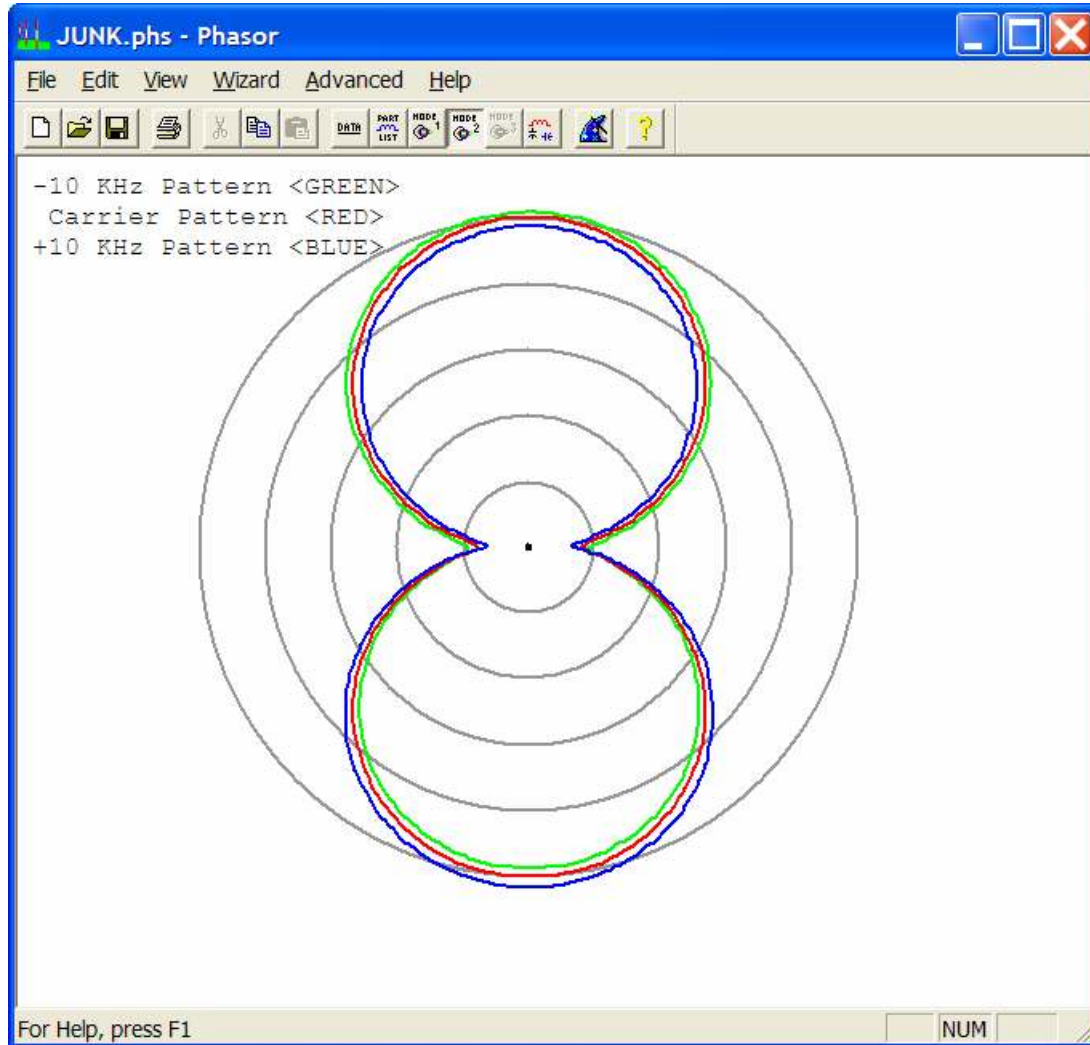


# Encrypting Flex, Protecting Revenue, and Making Cash!

Andrew Westberg  
360|Flex – San Jose



# Encrypting Flex

# Code Protection Level 1 – Embedding sensitive data

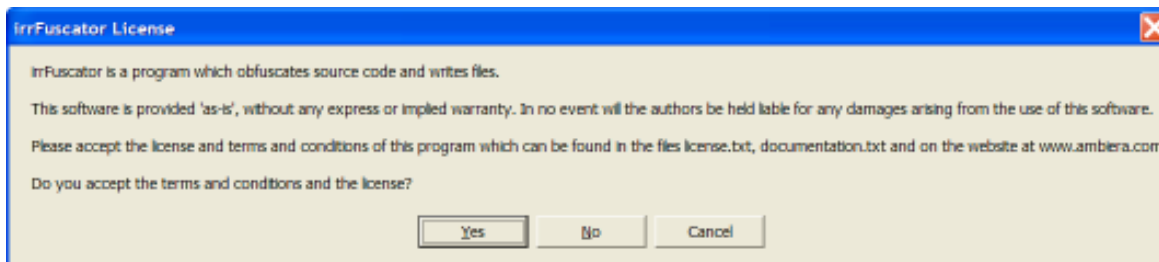
- Most decompilers don't look at embedded data.
- Given that SWF is an open file format, nothing is really stopping them from doing this in the future.
- Useful for quick/dirty storage of Webservice credentials.

# Code Protection Level 2 - Obfuscation

- Main Concept – Scramble the SWF bytecode so it looks like crap when decompiled.

# irrFuscator

- Scrambles source code **BEFORE** compiling



- Not really useful as you can see

```
<mx:WebService
  id="adminService"
  wsdl="{baseUrl}/wsdl/NitroAdmin.wsdl"
  useProxy="false"
  showBusyCursor="true">

  <mx:operation name="_yl1396"/>
  <mx:operation name="_dn817"/>
  <mx:operation name="_la1346"/>
  <mx:operation name="_kd231"/>
  <mx:operation name="_rl1128"/>
  <mx:operation name="_uh1320"/>
  <mx:operation name="_qx832"/>
  <mx:operation name="_so890"/>
  <mx:operation name="_np39"/>
  <mx:operation name="userGraph"/>
  <mx:operation name="_le1274"/>
  <mx:operation name="saveNotification"/>
  <mx:operation name="licenseGraph"/>
  <mx:operation name="_gu1317"/>

</mx:WebService>
```

# Decompiled Cairngorm Delegate (without obfuscation)

```
package com.simplifiedlogic.nitroadmin.cairngorm.business
{
    import com.universalmind.cairngorm.business.*;
    import mx.rpc.*;

    public class LoginDelegate extends Delegate
    {

        public function LoginDelegate(:IResponder = null, :String = "adminService")
        {
1 0 101 1 10      super(, );
            return;
        }// end function

        public function loginAppid(_2022732339loginBtn:String) : void
        {
1 0 151 1 17      var _loc_2:* = service.loginAppid(_2022732339loginBtn);
            prepareHandlers(_loc_2);
            return;
        }// end function

    }
}
```

# SWFProtect 1.6 (demo version)

- Didn't do much for Flex swfs

```
