CUPS Translation Guide

CUPS−TRANS−1.1

Easy Software Products

Table of Contents

1 Scope

1.1 Identification

This translation guide provides instructions for creating translations of the CUPS message catalogs and web pages for the Common UNIX Printing System ("CUPS") Version 1.1 software.

1.2 System Overview

CUPS provides a portable printing layer for UNIX®−based operating systems. It has been developed by Easy Software Products to promote a standard printing solution for all UNIX vendors and users. CUPS provides the System V and Berkeley command−line interfaces.

CUPS uses the Internet Printing Protocol ("IPP") as the basis for managing print jobs and queues. The Line Printer Daemon ("LPD") Server Message Block ("SMB"), and AppSocket (a.k.a. JetDirect) protocols are also supported with reduced functionality. CUPS adds network printer browsing and PostScript Printer Description ("PPD") based printing options to support real−world printing under UNIX.

CUPS also includes a customized version of GNU Ghostscript (currently based off GNU Ghostscript 5.50) and an image file RIP that are used to support non−PostScript printers. Sample drivers for HP and EPSON printers are included that use these filters.

1.3 Document Overview

This translation guide is organized into the following sections:

- 1 − Scope
- 2 − References
- 3 − Character Sets
- 4 − Message Catalogs
- 5 − Web Interfaces
- A − Glossary

2 References

2.1 CUPS Documentation

The following CUPS documentation is referenced by this document:

- CUPS−CMP−1.1: CUPS Configuration Management Plan
- CUPS−IDD−1.1: CUPS System Interface Design Description
- CUPS−IPP−1.1: CUPS Implementation of IPP
- CUPS−SAM−1.1.x: CUPS Software Administrators Manual
- CUPS−SDD−1.1: CUPS Software Design Description
- CUPS−SPM−1.1.x: CUPS Software Programming Manual
- CUPS−SSR−1.1: CUPS Software Security Report
- CUPS−STP−1.1: CUPS Software Test Plan
- CUPS−SUM−1.1.x: CUPS Software Users Manual
- CUPS−SVD−1.1: CUPS Software Version Description

2.2 Other Documents

The following non−CUPS documents are referenced by this document:

- Adobe PostScript Printer Description File Format Specification, Version 4.3.
- Adobe PostScript Language Reference, Third Edition.
- IPP: Job and Printer Set Operations
- IPP/1.1: Encoding and Transport
- IPP/1.1: Implementers Guide
- IPP/1.1: Model and Semantics
- RFC 1179, Line Printer Daemon Protocol
- RFC 2567, Design Goals for an Internet Printing Protocol
- RFC 2568, Rationale for the Structure of the Model and Protocol for the Internet Printing Protocol
- RFC 2569, Mapping between LPD and IPP Protocols
- RFC 2616, Hypertext Transfer Protocol −− HTTP/1.1
- RFC 2617, HTTP Authentication: Basic and Digest Access Authentication

3 Character Sets

CUPS uses character set files to define the mapping of local character sets to Unicode code points, as well as the fonts that should be used for different ranges of characters.

4 Message Catalogs

5 Web Interfaces

The CUPS scheduler provides a web interface that can be used to do many common printing and administration tasks. The built−in web server supports localization of web pages through the use of subdirectories for each locale, e.g. "fr" for French, "de" for German, "fr_ca" for French in Canada, and so forth.

5.1 Template Files

The CUPS CGI programs (`admin.cgi`, `classes.cgi`, `jobs.cgi`, and `printers.cgi`) are responsible for providing dynamic content for the web interface. To facilitate this process, a series of HTML template files are used that build the final output page. Template files are installed in the */usr/share/cups/templates* directory by default.

Translated versions of the template files should be installed in the appropriate subdirectories under */usr/share/cups/templates*. Template files consist of HTML with variable substitutions for named inside curley braces "{name}". See the next section for a complete description of the syntax. Variable names are generally the IPP attribute names with the hyphen ("−") replaced by the underscore ("_") character.

5.1.1 What Are Templates?

Templates are HTML files with special formatting characters in them that allow substition of variables and arrays.

> **Note:** LIBCGI doesn't actually care if the template file is HTML or not. You can just as easily use a plain text file, XML file, or any other type of text file with the template code.

5.1.2 How Do I Use a Template?

LIBCGI provides a single template function called `cgiCopyTemplateFile()`.

5.1.3 void cgiCopyTemplateFile(FILE *out, const char *filename)

The `cgiCopyTemplateFile()` function copies the template file (`filename`) to the output file (`out`.)

The `out` argument is usually `stdout` for CGI programs, however you can write to any file or pipe with this function.

If `cgiCopyTemplateFile()` is unable to open the specified file then no output is produced.

5.1.4 Special Characters

LIBCGI uses the curley braces ("{" and "}") to indicate substitutions, and the backslash ("\") character for quoting. To insert any of these special characters as−is you need to use the HTML `&name;` mechanism or prefix each special character with the backslash ("\".)

