SwiftStamp User Manual

Introduction

SwiftStamp licenses document and drawing files so they can be viewed for free using any copy of the SwiftView Viewer. The Viewer can be installed for free from www.swiftview.com or the customer's website, or included with the document e.g. in an email message, on a file server, or burned onto a CD.

SwiftStamp can also add metadata to the file that can be viewed in SwiftView or extracted later with SwiftStamp. This includes SwiftView ICS commands such as markups, or textual or XML information such as the identification of the author.

SwiftStamp can be rented or purchased on a stamps/year basis. Either way you get both a batch utility and a Windows Multi-File stamping GUI. The batch utility is a very small program which can only be executed from a command line or another program. It has no GUI because it is designed for integration into automatic processes. SwiftStamp batch utility is available for Win32, Sparc Solaris, HP-UX, RS/6000 AIX, Linux Intel, and other UNIX systems.

SwiftStamp is an alternative to a website license that also allows the user to resend files downloaded from a website to anyone for subsequent viewing. Note that SwiftStamp licenses do not apply to ICS files, so using SwiftStamp licensing on a website requires that each actual document file be stamped.

Effect of Stamping

A SwiftStamp license is a small license block appended to the end of a PCL, HPGL, TIFF, or ZIP file. It does not change the appearance or operation of the file in any way; for example, print files can still be copied to a printer. However, certain illegal (improperly created) TIFF files (in particular those which can be read only by assuming the remainder of the file is data rather than using data lengths stored in TIFF IFD's) could be affected. Contact SwiftView www.swiftview.com if you have a concern about this.

Distributing SwiftView to document recipients

We strongly suggest you provide an installer to your end users.

When you provide the installer, your users will always get the version of SwiftView that is appropriate for your setup. This also avoids users going to the SwiftView website looking for a download, which can cause confusion as your users browse through our website to find an installer. They could inadvertently download one that takes conflicting file extensions. When it comes time to provide SwiftView to your user base please contact SwiftView Support at www.swiftview.com/support/contact-support for the correct installer.

If you have any questions on how to do this please SwiftView Support at www.swiftview.com/support/contact-support

If you do not know your companies Customer ID, please contact licensing@paperbos.com.

Evaluation

If you want to try the SwiftStamp Multi-File GUI for Windows or if you need to integrate the SwiftStamp batch utility into your server-based application on Windows or UNIX, contact SwiftView Sales www.swiftview.com/sales/ for an evaluation copy. SwiftView can provide an evaluation version that will run for two weeks without a license and stamp up to 100 files on any system on which it is installed.

Because of its unique power to make SwiftView viewer available to anyone, anywhere, anytime, batch SwiftStamp requires annual maintenance fees, and special ASP pricing where applicable. Be sure to discuss SwiftStamp pricing with SwiftView Sales www.swiftview.com/sales before you begin your project.

Licensing and Installation

SwiftStamp can be licensed for unlimited stamping, a limited number of stamps per year, or a fixed block of stamps. It keeps a count of all stamps ever issued on the licensed system, in a shared, system-wide file. If the stamp count exceeds 85% of a licensed limit, SwiftStamp puts out a warning message (see Program Exit Codes below).

If you have purchased a limited license, you must install your licensed software within a few days of our shipping the software to you: 2 days for a block of stamps or 7 days for stamps per year. If you are using an evaluation copy, you must install within 2-16 days of downloading the evaluation copy. (This installation step is recommended, but not required for unlimited licenses, and can be done any time with unlimited licenses.)

When you install, you must run SwiftStamp once as Administrator (Windows) or root (Unix) in order for it to initialize the system-wide stamp count:

```
sv_stamp -I-1 -lyourpassword (if licensed)

sv_stamp -I-1 (if evaluation)
```

All of your first year's stamps, or your full block of stamps, may be used after the installation period ends. If you don't use up a year's stamps, they remain available for future use. If you need to increase your stamp rate, SwiftView simply issues you a new license string.