package com.simplifiedlogic.nitroadmin.cairngorm.business
{
    import com.universalmind.cairngorm.business.*;
    import mx.rpc.*;

    public class LoginDelegate extends Delegate
    {

        public function LoginDelegate(:IResponder = null, :String = "adminService")
        {
1 0 101 1 10      super(, );
            return;
        }// end function

        public function loginAppid(_2022732339loginBtn:String) : void
        {
1 0 151 1 17      var _loc_2:* = service.loginAppid(_2022732339loginBtn);
            prepareHandlers(_loc_2);
            return;
        }// end function

    }
}
```

# SWFEncrypt 5.0

- Slightly misleading name

```
package com.simplifiedlogic.nitroadmin.cairngorm.business
{
    import com.universalmind.cairngorm.business.*;
    import mx.rpc.*;

    public class LoginDelegate extends Delegate
    {

        public function LoginDelegate(:IResponder = null, :String = "adminService")
        {
1 0 101 1 10      // Jump to 14;
            super(, );
            return;
            // Jump to 45;
        } // end function

        public function loginAppid(□□□□:String) : void
        {
1 0 151 1 17      // Jump to 14;
            var _loc_2:* = service.loginAppid(□□□□);
            prepareHandlers(_loc_2);
            return;
            // Jump to 62;
        } // end function
    }
}
```

# Why Obfuscators SUCK!

- The code you WRITE isn't the code you RUN.
- Runtime debugging is a PITA because all your methods and vars have (in theory) been renamed.
- I don't know of a single good one for Flex.
- Totally ignore powerful Flex concepts such as Modules, RSLs, etc...
- They will ALWAYS be behind the Flex technology curve.

# Code Protection Level 3 – Encrypted Interfaces

- Create a SWC
- Encrypt its library.swf file
- Have a Factory class decrypt and load it
- Provide an interface class to use it
- Works well for API type of SWCs
  
- Doesn't work for MXML – can't use an Interface in MXML, and flashplayer doesn't let you override a stubbed-in version with loaded content for security reasons.

