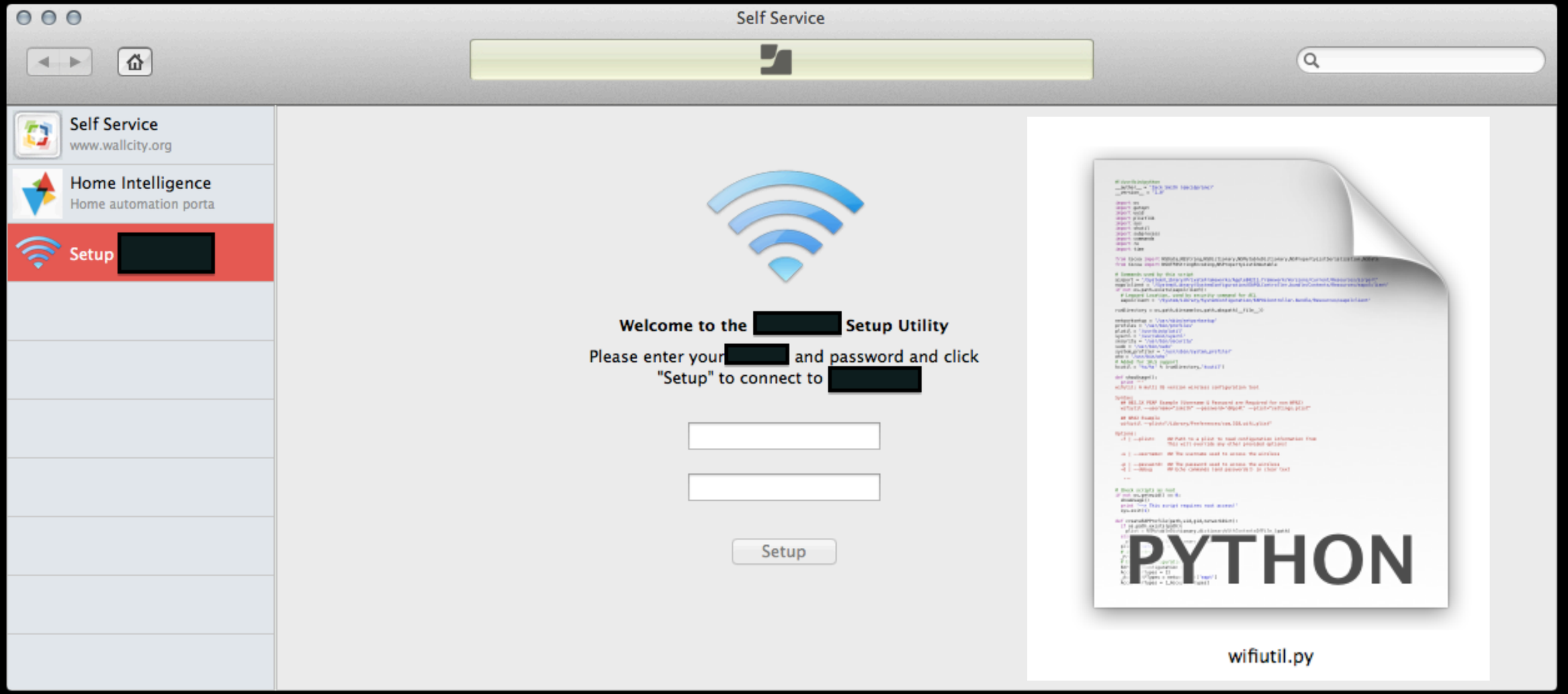
Server AD Migration & integration

@acidprime

For the record... #!/bin/bash



https://github.com/acidprime/WirelessConfig

Once upon a time...

Once upon a time...

I wish I could automate moving to a new authentication system.

Why Active Directory?

Why Active Directory?

- Because when you google for problems, you find answers, not your own blog posts.

Why Active Directory?

- Because when you google for problems, you find answers, not your own blog posts.
- Most any enterprise solution that supports external auth normally works with it or syncs with it.

Why Active Directory?

- Because when you google for problems, you find answers, not your own blog posts.
- Most any enterprise solution that supports external auth normally works with it or syncs with it.
- You need to manage windows machines

Why Active Directory?

- Because when you google for problems, you find answers, not your own blog posts.
- Most any enterprise solution that supports external auth normally works with it or syncs with it.
- You need to manage windows machines
- You want to run critical parts of your infrastructure on something thats not the server equivalent of a toaster.

“By failing to prepare, you are preparing to fail.”

-Benjamin, Franklin

Planning is half the battle

Planning is half the battle

- I once migrated 5,000 workstations in 5 days with 3 errors (of which we knew would happen going in)

Planning is half the battle

- I once migrated 5,000 workstations in 5 days with 3 errors (of which we knew would happen going in)
- I also once migrated 10,000 users with 1,000+ errors.

Planning is half the battle

- I once migrated 5,000 workstations in 5 days with 3 errors (of which we knew would happen going in)
- I also once migrated 10,000 users with 1,000+ errors.
- Start small, work out kinks with testing groups

Planning is half the battle

- I once migrated 5,000 workstations in 5 days with 3 errors (of which we knew would happen going in)
- I also once migrated 10,000 users with 1,000+ errors.
- Start small, work out kinks with testing groups
- Try and not change everything at once.

