux users will notice that some debug information is printed to their terminal window. Also, the files `/tmp/pg_log.txt` and `/tmp/pg_errlog.txt` will be created.

Windows users will find four files in their working directory (`puffin5_log.txt`, `puffin5_errlog.txt`, `pg_log.txt` and `pg_cblog.txt`). The first two of these correspond to redirection of `stdout` and `stderr`, respectively; the last two are the same as the Linux `/tmp/pg_log.txt` and `/tmp/pg_cblog.txt`.

These files contain debugging information used by the Puffin5 developers. You can delete them without effecting the behavior of Puffin5. Future releases of Puffin5 will eliminate these files, as well as the Linux verbosity.

Beta Gotcha: In addition to the suite of windows described in this manual, you will also get a debug information window. This window can be closed without affecting the behavior of Puffin5. It will be eliminated in a future release.

Note that your target development system must be turned on for Puffin5 to function. If it is not, you will see the error message dialog shown at right.

After you click OK, Puffin5 will exit. Before attempting to restart it, check the following:

- ? make sure that your target dev system is powered on
- ? make sure that your target dev system is connected to the network
- ? make sure that your workstation is connected to the network
- ? make sure that your `MD_PORT` environment variable is set to the IP address of your dev system
- ? make sure that your dev system's IP address is what you think it should be.



Illustration 1 - What happens when Puffin5 can't find your dev system.

2.4.1- Setting up the MMP

Beta Gotcha: Automatic processor initialization is not implemented in this version of Puffin5. This means that you will need to perform the following steps to let Puffin5 determine the state of the MMP.

In the Puffin5 Controls window do the following:

1. Hit Reset. Wait for the dev system to reset
2. Hit RunAll. The MPE1 & MPE2 windows may pop to the top of the window stack.

At this point, Puffin5 is aware of the MMP state, and you can proceed to load, execute, and

debug your program.

2.5- Loading an object file

Perform the following steps to load an object file.

1. Open the MPE window for the MPE you want to load your object file onto.
2. If you are running on MPE3, make sure the Use BIOS loader option in the File menu is set appropriately.
3. From the File menu, select Load object file...
4. When the file chooser pops up, select the `.cof` file you want to load and click Open.

Wait for the loading process to complete. A progress bar will keep you informed of the progress. When it is done, the progress bar will disappear.

2.5.1- Loading the room demo

Perform the following steps to load the room demo:

1. Open the MPE3 window.
2. Ensure that the Use BIOS loader option in the File menu is selected (red box in Linux, checkmark in Windows).
3. Do File->Load Object file... A file chooser window will pop up.
4. Select `room.cof` and click Open.

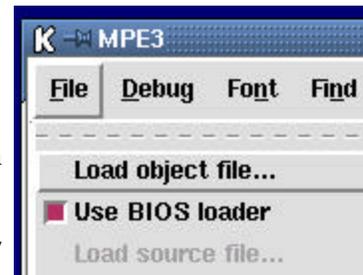


Illustration 2 - Use BIOS Loader selected

When loading is complete, the MPE3 source view will contain the disassembly code for the stopped MPE, with address where the MPE is stopped highlighted in red. The window will look something like the one shown in Illustration 3 on page 15.

Note that if you have started Puffin5 in a directory other than the `sample/mgl/room`, you can navigate to that directory in the file chooser window that popped up in step 3.

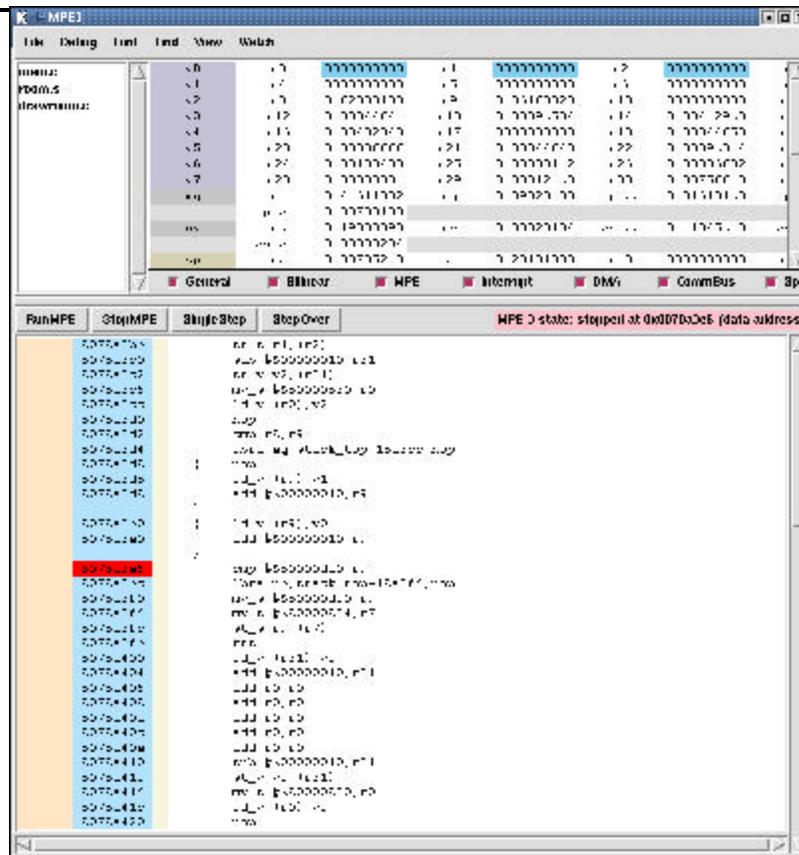


Illustration 3 - MPE3 window after loading *room.cof*

2.6- Loading a source file

In the previous sample, you'll notice a box in the upper left hand portion of the MPE window which contains a list of source files. When a *.cof* file is loaded, Puffin5 searches for the source files that were used to build that *.cof* file. It looks for these files in the following directories, in the given order:

- ? the directory where the *.cof* file was found;
- ? the working directory you were in when Puffin5 was launched;
- ? any directories you have specified with the Define Search Path... command.

Define Search Path... is now implemented in version 5.0.02 Beta.

Beta Gotcha: Note that in Puffin5 version 5.0.02 Beta, search path capability is not completely implemented. However, extensive progress has been made with the search path capabilities. Refer to the Search Path Manager section on page 24 for more information.

Any source files that Puffin5 finds are listed in this box. If you click on a filename in this list, the filename will be highlighted and the file will be loaded in to the source view window.

2.6.1- Loading main.c from the room demo

There should be three files in the source file list: main.c, room.s and drawroom.c. Click on main.c. The list entry will be highlighted, and main.c will be loaded into the source view.



Illustration 4 - main.c selected.

2.7- Looking at the Source View

There are two different Source Views: the disassembly view and the file view. These are described in detail in the MPE Window chapter, so we'll just cover the basics.

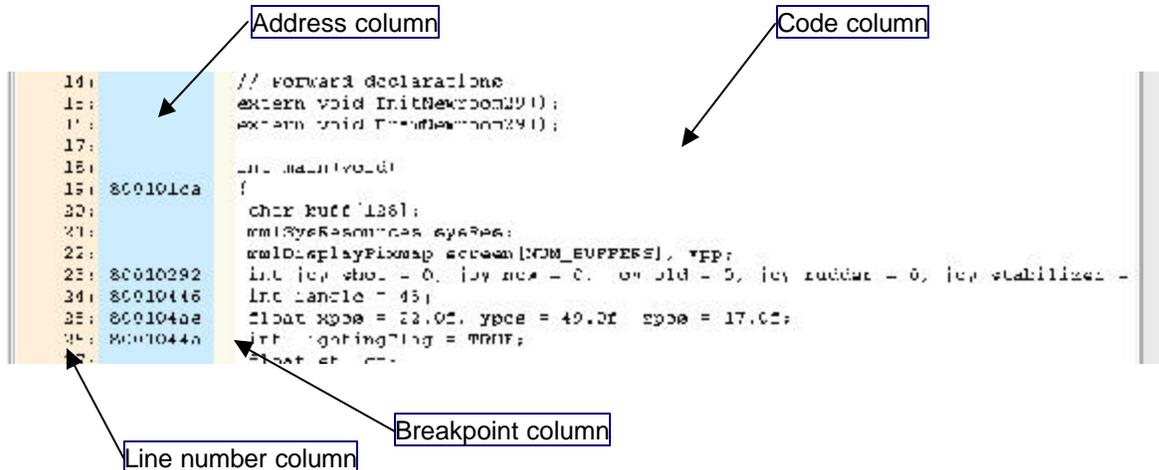


Illustration 5 - The parts of the source view

Both views are divided into four irregular columns. From left to right, these are

- ? line number
- ? address
- ? breakpoint
- ? code

The line number column gives the line number of the source file you are viewing. This column is blank in the disassembly view.

The address column gives the code address (if applicable) for that line. This line normally has a sky blue background, but will be highlighted in red if the MPE is halted at that address.

The breakpoint column is used for setting/clearing breakpoints, and for indicating set breakpoints.