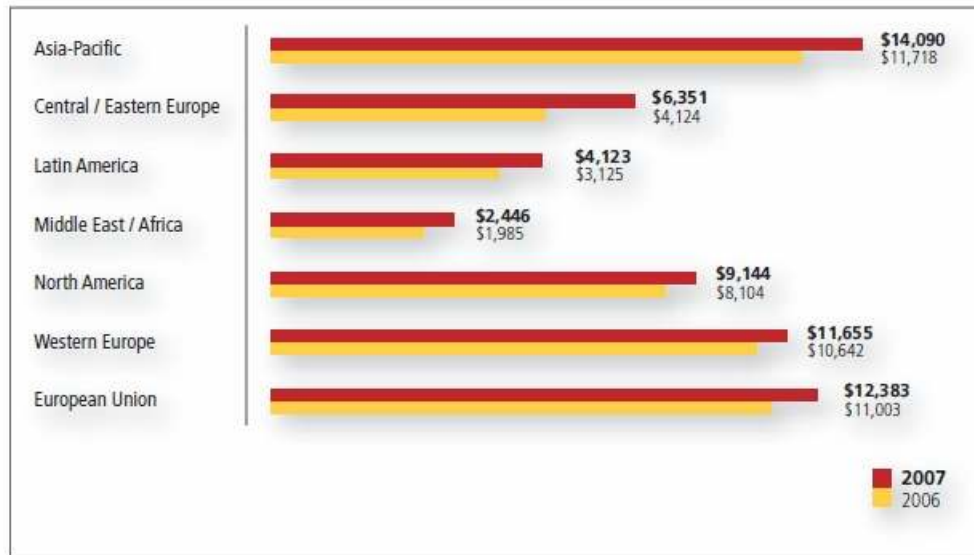
# Code Protection Level 4 – Nitro-LM Encrypted Modules

- Asymmetric vs. Symmetric Encryption
- Works with MXML-based classes
- Gives you a ton of other features beyond just encryption.
  - Talk to the guys at the Simplified Logic booth if you want to see a more feature-complete demo.
  - [Imsales@simplifiedlogic.com](mailto:Imsales@simplifiedlogic.com)

Nitro-LM AIR app /w encrypted  
modules

# Protecting Revenue

## 2007 Dollar Losses by Region (\$M)



### 2007 Software Titles Most Frequently Pirated

#### -By Companies

1. Symantec Norton Anti-Virus
2. Adobe Acrobat
3. Symantec PC Anywhere
4. Adobe PhotoShop
5. Autodesk AutoCAD
6. Adobe DreamWeaver
7. Roxio Easy CD/DVD Creator
8. Roxio Toast Titanium
9. Ipswitch WS\_FTP
10. Nero Ultra Edition

