

# **EXHIBIT 15**



**WIKIPEDIA**  
The Free Encyclopedia

Navigation

- [Main page](#)
- [Contents](#)
- [Featured content](#)
- [Current events](#)
- [Random article](#)
- [Donate to Wikipedia](#)

Interaction

- [Help](#)
- [About Wikipedia](#)
- [Community portal](#)
- [Recent changes](#)
- [Contact Wikipedia](#)

Toolbox

- [What links here](#)
- [Related changes](#)
- [Upload file](#)
- [Special pages](#)
- [Permanent link](#)
- [Cite this page](#)

Print/export

- [Create a book](#)
- [Download as PDF](#)
- [Printable version](#)

Article [Discussion](#)

Read [Edit](#) [View his](#)



# ActionScript code protection

From Wikipedia, the free encyclopedia



This article has multiple issues. Please help [improve it](#) or discuss these issues on the [talk page](#).

- It is written like an [advertisement](#) and needs to be rewritten from a [neutral point of view](#). Tagged since January 2010.
- It may be [confusing or unclear](#) for some readers. Tagged since January 2010.

ActionScript code protection. [ActionScript](#) is the main language for developing flash products. [Code obfuscation](#) is the process of transforming code into a form that is unintelligible to human readers while preserving the functionality and structure for computers. Flash developers use a variety of code encryption and obfuscation methods in order to but most obfuscation methods are kept secret because they are used in popular ActionScript obfuscators.

Contents [\[hide\]](#)

- [1 Overview](#)
- [2 Examples](#)
  - [2.1 ActionScript 2.0](#)
  - [2.2 ActionScript 3.0](#)
- [3 See also](#)
- [4 External links](#)
  - [4.1 Adobe documentation and references](#)

## Overview

[\[edit\]](#)

Obfuscation is the process of modifying ActionScript code. Obfuscation engineering makes it impossible to use the original code for programming by mixing up functions, arguments and variables names; however the SWF file with obfuscated code stays readable for Flash players.

ActionScript obfuscation algorithms are aimed to resist Flash decompilation and protect development's work from stealing. Nowadays, there are many different freeware and shareware tools that provides different obfuscation options. Some SWF Protectors implement lexical transformations - such as identifiers renaming, control flow transformation, and data abstraction transformation - that make it harder for decompilers to generate correct and usable output. Others - insert certain traps based on decompilers imperfections. Some decompilers manage to open protected files and maybe extract some of its resources (sounds, images, etc.), but they cannot read the ActionScript code correctly. It often happens that when protected SWF file is added to decompilation software, the latter may crash or quit unexpectedly.

## Examples

[\[edit\]](#)

The easiest way to understand the benefit of SWF obfuscation is to compare a non-protected SWF ActionScript code with protected one.

## ActionScript 2.0

[edit]

Code before obfuscation:

```
stop();
if (this.holder1 == Number (this.holder1)) {
    loadMovieNum ("jopeClipLoader5.swf", this.holder1);
} else {
    this.holder1.loadMovie("jopeClipLoader5.swf");
}
```

Code after obfuscation:

```
do {
    if (false) {
        ((add()).holder1); // not popped
        if (true) {
            // unexpected jump
            do {
                stop();
            } while (false);
            // unexpected jump
            if (false) {
                Set("\x0B\x1A\x13\x16", true);
            } while (true);
            do {
                if (true) {
                    // unexpected jump
                    // unexpected jump
                    if (false) {
                        } while (true);
                    } do {
                        // unexpected jump
                    } while (true);
                    // swfAction0xAD hexdata
                    0x52,0x17,0x99,0x02,0x00,0x39,0x00,0x9A,0x01... // Unknown action
                }
                ((add()).holder1)[(!"holder1".holder1()).holder1].loadMovie();
                // unexpected jump
                /* Error1016 */
                // unexpected jump
                do {
                    (this); // not popped
                    if (true) {
                        // unexpected jump
                    } while (this);
                    (this); // not popped
                    // unexpected jump
                }
            }
            if (false) {
            } while (true);
            do {
                // unexpected jump
            } while (true);
            // swfAction0xAD hexdata
            0x52,0x17,0x99,0x02,0x00,0x39,0x00,0x9A,0x01,0x00,0x00,0x99... // Unknown action
        }
        "holder1".holder1.loadMovie();
        // unexpected jump
        /* Error1016 */
        // unexpected jump
        do {
            (this); // not popped
            if (true) {
                // unexpected jump
            } while (this);
            (this); // not popped
            // unexpected jump
        }
    }
}
Set("\x0B\x1A\x13\x16", true);
} while (true);
do {
    ( /* Error1010 */ ); // not popped
    if (true) {
```

```
// unexpected jump
// unexpected jump
if (false) {
} while (true);
do {
    // unexpected jump
} while (true);
// swfAction0xAD hexdata
0x52,0x17,0x99,0x02,0x00,0x39,0x00,0x9A,0x01,0x00,0x00,0x99,0x02... // Unknown action
}
```

ActionScript 3.0

[edit]

Code before obfuscation:

```
private function getNeighbours(i:int, j:int):Array{
    var a:Array = new Array();
    for (var k = 0; k < 8; k++){
        var ni = i + int(neighbour_map[k][0]);
        var nj = j + int(neighbour_map[k][1]) ;
        if (ni < 0 || ni >= xsize || nj < 0 || nj >= ysize)
            continue;
        a.push(Cell(cells[ni][nj]));
    }
    return a;
}
```

Code after obfuscation:

```
private function getNeighbours(_arg1:int, _arg2:int):Array{
    var _local3:Array = -(((null - !NULL!) % ~(undefined)));
    var _local4:*;
    var _local5:*;
    var _local6:*;
    _local3 = new Array();
    _local4 = 0;
    for (;//unresolved jump
    , _arg2 < 8;_local4++) {
        _local5 = (_arg1 + int(!NULL!));
        _local6 = (_arg2 + int(!NULL!));
        if (true){
            _arg1 = (((//unresolved nextvalue or nextname << !NULL!) + !NULL!)
<< undefined);
            _arg1 = (!(!NULL!) ^ !NULL!);
            (!NULL! instanceof !NULL!);
            var _local1 = (((!NULL! as !NULL!) + !NULL!) == this);
            if (!(!NULL! == !NULL!)){
                -((true << !NULL!).push(Cell(cells[_local5][_local6])));
            }
        }
        if (!true){
            (_local6 < 0);
            (_local6 < 0);
            (_local5 < 0);
        }
    }
    return (_local3);
}
```

See also

[edit]

- ActionScript
- ActionScript Foundry
- Adobe Flash
- Obfuscated code
- SWF

External links





[edit]


Adobe documentation and

[edit]


Wikibooks has a book on the

references

- [ActionScript Technology Center](#) 
- [ActionScript 2.0 Language Reference](#) 
- [ActionScript 3.0 Language & Component Reference](#) 
- [Flex 3 LiveDocs: Programming ActionScript 3.0](#) 



topic of  
[ActionScript Programming](#)



Wikiversity has learning materials about  
[ActionScript:Introduction](#)

Categories: [Adobe Flash](#) | [Scripting languages](#) | [Obfuscation](#)

This page was last modified on 2 March 2011 at 23:17.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Contact us](#)

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#)