After the demo period sv_stamp may only execute on a single computer, using a SwiftView-supplied password locked to the system name. If you are having problems, be certain that the system name in the stamp program exactly matches your system name, and be sure you are actually running it on that system.

Stamping Files with batch SwiftStamp

Simply execute the program at the command prompt for a brief description of how to use it. The program is followed by parameters which specify the different operations. For example, to stamp a file called document.pcl you would execute the following line:

```
sv_stamp -lYourPasswordhere -fdocument.pcl
```

Note that there are no spaces between the parameter flag and its value.

Command Line Help

Typing 'sv_stamp -h' on a Win32 system gives full command line help as follows:

```
I:\temp_win32>sv_stamp -h

 The SwiftStamp file licensing program. Version 7.1.1.90e
 Copyright (c) 1989-2004 SwiftView, Inc.
 All Rights Reserved

This program is licensed for use by the company listed below.
Use by any other party for any purpose is forbidden.

SwiftView, Inc.


 usage: sv_stamp -lpassword -ffile -Ffile -v -M -s -u -h -d
      -ics"cmds" -c"icsfile" -i"infofile" -x"xmldatafile"
      -nuser -eemail -Ccompany -Icount

   -l  = Your license password provided by SwiftView, Inc.  It
is required to stamp
         after the evaluation timeout.
   -f  = The name of the file being licensed.  If not present
         SwiftStamp assumes stdin and stdout for data.
   -v  = Verify+extract metadata from an already-stamped file
       = or verify license password.
   -s  = Restamp an already-stamped file, or add a new stamp
         to an unstamped file.
   -u  = Remove an existing stamp.
   -F  = Initialize metadata and ICS commands from this already-
stamped file.
   -M  = Use macro ID # in the stamp. This option is rarely
         used. Macro ID 0 is the normal (default) case.
   -h  = Display this message and quit.
   -d  = Dummy parameter, prevents this message.
  -ics = Pipe-separated ICS commands
   -c  = add/extract ICS commands from/to a file
   -i  = set new metadata info text from a file
   -x  = set new metadata xml from a file or save it to file if
-v
   -n  = set new metadata full user name
   -e  = set new metadata user email address
   -C  = set new metadata company name
   -I  = set the stamp count higher (-I-1 initializes to zero)

   examples: sv_stamp -lYourPassword -fYourFile
             sv_stamp -lYourPassword < YourFile > StampedFile

   Host name: mustang
  IP address: 192.168.4.14
```

Verifying the Stamp

The -v option verifies that a file you have already attempted to stamp is indeed stamped. The -v option should not be used when the user is trying to stamp a document. Options listed after the -v option in the example above are ignored when the -v option is used.

Restamping / Unstamping

If SwiftStamp encounters a file which is already stamped, it will not add another stamp to that file. As a result, batch scripts such as those shown below can be used and reused to ensure an entire group of files are stamped without concern. Please note that SwiftStamp also has commands to 'unstamp' and 'restamp' files. Restamping might be desired to put new ICS commands on an already stamped file.

Zipping and Passwording Documents

SwiftView can view files that are zipped without unzipping them, allowing faster viewing and downloading of files. To use SwiftStamp in combination with ZIP, you should ZIP first and then stamp.

SwiftView also supports viewing standard password-locked zip files. Upon viewing in SwiftView the user will be prompted to enter the password.

Open-source zip utilities (Info-ZIP) are available as free downloads from www.swiftview.com/product/utilities/

Adding ICS Commands

ICS commands can be inserted in a stamp to control the appearance and function of SwiftView, to disable printing, to cause the document to open to a certain page at startup, to add markups, etc. For example, to stamp a file so that it cannot be printed and so that a message box comes up and alerts the user to that fact, you would execute the following as **one line**:

```
sv_stamp -lYourPassword -fdoc.pcl -ics"set output
disable|message \"You are not allowed to print this draft.\""
```

Note the method for placing quotes within quotes using the backslash. Also, the command "set output disable" was introduced in SwiftView Version 5.2.1 and will not work with older versions.