5.1.5 Variable Substitutions

You substitute the value of a variable using `{NAME}` in your template file. If the variable is undefined then the `{NAME}` string is output as−is.

To substitute an empty string if the variable is undefined, use `{?NAME}` instead.

5.1.6 Array Substitutions

The number of array elements can be inserted using `{#NAME}`. If the array is undefined then 0 is output. The current array element (starting at 1) is inserted with `{#}`.

Arrays are handled using `{[NAME]` at the beginning of a section and `}` at the end. The information between the closing bracket ("]") and closing brace ("}") is repeated for as many elements as are in the named array. For example, the following template would display an invoice for each item defined in the `ITEM` array:

```
<TABLE>
<TR>
        <TH>Quantity</TH>
        <TH>Item</TH>
        <TH>Price</TH>
        <TH>Amount</TH>
</TR>

{[ITEM]
<TR>
        <TD>{?QUANTITY}</TD>
        <TD>{?ITEM}</TD>
        <TD>{?PRICE}</TD>
        <TD>{?AMOUNT}</TD>
</TR>
}

<TR>
        <TD COLSPAN="3" ALIGN="RIGHT">Total</TD>
        <TD>{?TOTAL}</TD>
</TR>
</TABLE>
```

Arrays can be nested, however all elements within the curley braces ("{" and "}") are indexed using the innermost array.

5.1.7 Tests

Templates can also test variables against specific values and conditionally include text in the template. The format is:

```
{variable?true:false}
{variable=value?true:false}
{variable!value?true:false}
{variable<value?true:false}
{variable>value?true:false}
```

where *true* is the text that is included if the condition is true and *false* is the text that is included if the condition is false. A value of # is replaced with the current element number (starting at 1.)

The character after the variable name specifies the condition to test:

Char	Condition
?	True if *variable* exists.
=	True if *variable* is equal to *value*.
!	True if *variable* is not equal to *value*.
<	True if *variable* is less than *value*.
>	True if *variable* is greater than *value*.

5.1.8 Template Files

The following template files are used by the web interface:

add−class.tmpl
: This is the initial form that is shown to add a new printer class.

add−printer.tmpl
: This is the initial form that is shown to add a new printer.

admin−op.tmpl
: This is the template that is used to display an error message when the admin interface sees an undefined operation name.

admin.tmpl
: This is the template that shows the initial menu of operations (add a class, manage classes, etc.)

choose−device.tmpl
: This is the form that shows the list of available devices.

choose−make.tmpl
: This is the form that shows the list of available manufacturers.

choose−members.tmpl
: This is the form that shows the list of available printers that can be added to a class.

choose−model.tmpl
: This is the form that shows the list of available printer models/drivers.

choose−serial.tmpl
: This is the form that allows the user to choose a serial port and any options.

choose−uri.tmpl
: This is the form that allows the user to enter a device URI for network printers.

class−added.tmpl
: This template shows the "class added" message.

class−confirm.tmpl
: This is the template used to confirm the deletion of a class.

class−deleted.tmpl
: This template shows the "class deleted" message.

classes.tmpl
: This template shows one or more printer classes.

class−modified.tmpl
: This template shows the "class modified" message.

config−printer2.tmpl

config−printer.tmpl

error.tmpl

header.tmpl

job−cancel.tmpl

job−hold.tmpl
job−release.tmpl
job−restart.tmpl
jobs.tmpl
modify−class.tmpl
modify−printer.tmpl
option−boolean.tmpl
option−header.tmpl
option−pickmany.tmpl
option−pickone.tmpl
option−trailer.tmpl
printer−accept.tmpl
printer−added.tmpl
printer−configured.tmpl
printer−confirm.tmpl
printer−deleted.tmpl
printer−modified.tmpl
printer−reject.tmpl
printer−start.tmpl
printers.tmpl
printer−stop.tmpl
test−page.tmpl
trailer.tmpl

A Glossary

A.1 Terms

C
: A computer language.

parallel
: Sending or receiving data more than 1 bit at a time.

pipe
: A one−way communications channel between two programs.

serial
: Sending or receiving data 1 bit at a time.

socket
: A two−way network communications channel.

A.2 Acronyms

ASCII
: American Standard Code for Information Interchange

CUPS
: Common UNIX Printing System

ESC/P
: EPSON Standard Code for Printers

FTP
: File Transfer Protocol

HP−GL
: Hewlett−Packard Graphics Language

HP−PCL
: Hewlett−Packard Page Control Language

HP−PJL
: Hewlett−Packard Printer Job Language

IETF
: Internet Engineering Task Force

IPP
: Internet Printing Protocol

ISO
: International Standards Organization

LPD
: Line Printer Daemon

MIME
: Multimedia Internet Mail Exchange

PPD
: PostScript Printer Description

SMB
: Server Message Block

TFTP
: Trivial File Transfer Protocol