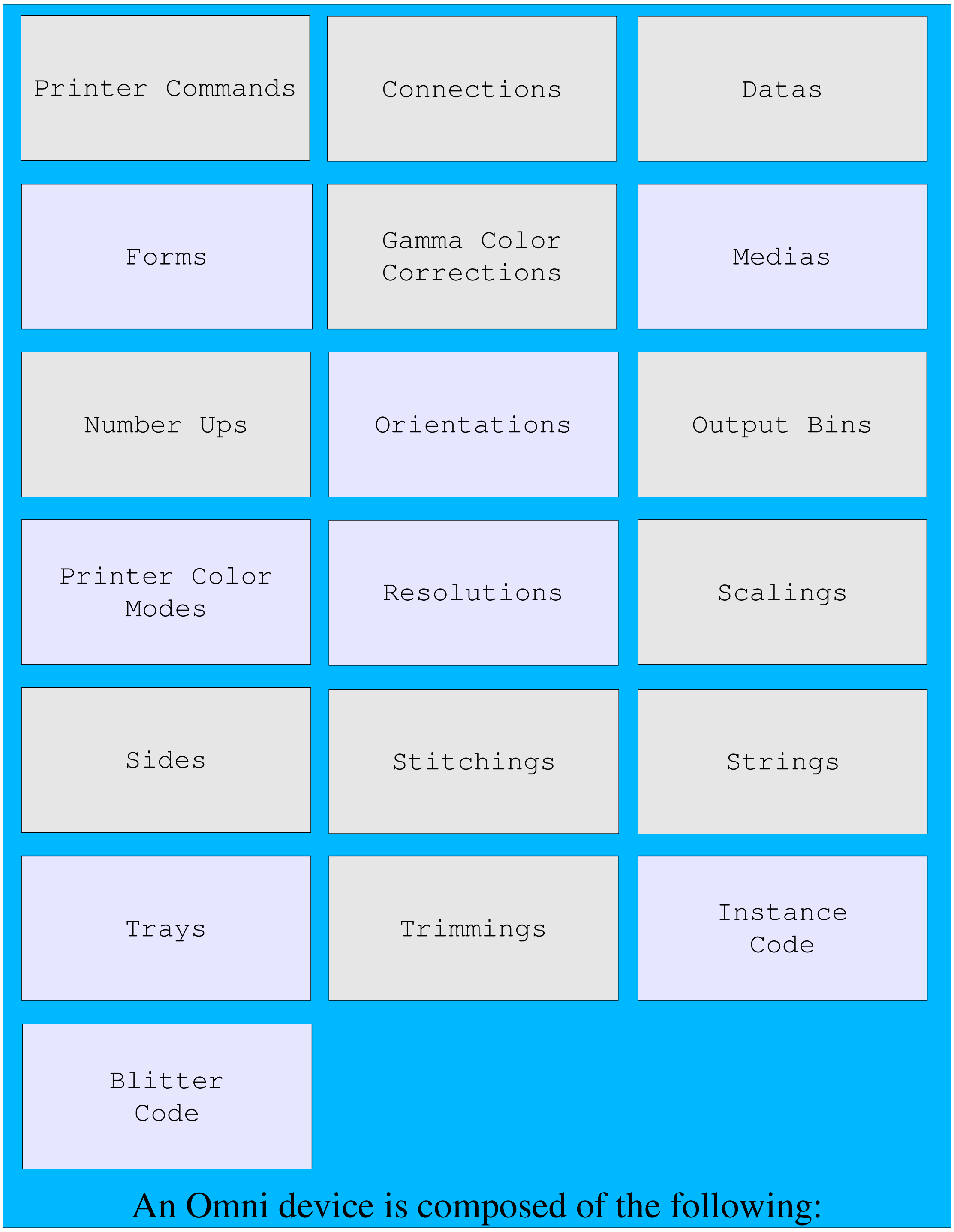
General Overview

The Omni driver is composed of many different devices. Each device contains a set of files that describe how the device operates and what features it contains. These files are self-contained and may be referenced by any device. This allows for data reuse. A set of devices that all share the same paper trays, for example, can each point to the same data file. When a bug is found in one feature, it will be fixed for each device that references that data file.



Printer Commands

Connections

Datas

Forms

Gamma Color Corrections

Medias

Number Ups

Orientations

Output Bins

Printer Color Modes

Resolutions

Scalings

Sides

Stitchings

Strings

Trays

Trimmings

Instance Code

Blitter Code

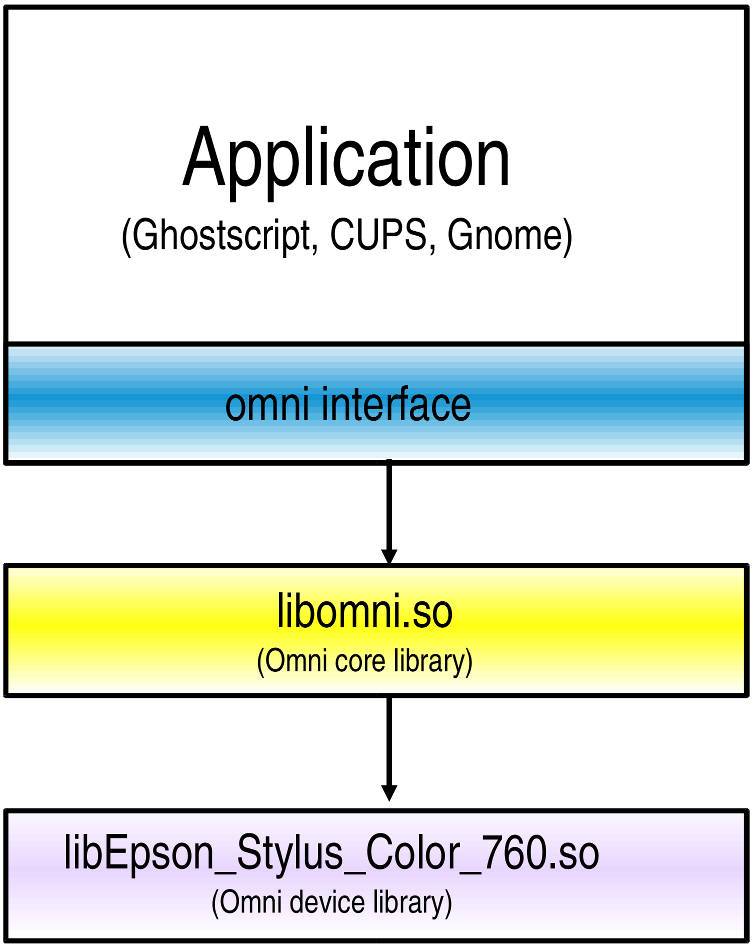
An Omni device is composed of the following:

There are two categories of information for a device in the Omni driver. The first category is a set of files that contain data that describe a device. This data is in XML format. The other category is a set of code routines that are executed when a job is submitted to the Omni driver. The code is written in C++ and is broken into two parts: code that handles instance data and code that translates bit mapped images into printer language. The code that handles instance data is also notified of events when a job is printed such as begin job, new page, and end job.

The Omni driver does not take your XML data and handle the print job for you. Instead, it takes the XML data and creates object oriented code for you. These objects are accessed from within your instance and blitter code.

Linkage Overview

The Omni driver was designed as an architecture for a general printer driver under Linux. It allows applications to print to many different devices under one common API. The common method is for an application to submit an already formatted Postscript job to a Postscript renderer such as Ghostscript. However, this is not the only way to print to Omni. If an application can create a raster image of the page, then the application can send that bit mapped image directly to Omni. Other examples of applications that can link directly to Omni are CUPS and Gnome.



The omni device library can be one of two different classes. The first class of device library source is a compiled device. This library is created from an XML data file that describes a device which is turned into C++ code which is then compiled into a linkable library. The second class of device library source is a generic library that reads an XML data file that describes a device. This library is a general purpose library that reads the data that describes a device directly and simulates the results that would have been obtained from executing the compiled library.

Source Code Overview

Download the latest source code from http://sourceforge.net/project/showfiles.php?group_id=18713 . Unpack the source tar ball and change to the Omni directory. Devices are separated by manufacturer into directories. A file called "devicesAll.list" in the root directory tells the build process what directories it needs to examine. In each directory, there is a file called "Device List". This file tells the build process which devices it needs to build.

The file, "devicesAll.list", is a shell script that sets one environment variable, DEVICELIST. This variable contains a space separated list of names. Each name is the name of a manufacturer directory in a special format.

In this format, spaces are converted to underscores ('_') and at the end of the name is a tilde ('~').

The file, “Device List”, is a text file that contains devices names. Each device name is on a separate line. Lines that start with the pound sign ('#') are ignored.

In order to add a device to the Omni driver, you need to perform the following:

- If the manufacturer's directory does not exist, then create it and add it to “devicesAll.list”. Run the script “./setupDevices add”.
- Add the device name to the “Device List” file in the correct directory.