Planning is half the battle

- I once migrated 5,000 workstations in 5 days with 3 errors (of which we knew would happen going in)
- I also once migrated 10,000 users with 1,000+ errors.
- Start small, work out kinks with testing groups
- Try and not change everything at once.
- Communication is key

LDIF

- Alter the data stream to re-write the file
- ldifde can do some of the work for you
- RFC 2307
- Intentional concerns (Base64)

BASE64 vs LDIF

- d:: denotes BASE64 encoded attributes.
- 72 Characters with line endings
- LDIF does not support line breaks like this without a leading space

Extending your schema

- Its simple
- Political not technical
- Apple prefixes don't conflict
- easier migration

Batch File

```
if %1!==!  goto Usage

REM First load the attributes before we go to the classes
ldifde /j . /k /i /f OpenDirectory.ldf /v /c "dc=X" %1

goto end

:Usage

@echo Usage:  Load_Apple ["Domain Path"]
@echo.
@echo For example:
@echo.
@echo Load_Apple "DC=MyDomain,DC=Com"
@echo.
@echo.
:end
```

Simple VBS Autorun

```
Set rootDSE = GetObject("LDAP://RootDSE")

domain = rootDSE.Get("rootDomainNamingContext")

RunLine = "load_apple " & " " & """" & domain & """"

CreateObject("Wscript.Shell").Run RunLine, 1, True


wscript.echo "Done"
```

Computer Management

Computer Management

- computer lists not groups

Computer Management

- computer lists not groups
- $guest computer account?

Computer Management

- computer lists not groups
- $guest computer account?
- CN=Groups does not exist

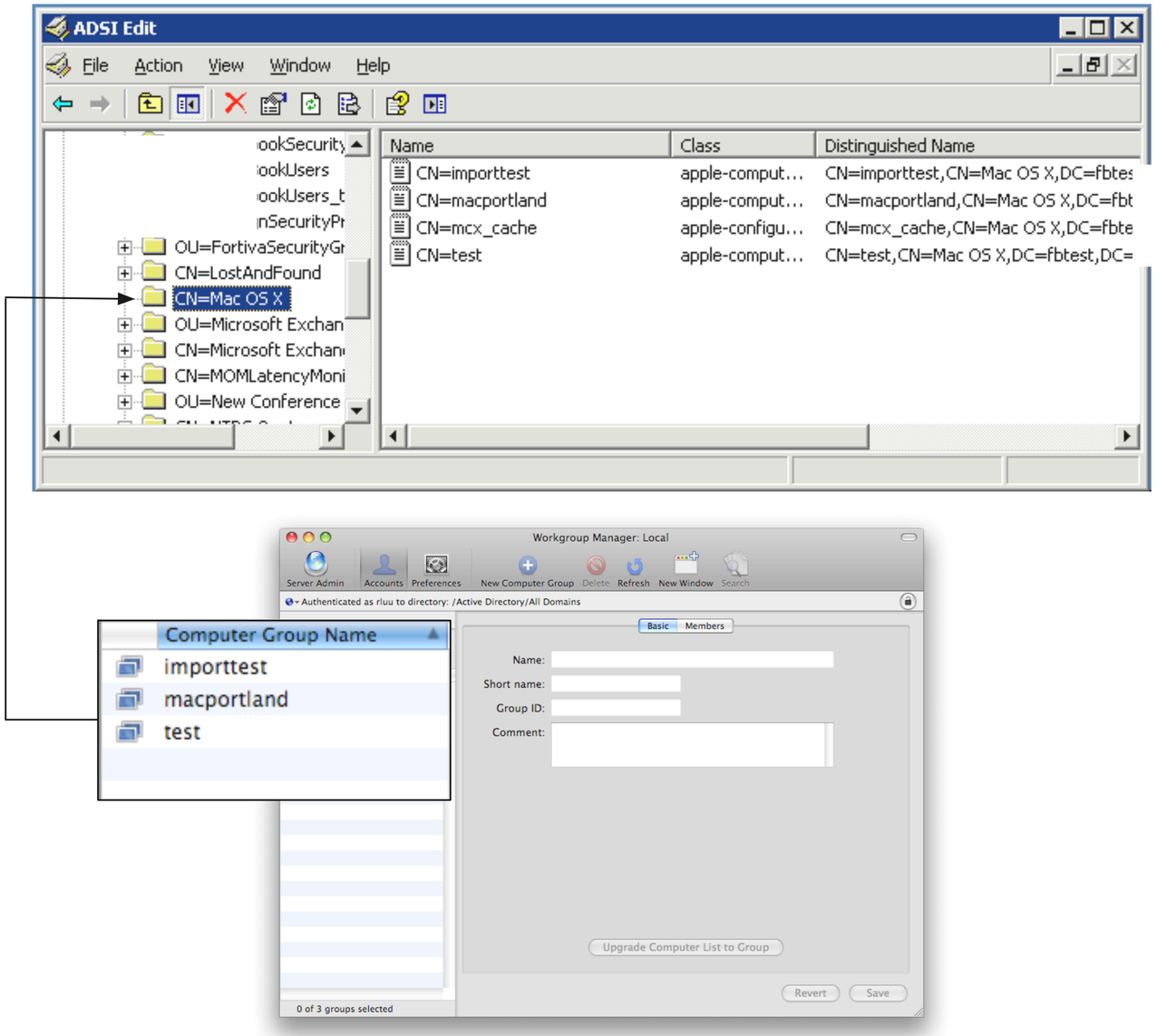
Computer Management

- computer lists not groups
- $guest computer account?
- CN=Groups does not exist
- 10.6+ Workgroup manager

Computer Management

- computer lists not groups
- $guest computer account?
- CN=Groups does not exist
- 10.6+ Workgroup manager
- Profile management

