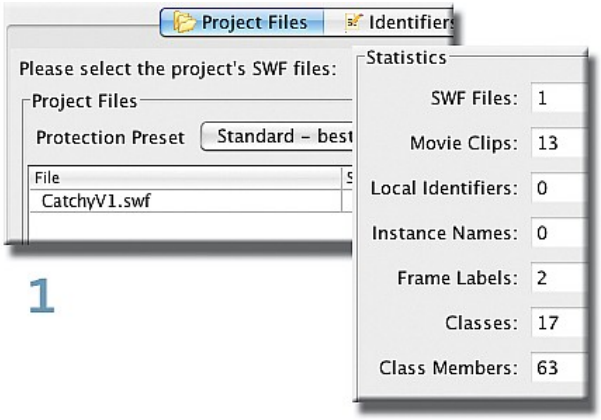
KINDISOFT

Getting Started Guide

Protecting your Flash™ applications is as easy as 1 - 2 - 3

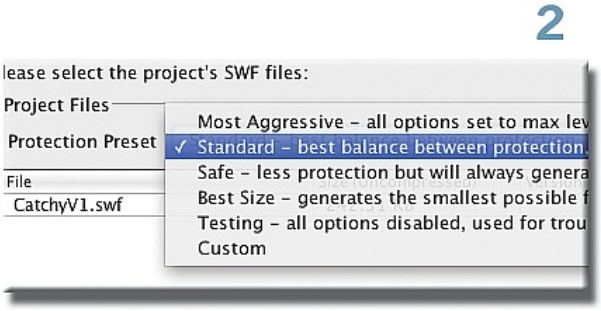
Getting Started

While secureSWF provides a wide array of protection for Flash applications, getting started with a level of security that could well be all you need is a simple 3-step operation.



Secure your SWFs – Easy as 1-2-3!

1. Add SWF file(s) into your project (Project Files tab)
You can add multiple files which will all be processed with the same settings. You should add all files that interact with each other so secureSWF can detect the references and make sure the obfuscated files will work the same way they did before obfuscation. The Statistics panel shows you info about the file or files in the project.



2. Look at Protection Presets choices
These are pre-defined security levels that may be enough for many needs. They offer different protection-to-file size-to-performance ratios. You can optionally set up custom levels exactly suited to your needs if presets are not quite what you want.

3. Specify prefix or postfix for output file and Protect
Add a prefix (e.g. secure_) and/or postfix (suffix) for the protected output version of your SWF file so that the original is not overwritten with the protected version. Then click the Protect button.

THAT'S ALL IT TAKES TO GET STARTED PROTECTING YOUR SWFs FILE!

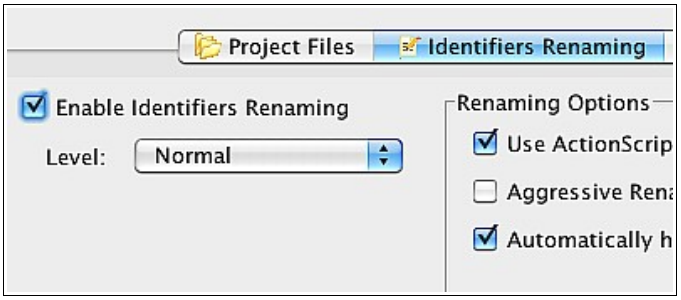
Of course you will want to explore the many protection options available in secureSWF to get the maximum benefits for your situation. The rest of this guide will introduce you to the major areas of the product to explore next.

At the conclusion of this guide you should be well on your way to understanding how to use secureSWF to achieve the security and performance levels that are right for your Flash applications.

Getting Started with secureSWF Protection Settings

While you can get a pretty good level of protection for your SWF files using presets, these are only the tip of the iceberg when it comes to your options for securing your Flash apps. It will of course take some time to master everything secureSWF has to offer, but it shouldn't take you too long to get acquainted with the major product features and settings and arrive to the level of protection that's good for your needs. Let's look briefly at the main things you should look at when devising a security scheme for your applications.

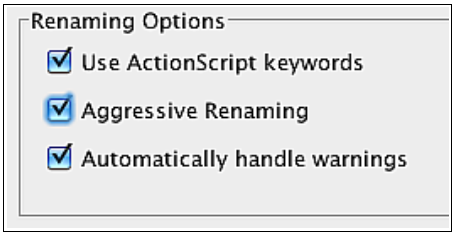
Identifiers Renaming Tab



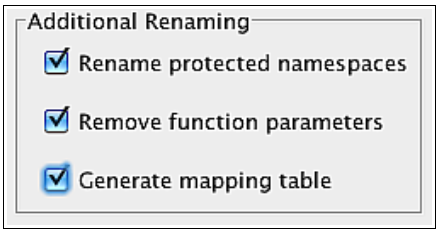
Renaming of identifiers in your application source code is one of the main ways secureSWF protects your apps from reverse engineering. The Identifiers Renaming tab provides a number of options that control how SecureSWF handles renaming as it obfuscates your source code.