Run the script “./setupOmni”. This should configure the build for your system and then build the Omni driver.

Standard Job Properties Overview

Job properties are properties that can change on a per-job basis. There is one property for each feature that your printer supports. Some examples of job properties are what form (letter, legal, A4) your job is printed on, what orientation is used (portrait, landscape), what resolution is used (300x300, 150x150), what media is used (plain, transparency), what tray the paper is pulled from (tray-1, auto), and what output bin the paper is sent to (top, bottom). The job properties that are common among different printer driver are being standardized at the Free Software Group (www.openprinting.org). When these job properties are referenced in this document, they will use the SJP acronym.

Device Overview

For each device, there is an XML file. This file describes what components comprise the device. For example, for the device “New Device”, there is a device file called “New Device.xml”. This XML file has a root node called <Device>. Within that node, there are some name space attributes and one other attribute called device. This attribute has the value of “New Device”. Child nodes of <Device> have the following tags:

XML tag	***description***
DriverName	The name of the driver
Capability	Optional capabilities that an application can query. There can be 0 or more of these tags.
RasterCapability	Optional capabilities that the raster engine can query. There can be 0 or more of these tags.

XML tag	***description***
DeviceOptions	Optional capabilities that your printer code can query to help distinguish between different devices. There can be 0 or more of these tags.
PDL	Programmatic description of your device. This has four attributes: “level”, “sublevel”, “major”, and “minor”. See the file PDL.hpp for expected values.
Uses	This is an alias of Has. See Has for a definition.
Has	Loads an external XML file into your device.
Instance	Points to a C++ header file or a C++ code file that controls the device's instance code. There are two of these tags. One for the header file and one for the code file. If this is present, then Blitter must follow and PluggableInstance or PluggableBlitter cannot occur.
Blitter	Points to a C++ header file or a C++ code file that controls the device's blitter code. There are two of these tags. One for the header file and one for the code file. If this is present, then Instance must already has been seen and PluggableInstance or PluggableBlitter cannot occur.
PluggableInstance	Points to a executable program that will control the device's instance and blitter code. If this is present, then PluggableBlitter must follow and Instance or Blitter cannot occur.
PluggableBlitter	Empty. PluggableInstance has all of the information. If this is present, then PluggableBlitter must already has been seen and Instance or Blitter cannot occur.

XML tag	***description***
DefaultJobProperties	Contains the device's initial job properties.

DefaultJobProperties structure:

XML tag	***description***
Copies	The default element in the corresponding XML file.
dither	The default element in the corresponding XML file.
Form	The default element in the corresponding XML file.
omniForm	This is an optional element which informational in nature and should not be used.
media	The default element in the corresponding XML file.
deviceNumberUp	The default element in the corresponding XML file.
Rotation	The default element in the corresponding XML file.
omniOrientation	This is an optional element which informational in nature and should not be used.
OutputBin	The default element in the corresponding XML file.
printmode	The default element in the corresponding XML file.
Resolution	The default element in the corresponding XML file.
omniResolution	This is an optional element which informational in nature and should not be used.
deviceScaling	The default element in the corresponding XML file.
SheetCollate	The default element in the corresponding XML file.
Sides	The default element in the corresponding XML file.
InputTray	The default element in the corresponding XML file.
omniTray	This is an optional element which informational in nature and should not be used.

XML tag	***description***
Trimming	The default element in the corresponding XML file.

The following is an example of a new device XML file.

New Device.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
      IBM Omni driver
      Copyright (c) International Business Machines Corp., 2000

      This library is free software; you can redistribute it and/or modify
      it under the terms of the GNU Lesser General Public License as published
      by the Free Software Foundation; either version 2.1 of the License, or
      (at your option) any later version.

      This library is distributed in the hope that it will be useful,
      but WITHOUT ANY WARRANTY; without even the implied warranty of
      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
      the GNU Lesser General Public License for more details.

      You should have received a copy of the GNU Lesser General Public License
      along with this library; if not, write to the Free Software
      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-->
<Device xmlns="http://www.ibm.com/linux/ltc/projects/omni/" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" name="New Device" xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../OmniDevice.xsd">
   <DriverName>New Manufacturer</DriverName>
   <Capability type="COLOR"/>
   <Capability type="MONOCHROME"/>
   <Capability type="MIRROR"/>
   <RasterCapabilities type="TOP_TO_BOTTOM"/>
   <DeviceOptions type="USB_PORT_SUPPORT"/>
   <PDL level="PDL_Epson" sublevel="LEVEL_ESCP_2" major="1" minor="0"/>
   <Uses>New Device Commands.xml</Uses>
   <Has>New Device Copies.xml</Has>
   <Has>New Device Forms.xml</Has>
   <Has>New Device Medias.xml</Has>
   <Has>New Device Number Ups.xml</Has>
   <Has>New Device Orientations.xml</Has>
   <Has>New Device Output Bins.xml</Has>
   <Has>New Device Print Modes.xml</Has>
   <Has>New Device Resolutions.xml</Has>
   <Has>New Device Scalings.xml</Has>
   <Has>New Device Sheet Collations.xml</Has>
   <Has>New Device Sides.xml</Has>
   <Has>New Device Stitchings.xml</Has>
   <Has>New Device Trays.xml</Has>
   <Has>New Device Trimmings.xml</Has>
   <Instance>New Device Instance.hpp</Instance>
   <Instance>New Device Instance.cpp</Instance>
   <Blitter>New Device Blitter.hpp</Blitter>
   <Blitter>New Device Blitter.cpp</Blitter>
   <DefaultJobProperties>
      <Copies>1</Copies>
      <dither>DITHER_STUCKI_DIFFUSION</dither>
```

