

# **BLAZEGUIDE**

Rick Pratt

**COLLABORATORS**

	<i>TITLE :</i> BLAZEGUIDE		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Rick Pratt	August 24, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>BLAZEGUIDE</b>	<b>1</b>
1.1	BlazeWCP . . . . .	1
1.2	It's a floor wax, it's a dessert topping.... . . . . .	1
1.3	I'm a glutton for punishment... . . . . .	3
1.4	300THz cpu with 512 GB mem... . . . . .	4
1.5	If it blows up, it's your fault . . . . .	4
1.6	You'll need a hammer . . . . .	4
1.7	Errors . . . . .	4
1.8	This program brought to you by MicroSof..... . . . . .	5
1.9	Version history . . . . .	5
1.10	Future . . . . .	5
1.11	If I mention your name, pay up :) . . . . .	6
1.12	You will be assimilated..... . . . . .	6
1.13	Contacting Me . . . . .	6

## Chapter 1

# BLAZEGUIDE

### 1.1 BlazeWCP

BlazeWCP - Very fast C2P patch for the OS chunky pixel functions

What is it

Why another patch

Requirements

Disclaimer

Installation

Error Codes

Bugs

Version History

Future stuff

People I'd like to thank

Other Stuff By Me

Please read WPATest guide before using WPATest.

Flames, bugs, congratulations...

find me here.

### 1.2 It's a floor wax, it's a dessert topping....

BlazeWCP is a highspeed C2P patch for the OS chunky functions

---

WritePixelLine8(), WritePixelArray8() and on OS 3.1+ WriteChunkyPixels(). It calls DoHookClipRects() to work on the destination bitmap directly and aligns all chipram writes to 32bit boundries. All of this makes it fast on 200+ cpus and REALLY fast on machines with 32bit chipram. Using the included wpatest, a benchmark of these functions written by Stephen Brookes (programs FBlit), I compared BlazeWCP to the OS routines and other patches of this type. Using an 8bit (256 color) 640 x 512 hires laced PAL screen I get the following results: ( #'s are thousands of pixels/s so 100k is 100,000 pixels/s)

Please read the wpatest guide before using it. It's a little buggy. The pause after hitting the Speed button is normal.

	not 32bit aligned	32bit aligned	
OS 3.0	111k pixels/s	272k pixels/s	
PatchWPA8	152k pixels/s	593k pixels/s	<- FBlit's why it's this fast on this test
NewWPA8	200k pixels/s	569k pixels/s	
BlazeWCP	432k pixels/s	2043k pixels/s	

As you can see BlazeWCP is from 2.15 - 3.59 times faster then NewWPA8 which is the next fastest. PatchWPA8 only seems faster on the 32bit aligned test because of FBlits BltBitMap() patch. Keep in mind that I only have a 030 40MHz. Also realize that since the test was done on an 8bit screen then # of pixels/s = the # bytes/s. According to SysSpeed, at the resolution and bit depth the test was done at copyspeed from fastram to chipram using longword (32bit) writes is 3.23mb/s. On aligned conversions i get 2.043mb/s so simple math shows that on my 030 40Mhz the patch runs at ~63% copyspeed. On my friend Iain Barclays' (programs 8nl.device) 030 50MHz he gets 547k pixels/s non-aligned and 2478k pixels/s aligned at this resolution/ bit depth. Stephen Brookes gets 876k pixels/s non-aligned and 4124k pixels/s aligned at the same res/depth on his 060 50MHz. I should have known this patch didn't run at copyspeed on a 040 25MHz even tho the c2p routines themselves do. I can dream tho :) Keep the speed results coming.

Speeds Reported by users, (non-aligned) - (aligned) in thousand pixels/s

Test Screen - PAL 8bit (256 color) hires laced 640 x 512:

WPL8          WPA8          WCP

```
030 40MHz      255 - 1388  432 - 2043  430 - 2043
030 50MHz      xxx - xxxx  547 - 2478  xxx - xxxx
040 28MHz      453 - 2027  636 - 2490  634 - 2490
060 50MHz      xxx - xxxx  876 - 4124  xxx - xxxx
```

I also had a report from a man with an ECS equipped A2000 that BlazeWCP works very well. His results were:

```
030 40MHz      92 - 430  157 - 667  156 - 663
```

Of course he was only able to run the test at 16 colors on a 640 x 512 PAL screen but he reported a speedup of 4x for the non-aligned test and 10x for the aligned test compared to the OS routines. For comparison I get 359 - 2199 , 859 - 4526 and 849 - 4521 on the same type screen.