CN=Mac OS X


ADSI Edit
File Action View Window Help
OU=FortivaSecurityGr
CN=LostAndFound
CN=Mac OS X
OU=Microsoft Exchan
CN=Microsoft Exchan
CN=MOMLatencyMoni
OU=New Conference
Name Class Distinguished Name
CN=importtest apple-comput... CN=importtest,CN=Mac OS X,DC=fbtes
CN=macportland apple-comput... CN=macportland,CN=Mac OS X,DC=fbt
CN=mcx_cache apple-configu... CN=mcx_cache,CN=Mac OS X,DC=fbte
CN=test apple-comput... CN=test,CN=Mac OS X,DC=fbtest,DC=
Workgroup Manager: Local
Server Admin Accounts Preferences New Computer Group Delete Refresh New Window Search
Authenticated as rluu to directory: /Active Directory/All Domains
Basic Members
Computer Group Name
importtest
macportland
test
Name:
Short name:
Group ID:
Comment:
Upgrade Computer List to Group
Revert Save
0 of 3 groups selected

Someone rewrite this

Someone rewrite this

```
#!/bin/bash
LDAP_SERVER="ldap.example.com"
IFS=$'\n'
ldapsearch -LLL -h "$LDAP_SERVER" -x -D uid=diradmin,cn=users,dc=example,dc=com -w password -b
cn=groups,dc=example,dc=com '(&(objectClass=posixGroup)(!(cn=Domain*))(!(cn=com.apple.*))(!(cn=admin))(!
(cn=staff)))' description apple-group-realname memberUid |
sed 's/^apple-group-realname:/name:/g' |
sed 's/cn=groups,dc=example,dc=com/ou=ImportedUsers,dc=ad,dc=example,dc=com/g' |
while read line ; do
  if [ "$line" != "${line/dn://}" ] ; then
    declare MY_UID="$(echo "$line" | awk -F'[=,]' '/^dn:/{print $2}' )"
    declare MY_CN="$(ldapsearch -LLL -h "$LDAP_SERVER" -x -D uid=diradmin,cn=users,dc=example,dc=com -w
password -b cn=groups,dc=example,dc=com "(&(objectClass=posixGroup)(&(cn=$MY_UID)))" apple-group-realname
|

  awk /^apple-group-realname:/ |

  sed 's/^apple-group-realname: //g')"
  declare MY_NEWLINE="$(echo "$line" | sed "s/$MY_UID/$MY_CN/g" )"
  echo "$MY_NEWLINE"
  echo "changetype: add"
  echo "objectClass: top"
  echo "objectClass: group"
  echo "objectClass: apple-group"
  echo "groupType: -2147483646"
  echo "cn: $MY_CN"
```

OD vs AD object constraints

- telephoneNumber
- aliases vs sAMAccountName

Stale users in password server

```
#!/bin/bash
declare -x STALE_USERS_REPORT="/Library/Caches/NonMigratedLastLogin.csv"
rm "$STALE_USERS_REPORT"
IFS=$'\n'
for LINE in `ssh root@od.master.com mkpassdb -dump` ; do

  declare USER_NAME="$(echo "$LINE" | awk '{print $4}')"
  declare LAST_LOGIN_DATE="$(echo "$LINE" | awk '{print $5}')"
  declare LAST_LOGIN_TIME="$(echo "$LINE" | awk '{print $6}')"

  echo "Processing: $USER_NAME"
  [ "${#USER_NAME}" -eq 0 ] && continue

  if grep "$USER_NAME" /Library/Caches/NotMigratedLog.log &>/dev/null ; then
    echo "$USER_NAME,$LAST_LOGIN_DATE,$LAST_LOGIN_TIME" >>"$STALE_USERS_REPORT"
  fi

done
```

```
dn: cn=Zack Smith,ou=ImportedUsers,dc=ad,dc=318,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
objectClass: apple-user
instanceType: 4
objectCategory: cn=Person,cn=Schema,cn=Configuration,dc=ad,dc=318,dc=com
uidNumber: 1082
sAMAccountName: zsmith
cn: Zack Smith
givenName: Zack
sn: Smith
street: 44 Beer blvd
l: Portland
st: OR
postalCode: 97212
c: USA
apple-imhandle: Jabber:zsmith@318.com
mail: zsmith@318.com
description::
ABUlhoaFkzUnNlU0JvYjNjZ1NTQm1aV1ZzSUdGaWIzVjBJRUZDT2dwSVpTQm9ZWE1nWVd4c0lHOW1JSFJvWlNCMmFYSjBkV1Z6SUVrZ1
pHbHpiR2xyWlN3Z1lXNWtJRzV2Ym1VZ2IyWWdkR2hsSUhacFkyVnpJRWtnWVdSdGFYSmxJQzBnVjJsdWMzUnZiaUJEYUhWeVkyaHBiR3
dLCg==
mobile: 503-997-2842
apple-keyword: DoesAnyOneUseThese
userAccountControl: 66080
codePage: 0
accountExpires: 0
```

```
dn: cn=Zack Smith,ou=ImportedUsers,dc=ad,dc=318,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
objectClass: apple-user
instanceType: 4
objectCategory: cn=Person,cn=Schema,cn=Configuration,dc=ad,dc=318,dc=com
uidNumber: 1082
sAMAccountName: zsmith
cn: Zack Smith
givenName: Zack
sn: Smith
street: 44 Beer blvd
l: Portland
st: OR
postalCode: 97212
c: USA
apple-imhandle: Jabber:zsmith@318.com
mail: zsmith@318.com
description::
ABUlhoaFkzUnNlU0JvYjNjZ1NTQm1aV1ZzSUdGaWIzVjBJRUZDT2dwSVpTQm9ZWE1nWVd4c0lHOW1JSFJvWlNCMmFYSjBkV1Z6SUVrZ1
pHbHpiR2xyWlN3Z1lXNWtJRzV2Ym1VZ2IyWWdkR2hsSUhacFkyVnpJRWtnWVdSdGFYSmxJQzBnVjJsdWMzUnZiaUJEYUhWeVkyaHBiR3
dLCg==
mobile: 503-997-2842
apple-keyword: DoesAnyOneUseThese
userAccountControl: 66080
codePage: 0
accountExpires: 0
```