```
50:       <Form>na_letter_8.50x11.00in</Form>
51:       <media>MEDIA_PLAIN</media>
52:       <deviceNumberUp>
53:          <NumberUp FORMAT="XbyY">
54:             <x>1</x>
55:             <y>1</y>
56:          </NumberUp>
57:
<NumberUpPresentationDirection>TorightTobottom</NumberUpPresentation
Direction>
58:       </deviceNumberUp>
59:       <Rotation>Portrait</Rotation>
60:       <OutputBin>Top</OutputBin>
61:       <printmode>PRINT_MODE_24_CMYK</printmode>
62:       <Resolution>360x360</Resolution>
63:       <deviceScaling>
64:          <ScalingPercentage>0</ScalingPercentage>
65:          <ScalingType>None</ScalingType>
66:       </deviceScaling>
67:       <SheetCollate>SheetUncollated</SheetCollate>
68:       <Sides>OneSidedFront</Sides>
69:       <deviceStitching>
70:          <StitchingPosition>0</StitchingPosition>
71:          <StitchingReferenceEdge>Top</StitchingReferenceEdge>
72:          <StitchingType>None</StitchingType>
73:          <StitchingCount>0</StitchingCount>
74:          <StitchingAngle>0</StitchingAngle>
75:       </deviceStitching>
76:       <InputTray>Continuous</InputTray>
77:       <Trimming>None</Trimming>
78:    </DefaultJobProperties>
79: </Device>
```

On line 21, the driver name (Manufacturer's name) is specified in the tag and on line 20, the device name is specified with the name= attribute.

On line 22 is the start of the capability flags. These are defined in Capability.hpp. They are returned to an application to simply determine the capabilities of a device.

On line 25 is the start of the raster capability flags. These are defined in RasterCapabilities.hpp. These will be used in the future by the raster engine when it sends bit mapped data to the device.

On line 26 is the start of device options. These options are queried by the device specific code to determine which category that this device falls into.

On line 27 is the PDL tag. This is a simple classification of what the raster language type is that this device uses.

On lines 28-42 is the meat of the device definition. It lists the individual files of features that this device uses. Each file is an XML file that describes a particular feature such as what forms a device can print on or what input trays that it can feed paper from, etc.

On lines 43-44 and 45-46 are the definition of what instance and blitter code that the device uses.

On line 47 is the structure of what defaults are for each of the allowable job properties. These job properties must be defined as acceptable for this device in the above XML files (lines 28-42).

Commands for a device

XML tag	***description***
command	A command for a device.

New Device Commands.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it 
and/or modify
 7:      it under the terms of the GNU Lesser General Public License 
as published
 8:      by the Free Software Foundation; either version 2.1 of the 
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be 
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty 
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General 
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 
02111-1307 USA
19: -->
20:
21: <deviceCommands
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:    
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <command name="cmdSetColor">_ESC_ &quot;r%d&quot;</command>
26:    <command name="cmdSetDotSizeInt">_ESC_ &quot;(e&quot; HEX(02) 
HEX(00) HEX(00) &quot;%c&quot;</command>
27:    <command name="cmdSetDotSize">_ESC_ &quot;(e&quot; HEX(02) 
HEX(00) &quot;%w&quot;</command>
28:    <command name="cmdSetFormSize">_ESC_ &quot;(C&quot; HEX(02) 
HEX(00) &quot;%W&quot;</command>
29:    <command name="cmdSetPageSize">_ESC_ &quot;(S&quot; HEX(08) 
HEX(00) &quot;%W&quot; HEX(00) HEX(00) &quot;%W&quot; HEX(00) HEX
(00)</command>
30:    <command name="cmdSetHardwareMicroweave">_ESC_ &quot;(i&quot; 
HEX(01) HEX(00) &quot;%c&quot;</command>
31:    <command name="cmdUSBInit">HEX(00) HEX(00) HEX(00) _ESC_ HEX
(01) &quot;@EJL&quot; HEX(20) &quot;1284.4&quot; _LF_ 
&quot;@EJL&quot; HEX(20) HEX(20) HEX(20) HEX(20) HEX(20) 
_LF_</command>
32:    <command name="cmdInit">_ESC_ &quot;@&quot; _ESC_ 
&quot;@&quot;</command>
33:    <command name="cmdRemote">_ESC_ &quot;(R&quot; HEX (08) HEX
(00) HEX(00) &quot;REMOTE1PM&quot; HEX (02) HEX (00) HEX (00) HEX
(00) &quot;SN&quot; HEX (03) HEX (00) HEX (00) HEX (00) HEX (01) 
&quot;SN&quot; HEX (03) HEX (00) HEX (00) HEX (01) HEX (00) 
&quot;SN&quot; HEX (03) HEX (00) HEX (00) HEX (02) HEX (00) _ESC_ 
HEX (00) HEX (00) HEX (00)</command>
```

```
34:    <command name="cmdUnknown">_ESC_ HEX (19)
&quot;1&quot;</command>
35:    <command name="cmdTerm">_ESC_ HEX (40)</command>
36:    <command name="cmdReset">_ESC_ HEX (40)</command>
37:    <command name="cmdAbort">_FF_ _ESC_ HEX (40)</command>
38:    <command name="cmdPageEject">_FF_</command>
39:    <command name="cmdSetXPos">_ESC_ HEX (24) HEX (25) HEX (77)
</command>
40:    <command name="cmdSetYPos">_ESC_ HEX (28) HEX (56) HEX (02)
_NUL_ HEX (25) HEX (77)</command>
41:    <command name="cmdBidi">_ESC_ HEX (55) HEX (25) HEX (63)
</command>
42:    <command name="cmdBeginRasterGraphics">_ESC_ HEX (28) HEX
(47) HEX (01) _NUL_ HEX (01)</command>
43:    <command name="cmdEndRasterGraphicsLine">_CR_</command>
44:    <command
name="cmdMoveToNextRasterGraphicsLine">_LF_</command>
45:    <command name="cmdSetResolution">_ESC_ &quot;(U&quot; HEX
(05) HEX (00) &quot;%c%c%c%w&quot;</command>
46:    <command name="cmdSetPageFormFactor">_ESC_ &quot;(S&quot; HEX
(08) HEX (00) &quot;%d%d&quot;</command>
47:    <command name="cmdSetSpecialResolution">_ESC_ &quot;(D&quot;
HEX (04) HEX (00) &quot;%w%c%c&quot;</command>
48:    <command name="cmdVariRaster">_ESC_ &quot;i%c%c%c%w%
w&quot;</command>
49:    <command name="cmdPositionX">_ESC_ &quot;($&quot; HEX (04)
HEX (00) HEX (00) HEX (00) HEX (00) HEX (00)</command>
50:    <command name="cmdPositionX1">_ESC_ &quot;($&quot; HEX (04)
HEX (00) &quot;%c%c%c%c&quot;</command>
51:    <command name="cmdMoveRelativeY">_ESC_ &quot;(v&quot; HEX
(04) HEX (00) &quot;%d&quot;</command>
52:    <command name="cmdSetSixthLineSpacing">_ESC_
&quot;2&quot;</command>
53:    <command name="cmdSetPageLengthInLines">_ESC_ &quot;C%
c&quot;</command>
54: </deviceCommands>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceCommands>.

On lines 25-53 are the commands that are valid for this device. The format for elements in the structure is the name of the tag (<command>) followed by an attribute (name=) which contains the name of the command. The command data is between the start and end command tags. The command data can be composed of one or more of the following:

ESC	- hexadecimal 0x1b
_FF_	- 0x0c
_LF_	- 0x0a
_CR_	- 0x0d
_EM_	- 0x19
_FS_	- 0x1c
CAN	- 0x18
NUL	- 0x00
HEX (xx)	- any hexadecimal value 2 hex digits long
ASCII (c)	- an ASCII character c.
"string"	- an ASCII string

Usage:

The command structure comes from the Device's getCurrentCommand method. Individual commands are then accessed by calling the DeviceCommand's getCommandData method. You can then send the command by itself via Device's sendBinaryDataToDevice method, or you can use a printf style call with Device's sendPrintfToDevice method. Valid printf arguments are:

```
   %d - little endian 32-bit value
   %D - big    endian 32-bit value
   %w - little endian 16-bit value
   %W - big    endian 16-bit value
   %c -                8-bit value
   %% - the % character
```

an example would be:

```
   DeviceCommand *pCommands = getCommands ();
   BinaryData    *pCmd      = 0;

   pCmd = pCommands->getCommandData ("cmdSetColor");
   pDevice_d->sendPrintfToDevice (pCmd, 4);
```

Connections for a device

This is a unused device feature. It exists to provide a mapping from a number of separate but related features into one easy to select item.

XML tag	***description***
name	The name of the connection.
connectionForm	What form the connection uses.
omniForm	This is an optional element which informational in nature and should not be used.
connectionTray	What tray the connection uses.
omniTray	This is an optional element which informational in nature and should not be used.
connectionMedia	What media the connection uses.

New Device Connections.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
     IBM Omni driver
     Copyright (c) International Business Machines Corp., 2000

     This library is free software; you can redistribute it and/or modify
     it under the terms of the GNU Lesser General Public License as published
     by the Free Software Foundation; either version 2.1 of the License, or
     (at your option) any later version.
```

```
 10:
 11:      This library is distributed in the hope that it will be
useful,
 12:      but WITHOUT ANY WARRANTY; without even the implied
warranty of
 13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
 14:      the GNU Lesser General Public License for more details.
 15:
 16:      You should have received a copy of the GNU Lesser General
Public License
 17:      along with this library; if not, write to the Free
Software
 18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
 19: -->
 20: <deviceConnections
xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
 21:    <deviceConnection>
 22:       <name>EPSON_CONN_ID_1</name>
 23:       <connectionForm>na_letter_8.50x11.00in</connectionForm>
 24:       <connectionTray>Continuous</connectionTray>
 25:       <connectionMedia>MEDIA_PLAIN</connectionMedia>
 26:    </deviceConnection>
 27:    <deviceConnection>
 28:       <name>EPSON_CONN_ID_2</name>
 29:       <connectionForm>na_legal_8.50x14.00in</connectionForm>
 30:       <connectionTray>Continuous</connectionTray>
 31:       <connectionMedia>MEDIA_PLAIN</connectionMedia>
 32:    </deviceConnection>
     ...
309: </deviceConnections>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On line 20 is the root tag of what XML data is contained in this file. The root tag is <deviceConnections>.

On lines 21-26 is one connection that is defined for this device. The data about the connection is enclosed within the <deviceConnection> and </deviceConnection> tags.

On lines 27-32 is another connection that is defined for this device.

The last connection ends at line 308 and line 309 is the end of the root tag.

Copies for a device

This provides information about what kind of copies a device can support. Copies can be supported either through a hardware command or through software simulation. If simulation is requested, then the Omni driver will send multiple copies of the pages to the device.

XML tag	***description***
minimum	The minimum number of copies that this device supports.

XML tag	***description***
maximum	The maximum number of copies that this device supports.
command	The command that corresponds to this element.
simulationRequired	Asks the Omni driver to simulate this feature to the device.

<u>New Device Copies.xml</u>

```
  1: <?xml version="1.0" encoding="UTF-8"?>
  2: <!--
  3:      IBM Omni driver
  4:      Copyright (c) International Business Machines Corp., 2000
  5:
  6:      This library is free software; you can redistribute it
and/or modify
  7:      it under the terms of the GNU Lesser General Public
License as published
  8:      by the Free Software Foundation; either version 2.1 of the
License, or
  9:      (at your option) any later version.
 10:
 11:      This library is distributed in the hope that it will be
useful,
 12:      but WITHOUT ANY WARRANTY; without even the implied
warranty of
 13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
 14:      the GNU Lesser General Public License for more details.
 15:
 16:      You should have received a copy of the GNU Lesser General
Public License
 17:      along with this library; if not, write to the Free
Software
 18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
 19: -->
 20: <deviceCopies
 21:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
 22:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
 23:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
 24:    <deviceCopy>
 25:       <minimum>1</minimum>
 26:       <maximum>10</maximum>
 27:       <command>_NUL_</command>
 28:       <simulationRequired>false</simulationRequired>
 29:    </deviceCopy>
 30: </deviceCopies>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On line 20 is the root tag of what XML data is contained in this file. The root tag is <deviceCopies>.

