

Re: Page Orientation capability in Unidrv? Reverse Print direction

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2006-03/msg00342.html>

- *From:* DaveK <dave.kenyon@xxxxxxxxxxxxx(donotspam)>
 - *Date:* Tue, 14 Mar 2006 13:23:27 -0800
-

Hi Dan,

Here is my INF file:

```
-----
;
; Printer Installation File
;
; List of supported printers, manufacturers
;

[Version]
Signature="$Windows NT$"
Provider=%TP%
ClassGUID={4D36E979-E325-11CE-BFC1-08002BE10318}
Class=Printer

[ClassInstall32.NT]
AddReg=printer_class_addreg

[printer_class_addreg]
HKR,,,%PrinterClassName%
HKR,,Icon,,-4"
HKR,,Installer32,,"ntprint.dll,ClassInstall32"
HKR,,NoDisplayClass,,1
HKR,,EnumPropPages32,,"printui.dll,PrinterPropPageProvider"

;
; Manufacturer section.
;
; This section lists all of the manufacturers
; that we will display in the Dialog box
;

[Manufacturer]
"Telpar"

;
```

Re: Page Orientation capability in Unidrv? Reverse Print direction

; Model Specifications Section

;

[Telpar]

"MTP-2242PP" = MTP-2242PP.GPD

;

; Installer section(s) referenced above.

;

[MTP-2242PP.GPD]

CopyFiles=@xxxxxxxxxxxxx,@MTP22PP.DLL,@MTP-2242PP.GPD

DataSection=UNIDRV_DATA

DataFile=MTP-2242PP.GPD

Include=NTPRINT.INF

PrintProcessor="MTP-22PP, MTP22PP.dll"

Needs=UNIDRV.OEM

[DestinationDirs]

DefaultDestDir=66000

MTP22PP.DLL=66001 ; print processor directory

[SourceDisksNames.x86]

1 = %disk1%,,\i386

[SourceDisksFiles]

MTP-22X2.Dll = 1

MTP-2242PP.GPD = 1

MTP22PP.dll = 1

[Strings]

PrinterClassName="Printers"

TP="Telpar"

disk1="Telpar Printer Driver Setup Disk"

If I comment out the PrintProcessor= line, the printer driver will install but without the print processor of course. The dll is getting copied to the print processor directory.

Thanks

DaveK

"Dan Negrescu" wrote:

Dave,

Can you show your INF file ?

Dan

"DaveK" wrote:

Thanks for all you insight on this.

Regarding the use of a print processor, I am having difficulty installing it with the INF file and the Add Printer Wizard using the statement `PrintProcessor="PrintProcessorName, PrintProcessorDLLName"` in the INF file. I get the following error:
'Printer driver not installed, operation could not be completed'

If I manually install the PrintProcessor by copying the dll to the appropriate print processor directory and update the registry to include the print processor, I can select it and debug it with the debugger. And it behaves as I desire.

Do you have any ideas on what I could be doing wrong or missing?

Thanks
DaveK

"Dan Negrescu" wrote:

Dave,

The print processor has access to the DEVMODE structure that was used to create the print job in the `OpenPrintProcessor()` function (within the `PPRINTPROCESSOROPENDATA` parameter). The content of the DEVMODE can be saved and used in the `PrintDocumentOnPrintProcessor()` function to decide if the page reversal has to be performed or not.

Regarding `OEMStartBanding()/OEMNextBand()` not being called:

If the entire page can fit into the max memory buffer size defined by the Unidrv (AFAIK it is about 6 Mb) then no banding is done. You mentioned that your printer is thermal printer so (my guess that) it's page size is at most Letter/A4 size (FAX or POS printer), black and white and a resolution between 150 and 300 dpi, so it would need about 1 Mb/per page. In this case the

Re: Page Orientation capability in Unidrv? Reverse Print direction

Unidrv will allocate memory for the whole page and OEMStartBanding()/OEMNextBand() are not called. In order to perform your page reversal you could hook the OEMSendPage() and flip the entire page.

Hope this helps,
Dan.

"DaveK" wrote:

Thanks for all the info. You re
overwhelming me with info,
but I sure am not complaining. Its been
helping me understand
all the pieces to the printer driver puzzle.

I was able to successfully implement the
print processor suggestion
using the print processor sample and
modifying emf.c to call
SetWorldTransform() with a Xform matrix
set to reflect about
the vertical and horizontal and adjust the
origin to place the
print within the page rect before calling
GDIPlayPageEmf().