The code column displays your source file or the disassembled code, depending on the selected view.

2.8- Breakpoints

There are a lot of other ways to manage breakpoints in Puffin5. These are detailed section 4.5, starting on page 33.

One way to set and clear breakpoints is to use the breakpoint column of the source view. Moving the cursor over the breakpoint column changes it to a circle shape. If a breakpoint can be set on a given line, the circle will turn red (it is normally black), indicating you can set the

breakpoint by clicking.. If there is already a breakpoint set on that line, the circle will turn green, indicating that you can clear the breakpoint by clicking.

When a breakpoint is set, a red blob (it's stop sign shaped if you look close) will appear in the breakpoint column. When it is cleared, the blob is removed.

*Illustration 6 -
The breakpoint
blob*

Beta Gotcha: Under Windows, the cursor changes shape to a circle whenever it is in the breakpoint column, and does not change color for setting or clearing breakpoints.

When you have a lot of breakpoints to manage, or are managing breakpoints on more than one MPE or source file, the Breakpoints window provides a convenient way to view summary information about your breakpoints. You can bring it up by clicking on the Breakpoints button in the Puffin5 Control window.

2.8.1- Setting a breakpoint in main.c

Assuming `main.c` is still loaded in the MPE3 window, perform the following steps:

1. Move the mouse over the breakpoint column for line 130. It should turn into a red circle.
2. Right click to set the breakpoint. A red blob should appear on the line, indicating the breakpoint has been set.

```

126: 800106a2      if (lightingFlag) {
127: 800106b2      glEnable(GL_LIGHTING);
128: 800106c6      glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
129: 800106dc      } else {
130: 800106e0      glDisable(GL_LIGHTING);
131: 800106f6      glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
132:              }
133:              }

```

Illustration 7 - A breakpoint at line 130 of main.c

3. Bring up the Puffin5 Control window.
4. Click on the Breakpoints button. The Breakpoints window should pop up, showing a breakpoint on MPE3 set at line 130 of `main.c`.

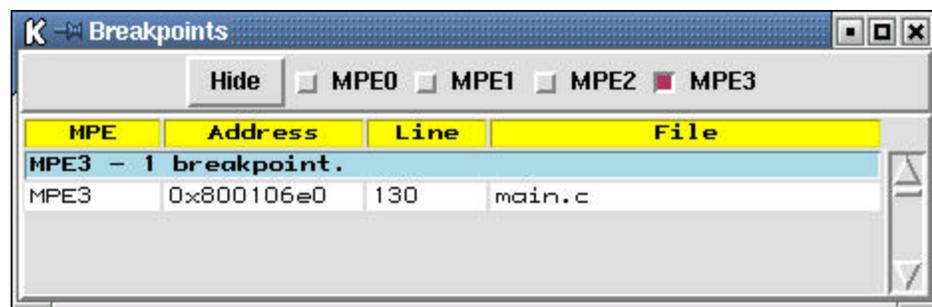


Illustration 8 - Breakpoints window example.

2.9- Watching variables

You can set a watch on a given memory location, specified either by global variable name or by

address. To set a watch, perform the following steps.

1. Move the mouse into the source view and right click to bring up the Source Menu.
2. Select the Watch command. The watch dialog will pop up.
3. Configure the watch as needed. In most cases, the default settings will suffice. See section 5.3 on page 35 for more details on the various options.
4. Click OK.



Illustration 9- The source menu

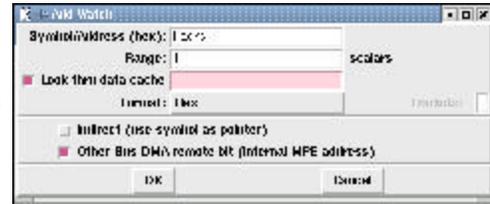


Illustration 10 - The Watch dialog window

You can view the watches you have set by bringing up the Watch window.

1. Got to the Puffin5 Controls window.
2. Click on Watch to bring up the Watch window.

2.9.1- Watching a variable in the room sample

Beta Gotcha: This sample is incomplete. It requires checking, additional text, formatting and maybe screenshots.

1. Go to the file list and choose `drawroom.c`
2. After `drawroom.c` loads, go to line 7 and select "`tracksj_pg_start`"
3. Right click to bring up the file ops menu and select Watch. The watch dialog box should pop up.
4. Click OK.
5. Select "`tracks`" on line 6.
6. Right click to bring up the file ops menu and select Watch. The watch dialog box should pop up.
7. Select the "Indirect" button, then click OK. If it complains about not being able to look up the value, click OK.

2.9.2- Clearing a Watch

You can clear an active variable watch by the following steps.

1. Right click over any field of the variable's display in the Watch window. A menu will pop up (the exact contents of this menu varies depending on the field. The example at right shows it for the Variable field).
2. Select Clear watch



Illustration 11 - Clearing a watch

The watch will be removed from the Watch display, and no longer watched.

2.10- Running a program

Running a program is fairly simple: just click the Run button for each MPE that you want to start.

In some cases, it may be more convenient to click the RunAll button in the Puffin5 Control window.

2.10.1- Running the room sample

Presuming you've got the room.cof program loaded, and the breakpoint and watch set as previously described, perform the following steps.

1. Click RunMPE in the MPE3 window. The room demo should start running.
2. Click the Z button (this is the one that toggles lighting. On my joystick (an HPI), this is the lefthand of the two red buttons labeled "-- N --".)

A breakpoint should happen: `main.c` will be loaded into the source window, the address of line 130 will be highlighted in red, and values in the Watch window and variable browser will be updated.

From here, you can set and clear breakpoints, add more watches, run, stop or step into.

2.11- Exiting Puffin5

There are three ways to exit Puffin5.

- ? Push CONTROL-Q in any Puffin5 window;
- ? Select Exit from the File menu of the Puffin5 Controls window.
- ? Close the control window using the window manager. Under most window managers, this is done by clicking an X shaped icon in the upper right corner of the window.³

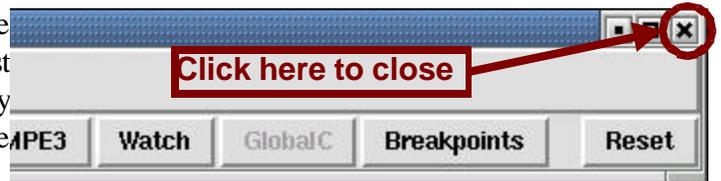


Illustration 12 - Where to click to close the window.

When you exit Puffin5, you will be presented with a confirmation window. Click the Confirm button to confirm that you really want to exit, and Puffin5 will exit. Click the Cancel button if you change your mind and want to return to Puffin5.



Illustration 13 - Exit confirmation window.

See section 3.3 on page 23 for more information on the exit confirmation window.

³ Actually, just about every window manager implements half a dozen or more ways to close a window. They are all effectively equivalent to clicking this button.

3- Puffin5 Controls

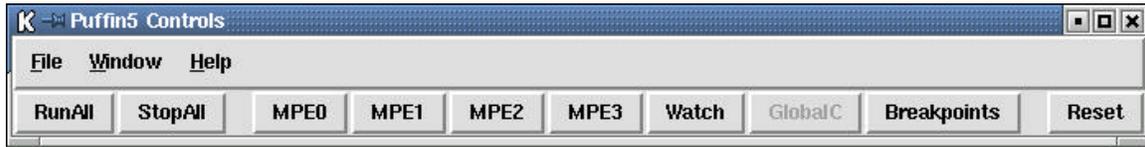


Illustration 14 - Puffin5 Controls window

The Puffin5 Controls window has a main menu bar and a control panel.

3.1- The Menu Bar

There are three menus in the menu bar:

- ? the File menu, which is used for overall control of Puffin5;
- ? the Window menu, which is used to control Puffin5's major windows; and
- ? the Help menu, which can be used to call up the Puffin5 User Manual or information about Puffin5 itself.

3.1.1- The File menu



The File menu has three entries.

- ? Load Debug File...
- ? Define Search Paths...
- ? Exit

These commands can be used to control the overall behavior of Puffin5.

3.1.1.1- Load Debug File...

Use this command to load and run a Bob debugging script. A file chooser window will pop up where you can select the script file to be run, and feedback from the

script will be displayed in the Puffin5 Console window.

See Chapter TBD for more information about the Bob scripting language.

3.1.1.2- Define Search Paths...

This command will bring up a window that you can use to configure the directories where Puffin5 will look for source files, scripts, executables, and other files it needs.

Refer to the Search Path Manager section on page 24 for more information.

3.1.1.3- Exit

This command will terminate your debug session, close all of Puffin5's windows, and exit Puffin5.

Load Debug File... is not implemented in version 5.0.02 Beta.

Define Search Path... is now implemented in version 5.0.02 Beta.

You can invoke this command from any Puffin5 window by pressing CONTROL-Q.

3.1.2- The Window menu

The GlobalC window is not implemented in this version.

The Window menu can be used to pop up any of the other major Puffin5 windows. Just click on the menu entry for the window you want to view.

