

# ProPal

---

## Documentation

---

Current Version: 01.02.03

Released: 02/08/11

Fairlight Consulting is a proud participant in the



We offer not only this core integration kit, but can code custom modules to suit your needs. [Contact us](#) with any specific inquiries.

## Table of Contents

- [Introduction](#)
- [Feature List](#)
- [Requirements](#)
- [Licensing, Downloading, and Warranty Information](#)
- [Support](#)
- [Program Flow](#)
- [ProPal DTD](#)
- [Installation](#)
- [Upgrading](#)
- [Configuration Instructions](#)
  - [Environment File](#)
  - [Main Configuration File](#)
  - [Command File](#)
- [Configuration Options](#)
  - [max\\_session\\_count](#)
  - [wait\\_interval](#)
  - [wait\\_attempts](#)

- [apache2](#)
  - [sandbox](#)
  - [logging](#)
  - [log\\_size](#)
  - [ProPal Logs](#)
  - [Security Notes](#)
  - [ProPal Testing Tips](#)
  - [Change Log](#)
  - [OFFICIAL REFERENCE: PayPal Integration Guide](#)
- 

## Introduction

ProPal is an enterprise-class integration kit that assists with integrating [PayPal](#) with any solution you need to program, utilising PayPal's IPN (Instant Payment Notification) functionality. This IPN technology lets you receive information about PayPal transactions "behind the scenes" so that you may take further automated action based on what transpired for any given transaction. Their technology is quite robust, occurs within seconds, and will retry upon failure to contact your site with the details. Of their solutions along these lines of integration, we at Fairlight Consulting are convinced it's the "best way to fly".

ProPal is not limited to being used only for shopping carts, or only for auctions. It handles every type of IPN transaction documented by PayPal in their Integration Guide. Using ProPal, you can very easily integrate PayPal into any solution your business or organisation may need, including:

- shopping carts
- custom store-fronts
- download access activation upon receipt of online payment
- tracking mass payments of affiliates or employees
- subscription management
- adjusting bookkeeping

...and much more!

If PayPal generates IPN data for a transaction, ProPal makes it easy for you to access all relevant data associated with it.

ProPal is small, fast, and robust. It installs and can be configured easily in about five minutes, and what you do with it from there is up to you. Your possibilities for use of ProPal are unlimited because you can use any language you choose to facilitate your tasks in a very modular fashion.

ProPal consists of a program named *flipna* and filePro files and processing necessary to import the IPN data into filePro. The *flipna* program handles all data receipt and validation of IPN's and stores the resultant data in an XML file. It then runs whatever commands you tell it to run upon receipt of an IPN. The filePro processing table *propal* in the table *flt\_propalctl* is responsible for data import into filePro. This processing will usually be the first command you run if you wish to track and maintain your IPN data locally in a database. Any "business rules" programs and their attendant logic (ie., creating an online account, or automating printing shipping labels) falls under business rules, and is written by the developer. However, you simply use filePro lookups to access the relevant data for the specific transaction. The grunt-work of tracking all your information and making it available in an easy-to-utilise form is handled for you by ProPal, leaving you free to handle your specific operations rather than reinventing the wheel.

ProPal is modularly designed, so adding and removing modules is a trivial task. Each module can be written in any language you like. You might start off simply activating accounts for people to download software from your web site once they've paid. Later on, you can add another module that tracks and handles subscriptions for you, and simply "plug it in" as a separate module. Because ProPal handles all transaction types, you can dynamically enable or disable any part of your enterprise at any given time, making it an utterly flexible solution. You don't have to start out with everything designed. You can start with one modular application and add more as you need them. Your business rules can change without having to redesign one monolithic program, giving you flexibility and ease of mind knowing that your current solution will simply be augmented by whatever you add, not potentially harmed by an entire redesign of one single module or application. The engine remains the same, but you can change everything else at will. That is one of ProPal's strengths.

ProPal currently runs on Windows or any Unix-like system that can run perl 5.6 or higher.

---

## Feature List

- Supports **all** PayPal transaction types which transmit details over IPN
  - Provides output to your applications in XML with a well-defined DTD, if access to the data directly from a file rather than (or in addition to) a database is desired
  - Supports unlimited commands per transaction
  - Supports business logic modules written in any language
  - Supports access logging (unlimited, rolling, or disabled)
  - Supports session-count limiting of command execution, with fully configurable retries
  - Supports user-defined environment variables
  - Designed with system security in mind
  - Source code immediately available for modifications, if necessary or desired
- 

## Requirements

This package requires Perl version 5.6 or higher to run.