### 2007 Software Titles Most Frequently Pirated

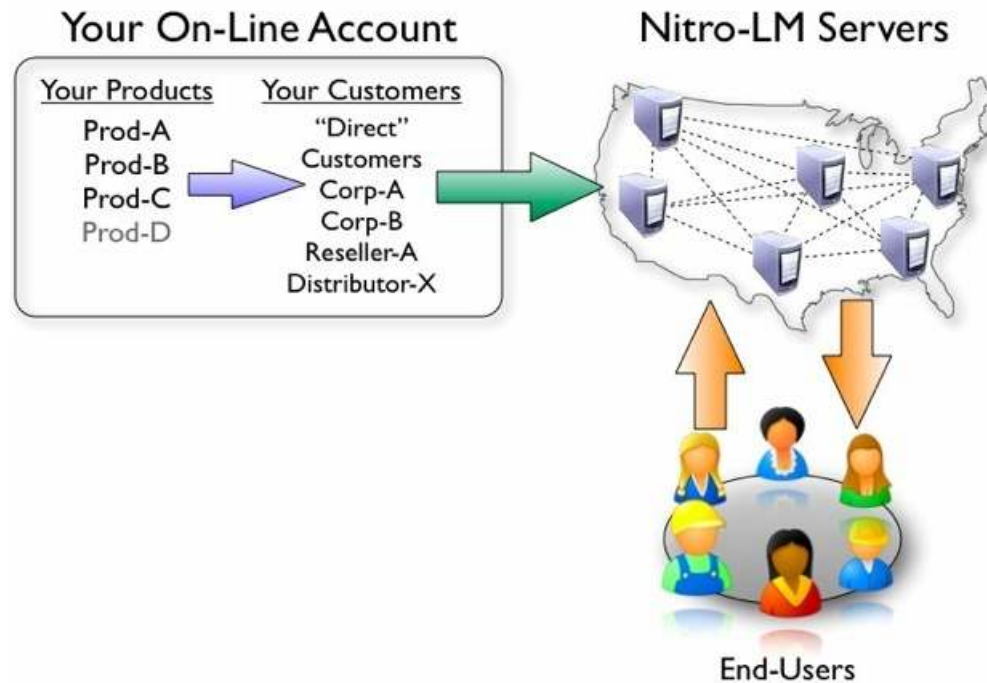
#### -on the Internet

1. McAfee VirusScan
2. Symantec Norton Anti-Virus
3. McAfee Internet Security Suite
4. Intuit TurboTax
5. Adobe Photoshop
6. Adobe Acrobat
7. Intuit Quicken Home and Business
8. Symantec Norton pcAnywhere
9. Symantec Norton Ghost
10. Adobe Creative Suite

# Verify the user's identity

- The “duh” of protecting revenue
- Make sure the user is who they say they are and verify their e-mail address.
- If they have a valid account for using a license of your software, make it **painless** for them to get it.
- Ensure they can't share accounts or ghost machines.

# How Nitro-LM verifies the user



(Show LM Admin Pool Screen & Restrictions)