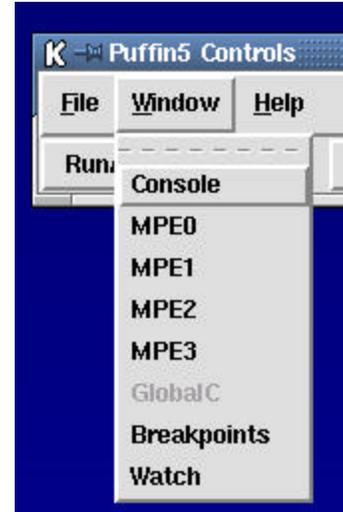


Illustration 16 - Puffin5 Controls: the Window menu

3.1.3- The Help menu

There are two entries in the help menu.

3.1.3.1- Puffin5 manual...

Online help is not implemented in version 5.0.02 Beta.

Clicking on this entry will bring up an online version of this manual in a web browser.



Illustration 17 - Puffin 5 Controls: the Help menu

3.1.3.2- About Puffin5...

Clicking on this entry will bring up a "splash window" with information about Puffin5. This is where you can find build and version number information for Puffin5.

Click on the **OK** button to dismiss this window.

See Illustration 18 for screenshot of this window.

3.2- The Control Panel

There are three groups of buttons in the control panel:

- ? RunAll/StopAll, which are shortcuts for controlling the MPEs run state;
- ? a collection of buttons providing quick access to the other major windows of Puffin5; and

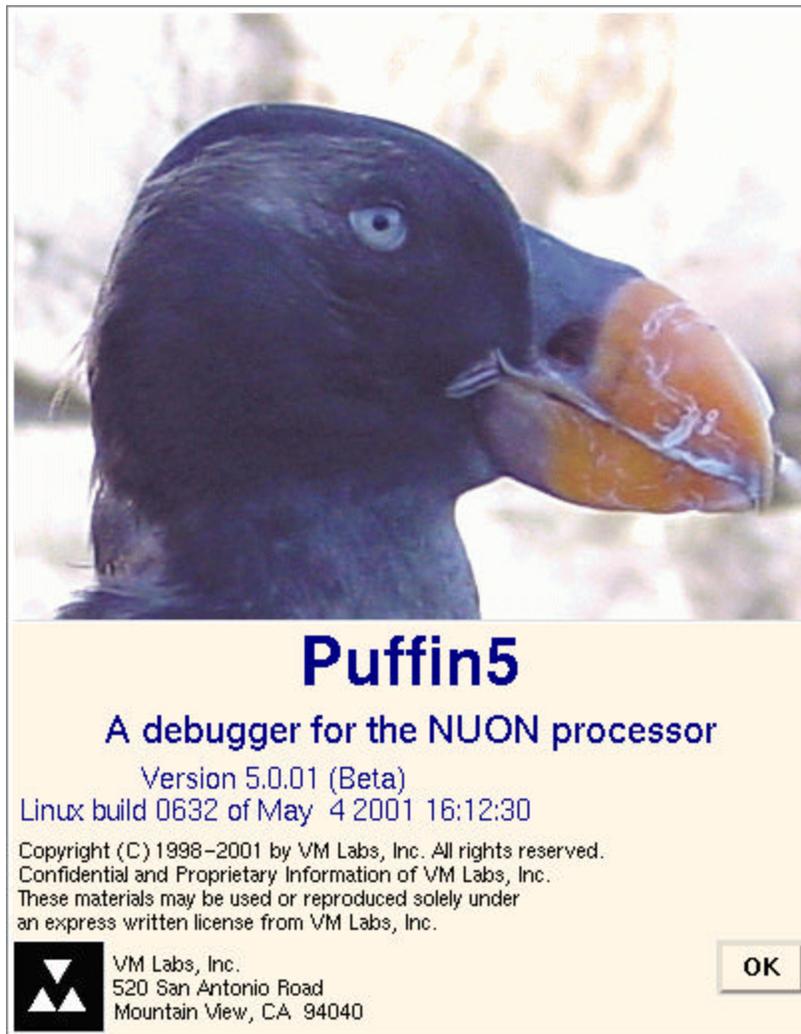


Illustration 18 - Puffin5 Splash window

? Reset, which resets the MMP.

3.2.1- RunAll and StopAll



Illustration 19 - Puffin5 Controls: RunAll and StopAll

The RunAll and StopAll buttons are fairly self explanatory.

Pushing RunAll causes each MPE in turn to be placed into run mode. The order is MPE0, MPE1, MPE2, and MPE3. If your code requires the MPEs to be started in a different order, or request to start only selected MPEs, you should use the Run button in the appropriate MPE

window(s).

Pushing StopAll causes each running MPE in turn to be placed into stop mode. The order is MPE0, MPE1, MPE2, and MPE3. If your code requires that you stop the MPEs in a different order, or if you need to stop only a particular MPE, you should use the Stop button in the appropriate MPE window(s).

3.2.2- The Window Control buttons



Illustration 20 - Puffin5 Controls: window control buttons

These buttons will bring up the specified Puffin5 window:

- ? MPE0, MPE1, MPE2, and MPE3 buttons will bring up the window for the appropriate MPE;
- ? Watch will bring up the Watch window;
- ? GlobalC will bring up the global C variable window;
- ? Breakpoints will bring up the Breakpoint window; and
- ? Console will open the Puffin5 Console window.

The GlobalC window is not implemented in this version. Its button is greyed out.

The Console button is not implemented in this version.

Please refer to the chapter details for the particular window you are interested in for more information.

3.2.3- The Reset Button



Illustration 21 - Puffin5 Controls: the Reset button

The Reset button resets the MMP, clears all breakpoints, watches, loaded source windows, and so on.

Beta Gotcha: Reset is not completely implemented in this version of Puffin5. If you want to reset the processor and reload your `.cof` files, you should instead exit and restart Puffin5, and then hit the Reset button.

Beta Gotcha: Reset does not completely initialize the internal representation of the processor's state in this release of Puffin5. It is recommended that you hit the RunAll button after hitting Reset - this will cause Puffin5 to load the current processor state.

3.3- Exit Confirmation Window

The Exist Confirmation Window is a new feature in Puffin5 5.0.02 Beta.

When you exit Puffin5 through one of the normal methods (CONTROL-Q, using the Exit button in the File menu, or closing the Controls window via your window manager), you will be presented with the Exit Confirmation Window.

This window has two buttons:

- ? Confirm - click this to complete exit.
- ? Cancel - click this if you've changed your mind and want to return to Puffin5.

There are also two options available in this window:

These options are not implemented in this version. The options are greyed out.

- ? Don't show this dialog again. If you select this option, the Exit Confirmation Window will not be shown again. You can change this preference back in the Preferences Editor.
- ? Save settings and preferences. Your current search path, variable display preferences, font preferences, and so on will be remembered the

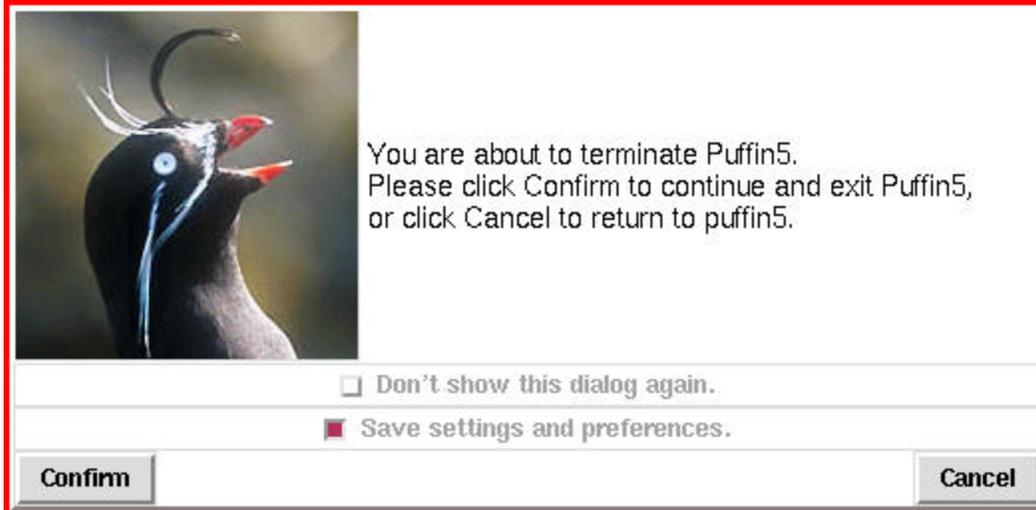


Illustration 22 - Puffin5 Exit Confirmation Window

next time you start Puffin5. This option is selected by default.

Beta Gotcha: These options are not implemented in this release of Puffin5 and are greyed out.

3.4- The Search Path Manager

You can use the Search Path Manager to tell Puffin5 where to look for source files and other files it might need. Click on the Define Search Paths... button in the File menu to bring up the Search Path Manager.