On lines 24-29 is the copies information that is defined for this device. The data about the copies is enclosed within the <deviceCopy> and </deviceCopy> tags.

an example would be:

```
   DeviceCopies *pDC              = getDeviceCopies ();
   int           iNumCopies       = 0;

   if (pDC)
   {
      iNumCopies = pDC->getNumCopies ();
   }
```

Data for a device

XML tag	***description***
deviceData	A data entry for a device.

New Device Data.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20:
21: <deviceDatas
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <deviceData name="Nozzle_Spacing"
type="integer">6</deviceData>
26:    <deviceData name="Nozzle_Number"
type="integer">48</deviceData>
27:    <deviceData name="Positioning_x"
type="integer">360</deviceData>
28:    <deviceData name="DotSize360" type="integer">18</deviceData>
29:    <deviceData name="DotSize720" type="integer">17</deviceData>
30:    <deviceData name="DotSize1440" type="integer">16</deviceData>
31: </deviceDatas>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceDatas>.

On lines 25-30 are the data entries that are valid for this device. The format for elements in the structure is the name of the tag (<deviceData>) followed by an attribute (name=) which contains the name of the data value followed by an attribute (type=) which describes how to interpret the contents. The type can be the following: string, boolean, integer, byte, binary, or bytearray.

an example would be:

```
   DeviceData *pData              = getDeviceData ();
   char       *pszDataTestString = 0;
   bool        fDataTestBool     = false;
   int         iDataTestInt      = 0;
   bool        fRc;

   if (pData)
   {
      fRc = pData->getStringData ("dataTestString",
&pszDataTestString);
      fRc = pData->getBooleanData ("dataTestBool", &fDataTestBool);
      fRc = pData->getIntData ("dataTestInt", &iDataTestInt);
   }
```

Forms for a device

XML tag	***description***
name	The job property value for this form.
omniName	This is an optional element which informational in nature and should not be used.
formCapabilities	It contains the programmatic description of the form. It is for future use.
command	The command that corresponds to this element.
hardCopyCap	See the below table for the description of this structure.
deviceID	An optional string identifier that will replace the id for the device job properties.

hardCopyCap structure:

XML tag	***description***
hardCopyCapLeft	The left unprintable area in thousands of a millimeter.
hardCopyCapTop	The top unprintable area in thousands of a millimeter.
hardCopyCapRight	The right unprintable area in thousands of a millimeter.

XML tag	***description***
hardCopyCapBottom	The bottom unprintable area in thousands of a millimeter.

<u>New Device Forms.xml</u>

```
  1: <?xml version="1.0" encoding="UTF-8"?>
  2: <!--
  3:      IBM Omni driver
  4:      Copyright (c) International Business Machines Corp., 2000
  5:
  6:      This library is free software; you can redistribute it
and/or modify
  7:      it under the terms of the GNU Lesser General Public
License as published
  8:      by the Free Software Foundation; either version 2.1 of the
License, or
  9:      (at your option) any later version.
 10:
 11:      This library is distributed in the hope that it will be
useful,
 12:      but WITHOUT ANY WARRANTY; without even the implied
warranty of
 13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
 14:      the GNU Lesser General Public License for more details.
 15:
 16:      You should have received a copy of the GNU Lesser General
Public License
 17:      along with this library; if not, write to the Free
Software
 18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
 19: -->
 20: <deviceForms
xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
 21:    <deviceForm>
 22:       <name>iso_a4_210.00x297.00mm</name>
 23:       <formCapabilities>NO_CAPABILITIES</formCapabilities>
 24:       <command>_NUL_</command>
 25:       <hardCopyCap>
 26:          <hardCopyCapLeft>3000</hardCopyCapLeft>
 27:          <hardCopyCapTop>8500</hardCopyCapTop>
 28:          <hardCopyCapRight>3000</hardCopyCapRight>
 29:          <hardCopyCapBottom>9220</hardCopyCapBottom>
 30:       </hardCopyCap>
 31:    </deviceForm>
 32:    <deviceForm>
 33:       <name>iso_b5_176.00x250.00mm</name>
 34:       <formCapabilities>NO_CAPABILITIES</formCapabilities>
 35:       <command>_NUL_</command>
 36:       <hardCopyCap>
 37:          <hardCopyCapLeft>3000</hardCopyCapLeft>
 38:          <hardCopyCapTop>8500</hardCopyCapTop>
 39:          <hardCopyCapRight>3000</hardCopyCapRight>
 40:          <hardCopyCapBottom>9200</hardCopyCapBottom>
 41:       </hardCopyCap>
 42:    </deviceForm>
      ...
236:    </deviceForm>
237: </deviceForms>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On line 20 is the root tag of what XML data is contained in this file. The root tag is <deviceForms>.

On lines 21-31 is one form that is defined for this device. The data about the form is enclosed within the <deviceForm> and </deviceForm> tags.

On line 22 is the <name> tag. It contains the SJP identifier for this form.

On line 23 is the <formCapabilities> tag. It contains the programmatic description of the form. It is for future use.

On line 24 is the <command> tag. It contains binary command data (described above) for this form.

On lines 25-30 is the hard copy capabilities for the form. The data is enclosed within the <hardCopyCap> and </hardCopyCap> tags.

On line 26 is information about the left unprintable area. The units are in thousands of a millimeter.

On line 27 is information about the top unprintable area. The units are in thousands of a millimeter.

On line 28 is information about the right unprintable area. The units are in thousands of a millimeter.

On line 29 is information about the bottom unprintable area. The units are in thousands of a millimeter.

an example would be:

```
DeviceForm  *pDF  = getCurrentForm ();
HardCopyCap *pHCC = pDF->getHardCopyCap ();

if (pDF->isEqual (“form=iso_a4”))
{
   ...
}
```

Gamma color corrections for a device

XML tag	***description***
Resolution	The resolution that corresponds to this gamma correction.
omniResolution	This is an optional element which informational in nature and should not be used.
media	The media that corresponds to this gamma correction.

XML tag	***description***
printMode	The print mode that corresponds to this gamma correction.
gammaTableDitherCatagory	The category of dither values that corresponds to this gamma correction.
gammaTableCGamma	The cyan gamma value in tenths.
gammaTableMGamma	The magenta gamma value in tenths.
gammaTableYGamma	The yellow gamma value in tenths.
gammaTableKGamma	The black gamma value in tenths.
gammaTableCBias	The cyan gamma bias.
gammaTableMBias	The magenta gamma bias.
gammaTableYBias	The yellow gamma bias.
gammaTableKBias	The black gamma bias.

New Device Gammas.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
     IBM Omni driver
     Copyright (c) International Business Machines Corp., 2000

     This library is free software; you can redistribute it and/or modify
     it under the terms of the GNU Lesser General Public License as published
     by the Free Software Foundation; either version 2.1 of the License, or
     (at your option) any later version.

     This library is distributed in the hope that it will be useful,
     but WITHOUT ANY WARRANTY; without even the implied warranty of
     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
     the GNU Lesser General Public License for more details.

     You should have received a copy of the GNU Lesser General Public License
     along with this library; if not, write to the Free Software
     Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-->
<deviceGammaTables xmlns="http://www.ibm.com/linux/ltc/projects/omni/" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../OmniDevice.xsd">
   <deviceGammaTable>
      <Resolution>360x360</Resolution>
      <media>MEDIA_PLAIN</media>
      <printmode>PRINT_MODE_1_ANY</printmode>
      <gammaTableDitherCatagory>DITHER_CATAGORY_MATRIX</gammaTableDitherCatagory>
      <gammaTableCGamma>10</gammaTableCGamma>
```

```
 27:       <gammaTableMGamma>10</gammaTableMGamma>
 28:       <gammaTableYGamma>10</gammaTableYGamma>
 29:       <gammaTableKGamma>10</gammaTableKGamma>
 30:       <gammaTableCBias>0</gammaTableCBias>
 31:       <gammaTableMBias>0</gammaTableMBias>
 32:       <gammaTableYBias>0</gammaTableYBias>
 33:       <gammaTableKBias>0</gammaTableKBias>
 34:    </deviceGammaTable>
 35:    <deviceGammaTable>
 36:       <Resolution>360x360</Resolution>
 37:       <media>MEDIA_PLAIN</media>
 38:       <printmode>PRINT_MODE_1_ANY</printmode>
 39:
<gammaTableDitherCatagory>DITHER_CATAGORY_DIFFUSION</gammaTableDithe
rCatagory>
 40:       <gammaTableCGamma>10</gammaTableCGamma>
 41:       <gammaTableMGamma>10</gammaTableMGamma>
 42:       <gammaTableYGamma>10</gammaTableYGamma>
 43:       <gammaTableKGamma>7</gammaTableKGamma>
 44:       <gammaTableCBias>0</gammaTableCBias>
 45:       <gammaTableMBias>0</gammaTableMBias>
 46:       <gammaTableYBias>0</gammaTableYBias>
 47:       <gammaTableKBias>0</gammaTableKBias>
 58:    </deviceGammaTable>
     ...
831: </deviceGammaTables>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On line 20 is the root tag of what XML data is contained in this file. The root tag is <deviceGammaTables>.

On lines 21-34 is one gamma table that is defined for this device. The data about the gamma table is enclosed within the <deviceGammaTable> and </deviceGammaTable> tags.

On line 22 is the <Resolution> tag. It contains the SJP identifier for the resolution.

On line 23 is the <media> tag. It contains the SJP identifier for the media.

On line 24 is the <printMode> tag. It contains the identifier for the print mode.

On line 25 is the <gammaTableDitherCatagory> tag. This is the dither category. For right now, valid categories are DITHER_CATAGORY_MATRIX, DITHER_CATAGORY_DIFFUSION, DITHER_CATAGORY_HSV_DIFFUSION, DITHER_CATAGORY_CMYK_DIFFUSION, DITHER_CATAGORY_VOID_CLUSTER, DITHER_CATAGORY_NEW_DIFFUSION, DITHER_CATAGORY_NEW_MATRIX.

On line 26 is the <gammaTableCGamma> tag. This is the Cyan gamma.

On line 27 is the <gammaTableMGamma> tag. This is the Magenta gamma.

On line 28 is the <gammaTableYGamma> tag. This is the

Yellow gamma.

On line 29 is the <gammaTableKGamma> tag. This is the Black gamma.

On line 30 is the <gammaTableCBias> tag. This is the Cyan bias.

On line 31 is the <gammaTableMBias> tag. This is the Magenta bias.

On line 32 is the <gammaTableYBias> tag. This is the Yellow bias.

On line 33 is the <gammaTableKBias> tag. This is the Black bias.

an example would be:

```
DeviceGamma *pDG        = getCurrentGamma ();
int          iCyanGamma;

iCyanGamma = pDG->getCGamma ();
```

Medias for a device

XML tag	***description***
name	The job property value for this media.
command	The command that corresponds to this element.
mediaColorAdjustRequired	This is passed to the dithering routine.
mediaAbsorption	This is passed to the dithering routine.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Medias.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
```

```
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20:
21: <deviceMedias
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <deviceMedia>
26:       <name>MEDIA_PLAIN</name>
27:       <command>_NUL_</command>
28:       <mediaColorAdjustRequired>0</mediaColorAdjustRequired>
29:       <mediaAbsorption>MEDIA_HEAVY_ABSORPTION</mediaAbsorption>
30:    </deviceMedia>
31:    <deviceMedia>
32:       <name>MEDIA_SPECIAL_360</name>
33:       <command>_NUL_</command>
34:       <mediaColorAdjustRequired>0</mediaColorAdjustRequired>
35:       <mediaAbsorption>MEDIA_LIGHT_ABSORPTION</mediaAbsorption>
36:    </deviceMedia>
37:    <deviceMedia>
38:       <name>MEDIA_SPECIAL_720</name>
39:       <command>_NUL_</command>
40:       <mediaColorAdjustRequired>0</mediaColorAdjustRequired>
41:       <mediaAbsorption>MEDIA_LIGHT_ABSORPTION</mediaAbsorption>
42:    </deviceMedia>
43: </deviceMedias>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceMedias>.

On lines 25-30 is one form that is defined for this device. The data about the form is enclosed within the <deviceForm> and </deviceForm> tags.

On line 26 is the <name> tag. It contains the Omni media identifier for this media.

On line 27 is the <command> tag. It contains binary command data (described above) for this media.

On line 28 is the <mediaColorAdjustRequired> tag. It contains information for the dithering routine.

On line 29 is the <mediaAbsorption> tag. It contains information for the dithering routine.

an example would be:

```
   DeviceMedia *pDM = getCurrentMedia ();

   if (pDM->isEqual (“media=MEDIA_PLAIN”))
   {
      ...
   }
```

Number Ups for a device

XML tag	***description***
NumberUp	See the numberUp table.
NumberUpPresentationDirection	The direction of the logical pages on the physical page.
simulationRequired	Asks the Omni driver to simulate this feature to the device.
deviceID	An optional string identifier that will replace the id for the device job properties.

XML tag	***description***
x	The number of logical pages along the x axis.
y	The number of logical pages along the y axis.

New Device NUps.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
      IBM Omni driver
      Copyright (c) International Business Machines Corp., 2000

      This library is free software; you can redistribute it and/or modify
      it under the terms of the GNU Lesser General Public License as published
      by the Free Software Foundation; either version 2.1 of the License, or
      (at your option) any later version.

      This library is distributed in the hope that it will be useful,
      but WITHOUT ANY WARRANTY; without even the implied warranty of
      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
      the GNU Lesser General Public License for more details.

      You should have received a copy of the GNU Lesser General Public License
      along with this library; if not, write to the Free Software
      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-->

<deviceNumberUps
   xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
   xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
   xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../OmniDevice.xsd">
   <deviceNumberUp>
      <NumberUp FORMAT="XbyY">
         <x>1</x>
         <y>1</y>
      </NumberUp>
      <presentationDirection>TorightTobottom</presentationDirection>
      <command>_NUL_</command>
```

```
32:       <simulationRequired>false</simulationRequired>
33:       <deviceID>a</deviceID>
34:    </deviceNumberUp>
35:    <deviceNumberUp>
36:       <NumberUp FORMAT="XbyY">
37:          <x>2</x>
38:          <y>2</y>
39:       </NumberUp>
40:
<presentationDirection>TobottomToleft</presentationDirection>
41:       <command>_NUL_</command>
42:       <simulationRequired>false</simulationRequired>
43:       <deviceID>a</deviceID>
44:    </deviceNumberUp>
    ...
45:    <deviceNumberUp>
46:       <NumberUp FORMAT="XbyY">
47:          <x>2</x>
48:          <y>2</y>
49:       </NumberUp>
50:
<presentationDirection>TotopToright</presentationDirection>
51:       <command>_NUL_</command>
52:       <simulationRequired>false</simulationRequired>
53:       <deviceID>a</deviceID>
54:    </deviceNumberUp>
55: </deviceNumberUps>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceNumberUps>.

On lines 25-34 is one print mode that is defined for this device. The data about the print mode is enclosed within the <deviceNumberUp> and </deviceNumberUp> tags.

an example would be:

```
DeviceNUp *pDNup = getCurrentNUp ();

if (  1 == pDNup->getXPages ()
   && 1 == pDNup->getYPages ()
   )
{
   ...
}
```

Orientations for a device

Each device has a hardware supported orientation that it supports. It may only be portrait, or it may be many orientations. These orientations are described in an XML file. The XML file looks like the following:

XML tag	***description***
name	The job property value for this orientation.

XML tag	***description***
omniName	This is an optional element which informational in nature and should not be used.
simulationRequired	Asks the Omni driver to simulate this feature to the device.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Orientations.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20: <deviceOrientations
xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
21:    <deviceOrientation>
22:       <name>Portrait</name>
23:    </deviceOrientation>
24: </deviceOrientations>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On line 20 is the root tag of what XML data is contained in this file. The root tag is <deviceOrientations>.

On lines 21-24 is one orientation that is defined for this device.

On line 22 is the <name> tag. It contains the SJP identifier for this orientation.

an example would be:

```
DeviceOrientation pDO = getCurrentOrientation ();

if (pDO->isEqual (“Rotation=portrait”))
{
   ...
}
```

Print modes for a device

XML tag	***description***
name	The job property value for this printmode.
omniName	This is an optional element which informational in nature and should not be used.
printModePhysicalCount	How many physical colors (in bits) are there for this mode?
printModeLogicalCount	How many logical colors (in bits) are these for this print mode?
printModePlanes	How many color planes are there for the print mode?
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device PrintModes.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
      IBM Omni driver
      Copyright (c) International Business Machines Corp., 2000

      This library is free software; you can redistribute it and/or modify
      it under the terms of the GNU Lesser General Public License as published
      by the Free Software Foundation; either version 2.1 of the License, or
      (at your option) any later version.

      This library is distributed in the hope that it will be useful,
      but WITHOUT ANY WARRANTY; without even the implied warranty of
      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
      the GNU Lesser General Public License for more details.

      You should have received a copy of the GNU Lesser General Public License
      along with this library; if not, write to the Free Software
      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-->

<devicePrintModes
   xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
   xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
```

```
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <devicePrintMode>
26:       <name>PRINT_MODE_1_ANY</name>
27:       <printModePhysicalCount>1</printModePhysicalCount>
28:       <printModeLogicalCount>1</printModeLogicalCount>
29:       <printModePlanes>1</printModePlanes>
30:    </devicePrintMode>
31:    <devicePrintMode>
32:       <name>PRINT_MODE_8_CMYK</name>
33:       <printModePhysicalCount>4</printModePhysicalCount>
34:       <printModeLogicalCount>8</printModeLogicalCount>
35:       <printModePlanes>1</printModePlanes>
36:    </devicePrintMode>
37:    <devicePrintMode>
38:       <name>PRINT_MODE_24_CMYK</name>
39:       <printModePhysicalCount>4</printModePhysicalCount>
40:       <printModeLogicalCount>24</printModeLogicalCount>
41:       <printModePlanes>1</printModePlanes>
42:    </devicePrintMode>
43: </devicePrintModes>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <devicePrintModes>.

On lines 25-30 is one print mode that is defined for this device. The data about the print mode is enclosed within the <devicePrintMode> and </devicePrintMode> tags.

On line 26 is the <name> tag. It contains the identifier for this print mode.

On line 27 is the <printModePhysicalCount> tag. This describes how many colors there are that are (encoded into bits-per-pel) in the source bitmap.

On line 28 is the <printModeLogicalCount> tag. This describes how many colors there are that are (encoded into bits-per-pel) in the destination bitmap.

On line 29 is the <printModePlanes> tag. This describes how many planes of color data there are in the destination bitmap.

an example would be:

```
DevicePrintMode *pDPM = getCurrentPrintMode ();

if (1 == pDPM->getLogicalCount ())
{
   // Printer wants a monochrome surface.
}
```

Resolutions for a device

XML tag	***description***
name	The job property value for this resolution.
omniName	This is an optional element which informational in nature and should not be used.
xRes	The x resolution in dots per inch.
yRes	The y resolution in dots per inch.
xInternalRes	The x resolution in dots per inch before scaling.
yInternalRes	The y resolution in dots per inch before scaling.
command	The command that corresponds to this element.
resolutionCapability	It holds information for future use that can describe this resolution in a programmatic fashion.
resolutionDestinationBitsPerPel	This value is used by the dithering routine.
resolutionScanlineMultiple	This value is used by the dithering routine.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Resolutions.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
     IBM Omni driver
     Copyright (c) International Business Machines Corp., 2000

     This library is free software; you can redistribute it and/or modify
     it under the terms of the GNU Lesser General Public License as published
     by the Free Software Foundation; either version 2.1 of the License, or
     (at your option) any later version.

     This library is distributed in the hope that it will be useful,
     but WITHOUT ANY WARRANTY; without even the implied warranty of
     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
     the GNU Lesser General Public License for more details.

     You should have received a copy of the GNU Lesser General Public License
     along with this library; if not, write to the Free Software
     Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-->
<deviceResolutions
```

```
xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
21:    <deviceResolution>
22:       <name>360x360</name>
23:       <xRes>360</xRes>
24:       <yRes>360</yRes>
25:       <command>_ESC_ &quot;.%c%c%c%c%w&quot;</command>
26:       <resolutionCapability>0</resolutionCapability>
27:
<resolutionDestinationBitsPerPel>2</resolutionDestinationBitsPerPel>
28:
<resolutionScanlineMultiple>288</resolutionScanlineMultiple>
29:    </deviceResolution>
30:    <deviceResolution>
31:       <name>720x720</name>
32:       <xRes>720</xRes>
33:       <yRes>720</yRes>
34:       <command>_ESC_ &quot;.%c%c%c%c%w&quot;</command>
35:       <resolutionCapability>0</resolutionCapability>
36:
<resolutionDestinationBitsPerPel>2</resolutionDestinationBitsPerPel>
37:
<resolutionScanlineMultiple>288</resolutionScanlineMultiple>
38:    </deviceResolution>
39:    <deviceResolution>
40:       <name>1440x720</name>
41:       <xRes>1440</xRes>
42:       <yRes>720</yRes>
43:       <command>_ESC_ &quot;.%c%c%c%c%w&quot;</command>
44:       <resolutionCapability>0</resolutionCapability>
45:
<resolutionDestinationBitsPerPel>2</resolutionDestinationBitsPerPel>
46:
<resolutionScanlineMultiple>288</resolutionScanlineMultiple>
47:    </deviceResolution>
48: </deviceResolutions>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On line 20 is the root tag of what XML data is contained in this file. The root tag is <deviceResolutions>.

On lines 21-29 is one resolution that is defined for this device. The data about the resolution is enclosed within the <deviceResolution> and </deviceResolution> tags.

On line 22 is the <name> tag. It contains the SJP identifier for this resolution.

On line 23 is the <xRes> tag which describes the X resolution in dots per inch.

On line 24 is the <yRes> tag which describes the Y resolution in dots per inch.

On line 25 is the <command> tag. It contains binary command data (described above) for this resolution.

On line 26 is the <resolutionCapability> tag. It holds information for future use that can describe this resolution in a programmatic fashion.

On line 27 is the <resolutionDestinationBitsPerPel> tag. It holds information for the dithering routine.

On line 28 is the <resolutionScanlineMultiple> tag. It holds information for the blitter routine. This is the number of scanlines that the blitter codes wants at a time. The actual number of scanlines will be a multiple of this number. For example, for 10, it could be 10, 20, 30, etc...

an example would be:

```
   DeviceResolution *pDR = getCurrentResolution ();

   if (  360 == pDR->getXRes ()
      && 360 == pDR->getYRes ()
      )
   {
      ...
   }
```

Scalings for a device

XML tag	***description***
default	The default value that this type will be scaled to.
minimum	The minimum value that this type is allowed to be scaled to.
maximum	The maximum value that this type is allowed to be scaled to.
command	The command that corresponds to this element.
allowedType	What percentage that this page is scaled to.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Scalings.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
```

```
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20:
21: <deviceScalings
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <deviceScaling>
26:       <default>50</default>
27:       <minimum>1</minimum>
28:       <maximum>100</maximum>
29:       <command>_NUL_</command>
30:       <allowedType>FitToPage</allowedType>
31:       <deviceID>a</deviceID>
32:    </deviceScaling>
33:    <deviceScaling>
34:       <default>73</default>
35:       <minimum>10</minimum>
36:       <maximum>90</maximum>
37:       <command>_NUL_</command>
38:       <allowedType>RotateAndOrFit</allowedType>
39:       <deviceID>b</deviceID>
40:    </deviceScaling>
41:    <deviceScaling>
42:       <default>100</default>
43:       <minimum>100</minimum>
44:       <maximum>100</maximum>
45:       <command>_NUL_</command>
46:       <allowedType>Clip</allowedType>
47:       <deviceID>c</deviceID>
48:    </deviceScaling>
49: </deviceScalings>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceScalings>.

On lines 25-32 is one print mode that is defined for this device. The data about the print mode is enclosed within the <deviceScaling> and </deviceScaling> tags.

an example would be:

```
DeviceScaling *pDS         = getCurrentScaling ();
double         dPercentage;

dPercentage = pDS->getScalingPercentage ();
```

Sheet Collations for a device

XML tag	***description***
name	The job property value for this sheet collation.

XML tag	***description***
command	The command that corresponds to this element.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Sheet Collates.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20:
21: <deviceSheetCollates
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <deviceSheetCollation>
26:       <name>SheetCollated</name>
27:       <command>_NUL_</command>
28:       <deviceID>kio</deviceID>
29:    </deviceSheetCollation>
30:    <deviceSheetCollation>
31:       <name>SheetAndJobCollated</name>
32:       <command>_NUL_</command>
33:       <deviceID>jrh</deviceID>
34:    </deviceSheetCollation>
35:    <deviceSheetCollation>
36:       <name>SheetUncollated</name>
37:       <command>_NUL_</command>
38:       <deviceID>pqo</deviceID>
39:    </deviceSheetCollation>
40: </deviceSheetCollates>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 25-29 is one print mode that is defined for this device. The data about the print mode is enclosed within the <deviceSheetCollation> and </deviceSheetCollation>

tags.

an example would be:

```
DeviceSheetCollate *pDS = getCurrentSheetCollate ();

if (pDS->isEqual (“SheetCollate=SheetUncollated”))
{
   ...
}
```

Sides for a device

XML tag	***description***
name	The job property value for this side.
simulationRequired	Asks the Omni driver to simulate this feature to the device.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Sides.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
     IBM Omni driver
     Copyright (c) International Business Machines Corp., 2000

     This library is free software; you can redistribute it and/or modify
     it under the terms of the GNU Lesser General Public License as published
     by the Free Software Foundation; either version 2.1 of the License, or
     (at your option) any later version.

     This library is distributed in the hope that it will be useful,
     but WITHOUT ANY WARRANTY; without even the implied warranty of
     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
     the GNU Lesser General Public License for more details.

     You should have received a copy of the GNU Lesser General Public License
     along with this library; if not, write to the Free Software
     Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-->

<deviceSides
   xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
   xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
   xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../OmniDevice.xsd">
   <deviceSide>
      <name>OneSidedBackflipX</name>
      <command>_NUL_</command>
      <simulationRequired>false</simulationRequired>
      <deviceID>a</deviceID>
   </deviceSide>
```

```
31:    <deviceSide>
32:       <name>OneSidedBackflipY</name>
33:       <command>_NUL_</command>
34:       <simulationRequired>false</simulationRequired>
35:       <deviceID>b</deviceID>
36:    </deviceSide>
    ...
37:    <deviceSide>
38:       <name>TwoSidedFlipY</name>
39:       <command>_NUL_</command>
40:       <simulationRequired>false</simulationRequired>
41:       <deviceID>e</deviceID>
42:    </deviceSide>
43: </deviceSides>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceSides>.

On lines 25-30 is one side that is defined for this device. The data about the side is enclosed within the <deviceSide> and </deviceSide> tags.

an example would be:

```
DeviceSide *pDS = getCurrentSide ();

if (pDS->isEqual (“Sides=OneSidedFront”))
{
   ...
}
```

Stitchings for a device

XML tag	***description***
stitchingPosition	The position of the stitch on the page.
stitchingReferenceEdge	The edge where the stitch is measured from.
stitchingType	The type of the stitch.
stitchingCount	The number of stitches.
stitchingAngle	The angle of the stitch.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Stitchings.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
```

```
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20:
21: <deviceStitchings
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <deviceStitching>
26:       <StitchingPosition>1</StitchingPosition>
27:       <StitchingReferenceEdge>Top</StitchingReferenceEdge>
28:       <StitchingType>Corner</StitchingType>
29:       <StitchingCount>1</StitchingCount>
30:       <StitchingAngle>0</StitchingAngle>
31:       <command>_NUL_</command>
32:       <deviceID>a</deviceID>
33:    </deviceStitching>
34:    <deviceStitching>
35:       <StitchingPosition>5</StitchingPosition>
36:       <StitchingReferenceEdge>Right</StitchingReferenceEdge>
37:       <StitchingType>Saddle</StitchingType>
38:       <StitchingCount>3</StitchingCount>
39:       <StitchingAngle>45</StitchingAngle>
40:       <command>_NUL_</command>
41:       <deviceID>b</deviceID>
42:    </deviceStitching>
43: </deviceStitchings>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceStitchings>.

On lines 25-33 is one print mode that is defined for this device. The data about the print mode is enclosed within the <deviceStitching> and </deviceStitching> tags.

an example would be:

```
   DeviceStitching *pDS    = getCurrentStitching ();
   int              iCount;
   int              iAngle;

   iCount = pDS->getStitchingCount ();
   iAngle = pDS->getStitchingAngle ();
```

Strings for a device

XML tag	***description***
name	The device independent string name
languages	A structure of what languages that this string is translated into.

New Device Strings.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20:
21: <deviceStrings
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <deviceString>
26:       <name>key1</name>
27:       <languages>
28:          <en>english1</en>
29:          <ja>japanese1</ja>
30:       </languages>
31:    </deviceString>
32:    <deviceString>
33:       <name>key2</name>
34:       <languages>
35:          <en>english2</en>
36:          <ja>japanese2</ja>
37:       </languages>
38:    </deviceString>
39: </deviceStrings>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceStrings>.

On lines 25-31 is one print mode that is defined for this device. The data about the print mode is enclosed within the <deviceString> and </deviceString> tags.

Trays for a device

XML tag	***description***
name	The job property value for this tray.
omniName	Optionally the old id of the tray.
trayType	It contains the programmatic description of the tag.
command	The command that corresponds to this element.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Trays.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
     IBM Omni driver
     Copyright (c) International Business Machines Corp., 2000

     This library is free software; you can redistribute it and/or modify
     it under the terms of the GNU Lesser General Public License as published
     by the Free Software Foundation; either version 2.1 of the License, or
     (at your option) any later version.

     This library is distributed in the hope that it will be useful,
     but WITHOUT ANY WARRANTY; without even the implied warranty of
     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
     the GNU Lesser General Public License for more details.

     You should have received a copy of the GNU Lesser General Public License
     along with this library; if not, write to the Free Software
     Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
-->
<deviceTrays xmlns="http://www.ibm.com/linux/ltc/projects/omni/" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../OmniDevice.xsd">
   <deviceTray>
      <name>Continuous</name>
      <trayType>TRAY_TYPE_AUTO</trayType>
      <command>_NUL_</command>
   </deviceTray>
   <deviceTray>
      <name>AutoSelect</name>
      <trayType>TRAY_TYPE_AUTO</trayType>
      <command>_NUL_</command>
   </deviceTray>
   <deviceTray>
      <name>AutoSelect</name>
      <trayType>TRAY_TYPE_MANUAL</trayType>
      <command>_ESC_ HEX (19) &quot;?&quot;</command>
```

```
35:    </deviceTray>
36: </deviceTrays>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On line 20 is the root tag of what XML data is contained in this file. The root tag is <deviceTrays>.

On lines 21-25 is one tray that is defined for this device. The data about the tray is enclosed within the <deviceTray> and </deviceTray> tags.

On line 22 is the <name> tag. It contains the SJP identifier for this print mode.

On line 23 is the <trayType> tag. It contains the programmatic description of the tag.

On line 24 is the <command> tag. It contains binary command data (described above) for this tray.

an example would be:

```
   DeviceTray *pTray = getCurrentTray ();

   if (pTray->isEqual (“InputTray=AutoSelect”))
   {
      ...
   }
```

Trimmings for a device

XML tag	***description***
name	The job property value for this trimming.
command	The command that corresponds to this element.
deviceID	An optional string identifier that will replace the id for the device job properties.

New Device Trimmings.xml

```
 1: <?xml version="1.0" encoding="UTF-8"?>
 2: <!--
 3:      IBM Omni driver
 4:      Copyright (c) International Business Machines Corp., 2000
 5:
 6:      This library is free software; you can redistribute it
and/or modify
 7:      it under the terms of the GNU Lesser General Public License
as published
 8:      by the Free Software Foundation; either version 2.1 of the
License, or
 9:      (at your option) any later version.
10:
11:      This library is distributed in the hope that it will be
useful,
12:      but WITHOUT ANY WARRANTY; without even the implied warranty
```

```
of
13:      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
14:      the GNU Lesser General Public License for more details.
15:
16:      You should have received a copy of the GNU Lesser General
Public License
17:      along with this library; if not, write to the Free Software
18:      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307 USA
19: -->
20:
21: <deviceTrimmings
22:    xmlns="http://www.ibm.com/linux/ltc/projects/omni/"
23:    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
24:
xs:schemaLocation="http://www.ibm.com/linux/ltc/projects/omni/ ../
OmniDevice.xsd">
25:    <deviceTrimming>
26:       <name>None</name>
27:       <command>_NUL_</command>
28:       <deviceID>a</deviceID>
29:    </deviceTrimming>
30:    <deviceTrimming>
31:       <name>Trim</name>
32:       <command>_NUL_</command>
33:       <deviceID>b</deviceID>
34:    </deviceTrimming>
    ...
35:    <deviceTrimming>
36:       <name>Tab</name>
37:       <command>_NUL_</command>
38:       <deviceID>e</deviceID>
39:    </deviceTrimming>
40: </deviceTrimmings>
```

On line 1 is the XML version of the file.

On lines 2-19 is the license.

On lines 21-24 is the root tag of what XML data is contained in this file. The root tag is <deviceTrimmings>.

On lines 25-29 is one print mode that is defined for this device. The data about the print mode is enclosed within the <deviceTrimming> and </deviceTrimming> tags.

an example would be:

```
   DeviceTrimming *pDT = getCurrentTrimming ();

   if (pDT->isEqual (“Trimming=None”))
   {
      ...
   }
```

Instance Code

All devices have to support a set of required routines. These routines are broken up into two sections. These are the instance and blitter routines. The instance routines are functions that support working with job properties and handling job and page notification.

```
void
initializeInstance ()
```

This function is called after the instance is created but before any user code is executed. This code allows routines to be called that are not safe to be called apon class instantiation.

```
bool
hasError ()
```

This function reports if an error has occurred since this instance code has been created.

```
std::string *
getJobProperties (bool fInDeviceSpecific = false)
```

This function asks the instance for any instance-specific job properties. The return value is a space separated key=value string. If fInDeviceSpecific is true, then for the returned string, the values should be in instance-specific form.