The *flipna* program requires the Perl CGI module (CGI.pm) and the Crypt::SSLeay module to be installed on your system to operate correctly:

In addition, you will need filePro installed to make use of the filePro import functionality.

You will also need a web server configured to allow CGI execution.

ProPal has been tested on Solaris, Linux, SCO UNIX OpenServer 5.0.6, and Windows 2003 Server. It should encounter no difficulties on any platform on which perl will compile and run correctly.

---

## Licensing

It should be noted that there is no demo version of ProPal, nor will there be a "Lite" version.

We will provide whatever pre-sales assistance is required to demonstrate whether ProPal is the right product for your needs. If you have questions or comments, please direct them to [sales@fairlite.com](mailto:sales@fairlite.com) without hesitation, and we will answer as promptly as possible.

We realise that software is an investment, and hope that you preview the documentation available for this product and make sure that it will suit your needs before purchasing a license. We also hope you avail yourself of the opportunity to ask any pre-sales questions you might have. All license sales are final and non-refundable.

ProPal is licensed at a cost of **\$495.00 USD** *per domain name under which it is used, per machine that serves said domain*. This means that if you have *www.widgets.com* and use three servers to load-balance your domain's web traffic, you require three ProPal licenses. If you also wish to serve *www.sprockets.com*, you would need an additional license for each machine which served that domain.

Bulk discounts may be negotiated for purchases of more than five copies at the same time. Special pricing is available for ISP deployment.

Each license fee entitles you to use ProPal under one domain name on one machine, in any role you require. You may modify the program to further suit your needs, and are under no obligation to release changes back to Fairlight Consulting. However, derivative works and/or modified versions **may not** be resold or otherwise distributed. Similarly, you may not copy ProPal, modify it, and run the altered version on another machine. You must purchase another license to use it in any form, altered or otherwise, on an additional machine. You may use an altered version and the original version on the same machine under a single license, however. ***"Machine" shall be defined as one instance of an operating system, for the purpose of this license. Machines which run multiple concurrent operating systems (virtual machines) count as multiple machines, and require additional licenses for each instance.*** The licensee agrees to keep the source code confidential and protected.

Please note that you may utilise more than one copy of ProPal per single domain, per server **for testing purposes only**. This would apply in the case of using it for live work as well as with PayPal's Sandbox concurrently. However, you may **not** use this special provision (intended for testing purposes) to "get around" the need for more licenses. You should not be running copies of ProPal for more than one organisation, all from inside one domain. The intent of the license is that licenses are counted by both domain names and machines which serve them, but that one domain name equals one organisation, except in the case of special ISP licensing.

Upon receipt of payment for a license, access to the program will be generated for the licensee, and such information shall be delivered to the email address associated with the PayPal payment.

Upgrades for the product are *currently* free when moving to new minor and major versions. Fairlight Consulting reserves the right to change this policy in the future, with no prior warning.

There is *no warranty* for this software. This software is offered "**AS-IS**" and without warranties as to performance or merchantability or any other warranties, whether expressed or implied.

Good computing practice dictates that any program should be thoroughly tested with non-critical data before deploying it for production. The user assumes the entire risk of using the program. In no event shall Fairlight Consulting be held liable for loss of data, failure of performance, or any other damages, be they real or perceived.

If you agree to these terms, [click here to order ProPal!](#)

Already registered and have your account information?

[Download ProPal!](#)

---

## Support

Extended (non-bug-related) support for ProPal is available at our standard hourly [rates](#). Because we offer pre-sales assistance in determining if ProPal is right for your needs, and because documentation is readily available, anything not covered by either of these is deemed an at-cost support issue.

You may request *any* kind of technical assistance with ProPal by sending email to [propal@fairlite.com](mailto:propal@fairlite.com), including bug reports and feature requests.

Feature requests may be commissioned for special functionality, if desired. Any non-commissioned requests are subject to being implemented solely at the discretion of Fairlight Consulting. This may include not being implemented at all, depending on how useful we think the feature would be in general. Commissioned requests can be price-negotiated based on whether the features requested are allowed to be re-integrated into the main product, or whether they shall remain proprietary and exclusive to the commissioning party.