Adding long ICS commands

If you find that the length of the command line limits the number of commands you can place in your stamp, you can use the -c parameter to specify a file which contains the ICS commands you would like to insert. For example, if you wrote a text file called Markup.ics containing ICS commands you would insert those commands into your stamp as follows:

```
sv_stamp -lYourPassword -cMarkup.ics
```

Adding Metadata

SwiftStamp has the ability to store sets of specific or arbitrary "metadata" information in a file's stamp. Except for the data stored as XML, this information can be viewed in the SwiftView Viewer F8 file info dialog. The information automatically includes the hostname and date of stamp generation. A user name, user email address, company name, and arbitrary text and XML data can be added with SwiftStamp command line options -n -e, -C, -i and -x.

The first set of metadata in a stamp is assumed to be about the document "author", and subsequent sets about "reviewers". If you use the -F option, all sets of metadata from the given stamped file are copied into the new stamp, then any metadata added with the -neCix options is added as a subsequent, "reviewer" metadata set.

The -v option prints a formatted display of all of this metadata information to stdout, except the XML data which is written to the -xfile, from all metadata sets in order.

STDIN and STDOUT operation

SwiftStamp can stamp files in a stream process from STDIN to STDOUT. This is a very normal manner of processing on UNIX systems but is also provided on Win32. Simply omit the '-f...' parameter and add the '-d' parameter. If you do not use '-d', the command line message will be displayed and no processing will be performed.

Program Exit Codes

The sv_stamp program displays error messages to stderr on any error condition, and returns the following exit codes, so your software can report errors:
 0: File not stamped, file empty, license password is valid (-v), or displayed usage (-h) (no error)
 1: File is stamped. (no error)
-1: Improperly generated sv_stamp program.
-2: SwiftStamp beta has expired.
-3: SwiftStamp requires a license password.
-4: Invalid license password or expired rental.
-5: Invalid stamp in file, or invalid -M parameter
-6: Could not open file.
-7: Out of memory.
-8: Could not read file.
-10: Could not write stamp count file, contact SwiftView Support.
-11: Error 38946, contact SwiftView Support.
-12: SwiftStamp trial/startup period ended without ever being run; please obtain a new copy from SwiftView.
-13: (Warning only) You have issued %d SwiftStamps since.
  At this rate you will exceed your purchased stamping capacity of
  N stamps/month on .
  Contact SwiftView Sales to purchase additional capacity.
-14: You have issued %d SwiftStamps since ,
  exceeding your allowed stamp rate.
  Contact SwiftView Sales to purchase additional stamping capacity.
-15: Could not initialize stamp count - are you logged in as root/Administrator?
-16: Could not open up access to the stamp count file, are you logged in as root/Administrator?
-17: Decreasing the stamp count is not allowed.

Special attention needs to be given to the exit codes -13 and -14, as these can occur during normal operation of rate-limited SwiftStamp. Your software should report these messages to the user so that the license can be upgraded to a higher rate before SwiftStamp stops functioning. We recommend that you capture all error messages on stderr and present them to the user if SwiftStamp returns a negative exit code (i.e. other than zero or one).

Stamp an entire directory of files

Trivial Windows/Unix commands can be used to do this. Note that the SwiftStamp program must be in the PATH in all of these examples.
Win32 example **command line:**

```
FOR %i IN (*.hpg) DO sv_stamp -f%i -l...password...
```

Win32 example **batch file** (no command line parameters to batch file):

```
FOR %%i IN (*.hpg) DO sv_stamp -f%%i -l...password...
```

On Windows NT and 2000 (but not 95/98), add the command line option '/r' after FOR to stamp an entire tree of files.

UNIX script file - stamp one directory of PCL files:

```
#!/bin/sh
for file in *.pcl
  do sv_stamp -f$file -l...password...
done
```

UNIX script file - stamp tree of PCL files:

```
#!/bin/sh
for file in `find . -name "*.pcl" -print`
  do sv_stamp -f$file -l...password...
done
```