### 1.3 I'm a glutton for punishment...

I made BlazeWCP after I got sick of how slow the other patches of this type were . PatchWPA8 was very slow on my 030 since it calls BltBitMapRastPort() after everyline it converts and since it uses a single scratch bitmap and buffer for its functions it gets SLOW !! when more than 1 program tries to use it's routines. NewWPA8 was faster but it only used 16 bit C2P code which is slower then it has to be when you have a 32bit cpu and 32bit chipram. NewWPA8 also has a bug that the author denies that causes line corruption with IBrowse since it calls WritePixelLine8() to render to the screen. The corruption is caused by the fact that NewWPA8 builds the hook it passes to DoHookClipRects() using memory internal to the patch itself. This works fine if only 1 program uses it's routines but when 2 or more use them the hook gets overwritten with the new callers data. This causes invalid pointers to be passed to DoHookClipRects() which causes the corruption me and others noticed. Since IBrowse spawns a task for every picture being rendered/decoded these routines DO get called by more than one task and that's why some lines get corrupted.

/BEGIN RANT

I emailed Michael van Elst, the author of NewWPA8, about this corrupt lines problem about

2 years ago, before I knew what caused it. He sent me an email back claiming ←  
that  
there was nothing wrong with NewWPA8 because it "worked for him". Mr. Elst (or ←  
van Elst),  
any good programmer will tell you that just because something "works for you" ←  
doesn't mean  
it works for everybody. Your complete lack of interest in a bug in your patch ←  
shows me  
that you truly didn't care, or even bother to check. I guess I'm lucky you even ←  
bothered  
to email me back at all.

/END RANT

## 1.4 300THz cpu with 512 GB mem...

A 020+  
Some fast ram  
OS 3.0+

Recomended:

A 030 50Mhz  
32bit chip ram  
AGA chipset  
FBlit

## 1.5 If it blows up, it's your fault

I take no responsibility for any death or destruction caused by this patch. Use it at your own risk.....you've been warned.

## 1.6 You'll need a hammer

Just copy BlazeWCP to your C directory and add C:BlazeWCP to your startup-sequence AFTER SetPatch BUT BEFORE any gfxcard support software like P96 or cybergfx. I doubt you'll be using this patch if you have a gfxcard tho.....lucky you :)

## 1.7 Errors

The patch will return error code 10 if it can't allocate memory for the patch, 20 if you don't have a 020 or better and 30 if it can't open graphics.library v39+ or layers.library V37+.

---

## 1.8 This program brought to you by MicroSof.....

None known, email me if you find some.

## 1.9 Version history

0.x - Internal testing versions, buggy and not optimized, not released.

1.0 - Was supposed to be the initial release.

1.1 - Sped up 8bit and 4bit conversion routines a little.

- Sped up the main loop a bit.

- Added support for rastports with masked planes.

- Initial Release.

1.2 - Streamlined the c2p code a little, constance are only loaded into registers once instead of by both routines of a conversion >4 planes. Makes the code smaller and possibly a little quicker.

- Now calculates # pixels plotted by adding up the pixels plotted for every rect the hook is called for. Should make the #s returned by WPL8 and WPA8 100% accurate....WCP now actually returns # pixels plotted too even tho it doesn't have to.

- 2nd Release.

1.3 - Reorderd register usage a little. This saves 12 bytes per routine and freed ↔

up a register which is now used to defer a chipram write for the 4/8 bit conversion routines instead of defering it to the stack.

- The above optimization made me think a little (not too much, i'd hurt myself ;) ↔

and allowed me to see a way to keep the other chipram write that was deferred to ↔

the stack in the 4/8 bit routines in a register. Now all the conversion routines ↔

run completely within the registers and are just complex mem copy loops, as they ↔ should be.

- 3rd Release

## 1.10 Future

---



- More speed.....that's a tough one....
- Sugestions welcome.

## 1.11 If I mention your name, pay up :)

- Iain Barclay: for 8n1.device, getting me into asm programming and putting up with stupid questions ;)
- Stephen Brookes: for FBlit and for the help you've given me. Also for ↔ responding to my emails even when you're swamped and I'm sure couldn't care less about ↔ them at the time.
- Michael Kalms: for the lightning fast chunky 2 planar routines that this ↔ patch uses.
- Michael van Elst: for adamantly denying the existance of a blantanly ↔ obvious bug in NewWPA8, which aside from that is a really good patch.

I'd also like to thank everyone for the feedback I've gotten :)

## 1.12 You will be assimilated.....

Other stuff by me on Aminet

FText V1.7 - Speeds up text by rendering to fastram instead of chip. Needs ↔ FBlit V3+.  
Get the latest FBlit from <http://www.tpec.u-net.com>

QBC V1.1 - A cpu only BlitClear() patch.

## 1.13 Contacting Me

Send praise, suggestions, bug reports, flames to:

rickprat@usit.net

or find me on DalNet channel #AmIRC nick: Kingamiga

Check out my poor excuse for a website at <http://kingamiga.webjump.com>

---