However, I noticed that other features of this
sample involved
with orientation such as Landscape mode or
NUp printing do
not get executed when the printer settings
are selected. I tested
NUp with MS Word and get the desired
NUp printed output, but
following the code in the debugger the NUp
routines are bypassed
due to dwTotalNumberOfPages always
equals 1, even when
printing multiple pages. Is there an
explanation for this?

The reason for the above concern is that I
would like to
make the page rotation a selectable feature.
Can printer
settings/configurations be retrieved from
within the print
processor with a call such as GetPrinter()?

Re: Page Orientation capability in Unidrv? Reverse Print direction

Or is there
another method for this type of request?
Such as within
PPRINTPROCESSOROPENDATA passed
to OpenPrintProcessor()
and the GetKeyValue() routine to look for
keys within the
pParameters of the
PPRINTPROCESSOROPENDATA?

Thanks Dan for the DrvStartBanding and
DrvNextBand
suggestion. I would like to look at this idea
further as
some of our customers prefer to print direct,
no spooling,
thus raw data and no emf data type. I am
developing
this driver for a family of our thermal
printers which use
an ESC/POS printer language, basically
printing raster
scan lines. Therefore this option looks like it
has potential.
However I need to understand how the
POINTL and
SURFOBJ variables operate within the
Banding routines
to understand how to set the starting scan
line and the
direction.

I built and installed the watermarkuni
sample to follow
through the code in the debugger of Visual
Studio 2005,
could hit break points in OEMEnablePDEV,
OEMStartDoc,
OemEndDoc. But could not hit break points
in the
OEMStartBanding and OEMNextBand
routines.
Is there something else needed to enable
these routines?

Thanks for all your assistance.
DaveK

"Dan Negrescu" wrote:

Re: Page Orientation capability in Unidrv? Reverse Print direction

Not sure to fully understand what are you saying here? If you meant to say that my solution applies if the device is a raster printer and the driver on host should perform the rasterization then yes that is correct.

My reading of the OP was about achieving bottom-up page orientation for a printer using a driver based on the Unidrv, and the printer doesn't have the ability to perform a full page rotation in the device, so I concluded that the rasterization would take place on the host.

The Unidrv supports raster and vector (HPGL/2 and some versions of PCL) devices. Usually a PCL or HPGL device has full access to the page content prior to print and it is quite easy to implement the page reversal (by using printer language commands), but not every printer out there has a PCL parser (or any high level printer language) built in.

I didn't mean to dismiss the print processor solution (this is a viable solution) but to mere present another way to achieve the requirements (that might be, in some cases, appropriate).

Dan.

"Vipin [MVP]" wrote:

Re: Page Orientation capability in Unidrv? Reverse Print direction

But that
would mean
you would
rasterise on
the
windows
machine
using the
GDI
graphics
engine.

--

Vipin
Aravind

"Dan
Negrescu"
<Dan
Negrescu@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in
message
news:D1DF06F0-AF94-4C24-91FF-296F0A57E871@xxxxxxxxxxxx

Sorry
for
jumping
in
late,
but
there
is
another
way
to
achieve
this
from
within
your
driver
(no
print
processor
magic
required).
You
would
have
to
hook/implement

Re: Page Orientation capability in Unidrv? Reverse Print direction

Re: Page Orientation capability in Unidrv? Reverse Print direction

the
DrvStartBanding
and
DrvNextBand
APIs
and
fill
the
"pptl"
structure
members
in
such
way
that
the
banding
is
performed
from
bottom-up
(instead
of
the
default
top-down).
This
way
you
get
access
to
the
bottom
of
the
page
first
and
build
your
page
up
to
the
top.
The
only
thing
left
is

Re: Page Orientation capability in Unidrv? Reverse Print direction

hook/implement
the
IPrintOemUni::FilterGraphics
method
(or
IPrintOemUni::ImageProcessing)
and
flip
the
scan
lines
inside
each
band.
This
way
you
don't
need
to
worry
about
the
spool
format
being
EMF
or
RAW.

Dan.

"DaveK"
wrote:

I
found
out
why
the
RAW
data
type
was
being
used
instead
of
EMF,
the

Re: Page Orientation capability in Unidrv? Reverse Print direction

printer
attributes
were
set
to
DIRECT.
I
re-enabled
spooling
and
got
the
EMF
data
type
back.

Is
it
true
that
the
data
type
must
be
EMF
to
perform
any
page
re-orientations,
and
print
a
page
in
reverse?

Thanks
DaveK

"DaveK"
wrote:

Thanks
Vipin,

After

reviewing
the
sample
further
I
see
that
the
EMF
data
type
has
the
ability
to
orient
the
printed
output
with
the
use
of
rect
variables
and
GDI
routines.

