MyWebUniversity
python
Red Hat Enterprise Linux Manual Pages
FPDF Library PDF generator
FPDF

Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'backend.7' command

$ man backend.7

```
backend(7)                    Apple Inc.                    backend(7)

NAME

       backend - cups backend transmission interfaces

SYNOPSIS

       backend

       backend job user title num-copies options [ filename ]

       #include <cups/cups.h>

       const char *cupsBackendDeviceURI(char **argv);

       void cupsBackendReport(const char *device_scheme,

                              const char *device_uri,

                              const char *device_make_and_model,

                              const char *device_info,

                              const char *device_id,

                              const char *device_location);

       ssize_t cupsBackChannelWrite(const char *buffer,

                                    size_t bytes, double timeout);

       int cupsSideChannelRead(cups_sc_command_t *command,

                               cups_sc_status_t *status, char *data,

                               int *datalen, double timeout);

       int cupsSideChannelWrite(cups_sc_command_t command,

                                cups_sc_status_t status, const char *data,

                                int datalen, double timeout);

DESCRIPTION
```

Backends are a special type of filter(7) which is used to send print data to and discover different devices on the system.

Like filters, backends must be capable of reading from a filename on the command-line or from the standard input, copying the standard input to a temporary file as required by the physical interface.

The command name (argv[0]) is set to the device URI of the destination printer. Authentication information in argv[0] is removed, so backend developers are urged to use the DEVICE_URI environment variable when?ever authentication information is required. The cupsBackendDeviceURI() function may be used to retrieve the correct device URI.

Back-channel data from the device should be relayed to the job filters using the cupsBackChannelWrite function.

Backends are responsible for reading side-channel requests using the cupsSideChannelRead() function and responding with the cupsSideChannel?Write() function. The CUPS_SC_FD constant defines the file descriptor that should be monitored for incoming requests.

DEVICE DISCOVERY

When run with no arguments, the backend should list the devices and schemes it supports or is advertising to the standard output. The out?put consists of zero or more lines consisting of any of the following forms:

```
device-class scheme "Unknown" "device-info"
device-class device-uri "device-make-and-model" "device-info"
device-class device-uri "device-make-and-model" "device-info" "device-id"
device-class device-uri "device-make-and-model" "device-info" "device-id" "device-location"
```

The cupsBackendReport() function can be used to generate these lines and handle any necessary escaping of characters in the various strings.

The device-class field is one of the following values:

direct

The device-uri refers to a specific direct-access device with no options, such as a parallel, USB, or SCSI device.

file The device-uri refers to a file on disk.

network

The device-uri refers to a networked device and conforms to the general form for network URIs.

serial

The device-uri refers to a serial device with configurable baud rate and other options. If the device-uri contains a baud value, it represents the maximum baud rate supported by the device.

The scheme field provides the URI scheme that is supported by the back? end. Backends should use this form only when the backend supports any URI using that scheme. The device-uri field specifies the full URI to use when communicating with the device.

The device-make-and-model field specifies the make and model of the de? vice, e.g. "Example Foojet 2000". If the make and model is not known, you must report "Unknown".

The device-info field specifies additional information about the de? vice. Typically this includes the make and model along with the port number or network address, e.g. "Example Foojet 2000 USB #1".

The optional device-id field specifies the IEEE-1284 device ID string for the device, which is used to select a matching driver.

The optional device-location field specifies the physical location of the device, which is often used to pre-populate the printer-location attribute when adding a printer.

PERMISSIONS

Backends without world read and execute permissions are run as the root user. Otherwise, the backend is run using an unprivileged user ac? count, typically "lp".

EXIT STATUS

The following exit codes are defined for backends:

CUPS_BACKEND_OK

The print file was successfully transmitted to the device or re? mote server.

CUPS_BACKEND_FAILED

The print file was not successfully transmitted to the device or remote server. The scheduler will respond to this by canceling

the job, retrying the job, or stopping the queue depending on the
state of the printer-error-policy attribute.

CUPS_BACKEND_AUTH_REQUIRED

The print file was not successfully transmitted because valid au?
thentication information is required. The scheduler will respond
to this by holding the job and adding the 'cups-held-for-authenti?
cation' keyword to the "job-reasons" Job Description attribute.

CUPS_BACKEND_HOLD

The print file was not successfully transmitted because it cannot
be printed at this time. The scheduler will respond to this by
holding the job.

CUPS_BACKEND_STOP

The print file was not successfully transmitted because it cannot
be printed at this time. The scheduler will respond to this by
stopping the queue.

CUPS_BACKEND_CANCEL

The print file was not successfully transmitted because one or
more attributes are not supported or the job was canceled at the
printer. The scheduler will respond to this by canceling the job.

CUPS_BACKEND_RETRY

The print file was not successfully transmitted because of a tem?
porary issue. The scheduler will retry the job at a future time -
other jobs may print before this one.

CUPS_BACKEND_RETRY_CURRENT

The print file was not successfully transmitted because of a tem?
porary issue. The scheduler will retry the job immediately with?
out allowing intervening jobs.

All other exit code values are reserved.

ENVIRONMENT

In addition to the environment variables listed in cups(1) and fil?
ter(7), CUPS backends can expect the following environment variable:

DEVICE_URI

The device URI associated with the printer.

FILES

/etc/cups/cups-files.conf

NOTES

CUPS backends are not generally designed to be run directly by the user.   Aside from the device URI issue (argv[0] and DEVICE_URI envi? ronment variable contain the device URI), CUPS backends also expect specific environment variables and file descriptors, and typically run in a user session that (on macOS) has additional restrictions that af? fect how it runs. Backends can also be installed with restricted per? missions (0500 or 0700) that tell the scheduler to run them as the "root" user instead of an unprivileged user (typically "lp") on the system.

Unless you are a developer and know what you are doing, please do not run backends directly.   Instead, use the lp(1) or lpr(1) programs to send print jobs or lpinfo(8) to query for available printers using the backend.   The one exception is the SNMP backend - see cups-snmp(8) for more information.

NOTES

CUPS printer drivers and backends are deprecated and will no longer be supported in a future feature release of CUPS. Printers that do not support   IPP   can   be   supported   using   applications   such   as ippeveprinter(1).

SEE ALSO

cups(1), cups-files.conf(5), cups-snmp(8), cupsd(8), filter(7), lp(1), lpinfo(8), lpr(1),

CUPS Online Help (http://localhost:631/help)

COPYRIGHT