

Operation	VGA	SVGA
Using FillRect	924.27	1564.745
Using ExtTextOut in place of FillRect	965.112	1625.859
Percentage Performance Gained using ExtTextout	-4.23%	-3.76%
Using TextOut	2299.163	3548.573
Using ExtTextOut	2050.231	3192.911
Percentage Performance Gained using ExtTextout	12.14%	11.14%
	===== VGA =====	
	TestIDM_TEXT_TEXTOUT	
	216:	4.061
	217:	
	218:	96.338
	219:	91.389
	220:	1312.952
	221:	702.363
	222:	92.06
		2299.163
	TestIDM_TEXT_ETO	
	259:	4.074
	260:	
	261:	94.93
	262:	91.861
	263:	1755.664
	264:	103.702
		2050.231
	TestIDM_RECT_FILLREC	
	150:	0.084
	151:	0.251
	152:	923.935
		924.27

	TestIDM_RECT_ETO	
	174:	0.085
	175:	0.251
	176:	
	177:	8.316
	178:	952.456
	179:	4.004
		965.112
	===== SUPERVGA =====	
	TestIDM_RECT_FILLREC	
	150:	0.341
	151:	0.277
	152:	1564.127
		1564.745
	TestIDM_RECT_ETO	
	174:	0.092
	175:	0.273
	176:	
	177:	7.031
	178:	1613.601
	179:	4.862
		1625.859
	TestIDM_TEXT_TEXTOUT	
	216:	4.969
	217:	
	218:	152.306
	219:	144.468
	220:	1826.176
	221:	1280.725
	222:	139.929
		3548.573
	TestIDM_TEXT_ETO	
	259:	4.651
	260:	

	261:	147.391
	262:	143.763
	263:	2756.344
	264:	140.762
		3192.911
	=====	8514 (large fonts
	TestIDM_RECT_FILLREC	
	150:	0.083
	151:	0.248
	152:	542.617
		542.948
	TestIDM_RECT_ETO	
	174:	0.082
	175:	0.248
	176:	
	177:	7.892
	178:	526.781
	179:	3.948
		538.951
	TestIDM_TEXT_TEXTOUT	
	216:	4.009
	217:	
	218:	129.21
	219:	130.044
	220:	2225.483
	221:	474.886
	222:	128.22
		3091.852
	TestIDM_TEXT_ETO	
	259:	4.216
	260:	
	261:	129.966
	262:	127.27
	263:	2856.132
	264:	127.629

		3245.213
	=====	8514 (small fon
	TestIDM_RECT_FILLREC	
	150:	0.08
	151:	0.228
	152:	561.634
		561.942
	TestIDM_RECT_ETO	
	174:	0.08
	175:	0.239
	176:	
	177:	8.128
	178:	545.074
	179:	3.733
		557.254
	TestIDM_TEXT_TEXTOUT	
	216:	3.745
	217:	
	218:	154.989
	219:	155.831
	220:	2405.826
	221:	600.355
	222:	153.971
		3474.717
	TestIDM_TEXT_ETO	
	259:	3.983
	260:	
	261:	153.921
	262:	156.18
	263:	3163.973
	264:	157.7
		3635.757

8514 Lrg	8514 Small	
542.948	561.942	
538.951	557.254	
0.74%	0.84%	
3091.852	3474.717	
3245.213	3635.757	
-4.73%	-4.43%	
===		
0	48	
0.7	1056	
0.7	1056	
9.6	1056	
5.1	1056	
0.7	1056	
0	48	
0.7	1056	
0.7	1056	
12.8	1056	
0.8	1056	
T		
0	1	
0	3	
6.7	48	

0.7	1440	
0.7	1440	
13.9	1440	
0.7	1440	
) =====		
T		
0	1	
0	3	
3.4	48	
0	1	
0	3	
0	48	
3.3	48	
0	48	
-		
0	48	
0.8	1488	
0.8	1488	
13.8	1488	
2.9	1488	
0.8	1488	
0	48	
0.8	1488	
0.8	1488	
17.7	1488	
0.8	1488	

ts) =====		
T		
0	1	
0	3	
3.4	48	
0	1	
0	3	
0	48	
3.3	48	
0	48	
0	48	
0.9	1920	
0.9	1920	
14.4	1920	
3.6	1920	
0.9	1920	
0	48	
0.9	1920	
0.9	1920	
19	1920	
0.9	1920	


```
for ( j = 0; j < 3; j++ )
for ( i = 0; i < 16; i++ )
{
SetBkColor ( hDC, dwRGB[i] );
ExtTextOut ( hDC, 0, 0, ETO_OPAQUE, &r, NULL, NULL, NULL );
}
```

```
for ( j = 0; j < 3; j++ )
for ( i = 0; i < 16; i++ )
FillRect ( hDC, &r, hRGBBrush[i] );
```

```
for ( j = 0; j < 3; j++ )
for ( i = 0; i < 16; i++ )
{
SetBkColor ( hDC, dwRGB[i] );
ExtTextOut ( hDC, 0, 0, ETO_OPAQUE, &r, NULL, NULL, NULL );
}
```

```
for ( k = 0; k < iNumLines; k++ )
{
r.top = k*wCharHeight;
r.bottom = r.top + wCharHeight;
TextOut ( hDC, 0, k*wCharHeight, szTest, cbTest );
FillRect ( hDC, &r, hRGBBrush[i] );
}
```

```
for ( k = 0; k < iNumLines; k++ )
{
```

r.top = k*wCharHeight;
r.bottom = r.top + wCharHeight;
ExtTextOut (hDC, 0, k*wCharHeight, ETO_OPAQUE, &r, szTest, cbTest, NULL);
}
for (j = 0; j < 3; j++)
for (i = 0; i < 16; i++)
FillRect (hDC, &r, hRGBBrush[i]);
for (j = 0; j < 3; j++)
for (i = 0; i < 16; i++)
{
SetBkColor (hDC, dwRGB[i]);
ExtTextOut (hDC, 0, 0, ETO_OPAQUE, &r, NULL, NULL, NULL);
}
for (k = 0; k < iNumLines; k++)
{
r.top = k*wCharHeight;
r.bottom = r.top + wCharHeight;
TextOut (hDC, 0, k*wCharHeight, szTest, cbTest);
FillRect (hDC, &r, hRGBBrush[i]);
}
for (k = 0; k < iNumLines; k++)
{
r.top = k*wCharHeight;
r.bottom = r.top + wCharHeight;
ExtTextOut (hDC, 0, k*wCharHeight, ETO_OPAQUE, &r, szTest, cbTest, NULL);
}

for (j = 0; j < 3; j++)
for (i = 0; i < 16; i++)
FillRect (hDC, &r, hRGBBrush[i]);
for (j = 0; j < 3; j++)
for (i = 0; i < 16; i++)
{
SetBkColor (hDC, dwRGB[i]);
ExtTextOut (hDC, 0, 0, ETO_OPAQUE, &r, NULL, NULL, NULL);
}
for (k = 0; k < iNumLines; k++)
{
r.top = k*wCharHeight;
r.bottom = r.top + wCharHeight;
TextOut (hDC, 0, k*wCharHeight, szTest, cbTest);
FillRect (hDC, &r, hRGBBrush[i]);
}
for (k = 0; k < iNumLines; k++)
{
r.top = k*wCharHeight;
r.bottom = r.top + wCharHeight;
ExtTextOut (hDC, 0, k*wCharHeight, ETO_OPAQUE, &r, szTest, cbTest, NULL);
}