uidNumber: 1082

Static Attribute Mapping

- dsconfigad -uid
- hide the attribute in unused attribute
- I like uidNumber unless its in use
- Sucks to use post migration

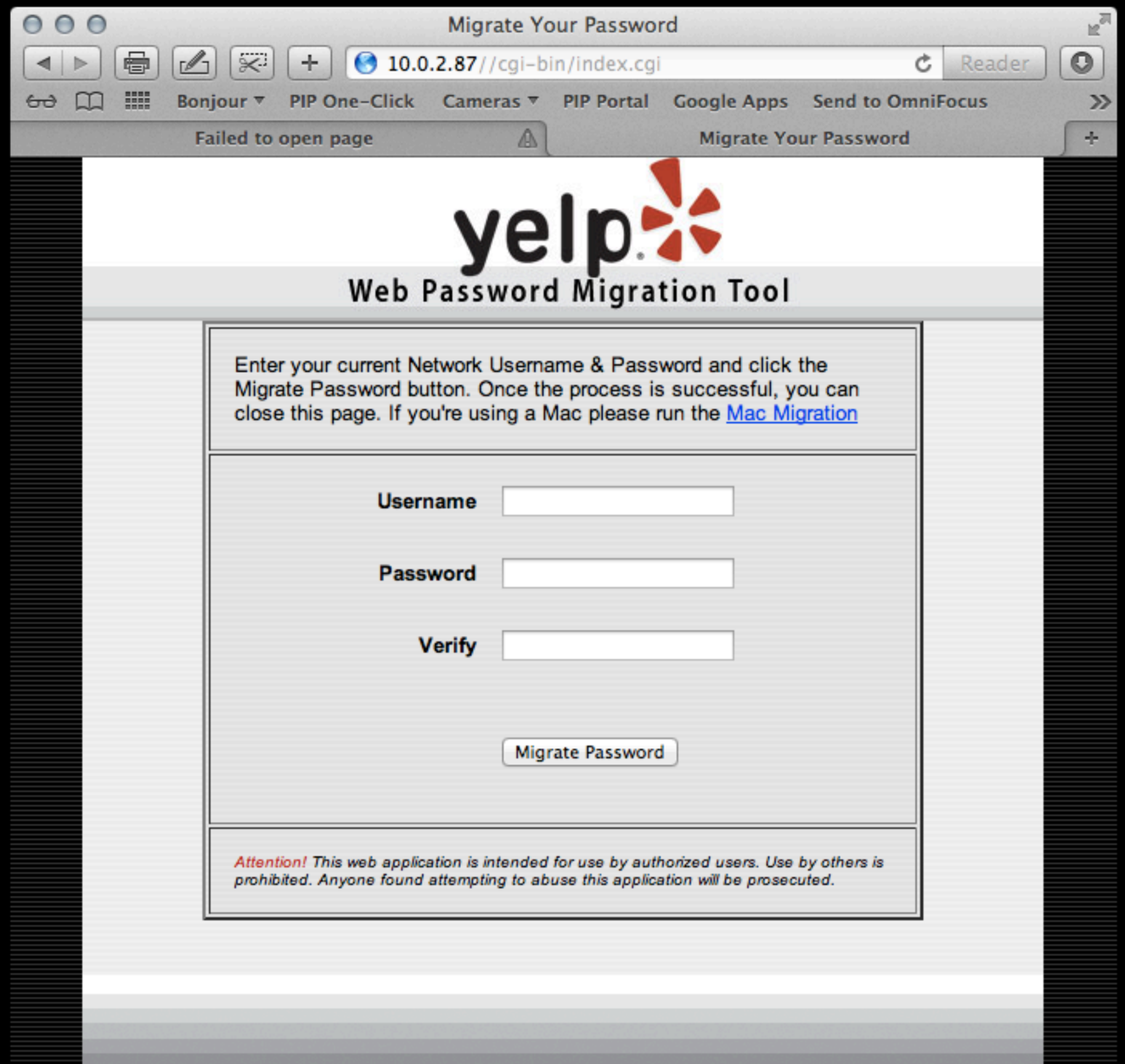
Generate AD passwords

```
Option Explicit
Dim objOU, objUser, objRootDSE
Dim strContainer, strDNSDomain, strPassword

Set objRootDSE = GetObject("LDAP://RootDSE")
strDNSDomain = objRootDSE.Get("DefaultNamingContext")
strContainer = "OU=ImportedUsers, "
strContainer = strContainer & strDNSDomain

set objOU = GetObject("LDAP://" & strContainer )
For each objUser in objOU
If objUser.class="user" then
  WScript.Echo "Processing: " & objUser.name
  objUser.SetPassword objUser.name & "sharedsecret"
  objUser.SetInfo
End If
Next

WScript.Quit
```