We also offer consulting on how to achieve specific results using the product through this support mechanism, and would be happy to assist you in this regard.

---

## Program Flow

The logic flow of ProPal is straightforward, and is as follows:

1. A PayPal transaction is sent to your *flipna* URL as configured in your PayPal account profile.
2. The web server (httpd) starts *flipna*.
3. The *flipna* program confirms the validity of the transaction with PayPal's servers.
4. The *flipna* program then writes an XML file into a spool directory, and then executes all commands listed in the *commands* file, in order. If the session count was exceeded for the configurable amount of retry time, the file is left in place. If all operations succeed, the XML file is removed after command execution.

This command execution phase is where database population is achieved with a call to *[d|r]report* and any other additional programs of your choice. You may insert other programs after the database population to handle "business rules" operations.

---

## ProPal DTD

Should you ever need to directly access the XML files that *flipna* generates, the ProPal DTD for incoming data is as follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE form_submission [
<!ELEMENT form_submission (form_field*)>
<!ELEMENT form_field (field_name,field_value+)>
<!ELEMENT field_name (#PCDATA)>
<!ELEMENT field_value (#PCDATA)>
<!ATTLIST form_field type (file|value|cookie|textarea)
"value">
<!ATTLIST form_field orig_filename CDATA #IMPLIED>
<!ATTLIST form_field size CDATA #IMPLIED>
]>
```

There are several things to note regarding how ProPal presents data:

- All fields that are submitted will be reported, even if empty. In the case of empty fields, the *field\_value* element will be an empty container.
- All *field\_value* contents will be encapsulated inside the `<![CDATA[]]>` syntax, so as to preserve any possible SGML input that might result from form data. In the case of empty containers, the CDATA block will not be present, and the container will truly be empty.

The type attributes **file**, **cookie**, and **textarea** are not used at this time. Currently, all fields are of type **value**.

---

## Installation

Installing ProPal is very straightforward, and involves a few simple steps.

ProPal's main program, *flipna*, should reside in a directory that has CGI execution enabled in your httpd configuration. The program's working directory, which will be named *propal* should reside somewhere **outside** of the web document space on your server for reasons of security.

**If you are installing on \*nix, make sure you log in as or su to the same user as should be running ProPal's component programs, prior to installation or any (re)configuration. If you do not, you will need to manually adjust permissions and ownerships accordingly.**

To install ProPal, follow these steps:

1. Copy the distribution tarball to a directory where you have write permission.
2. Untar or unzip the distribution, depending on your platform.
3. Change directory into *propal-install*.
4. Run **perl ./install.pl** to begin the installation process.
5. Answer the location questions.
6. Adjust the ownership of the files and directories under ProPal's directory tree if necessary.
7. Extract the *fpfiles.tar.gz* or *fpfiles.zip* archive, depending on your platform. Copy the *flt\_propal* and *flt\_propalctl* directories from the appropriate platform directory under the extracted *propal-filepro* directory to your actual filePro **\$PFDIR** location. Use the *chown* command to ensure that the *filepro* user owns all files within these installed directories, if required.
8. Proceed to the section on configuration.
9. After configuration is complete, configure the IPN section of your PayPal account profile to point to the URL for your copy of *flipna*

---

## Upgrading

To upgrade ProPal, follow the steps for installation.

Please note that ProPal's installer will overwrite the old program file, but your entire *propal* directory tree will remain untouched by the process. All existing configuration will be retained.

Despite this fact, good computing practice dictates that one should make a backup of one's configuration before performing an upgrade. It is not necessary, but it *is* wise.

---

## Configuration Instructions

In this section we will walk you through the relatively simple tasks involved in configuring ProPal for use.

Before we begin, it should be stated that in the following sections, the files mentioned reside in the main *propal* directory.

### Environment File

Global environment variables that your commands require may be set by using the *environment* file. The presence of this file is optional.

The format of this file is identical to Bourne shell syntax, with one **VAR=value** assignment per line. There are a few exceptions to the Bourne-like quality of the syntax. Exports are not needed. ProPal handles this for you, as the file is not processed by a shell. Comments may be on individual lines, and blank lines are ignored. You may *not* put comments after value contents. The other notable difference is that you do not need to quote your values. Any quotes included (be they single or double) will be literally interpreted as part of the value for your variable.