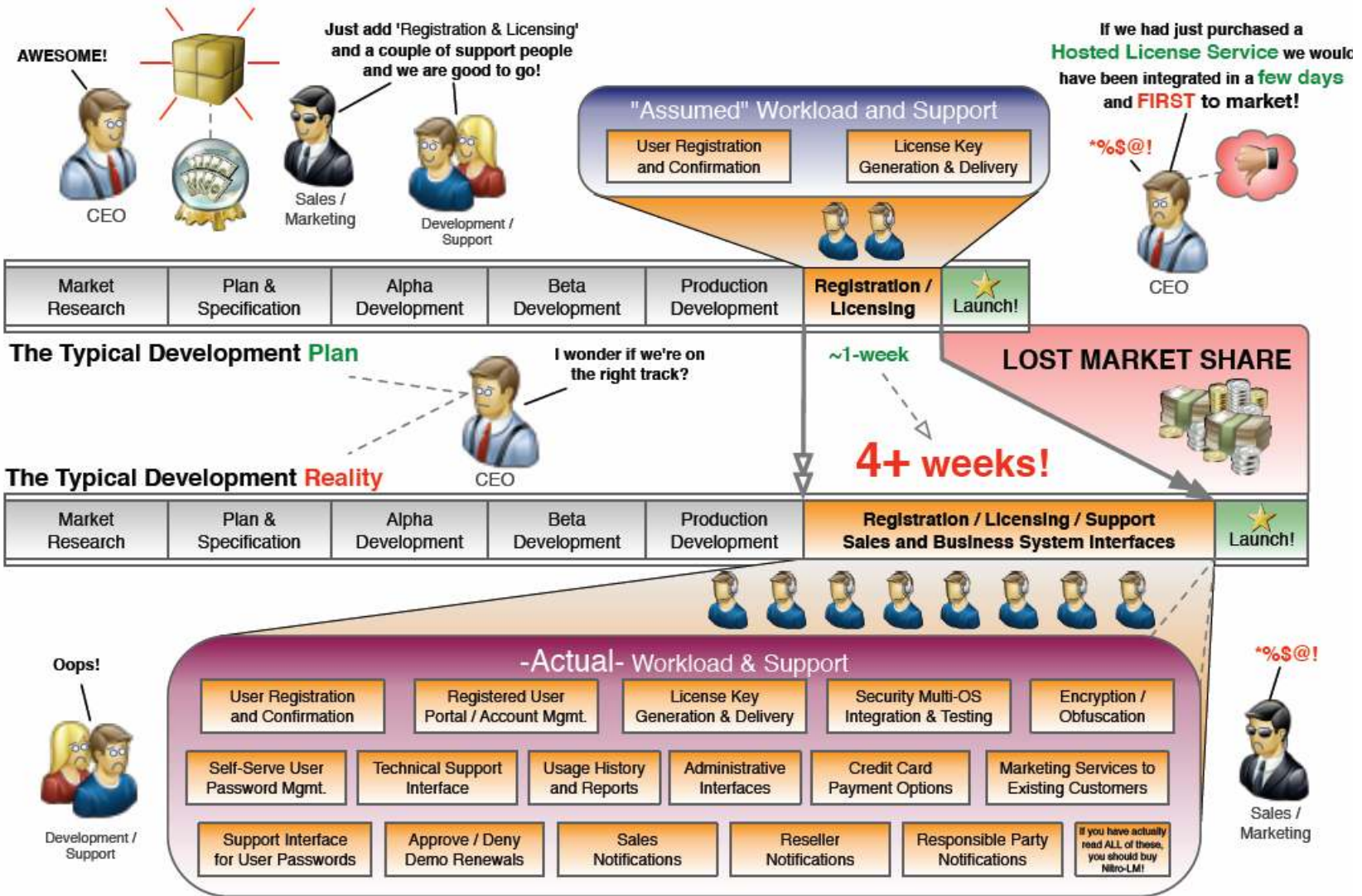
# Making Cash

# Making Cash through Notifications

- Lets you know when a customer runs out of licenses (sales should be on the phone)
- Lets you know when a potential customer demos your software.
- Allows your sales people to **stop bugging** your customers unless they actually need assistance.
- Sales people can focus on new sales instead of babysitting existing customers.

(Show New Notifications screen)

# Be first to market



# Common Objections to Nitro-LM

- It's BIG brother!
  - It's just an API, **YOU** can make it either a heavy-handed, or light touch.
- It does 50 things, I only need feature X
  - It's an API, just implement the features you need. You can always expand later.
- It doesn't do X. I need feature X.
  - We have an enhancement process. If you're willing to pay for an enhancement, you go to the front for consideration.

# Nitro-LM Roadmap

- Credit Card Processing

The screenshot displays the 'Merchant Interface <Licensee / Customer>' window, specifically the 'Account Profiles' tab. The interface is divided into two main sections: 'Payment Profiles' and 'Shipping Profiles'. Each section includes a 'Profile Status' dropdown (set to 'Active'), a 'CIM Profile ID' field, and a list of profiles with '+' and '-' icons. The 'Payment Profiles' section has a 'Send Test Transaction' button and a 'Test Transaction Mode' dropdown set to 'testMode'. The 'Shipping Profiles' section has a 'CIM Shipping Profile ID' field. At the bottom, there are 'Delete' and 'Save' buttons for both profile types, and 'Ok' and 'Cancel' buttons.

Callouts and annotations include:

- Name of Card Profile Profile**: Points to the profile name field in the Payment Profiles section.
- Visual Status of Card (Stop = Disabled Check = Active)**: Points to the profile status dropdown in the Payment Profiles section.
- Refresh Interface**: Points to the refresh icon in the top right corner.
- USER SELECTION!!!**: A yellow box pointing to the 'End User' dropdown.
- Selected Registered User with Permission to Create Credit Card Transactions - Selection - Adjusts Payment & Shipping Profiles**: Points to the 'End User' dropdown.
- General Information Tab for how to use/administer Payment Profiles**: Points to the 'Payment Profiles' tab header.
- Card Profile Status <Active / Disabled>**: Points to the profile status dropdown in the Payment Profiles section.
- Add New Profile Opens Dialog to New Profile Name**: Points to the '+' icon in the Payment Profiles section.
- Refresh display with CIM Payment Profile ID after successfully created**: Points to the 'CIM Payment Profile ID' field.
- Test Modes for AMI for Validating Credit Card Information testMode = field validation liveMode = \$0.01 transaction**: Points to the 'Test Transaction Mode' dropdown.
- Address Profile Status <Active / Disabled>**: Points to the profile status dropdown in the Shipping Profiles section.
- Refresh display with CIM Address Profile ID after successfully created customerAddressId=**: Points to the 'CIM Shipping Profile ID' field.

# Nitro-LM Roadmap

- Some form of **SWC encryption** so individuals/groups can sell Component Libraries with confidence.
- Make sure the libraries work the same as the original.
- Ensure they can't be decompiled.
- Ensure they can't be shared.
- Ensure there is no runtime dependency on the Nitro-LM license service (only compile-time license checks to ensure the developer has purchased a valid license to the swc component)
- Ensure that the SWC library developer gets **\$PAID\$**

# What does a fully blown cairngorm encrypted Flex app look like?



# Development Stack

- Uses `<nitroIm:EncryptedModuleLoader>`
- Skinned with Degrafa ([degrafa.com](http://degrafa.com))
- Graphs using Birdeye RaVis (google code)
- UM Cairngorm Extensions
- SOAP WebServices – (not Remoting)

# Thanks!

Visit the Simplified Logic booth for  
more information.

(If you're a "rockstar", drop off your resume too)