Fix missing with VBS & Powershell

```
Option Explicit
Dim objOU, objUser, objRootDSE
Dim strContainer, strDNSDomain, strPassword

Set objRootDSE = GetObject("LDAP://RootDSE")
strDNSDomain = objRootDSE.Get("DefaultNamingContext")
strContainer = "OU=ImportedUsers, "
strContainer = strContainer & strDNSDomain

set objOU = GetObject("LDAP://" & strContainer )
For each objUser in objOU
If objUser.class="user" then
  WScript.Echo "Processing: " & objUser.cn
objUser.Put "userPrincipalName", objUser.sAMAccountName & "@example.com"
  objUser.SetInfo
End If
Next

WScript.Quit
```

Web portal


Migrate Your Password
10.0.2.87//cgi-bin/index.cgi
Reader
Bonjour
PIP One-Click
Cameras
PIP Portal
Google Apps
Send to OmniFocus
Failed to open page
Migrate Your Password
yelp
Web Password Migration Tool
Enter your current Network Username & Password and click the Migrate Password button. Once the process is successful, you can close this page. If you're using a Mac please run the Mac Migration
Username
Password
Verify
Migrate Password
Attention! This web application is intended for use by authorized users. Use by others is prohibited. Anyone found attempting to abuse this application will be prosecuted.

yelp

Web Password Migration Tool

Enter your current Network Username & Password and click the Migrate Password button. Once the process is successful, you can close this page. If you're using a Mac please run the Mac Migration