### Configuration File

The file *config* controls global behaviour for ProPal's handling of all applications. The presence of this file is optional, and if it is absent or empty, default values will be used.

All available options are described in detail on the [Configuration Options](#) page. The format of this file is one **option=value** assignment per line. Comments may be on lines by themselves, and blank lines are ignored.

### Commands File

The heart of your interaction with ProPal, the file *commands* controls what commands are executed by *flipna*. This file **must** be present, and configured with the list of your commands for the program set.

The syntax allows for blank and separate comment lines. You may *not* place comments on the same line as a command. Each command should be on a separate line, and each will be processed in order at execution.

You may use any valid command on your system, and shell metacharacters (redirects, etc.) *are* allowed. Please note that ProPal clears the *\$PATH* environment variable at the beginning of its execution, so you effectively have no path unless you set one in either the main or program set environment variable files. If you do not define a path, you will need to explicitly state full paths to your commands.

There are three "magic cookies" that allow you to give your commands access to the relevant files and data associated with a ProPal transaction. Wherever these tokens appear in your commands, they will be replaced with the appropriate pathname.

- **%i** - This token represents the XML data file generated by ProPal. This file contains the IPN data.
- **%l** - This token represents the internal lock number ProPal uses for your current transaction.
- **%n** - This token represents a unique ID that is used by any of your other applications and commands in your command list. The ID that this token will expand to is the value for which you would search for in the **flt\_id** field within the database tables. You may also use this ID to create transaction-specific temporary files between commands.

Should you be unable to utilise the command line to pass filenames or the internal lock number to your application, an alternate method is available in the form of environment variables. These do not go in the configuration file, but instead may be accessed from within your application:

- **FLIPNA\_FILE** - Identical to the file that **%i** denotes in the commands file.
- **FLIPNA\_LOCK\_NUMBER** - Identical to the number that **%l** denotes in the commands file.
- **FLIPNA\_ID** - Identical to the program name that **%n** refers to in the commands file.

---

## Configuration Options

The following options are valid for ProPal's configuration files:

### **max\_session\_count**

This option dictates how many instances of *flipna* may be executing external commands at the same time. This option is to prevent license user counts from being exceeded in any software with license managers, as well as to help throttle the load on a very busy system.

### **wait\_interval**

When *flipna* gets to the phase where it executes external commands, it performs a check to make sure that no more than a configurable number of instances of itself are doing so at the same time. If it detects that number are already running, it will retry the condition a configurable amount of times. This option sets the granularity for these retries. The argument is an integer expression denoting seconds. The default for this option is **1** second.

### **wait\_attempts**

The companion to *wait\_interval*, this option denotes the number of retries to attempt before giving up and leaving the spool XML data file for the transaction in place for later perusal. The formula for total time spent retrying is *wait\_interval* \* *wait\_attempts*. The default for this option is **600** attempts (10 minutes, given the default setting for *wait\_interval*).

### **apache2**

This option **must** be enabled with the Apache 2.x servers. Apache 2.x kills child processes whenever they close their main three file descriptors. This results in stranded files which would be cleaned under other web servers such as Apache 1.3.x. Set to **yes** if you are using Apache 2.x. Otherwise set to **no**, which is the default. If you set this to **yes**, please make sure your commands in the *commands* file are redirected somewhere harmless (/dev/null in \*nix, for instance).

### **sandbox**

This option dictates whether *flipna* will run its transaction validations against the PayPal Sandbox at PayPal Developer Central, or whether it will run its transaction validations against the live PayPal site. A setting of **yes** activates PayPal Sandbox testing mode. A setting of **no** enables live mode against the real site. (**Note: As detailed in the official PayPal Integration Guide, transactions made in the Sandbox cannot be verified against the live site, and vice versa.**)

### **logging**

This option dictates whether *flipna* will log only fatal errors, or whether they'll log all transactions. A setting of **yes** enables logging of all transactions. The filename in the *propal* directory is **FLIPNA.LOG** for *flipna*.

## log\_size

This option sets the maximum number of lines in each logfile. If set to the word *unlimited* (or omitted), logs may grow infinitely. If set to an integer, logfile lines are kept to this many lines, with the oldest entries that exceed this threshold being eliminated.

---