This window contains several elements:

- ? A scrollable list of the directories in the search path.
- ? The directory name field, with a Browse... button.
- ? The following control buttons:

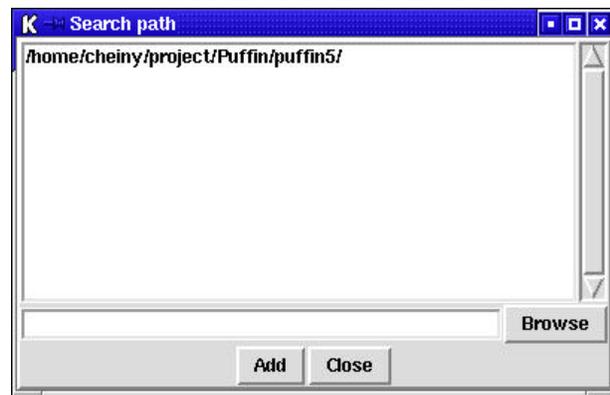


Illustration 23 - The Search Path Manager window

The Remove, Up and Down buttons are not implemented in this version. They are not displayed in the Search Path Manager window.

- ? Add and Remove, to put new directories into the path or remove an existing directory from the path.
- ? Up and Down, to change a directory's position within the search path.
- ? Close, to close the Search Path Window.

3.4.1- The Search Path directory list

This is a scrollable list of all the directories in the search path. When Puffin5 is searching for a file, it starts with the first directory in the search path, and continues examining the directories in order until it finds the file it is after or there are no more directories in the search path.

The list presents these files in order, with the first file at the top of the list. You can change a file's position in the search order by selecting the directory you want to move, and then using the Up or Down button to move it about within the list.

It is important to note that changes you make to the Search Path take effect immediately.

3.4.2- The Directory Name field and the Browse button

You can type the name of a directory you want to manipulate here, and then hit the RETURN key.

If the directory you want to work with is already on the list, you can select it directly from the list. Its name will be entered into this field automatically.

Alternatively, you can use the Browse... button to pop up the Directory Browser to choose a directory. Once you have chosen a directory and closed the browser, the directory name will be entered in this field.

The highlight/jump function is not implemented in this version.

If the directory you type or choose is already in the path, it will be highlighted and the list will be scrolled so it is displayed.

3.4.3- Add

Use this button to add a directory to the search path. The directory will be added to the bottom of the search path.

If the directory is already on the path, this button has no effect.

3.4.4- Remove

The Remove button is not implemented in this version.

Use this button to remove the selected directory from the list. If there is no directory selected, this button will be greyed out.

3.4.5- Up

The Up button is not implemented in this version.

Use this button to move the selected directory toward the top of the list. If there is no directory selected, or if the selected directory is already at the top of the list, this button will be greyed out.

3.4.6- Down

The Down button is not implemented in this version.

Use this button to move the selected directory toward the bottom of the list. If there is no directory selected, or if the selected directory is already at the bottom of the list, this button will be greyed out.

3.4.7- Close

Click this button to close the Search Path Manager window.

3.5- The Directory Browser

The Directory Browser is used to select directories for the Search Path Manager. It is accessed only by clicking the Browse... button in the Search Path Manager Window.

Beta Gotcha: The Directory Browser window operation will change significantly in an upcoming release of Puffin5.

There are 4 parts to the Directory Browser window:

- ? The location bar, which tells you what directory you are currently browsing in and allows you to change that directory.
- ? The directory list, which is a list of all the directories contained by the current directory.
- ? The directory entry field, which contains the name of the directory that we are currently interested in.
- ? The control buttons, where you can confirm or cancel your selection.

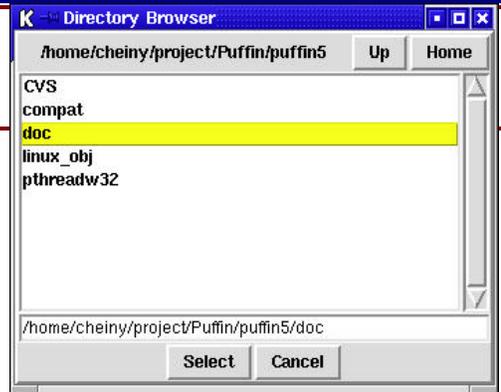


Illustration 24 - The Directory Browser

3.5.1- The Location Bar

This bar tells you which directory you are browsing in. Hit the Up button to move to its parent directory.

The Home button is not completely implemented in this version.

Hit the Home button to move to your home directory, as defined by the HOME environment variable.

3.5.2- The Directory List

This is a list of all the directories that are contained in the current directory.

Click on an entry in this list to select the directory.

Hit RETURN to browse into the selected directory.

The up/down arrow buttons are not completely implemented in this version.

Use the keyboard up arrow or down arrow keys to move the selection up or down in the list.

3.5.3- The Directory Entry Field

Type a directory name here, and hit RETURN to browse the directories within that directory.

This field will be automatically filled in when you select a directory from the directory list.

3.5.4- The Control Buttons

Click Select to close the browser and enter your selected directory in the Directory Field of the Search Path Manager.

Click Cancel to close the browser without selecting a directory.

4- The MPE Windows

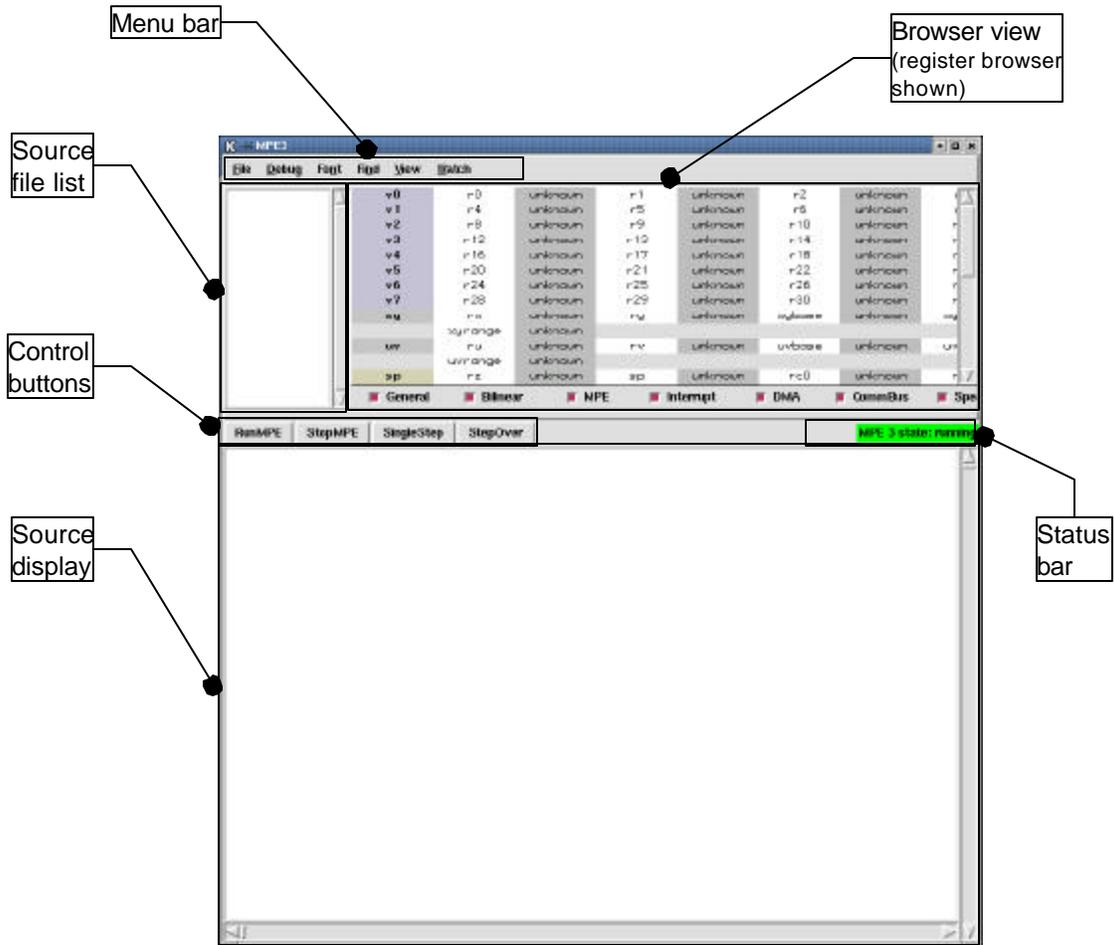


Illustration 25 - Parts of the MPE Window

Each MPE has its own window which is used to control the MPE and display information about the MPE and the file(s) loaded onto it.

There are 6 major parts to the MPE Window, as shown in the illustration above. These are:

- ? the menu bar;
- ? the browser view;
- ? the source file list;
- ? the control buttons;
- ? the status bar;
- ? the source view.

Each of these is described in greater detail in the following pages.

4.1- The Menu Bar

There are six menus in the MPE Window Menu Bar:

- ? File,
- ? Debug,
- ? Font,
- ? Find,
- ? View,
- ? Watch.