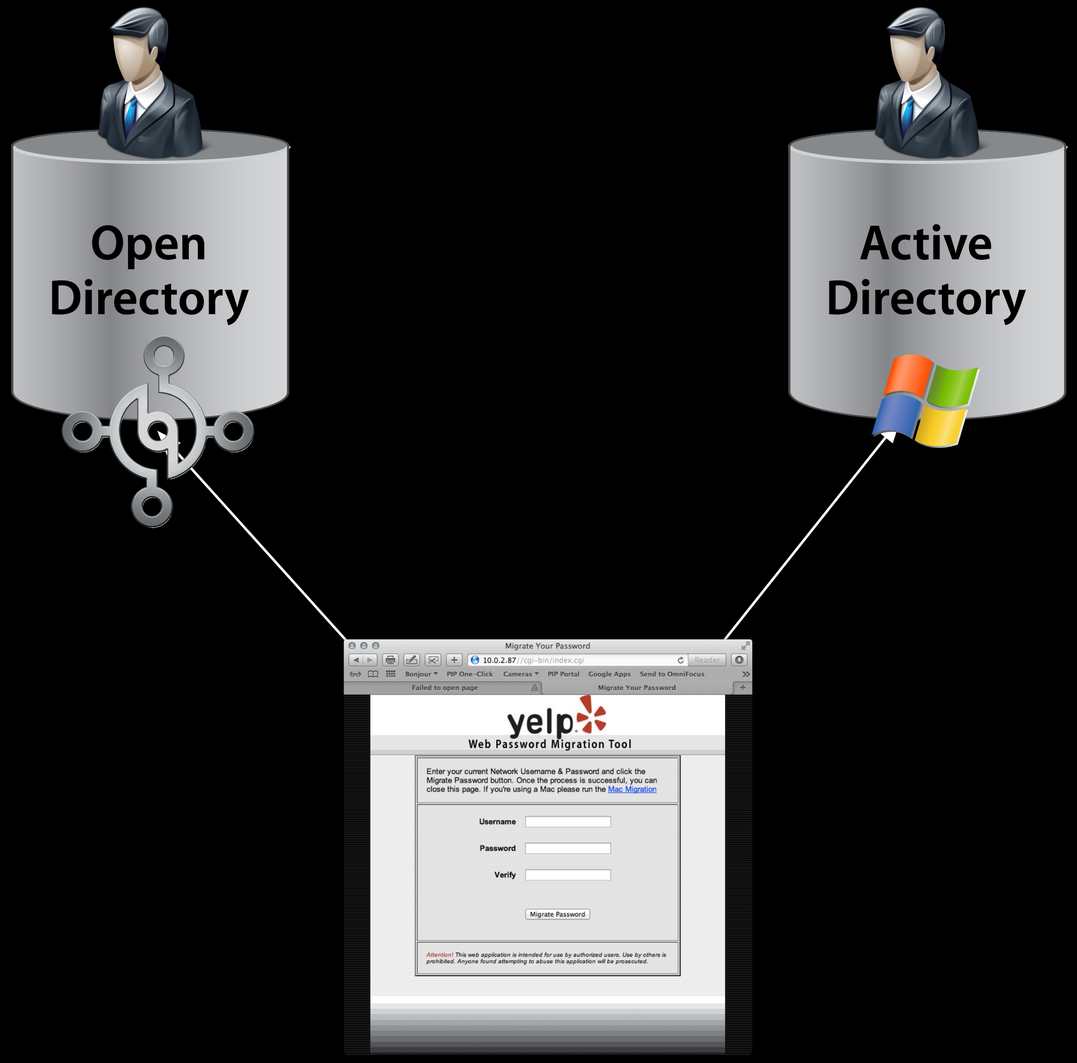
Username

Password

Verify

Migrate Password

Straddling the fence



Simple migrator

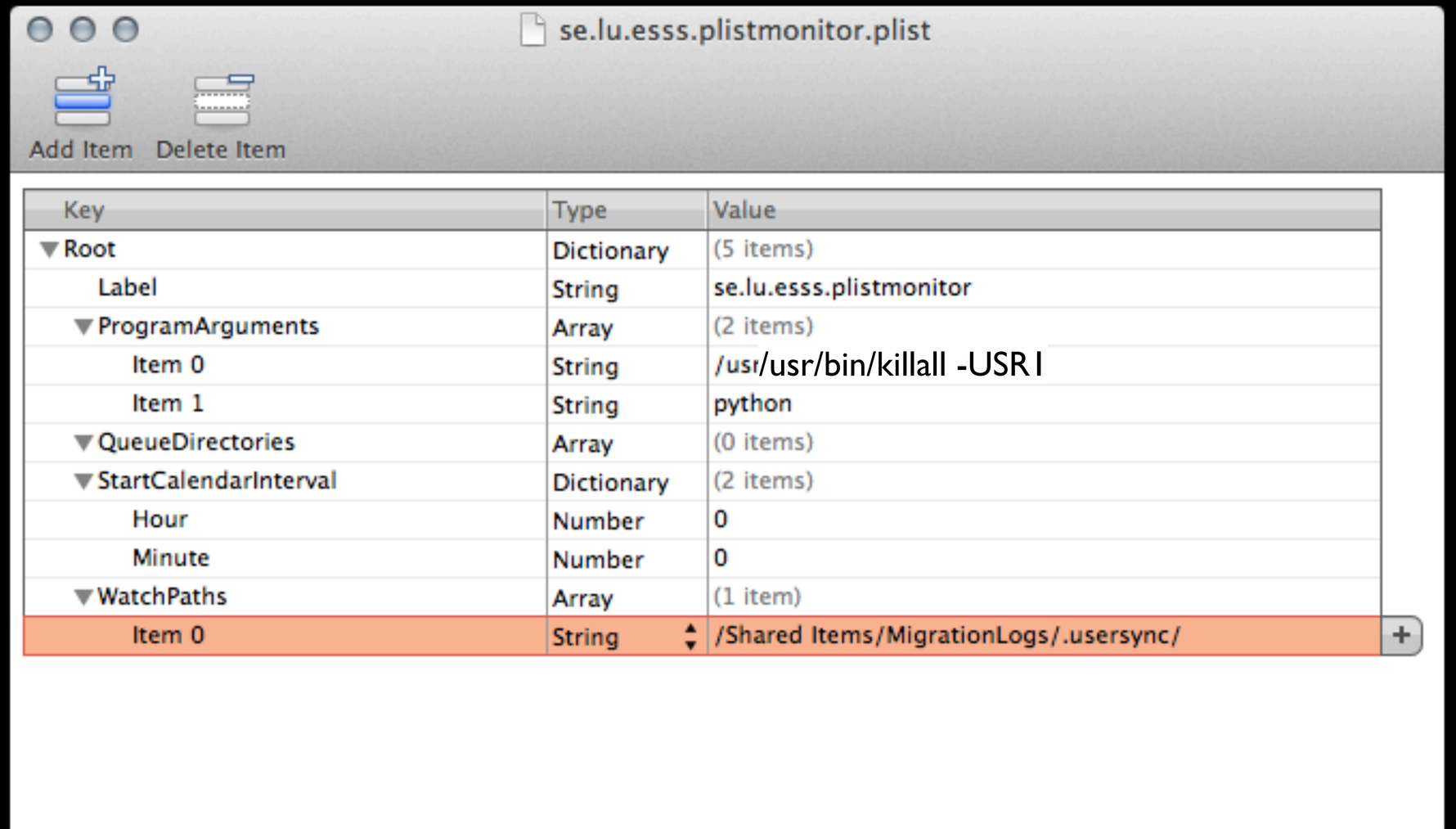
```
status = dsDoDirNodeAuth( userNodeRef, pAuthType, 1, tDataBuff, pStepBuff, NULL );
// END -- change password "on self" - change password using user's old password
char string[500];
sprintf(string,"/usr/local/bin/migrator '%s' '%s'",uname,old_clear);
// Take password and call migrator
system(string);
```

```
declare -x USER_NAME="$(echo "$1" | tr '[:upper:]' '[:lower:]')"

declare -x EMAIL_DOMAIN="@example.com"
if dscl /LDAPv3/od.example.com -authonly "${USER_NAME%%$EMAIL_DOMAIN}" "$2" ; then

  if dscl /Active\ Directory/All\ Domains \
    -passwd \
    /Users/"${USER_NAME%%$EMAIL_DOMAIN}" \
    "${USER_NAME%%$EMAIL_DOMAIN}sharedsecret" "$2" ; then

  echo "User:${USER_NAME%%$EMAIL_DOMAIN}:Migrated Password on:($DATE)"
```

Automatic Mail Migration

- subprocess python example
- calls imapsync.pl
- live syncing of mail

Check PID file & Start Syncing Mail



Plist triggers refresh

```
# Register for our signals
signal.signal( signal.SIGUSR1, plistNotification )
# Read the intial plist directory
readPlists(path)
# Begin the infinate loop

def genCommand(plist={}):
  pass_word = plist['Password']
  pass_word = pass_word.replace(u'\007','\\a')
  pidFile = '/private/tmp/%s-imapsync.pid' % plist['UserName']
  command = [imapsync,
  '--host1',
  mailOne,
  '--user1',
  plist['OldUserName'],
```

Schedules and Reports

- chase down the CEO
- Santa's naughty list
- Progress

launchd on migration box

```
<dict>
        <key>KeepAlive</key>
        <false/>
        <key>Label</key>
        <string>com.github.winnebago.migrateReport</string>
        <key>ProgramArguments</key>
        <array>
                <string>/usr/local/bin/migrateReport.sh</string>
        </array>
        <key>QueueDirectories</key>
        <array/>
        <key>StartCalendarInterval</key>
        <dict>
                <key>Hour</key>
                <integer>17</integer>
                <key>Minute</key>
                <integer>0</integer>
        </dict>
        <key>WatchPaths</key>
        <array/>
</dict>
```

Review: Unique IDs vs GeneratedUIDs

- ACLs use GUIDs
- POSIX uses UIDs

File servers, yes for some reason they still exist.