Main Options:

- **Enable Identifiers Renaming -** This is a toggle that controls whether or not secureSWF will rename identifiers as part of the obfuscation process. Check it to enable renaming of identifiers.
- **Level -** This setting controls how renaming happens.
- **Aggressive Renaming -** If you check it, identifiers are renamed with non-printing characters, illegal names, etc. This generally results in smaller names (reducing the byte count in the final obfuscated file). Decompilers can't print or display identifiers.
- **Use ActionScript Keywords -** this causes keywords inside ActionScript code to be changed to be unintelligible and unrecognizable as keywords.
- **Automatically Handle Warnings** − secureSWF will handle warnings resulting from renaming of identifiers. Turn off this option if you want to see warnings.



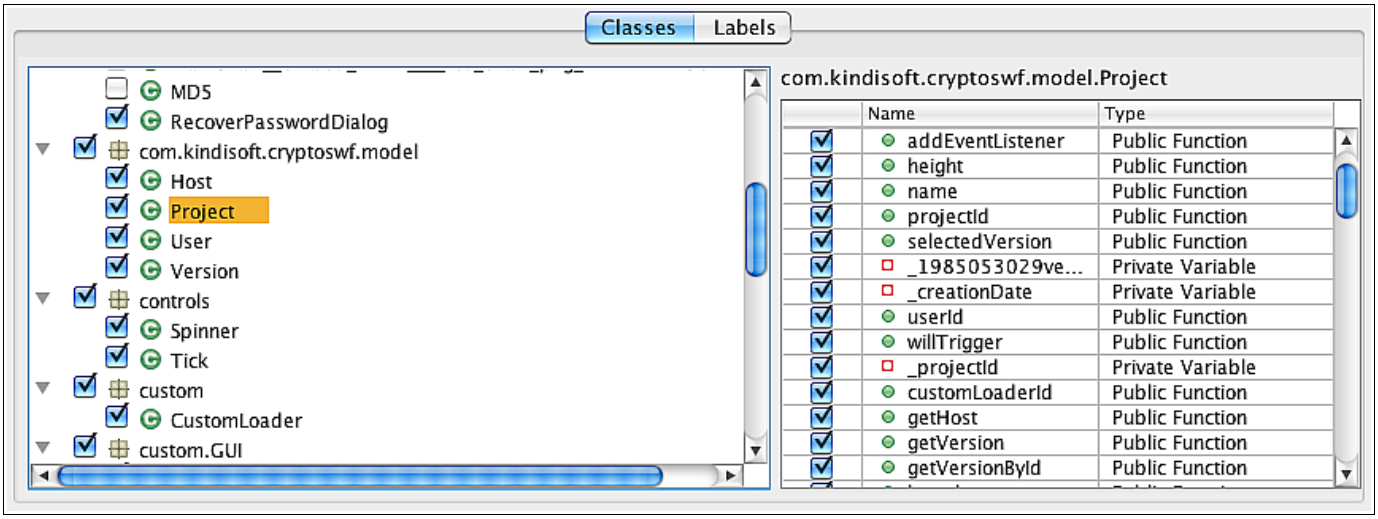
Additional Renaming Options



Fairly self explanatory. Mapping table is just a table of original identifiers mapped to renamed identifiers. It can Help de- obfuscate error message that might arise in your Flash app later. See Help if you need to delve deeper into these options.

Exploring Packages and Classes

The Identifiers Renaming tab displays a tree of the packages and classes in your application, and it shows which classes will have identifiers renamed with the standard renaming defaults. When you select any class on the left, its members are displayed on the right.



Basically what you want to do here is look through the classes, see which ones will have identifiers renamed by the default renaming (these are checked in the listing), and decide whether you want to have more classes processed, and set options for renaming. You can as get granular as you want by selecting individual classes for renaming and making settings for each selected class.

Protection Options Tab

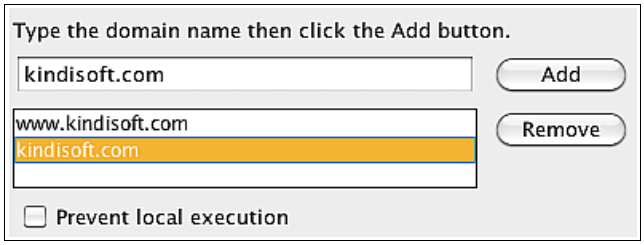
This tab provides the second major security features area of secureSWF. It provides options for code transformations, file optimization, domain locking, and encryption of embedded string literals.