Illustration 26 - Puffin5 MPE window: menu bar

4.1.1- The File Menu

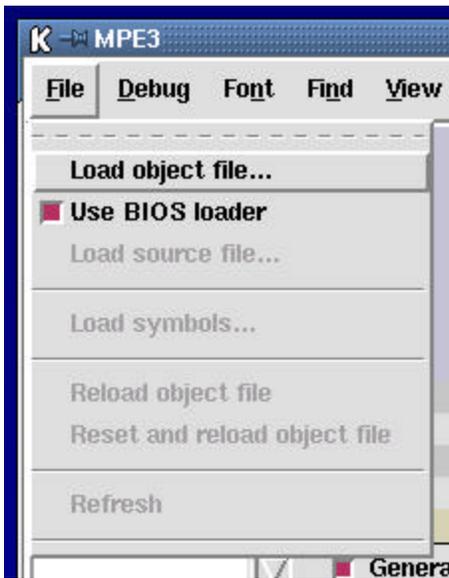


Illustration 27- Puffin5 MPE window: the file menu

There are 7 controls provided by the MPE file menu:

- ? Load object file...
- ? Use BIOS Load (MPE3 only)
- ? Load source file...
- ? Load symbols...
- ? Reload object file
- ? Reset and reload object file
- ? Refresh

Beta Gotcha: Some of these functions are not available in this version, and are greyed out on the menu.

4.1.1.1- Load object file...

Selecting this command will pop up a dialog box where you can select an object file to be loaded into the MPE.

The only object file format recognized by Puffin5 is `.cof`.

Beta Gotcha: This version of Puffin5 does not deal well with searching alternate directories for files yet. If you encounter problems with load files, resolving symbols or locating source files, it is recommended that you start Puffin5 from the directory where your object file and source files are located.

4.1.1.2- Use BIOS loader

This option applies only to MPE3. When it is selected, files will be loaded onto MPE3 using a special "helper app" known as the BIOS loader. The BIOS loader greatly speeds the process of transferring programs onto MPE3.

The BIOS loader is a kamikaze. Once its mission of loading your file is completed, it exits and is removed from memory, probably overwritten by your code. You must reset MPE 3 to reload the BIOS loader.

Note: The BIOS loader is only running if MPE3 has been restarted from the boot rom. If it is

not running when you attempt to load, Puffin5 will fail to load your file. Things that may cause the BIOS loader to not be running are:

- ? you have not reset MPE3 since you last loaded a program;
- ? MPE3 is halted from a software failure;
- ? MPE3 is halted because you pushed the Stop button; or
- ? You have a non-debug BIOS image loaded on your system.



Illustration 28 - The red progress bar of doom.

Regardless of the cause, if the BIOS loader is not running, Puffin5 will fail in its attempts to contact it. The progress meter will hang at 0% for while before flashing to red, after which an error dialog box will pop-up, similar to the one shown here in TBD.

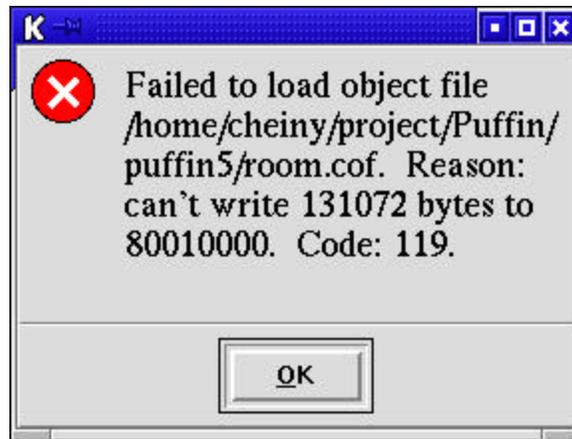


Illustration 29 - BIOS Loader failure error dialog.

4.1.2- The Debug Menu

Beta Gotcha: The Debug Menu is not completely implemented in this release of Puffin5. Layout and functionality is subject to change in a future release.

The Debug Menu allows you to invoke a number of Puffin5's MPE debug items.

4.1.2.1- Run, Stop, Step In, Step Over

These items have the same effect as the MPE control buttons with the same name. See section TBD for more information.

4.1.2.2- Disassemble at...

This item will pop up a dialog window that lets you specify an address and a number of instruction packets to that will be disassembled. The requested instructions are disassembled and displayed in the source view.

Beta Gotcha: This item may move to the View Menu at some later date.



Illustration 30 - The MPE Window Debug menu

4.1.2.3- Set/Clear breakpoint at...

The Set/Clear breakpoint item is not implemented in this version.

Pops up a dialog box that provides yet another way to set or clear a breakpoint. You specify an address, and click Set to set the breakpoint or Clear to clear the breakpoint, or Cancel to just forget the whole thing.

4.1.2.4- Clear all breakpoints

Clicking on the Clear all breakpoints item will remove all breakpoints associated with this MPE.

4.1.2.5- Show breakpoints

Clicking on the Show breakpoints item will pop up the Breakpoint Window (see chapter 6 on page 36), with the break point display for this particular MPE activated.

4.1.3- The Font Menu

The MPE Font Menu allows you to choose the preferred font family and size that will be used.

Note that some parts of the Puffin5 display are always presented in a fixed pitch font. These include the presentation of addresses, data values, and the source view window. These parts of the display will respond to your font size selection, but not to changes in the font family.

Beta Gotcha: The Font Menu is not implemented in this release of Puffin5. The layout, font selection, and functionality are all subject to change in a future release.

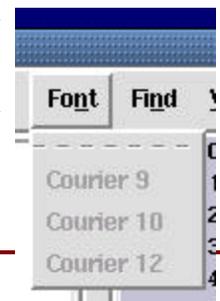


Illustration 31 - The MPE Window Font menu

4.1.4- The Find Menu

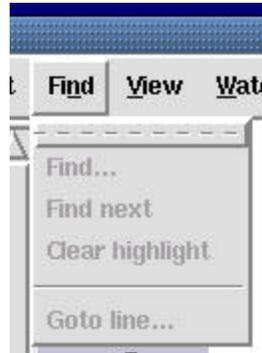


Illustration 32 - The MPE Window Find menu

Beta Gotcha: The Find Menu is not implemented in this release of Puffin5. The layout and available functions are subject to change in future releases.

4.1.5- The View Menu

Beta Gotcha: The View Menu is not implemented in this release of Puffin5. The layout and available functions are subject to change in future releases.

The View Menu enables you to select between alternative views for differing parts of the MPE window.

4.1.5.1- Show registers/Show variables

These two items control the information presented in the browser view.

Selecting Show registers will bring up the register browser.

Selecting Show variables will bring up the variable browser.

See section 4.2 on page 32 for more information about the browser view.

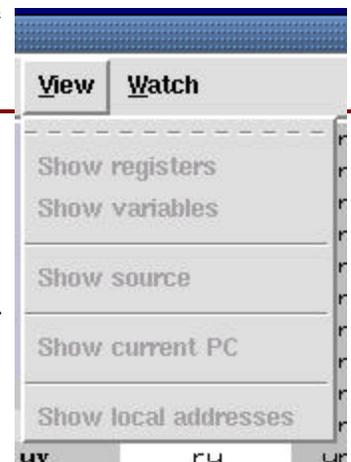


Illustration 33 - The MPE Window View Menu

4.1.5.1.1- Lock browser

Normally, the MPE browser view will automatically switch between the two browsers, depending on the context being displayed in the source view. You can lock the browser view to one or the other browser by selecting the Lock Browser item. In this case, the selected browser will always be presented in the browser view.

4.1.5.2- Show file/Show disassembly

These two items control the information presented in the source view.

Selecting Show file show the file corresponding to the current source view. If there is no file available, this menu item is greyed out.

Selecting Show disassembly will show the disassembly output corresponding to the current

source view.

See section 4.4 on page 32 for more information about the source view.

4.1.5.2.1- Lock source view

Normally, the MPE source view will automatically switch between file view and disassembly view. You can lock the source view to one of the other by selecting the Lock source view item. In this case, the selected view will always be presented.

4.1.6- The Watch Menu

The watch menu allows you to add and remove entries from the Watch window.

Selecting Add... will pop up the Watch Dialog window. See section 5.3 on page 35 for more information on how to use the Watch Dialog window.

Clicking on Remove selected will perform some action not yet adequately defined.

Beta Gotcha: The Watch menu is not completely implemented in this release of Puffin5.

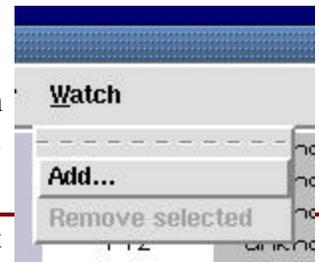


Illustration 34 - The MPE Window Watch menu

4.2- The Browser View

4.2.1- The Register Browser

4.2.2- The Variable Browser

Beta Gotcha: The Variable Browser is not implemented in this release of Puffin5.

4.3- The Source File List

The Source File List contains a list of all the source files that Puffin 5 could find relating to your loaded .cof file. The files are sorted alphabetically. Filenames are case sensitive.