I
have
built
and
installed
the
sample
to
follow
program
execution
with
the
VC7
debugger.
It
appears
our
printer
is
set
as

Re: Page Orientation capability in Unidrv? Reverse Print direction

using
the
Raw
data
type
and
the
data
read
from
the
spooler
is
already
formatted
as
PJL
commands.
Is
there
a
way
to
set
the
data
type
to
EMF
for
our
printer?

I
tried
to
set
this
to
an
EMF
data
type
in
the
printer
properties
page
under
the
print

Re: Page Orientation capability in Unidrv? Reverse Print direction

processor
settings,
but
when
closing
the
property
page
a
message
appears
indicating
the
properties
could
not
be
saved.

Thanks
for
your
help
on
this,
you
have
been
a
valuable
resource
in
pointing
me
in
the
right
direction
as
info
on
this
subject
is
hard
to
come
by.

DaveK

Re: Page Orientation capability in Unidrv? Reverse Print direction

"Vipin
[MVP]"
wrote:

What
I
am
saying
is
that,
you
can
use
the
print
processor
to
accomplish
the
rotating
and
it
would
just
work
fine
for
you.
You
don't
have
to
be
bothered
about
datatypes,
as
they
are
not
relevant
to
the
problem
in
question.
But
make
sure
the

Re: Page Orientation capability in Unidrv? Reverse Print direction

spooler
does
emf
spooling,
that
is
anyway
the
default
setting.

--
Vipin
Aravind

"DaveK"
<dave.kenyon@xxxxxxxxxxxxx(donot
wrote
in
message
<news:7FCFC11F-8025-4F0F-ADE>

Hi
Vipin,

Reviewing
the
sample
print
processor,
Genprint.dll,
I
see
the
data
being
sent
to
the
driver,
(Played),
in
the
data
type
files,
ie.
emf.c,
raw.c
and
text.c.

Re: Page Orientation capability in Unidrv? Reverse Print direction

The
example
most
related
to
what
I
am
trying
to
accomplish,
reverse
printing
the
page,
seems
to
be
the
methods
used
in
emf.c.

If
the
data
type
is
not
relevant,
where
else
in
the
print
processor
can
this
be
performed?
Are
you
simply
suggesting
that
each
data
type
can
perform

Re: Page Orientation capability in Unidrv? Reverse Print direction

the
reversing
operation,
or
is
there
a
method
to
perform
this
operation
independently
of
the
data
type?

Thanks
for
your
insight
and
help.
DaveK

"Vipin
[MVP]"
wrote:

Dave,
That's
unrelated,
your
printer
doesn't
need
to
be
using
emf
format
to
be
able
to
use
the
print
processor.

Re: Page Orientation capability in Unidrv? Reverse Print direction

The
print
processor
would
rotate
and
play
it
back
to
the
driver,
so
driver
gets
the
adjusted
coordinates
to
work
on.

--

Vipin
Aravind

"DaveK"
<dave.kenyon@xxx>
wrote
in
message
<news:81FD18A2-A>

Thanks
Vipin,

As
you
suggested
it
Looks
I
will
need
to
create
a
print
processor.
Our
printer

Re: Page Orientation capability in Unidrv? Reverse Print direction

18

does
not
have
enough
memory
to
store
a
page
and
rotate
it
internally
so
we
would
like
to
rotate
the
page
before
it
is
sent
to
the
printer.

Reviewing
the
sample
print
processor
Genprint.dll
I
see
the
emf.c
file
has
code
to
change
page
orientation
and
Nup
page
settings.
However,

I
am
not
sure
if
our
printer
is
using
the
emf
data
type.
Checking
DRIVER_IN
with
GetPrinterDr
for
our
printer
driver
there
is
no
default
data
type
set
.
Is
there
a
way
to
determine
which
data
type
our
driver
is
using?

If
the
data
type
is
EMF,
in
PlayEMFPag

Re: Page Orientation capability in Unidrv? Reverse Print direction

of
emf.c
is
rectPrinter
in
the
call
to
GdiPlayPage
the
ideal
method
for
reversing
the
orientation
of
the
page?

Thanks
DaveK

"Vipin
[MVP]"
wrote:

What
you
are
looking
for
is
a
job
of
printing
proc
not
a
driver
If
the
printer
can
support
com
and
driver

shou
emit
print
com
to
be
proc
on
the
devi
You
shou
look
at
the
devi
man
mos
pcl
print
have
this
capa
of
rotat
insic
it.