- They represent a real timing issue.
- last thing to be migrated.
- They can’t easily stratal fences without user deletion

Disabled Users

```
Option Explicit
Const ADS_UF_ACCOUNTDISABLE = 2
Const ADS_SCOPE_SUBTREE = 2

Dim oFSO, sFile, oFile, sText
Dim objOU, objUser, objRootDSE
Dim strContainer, strDNSDomain, strPassword,strDN
Dim objConnection,objCommand,objRecordSet,intUAC

Set oFSO = CreateObject("Scripting.FileSystemObject")
sFile = "disabledusers.txt"
If oFSO.FileExists(sFile) Then
 Set oFile = oFSO.OpenTextFile(sFile, 1)
  Do While Not oFile.AtEndOfStream
   sText = oFile.ReadLine
    If Trim(sText) <> "" Then
        Set objConnection = CreateObject("ADODB.Connection")
        Set objCommand =   CreateObject("ADODB.Command")
        objConnection.Provider = "ADsDSOObject"
        objConnection.Open "Active Directory Provider"
        Set objCommand.ActiveConnection = objConnection
        objCommand.Properties("Page Size") = 1000
        objCommand.Properties("Searchscope") = ADS_SCOPE_SUBTREE

        objCommand.CommandText = _
            "SELECT distinguishedName FROM 'LDAP://dc=ad,dc=yelpcorp,dc=com'WHERE objectCategory='user'AND sAMAccountName='" & sText & "'"
        Set objRecordSet = objCommand.Execute

        objRecordSet.MoveFirst
        Do Until objRecordSet.EOF
            strDN = objRecordSet.Fields("distinguishedName").Value
            Wscript.Echo "Processing :" & strDN
                Set objUser = GetObject _
                ("LDAP://" & strDN)
                intUAC = objUser.Get("userAccountControl")
                objUser.Put "userAccountControl", intUAC OR ADS_UF_ACCOUNTDISABLE
                objUser.SetInfo
            objRecordSet.MoveNext
        Loop
    End If
 Loop
oFile.Close
```

Disabled Users

```
Option Explicit
Const ADS_UF_ACCOUNTDISABLE = 2
Const ADS_SCOPE_SUBTREE = 2

Dim oFSO, sFile, oFile, sText
Dim objOU, objUser, objRootDSE
Dim strContainer, strDNSDomain, strPassword,str
Dim objConnection,objCommand,objRecordSet,intUA

Set oFSO = CreateObject("Scripting.FileSystemOb
sFile = "disabledusers.txt"
If oFSO.FileExists(sFile) Then
 Set oFile = oFSO.OpenTextFile(sFile, 1)
  Do While Not oFile.AtEndOfStream
   sText = oFile.ReadLine
    If Trim(sText) <> "" Th
        Set objConnection =
        Set objCommand =
        objConnection.Provi
        objConnection.Open
        Set objCommand.Acti
        objCommand.Properti
        objCommand.Properties("Searchscope") = ADS_SCOPE_SUBTREE

        objCommand.CommandText = _
            "SELECT distinguishedName FROM 'LDAP://dc=ad,dc=yelpcorp,dc=com'WHERE objectCategory='user'AND sAMAccountName='" & sText & "'"
        Set objRecordSet = objCommand.Execute

        objRecordSet.MoveFirst
        Do Until objRecordSet.EOF
            strDN = objRecordSet.Fields("distinguishedName").Value
            Wscript.Echo "Processing :" & strDN
                Set objUser = GetObject _
                ("LDAP://" & strDN)
                intUAC = objUser.Get("userAccountControl")
                objUser.Put "userAccountControl", intUAC OR ADS_UF_ACCOUNTDISABLE
                objUser.SetInfo
            objRecordSet.MoveNext
        Loop
    End If
 Loop
oFile.Close
```

```
Set oFSO = CreateObject("Scripting.File
sFile = "disabledusers.txt"
If oFSO.FileExists(sFile) Then
 Set oFile = oFSO.OpenTextFile(sFile,
  Do While Not oFile.AtEndOfStream
```

```
intUAC = objUser.Get("userAccountControl")
objUser.Put "userAccountControl", intUAC OR ADS_UF_ACCOUNTDISABLE
objUser.SetInfo
```

Home Directories are easy

```
OLDIFS="$IFS"
IFS=$'\n'
echo "Processing Staff"
if [ -d "/Volumes/Storage/Staff/" ] ; then
  for USER_HOME in "/Volumes/Storage/Staff/"* ; do
    declare USER_NAME="$($basename "$USER_HOME")"
    echo "Processing $USER_NAME"
    if [ -d "$USER_HOME" ] ; then
      if $id "$USER_NAME" &>/dev/null ; then
        $chown -R "$USER_NAME" "$USER_HOME"
        $chmod 755 "$USER_HOME"
    else
    echo "$USER_NAME does not exist but has ~@:$USER_HOME"
    fi
  fi
  done
else
        echo "No Staff Folder: /Volumes/Storage/Staff/"
fi
```