```
bool
setJobProperties (PSZCRO pszJobProperties)
```

This function is called when job properties are set to the Omni driver. The instance should handle any instance-specific job properties. If there was any invalid value for the job properties or any invalid key=value combination, then this function should return false.

```
Enumeration *
getGroupEnumeration (bool fInDeviceSpecific = false)
```

This function is called when job properties are queried by an application. Job properties are broken down into groups. For example, one group of job property keys is “NumberUp” and “NumberUpPresentationDirection”. Another group would be “Resolution”. And another group would be “ScalingType” and “ScalingPercentage”. The return value is an Enumeration* of Enumeration*. If fInDeviceSpecific is true, then for the 2nd level Enumeration*, the Key values should be in instance-specific form.

```
std::string *
getJobPropertyType (PSZCRO pszKey)
```

For a given job property key, return the type of the job property. The type is defined as the following:

string
integer
float
boolean

Following the type is a space and then the default value.

```
std::string *
getJobProperty (PSZCRO pszKey)
```

For a given job property key, return the current value.

```
std::string *
translateKeyValue (PSZCRO pszKey,
                   PSZCRO pszValue)
```

For a given job property key and value, return the translation of

```
bool
deviceOptionValid (PSZCRO pszDeviceOption)
```

For the given device option, return true if you support it or false otherwise.

```
bool
setOutputStream (FILE *pFile)
```

This function is called to change the output stream to a different file handle.

```
bool
setErrorStream (FILE *pFile)
```

This function is called to change the error stream to a different file handle.

```
bool
setLanguage (int iLanguageID)
```

This function is called when the default language is changed.

```
bool
beginJob ()
```

This function is called at the start of a print job.

```
bool
beginJob (bool fJobPropertiesChanged)
```

This function is called at the start of a print job and the job properties have changed. You will have already received a `setJobProperties` call.

```
bool
newFrame ()
```

This function is called at the start of another page in the print job.

```
bool
newFrame (bool fJobPropertiesChanged)
```

This function is called at the start of another page in the print job and the job properties have changed. You will have already received a `setJobProperties` call.

```
bool
endJob ()
```

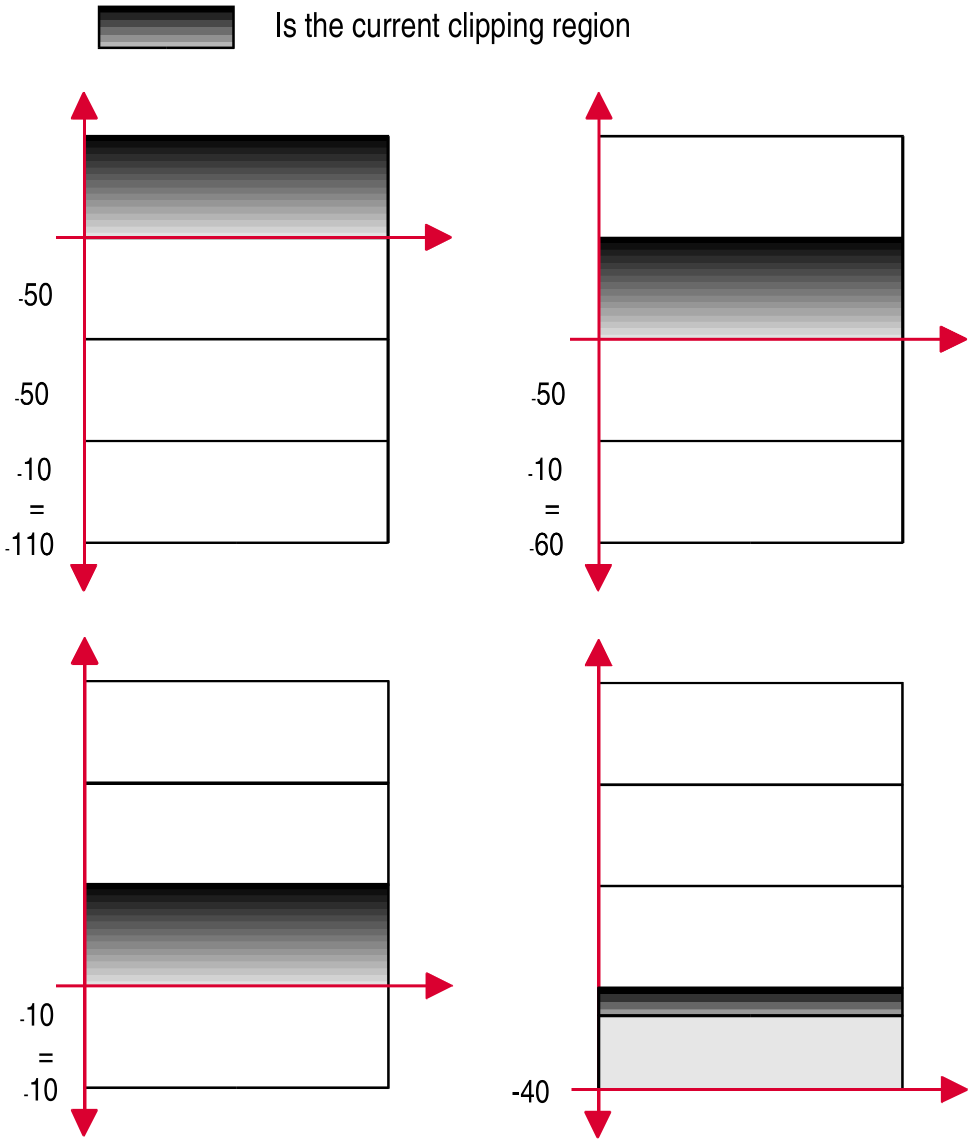
This function is called at the end of the print job.

```
bool
abortJob ()
```

This function is called when a print job is aborted.

Blitter Code

The blitter routines are involved in the transformation of a page of bitmap calls into the printer language. The page is broken up into a series of bitmaps called bands in order to save memory (consider how much memory is used for an 8.5''x11'' at 600 dpi and 24 bits per pel color). These bands start from the top of the page and work their way down to the bottom. For every band of bitmap data, you will receive a rasterize call.

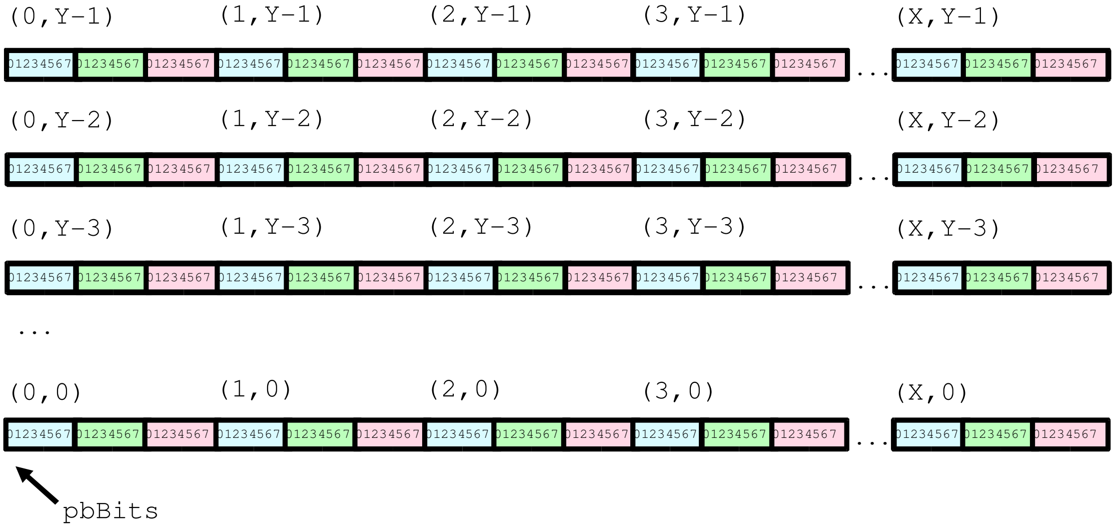


The bitmap bands are ordered as follows:

Scanlines are double word aligned. That is, the size will always be a multiple of 32 bits or 4 bytes. For example, a solid, 21 x 3 sized, 1 bit per pel bitmap would look like:

```
   BYTE abBitmapData[] = {
      255, 255, 248, 0,    // Scanline 0 (bottom)
      255, 255, 248, 0,    //          1
      255, 255, 248, 0     //          2 (top)
   };
```

Another example for 24 bit RGB data would be:



```
void
initializeInstance ()
```

This function is called after the instance is created but before any user code is executed. This code allows routines to be called that are not safe to be called apon class instantiation.

```
bool
hasError ()
```

This function reports if an error has occurred since this blitter code has been created.

```
bool
rasterize (PBYTE        pbBits,
           PBITMAPINFO2 pbmi,
           PRECTL       prectlPageLocation,
           BITBLT_TYPE  eType)
```

This function is called to handle a band of bitmap data.