Beta Gotcha: This could cause some problems on Windows systems, due to the case insensitivity of filenames under that OS. This will be addressed in a future release.

To reduce clutter and improve readability, duplicate file names are removed from the source file list. If 22 of your source files include the file `raccoonfeeder.h`, then `raccoonfeeder.h` will only be listed once. Note that this could cause some confusion if you have reused filenames among modules. See section 4.3.1 on Namespace Collisions, below.

4.3.1- Namespace Collisions

Note that the .cof file only records the name of the source file as it was passed to the compiler/assembler. This means that Puffin5 cannot distinguish between files with the same name that may exist in different components and/or versions. For example, various of the VM

Labs libraries incorporate files named `version.c` (a total of 6 such files at last count). If you find that you are not getting the file you expect when you go to load a source file, you might try the following remedies

- ? rearrange your search path appropriately;
- ? start Puffin5 from a different directory;

Beta Gotcha: This shouldn't be necessary anymore, starting with Puffin5 5.0.02 Beta.

- ? adopt a file naming convention that helps avoid namespace collisions (for example, rename the MGL project's `version.c` file to `mgl_version.c`).

4.4- The Source View

Beta Gotcha: When a large file is loaded into the file view, it can take a while to switch from that file to another file, or to the disassembly view. In fact, it can take an excruciatingly long time. During this time, a yellow progress bar will be displayed and it will appear that Puffin5 has locked up. Have patience though - the process will eventually complete and the new file will be displayed.



Illustration 35 - The yellow progress bar of sloth.

4.4.1- Disassembly View

4.4.2- Text View

4.5- Breakpoints

5- Watch Window

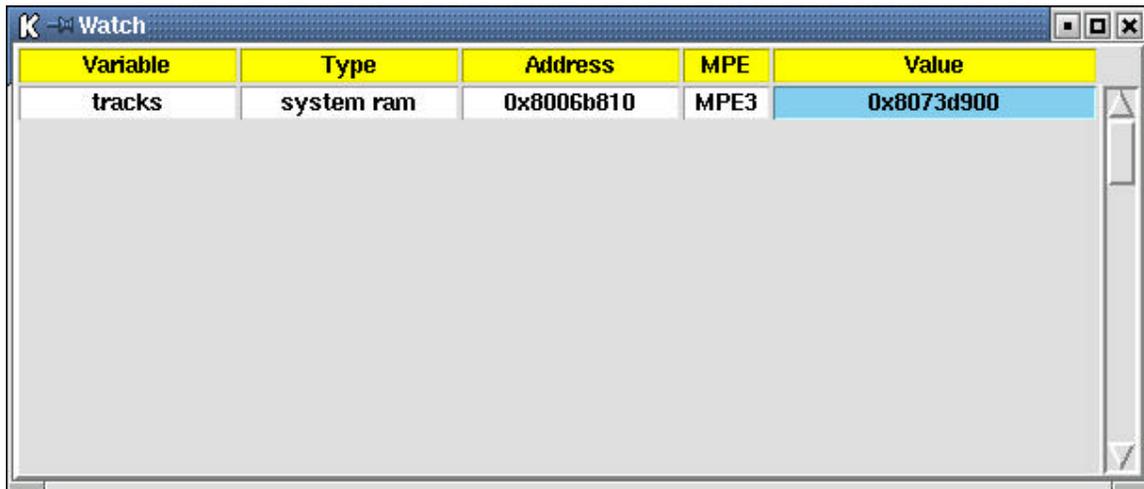


Illustration 36 - The Puffin5 Watch Window

The Watch Window displays a list of all the currently set watched variables.

5.1- Clearing a Watch

Clicking the right mouse button on any entry in the Watch Window will bring up a menu. This menu varies from field to field, but always contains a "Clear Watch" entry. Clicking on "Clear Watch" will remove the particular watch entry.

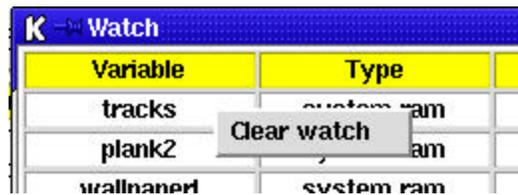


Illustration 37 - The "Clear watch" menu item

5.2- Changing the Format

Right clicking on the Value field of a watch entry will bring up a menu with a "Format" entry. Clicking on this entry will bring up a list of formats that you can choose from. These are

- ? decimal
- ? hex
- ? binary
- ? ascii
- ? real

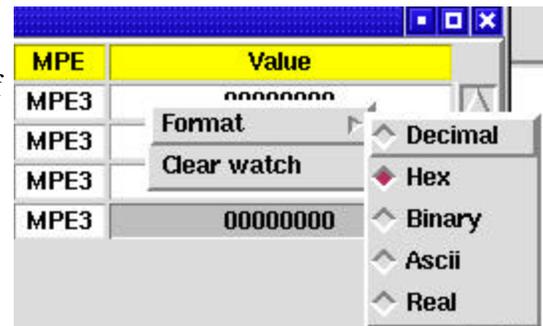


Illustration 38 - Formatting a watched value.

Beta Gotcha: Only decimal, hex, and binary formats are reliably implemented in Puffin5 5.0.02 Beta.

5.3- The Watch Dialog

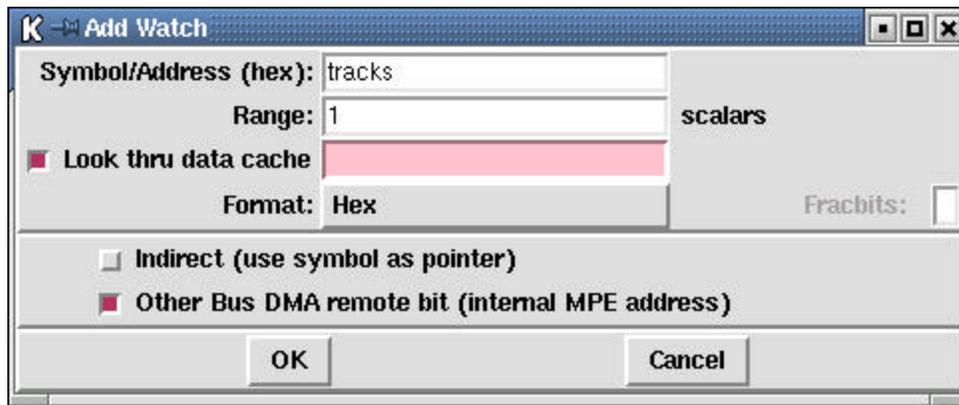


Illustration 39- The Watch Dialog window

The Watch Dialog is used to setup variable watches. It will pop up when you select Watch from the source view menu, or from the MPE Debug Menu.

Beta Gotcha: At this time, only a range of 1 scalar is supported.

Beta Gotcha: only hex decimal and binary formats are fully supported at this time.

If you try watching an uninitialized pointer, a warning message will pop up.

6- Breakpoint Window

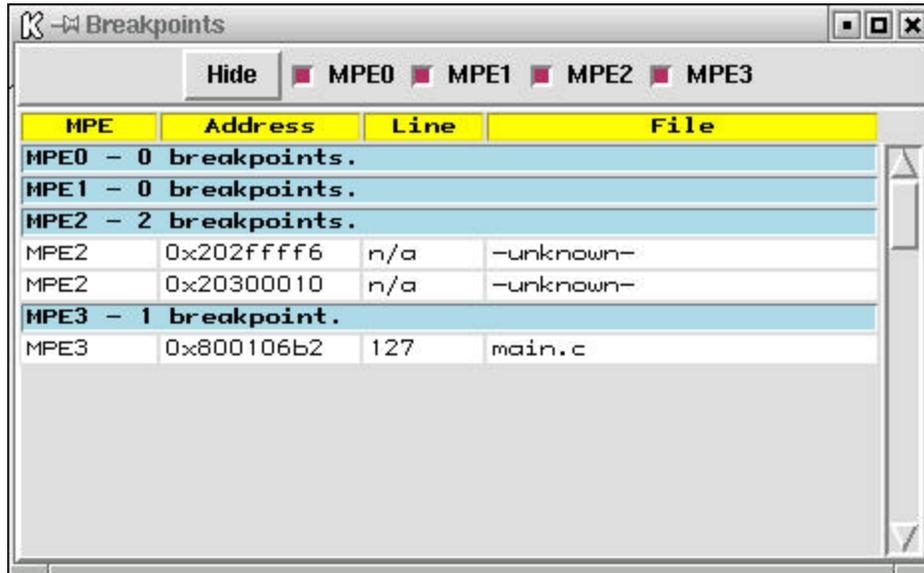


Illustration 40 - The Breakpoints Window

7- Global Variables Window

Beta Gotcha: The Global Variables Window is not implemented in this release of Puffin5.