Code Transformation

Code transformation foils decompilers by changing compiled code so that it can no longer be reverse engineered into source. You can set a number of options including Advanced options for a greater or lesser degree of transformation. Detailed information about each option can be found here.

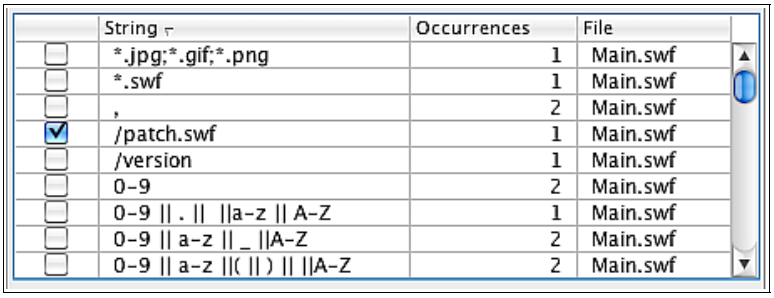
Encrypted Domain Locking

This feature enables you to specify the domains on which the protected SWF file(s) can be hosted. Files will not work if hosted on any other domain. Also can be used to prevent local execution offline.



Optimization

Optimizes the code itself and performs various operations to reduce file size and optimize performance. Detailed information about each option can be found here.



Literal Strings Encryption

Encrypts the literal data that's inside the code – hard-coded passphrases or URLs for example. Replaces literal strings with a special function call that returns the decrypted string, which is stored in an encrypted form.

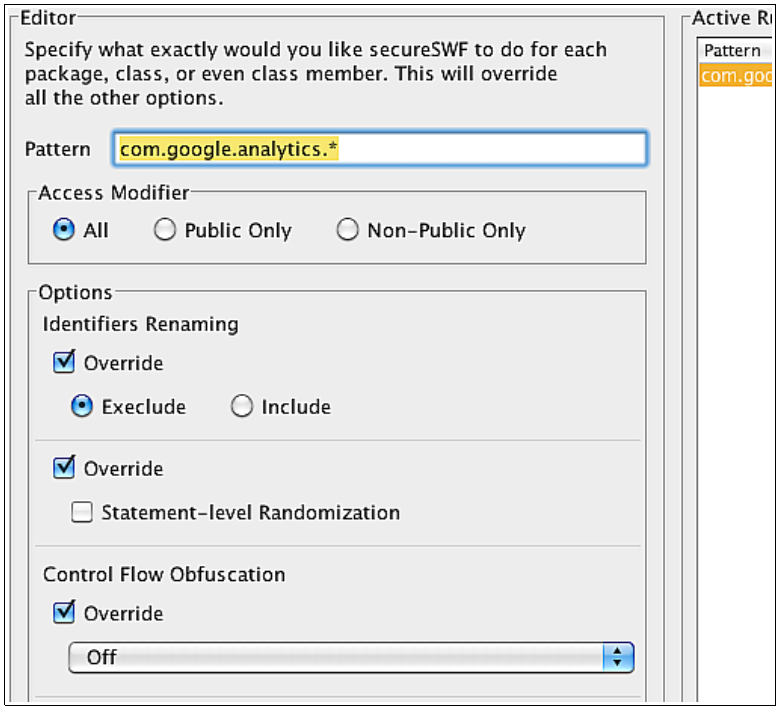
Rules Tab

While previous tabs enable you to specify how your code will be obfuscated down to a single function or identifier, we know that pin pointing every single one could be a very time consuming process. Because of this, we've added the Rules tab.

Here you can just tell secureSWF what to do with all the code in an entire package, or a single class. Just write in the pattern and check which options you want to override.

You can use this, for example, to avoid obfuscating code merged from SWC libraries provided by third-parties.

Rules will override all other settings.



Pulling it all together

Hopefully by now you have enough direction to begin working with secureSWF. Be sure to refer to the Help topics as you go along. The main thing you are looking to achieve is an optimal balance of security, performance, and file size.

To keep abreast of the latest news, pop in to the Kindisoft blog at **http://www.kindisoft.com/blog/** or tune in to Kindisoft on Twitter at **http://twitter.com/secureswf/** .

If you need support, don't hesitate to start a new ticket on **http://www.kindisoft.com/support/**. We are there to help!