## ProPal Logs

ProPal logs its activities in the *propal* directory, and each program has its own log. The log format is event-style, one line per event of any program. All program events are grouped together, between a beginning and end-of-processing pair event lines.

The format of any given line is:

- The program name as listed in the filesystem
- The process ID in square brackets
- The full date and time followed by a colon
- The text for the event.

Logging has three states: on, off, or rolling. If you set **logging** to on, logging will be performed. If you set it to off, only transactions with fatal errors will be logged. If you set **log\_size** to an integer, the log will roll off old entries when it reaches that number of lines. The **log\_size** will default to *unlimited* by default.

---

## Security Notes

ProPal was designed with security in mind. While it makes every attempt to be as secure as possible, you have to do your part as well.

The biggest pitfall of CGI programming is trusting external data and using it for system calls without vetting it first. This can be catastrophic, and lead to destruction of data. ProPal does *not* use any remote data in remote commands, nor does it provide you any facility to do so via the program set command file. However, it does not (and can not) prevent you from persuing this extremely hazardous practice within the applications you call from the command file. Those applications could conceivably call other programs unsafely, if you blindly trust data provided from outside your control and use it in a command line context. **DO NOT DO THIS.** Doing so is tantamount to leaving the keys to your car on the hood while you quickly nip into Wal\*Mart for even one small item.

There's a good chance your car won't be there when you get back. Likewise, bad things will inevitably happen if you commit this grave security error.

ProPal simply passes the data it obtains without parsing it or censoring it. As a generic application, it can not and should not do so. It is up to you to make sure that your application adequately scans and validates the incoming data to ensure it is safe for use in whatever context is applicable.

Security is one of the biggest issues with CGI programming. ProPal does its best to make the environment safe as possible without compromising the data integrity. It is your responsibility to ensure that your applications use the data safely.

For a complete list of points to check over in your integration efforts, please consult [this more comprehensive list](#) that we have released to the public to aid in CGI security education.

---

## Testing Tips

Using ProPal is quite easy. It installs and can be configured in a matter of minutes. After that, however, you need to program your business rules. Further, you need to **test** your business rules. This would ordinarily be a tedious process, were it not for PayPal's great testing facilities.

PayPal provides a "sandbox" area which lets you test everything except reversals (like chargebacks), and auctions. To utilise this resource, follow these steps:

1. Utilise the setting **sandbox=on** in your *config* file for ProPal.
2. Go to [PayPal Developer Central](#) and create an account there.
3. Log in and click on *Test Tools*.
4. Click on *Instant Payment Notification (IPN) Simulator*.
5. Set the URL for the IPN handler to point to *flipna* on your website.
6. Start testing. You can test any of the IPN samples provided in the *Transaction Type* selection box.
7. When you're done testing, utilise **sandbox=off** in your *config* file for ProPal. Configure your final solution with buttons that will utilise the live PayPal site (at **www.paypal.com** for all **webscr** locations within the buttons).

Please note that **flt\_verified** will be *VERIFIED* for Sandbox transaction IPN's tested against the Sandbox. However, you cannot test Sandbox transaction IPN's against the live system, nor can you test live IPN's against the Sandbox. They are mutually exclusive. If

you must test new modules while live ones remain in production, it is recommended that you utilise a second copy of ProPal in another location. This can be done by reinstalling to different locations on the same system using a different location for the *propal* directory and a different cgi-bin directory for *flipna*). In this manner, you can have one configuration for testing, and one for live use. This falls within acceptable use within the ProPal license.

---

## Accessing IPN Data in filePro

To access your data in filePro, simply perform a lookup to the *flt\_propal* table. The first field in the file (upon which automatic index *A* is built) will contain the filename found in the environment variable **FLIPNA\_FILE**. Simply look up all records with the string found in that variable within field 1 of the table. The relevant fields are as follows:

1. Application ID (999,\*)
2. Field Name (100,\*)
3. Field Value (999,\*)

There are other fields in the *flt\_propal* file, but they are extensions from a software package with similar requirements, and are not used by ProPal. ***Do not, however, delete the extra fields, as this will break the XML parser that does the importing of IPN data into filePro.***

---

## Change Log

v01.02.03 – 02/08/11

Original code base.

---

Copyright 2005-2011, Fairlight Consulting. All rights reserved.