8- Current Puffin5 Bug List

BugID	Build	OS	Summary	Steps to Reproduce	Status
1	0021-L	Linux Win2k	Crash if click on side outer scroll bar on any .mpe window	launch puffin5; when .mpe's windows pops-up, click on the left outer scroll bar on any of the windows. Crash with segmentation fault.	close
2	0021-L	Linux	Vertical scroll bar in the bottom part of the .mpe3 dialog is not getting displayed unless you click on MPE3 button on the.puffin5 window	launch puffin5; click on the Reset button; click on RunAll button(optional); on .mpe3 window execute Puffin->Load Object File. Load room.cof. File is loaded correctly. Click on RunMPE button. You can see the "room" picture on your devsystem. Click on StopMPE button. The "room" picture disappears. Note that the source code main.c file is displayed, but you cannot get to any line. Vertical scroll bar is disable. You need to close puffin5, and after clicking on Reset button, click on MPE3 button on .puffin5 window.	close
3	0021-L	Linux Win2k	Loading .cof file to mpe other than mpe3 causes application crash	launch puffin5; click on the Reset button; click on RunAll button; on .mpe2 window execute Puffin->Load Object File. Load room.cof. Application crashes with the msg. On terminal window: can't write 131072 bytes to 80010000.Code: 19	close
4	0021-L	Linux Win2k	If source files are NOT in the same directory as puffin5 executable, loading .cof application crash (even if .cof file is in the same same directory as puffin5)	1)Run puffin5k from the directory other than the one that has source code for room sample (mkdir test; cp puffin5k test; cd test; ./puffin5. Application crashes when you try to load room.cof file. 2) cp source for room sample (*.c *.s.*.o) to test directory, don't copy room.cof launch puffin5 from Test directory. Load room.cof that is located in different directory. Load and run successful.	close
5	0021-L	Linux	Launching puffin5 when dev system is turned off causes application crash	Turn off dev-system. Launch puffin5. Application crashes with segmentation fault.	close
6	0021-L	Linux	Loading .cof file before resetting mpe causes failure to run, but does not produce error message. If you click on Reset button after,and re-load same file, source files will be displayed twice.	Launch puffin5. Without clicking on Reset button, select mpe3, load room.cof. Click on RunMPE. No error message, but program does not run. Click on Reset button. Execute File->Load object file again. Load room.cof. You will notice that all source code files are displayed twice. You will see two main.c, room.s, drawroom.c	close
7	0021-L 0154 -W	Win/Li nux	If close MPE, Watch window, you cannot re-open it.	Launch puffin5. Select MPE3 button on .puffin5 window. It should open mpe3. Close MPE3 window by clicking on "x" window manager. Click on mpe3 button again. MPE window becomes unaccessible. Message " bad window path name ".mpe3" appears on the terminal. The same true for Watch dialog.	close
8	0021-L	Linux	Selecting .s file (room.s) in the source code list causes GUI to freeze	Launch puffin5. Click on Reset button . Click on mpe3 button on puffin5 dlg.. Load room.cof. Select room.s file in the source file's list. GUI is getting frozen.	close

BugID	Build	OS	Summary	Steps to Reproduce	Status
9	0021-L 0154 -W	Win/Li nux	If set breakpoint on the line that doesn't have the corresponding address, you will be given the correct error message:that it cannot find address corresponding to.. , but it DOES SET the breakpoint. When you try to clear by selecting Clear breakpoint from the pull-down list, it complains that it is not set.	Launch puffin5. Click on Reset button. Click on mpe3 button on puffin5 dlg. Load room.cof Select the line that does not have the corresponding address (e.g. Comments) Using right button pull-down Set Breakpoint. You will get a message that it cannot find the corresponding address. Using right button pull-down menu click on Clear Breakpoint. It complains that breakpoint has not been set	close
10	0115-W	Win	Windows Tcl/Tk Bug: In read-only text displays, Tcl/Tk displays the selection as white background and black text. You cannot see the line you selected.	Launch puffin5. Click on Reset button. Click on mpe3 button on puffin5 dlg. Load room.cof Select the line that does not have the corresponding address (e.g. Comments) You cannot see the selection.	open
11	0154 -W 0146 -L	Win/Li nux	If exit Puffin5 not by executing File->Exit but by closing Puffin windows through File Manager (x), Puffin processes are not getting killed.	Launch puffin5. Close all puffin5 windows by clicking on right-upper -corner (x) , Make sure all puffin windows are not displayed. Invoke Task Manager (press Cntr->Alt_Delete). Note that puffin5 process is there.	close
12	0154 -W 0146 -L	Win/Li nux	Tcl/Tk 083 is not statically linked	If you don't have tcl/tk intalled, puffin5 won't launch. The message "a required TCL83.dll is not found.	open
13	0154 -W 0146 -L	Win/Li nux	Debug menu items: Run, Stop, Step Over, Remove all breakpoints are not functioning. File->Refresh does not fork on Linux.	Those menu items are not greyed out	open
14	0154 -W 0146 -L	Win/Li nux	There is not confirmation msg.when exiting puffin5	When execute File->Exit or close the puffin5 windows through File Manager, there is no confirmation mgs.	open
15	0154 -W 0146 -L	Win/Li nux	Clicking on the left site out bar triggers clearing source window	launch puffin5; when .mpe's windows pops-up, click on the left outer scroll bar on any of the windows. The msg. saying that source window is being cleared is popping up.	close

CURRENT PUFFIN5 BUG LIST

BugID	Build	OS	Summary	Steps to Reproduce	Status
16	0154-W	Win	When breakpoint is reached, the file that has the breakpoint is not getting loaded if previously other file was displayed.	Following demo instructions, load room.cof. -click to load main.c file in the source window - set breakpoint on line 130 - click to load drawroom.c in the source window -invoke Watch window for "tracksjpg_start" on line 7 and "tracks" on line 6 -click on RunMPE in the MPE window. Click the Z button to toggle lighting. When breakpoint happens, main.c is not getting loaded into the source window unless you load this file before running RunMPE.	open
17	0153-W	Windows	.watchDialog does not display OK, Cancel button. You need to stretch the frame down to see it.	launch puffin5, load room.cof; click to load drawrom.c in the source window. Select "tracks" on line6, Right click to bring up the file ops menu and select Watch. The watch dialog does not display OK, Cancel buttons.	close
18	0153- W 0146-L	Win/Li nux	You have to exit and restart puffin5 to reload the object file	Launch puffin5. Click on Reset button. Click on mpe3 button on puffin5 dlg. Load room.cof Click on MPERun button. Click on Reset button. MPE3 is reset. Load another .cof file (e.g. simple). The file is not loaded. You need to exit and restart puffin5 to reload object file	open
19	0153- W 0146-L	Win/Li nux	When load room.s file and do single step GUI is getting frozen for couple of minutes, since the source window is cleared. There is no message notifying about this.	launch puffin5, load room.cof; click to load drawrom.s in the source window. Click on Single step button. GUI is getting frozen until it's clearing the file. No notification message	close
20	01360-W 0357-L	Win/Li nux	.watchDialog layout is munged. You need to expand the window to find the buttons.	You need to expand the window to find the buttons	close
21	01360-W 0357-L	Win/Li nux	If invoke Watch window from Puffin5 Controls BEFORE than selecting variable to watch and opeing .watchDialog, the variables' values are NOT getting displayed in the Watch window.	Launch puffin5. Click on Reset button. Click on Run All button.Click on Watch window from Puffin5 Ciontrols. Click on mpe3 button on Load room.cof. Select drawroom.c in the source window Go to line7 and select "tracksjpg_start". Right click to bring up the file ops menu and select Watch. Click OK on .watchDialog. Click on Watch button on Puffin5 Controls.It invokes empty Watch window. The same happens if you open and close Watch window instead of keeping it open.	close
22	01360-W 0357-L	Win/Li nux	If select the symbol to watch, while Watch window is closed, and open Watch window, it would not display any entry	Launch puffin5. Click on Reset button. Click on Run All button.Load room.cof. Select drawroom.c in the source window Go to line7 and select "tracksjpg_start". Right click to bring up the file ops menu and select Watch. Click OK on .watchDialog. Click on Watch button on Puffin5 Controls to bring Watch window. It displays the value of "tracksjpg_start". Close Watch window. Select "track" symbol.Right click to bring up the file ops menu and select Watch. Click OK on .watchDialog.Open Window dialog. It does not display any value. But if you click on StopMPE button all values are getting displayed.	close

BugID	Build	OS	Summary	Steps to Reproduce	Status
23	0359-W	Win	The breakpoint indicator is displayed if you try and set a breakpoint on a line with no corresponding address.	Launch puffin5. Click on Reset button. Click on mpe3 button on puffin5 dlg. Load room.cof Select the line that does not have the corresponding address (e.g. Comments) Using right button pull-down Set Breakpoint. The break point indicator is still displayed.	close
24	0360 -W 0357-L	Win/Li nux	If you close Watch window and re-open it, the restored window does not look like they should	Launch puffin5. Click on Reset button. Click on Run All button.. Load room.cof into mpe3. Click on Watch button on Puffin5 Controls to bring Watch window.Select drawroom.c in the source window Go to line7 and select "tracksjpg_start". Right click to bring up the file ops munu and select Watch. Click OK on .watchDialog. Click on Watch button on Puffin5 controls to open Watch dlg. Watch dlg displays System Type, Address, MPE, Value for the variable "tracksjpg_start". Close Watch dlg. Open it again. It does show the dialog differently that it was before.	close
25	0393-W	Win	pthreadVSE.dll is NOT statically linked	Launch puffin5. You will get an error that pthreadVSE.dllcannot be found unless you have it copied to the same directory from which you're running puffin5	open
26	0393 -W 0411-L	Win/Li nux	After Single Stepping and Stepping Over the source window is getting cleared.	Launch puffin5. Click on Reset button. Click on Run All button.. Load room.cof into mpe3. Click on SingleStep button. The source window is getting cleared	close
27	0393-W	Win	.watchDialog is unable to look up value for the pointer UNLESS you RunMPE, but It finds the value for a variable.	Launch puffin5. Click on Reset button. Click on mpe3 button on puffin5 dlg. Load room.cof . Select drawroom.c in the source window Go to line7 and select "tracksjpg_start". Right click to bring up the file ops menu and select Watch. Click OK on .watchDialog. Go to line 7 and select *tracks pointer. Right click to bring up the file ops menu and select Watch. .watchDialog complains that is "unable to look up value for tracks (addr: 0x800652a0). If you click on RunMPE button it displays the value for the pointer.	Not a problem
28	0393 -W 0411-L	Win/Li nux	Vertical scroll bar in .console and .debuglog windows is not working	Launch puffin5. Click on .console window. Vertical scroll bar does not scroll up or down. The same happens with .debuglog window.	close
29	0411-L	Win/Li nux	Invalid disassembly load if click in empty area of file	start puffin5. Click in the empty file list of any MPE window. Invalid disassembly information will be loaded	close
30	0393 -W 0411-L	Win/Li nux	Window menu in Puffin5 Controls is not functioning	When you select any of the sub-items in Window menu in Puffin5 Controls dialg, it does not pop-up the selected windows.	close
31	0454 -W	Win	The problem in setting breakpoint.	If load .cof file (room.cof), select the source file (main.c), select the line to set breakpoint, the error msg. Pops up saying "No file loaded	close

CURRENT PUFFIN5 BUG LIST

BugID	Build	OS	Summary	Steps to Reproduce	Status
32	0454-W 0411-L	Win/Li nux	There is no error msg that stays when load file fails	Load an object file into MPE2 with the "Use BIOS Loader" flag set. The file will fail to load, but the only indication of thi I a brief flicked of red from the progress bar.	close
33	0454-W 0411-L	Win/Li nux	Loading an object file into MPE2 with the "Use BIOS Loader" flag fails.	Load an object file into MPE2 with the "Use BIOS Loader" flag turned on. Loading fails.	close
34	0411-L	Linux	Selection in source file list does not persist	Select a file in the file list. After it loads, select some text in the displayed source. The selection in the file list will disappear. It should persist.	open
35	0491-W	Win	When clicking on RunMPE button it loads disassembly into MPE1 and MPE2	On Windows: start puffin5. Click on Reset button, click on RunMpe button. Invalid dissembly information is loaded into MPE1 and MPE2.	Not a proble m
36	0531-W 0529-L	Win/Li nux	Source window is not get updated after stopping MPE.	Launch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof file. Click on RunMPE button. Click on StopMPE button. The source window does not display the disaseembled code when halt occurred. You need to do Single Stepping to see the context.	close
37	0531-W 0529-L	Win/Li nux	Stepping Over does not properly function	Launch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof file. Select main.c file. Set breakpoint on line glEnable(GL_LIGHTING). and click on Z button, it highlighted the line and stopped. Click on StepOver button. It does not move to the next function on the source window.	open
38	0531-W 0529-L	Win/Li nux	Source window is not always get updated when Single Step into function.	Launch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof file. Select main.c file. Set breakpoint on line glEnable(GL_LIGHTING). and click on Z button, it highlighted the line and stopped. Click on Single Step button. Source window has the same line highlighted. It neither show source inside this function, nor dissabling.	close
39	0531-W 0529-L	Win/Li nux	Each Single Stepping is accompanied by green status bar "Disassamble done"	Launch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof file. Select main.c file. Set breakpoint on line glEnable(GL_LIGHTING). and click on Z button, it highlighted the line and stopped. Click on Single Step button. Green status bar is getting displayed saying "Disassamble done"	close
40	0531-W 0529-L	Win/Li nux	Console window displays misleading error message mpe_LoadObjectFile not completely implemented, though the load was successful	Lanch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof. Console windows displays highlighted message "mpe_LoadObjectFile not completely implemented". Click on RunMPE button. Devsystem displays the expected image. Load is successful.	close
41	0531-W 0529-L	Wiin/Li nux	When do Single stepping the highlight message on Console window says : Failed to find file index 61	Launch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof file. Select main.c file. Set breakpoint on line glEnable(GL_LIGHTING). and click on Z button, it highlighted the line and stopped. Click on Single Step button. Console window displays Failed to find index <> message every time you do single stepping.	close

BugID	Build	OS	Summary	Steps to Reproduce	Status
42	0531-W 0529-L	Wiin/Li nux	When close console window and re-open it from Puffin5 Control Window menu, it appears empty	Launch puffin5. One of the windows this invokes is the Console window. It displays information about build number, etc. Close this window. Select Console from Window menu in Puffin5 controls. The window appears empty.	open
43	0531-W 0529-L	Wiin/Li nux	Scroll bar in Console window does not work	Launch puffin5. Click on Console window. Vertical scroll bar does not work.	Not a problem
44	0531-W 0529-L	Wiin	Puffin5 crashes if click OK to the Watch window before dismissing error message dialog that pops up to notify about some error in locating the variable.	Load room.cof into mpe3, select drawroom.c. After drawroom.c loads, go to line 6 and select "tracks". Using right-bottom mouse select Watch on ops menu, select the "Indirect button" on Watch dialog. Click on OK button on Watch dialog. The msg. pops up saying Unable to look for value "track". BEFORE clicking OK to this error msg., click OK on Watch window. Program Error Message pops up, saying that you need to re-start puffin5. Program crashes. On Linux Watch window OK button is disable until you close "Unable to look for value "track" - it does not crash.	close
45	0531-W 0529-L	Wiin/Li nux	There is no way to remove variables from the Watch window.	Its very desirable to be able to remove variables from the Watch (Puffin Control) window.	close
46	0531-W	Win	MPE level File->Use Bios Loader menu displays Jse instead of Use if select this option.	Launch puffin5. . Click on Reset, RunAll, MPE3 buttons on Puffin Controls. On MPE3 window click to select Use Bios Load menu option. When this option is getting highlighted Use changes to Jse. This problem is due to the bug in TCL/TK windows implementation	open
47	0531-W 0529-L	Wiin/Li nux	Source window is not always get updated when Single Step into function.	Launch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof file. Select main.c file. Set breakpoint on line glEnable(GL_LIGHTING). and click on Z button, it highlighted the line and stopped. Click on Single Step button. Source window has the same line highlighted. It neither show source inside this function, nor dissabling.	close
48	0531-W 0529-L	Wiin/Li nux	There is no feedback when clicking on Single Step button while MPE is running.	Launch puffin5. Click on Reset, RunAll buttons on Puffin Controls. Load room.cof file. Click on RunMPE button. Click on Single Step button. Nothing happens. It should be an error message saying you need to StopMPE first. The same applies to Step Over	close
49	058-W	Wiin	Cursor does not change color when moving into the breakpoint zone. It does change shape.	Move the cursor in and out of the breakpoint zone. Under Linux, it changes shape and color (read to set, green to clear). Under windows, it only changes shape. This is due to tk bug, and cannot be fixed until tcl/tk is fixed. See problem 420510 at SourceForge.net.	open

Alphabetical Index

address column	16		
Beta Gotcha	8p., 13, 15, 17, 23p., 26, 28pp, 34p., 37		
BIOS loader	28		
Breakpoint	16		
breakpoint column	16		
breakpoint marker	17		
breakpoints			
clearing	33		
managing	33		
marker	17		
setting	33		
Breakpoints window	17		
Breakpoint Window	36		
Browser View	32		
Bug List	38		
code column	16		
cursor	16		
Debug Menu	29		
Directory Browser	25		
disassembly view	16		
3.3-Exiting Puffin5	19, 23		
Confirmation Window	23		
File Menu	28		
file view	16		
Find Menu	30		
Font Menu	30		
for breakpoints			
cursor	16		
Global Variables	37		
Launching	12		
line number column	16		
loading			
object file	14		
4-MPE Window	27		
File Menu	28		
Menu Bar	27		
object file	28		
Puffin5 Controls	20		
Register Browser	32		
search path			
Directory Browser	25		
Search Path Manager	24		
Search path	15,	20	
setting			
breakpoint	17		
source file			
loading	15		
Source File List	32		
Source View	16,	32	
Variable Browser	32		
View Menu	31		
watch			
clearing	18, 34		
formatting	34		
setting	35		
Watch	17		
Watch Dialog	35		
Watch Menu	31		
Watch Window	34		