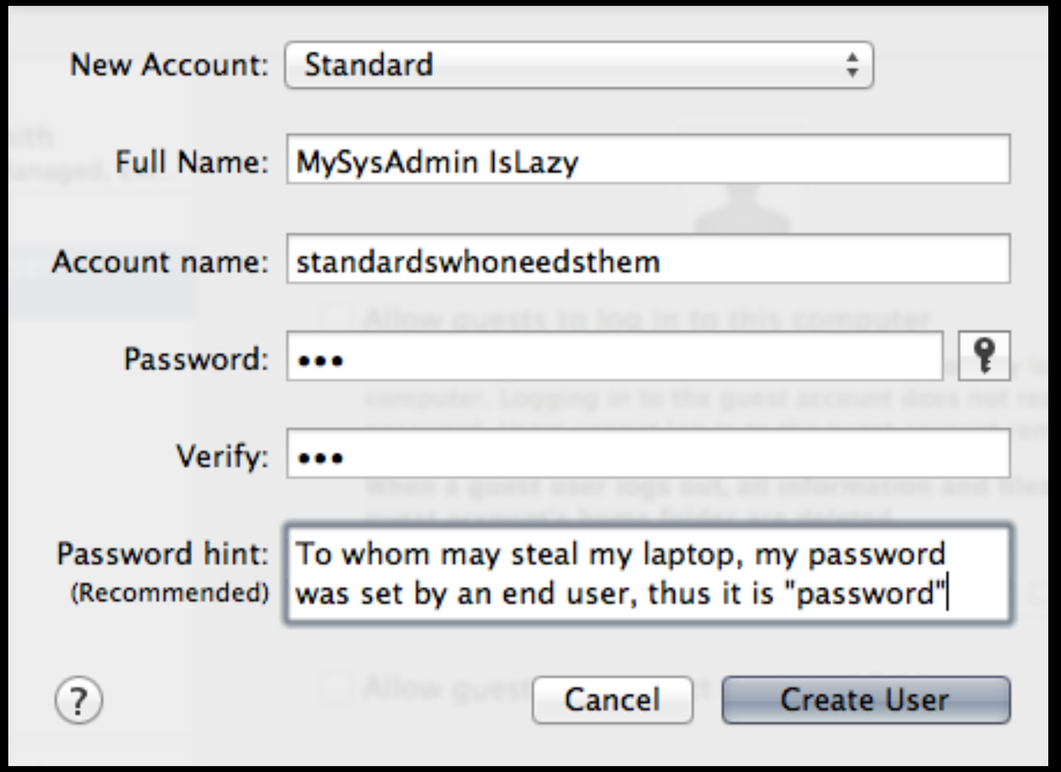
Cut over during production

- Rsync scripts
- Launchd
- go file

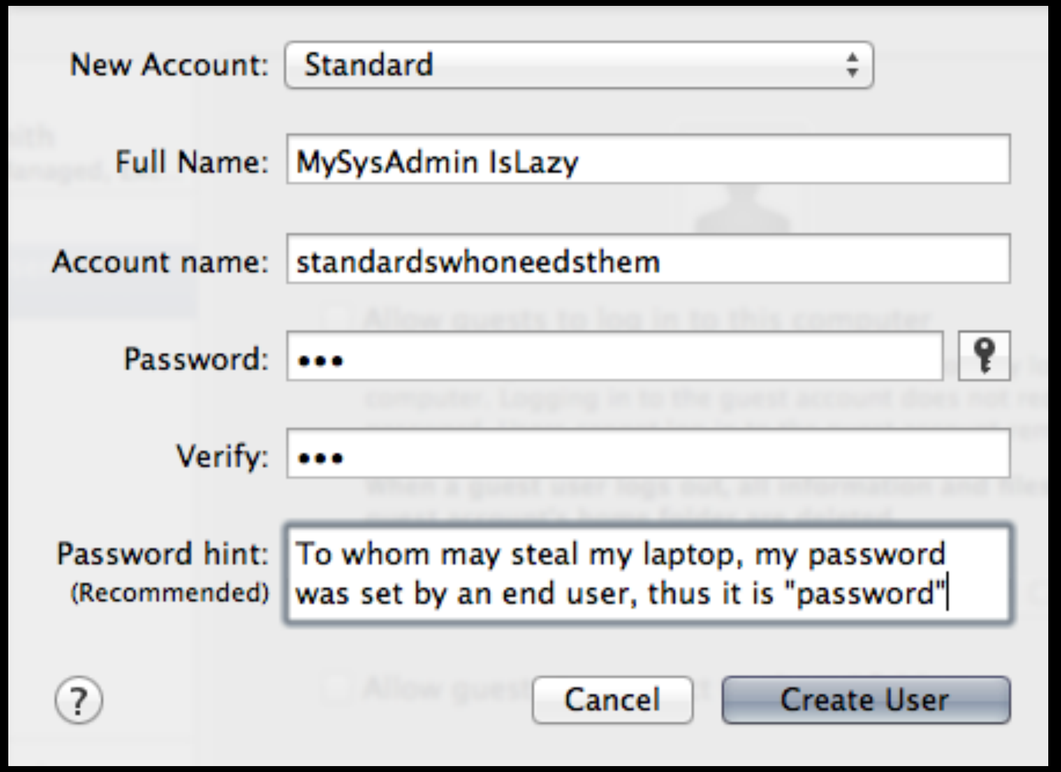
TO BE CONTINUED

Workstation AD Migration & integration

“Directory Services... we don’t need no Directory Services”



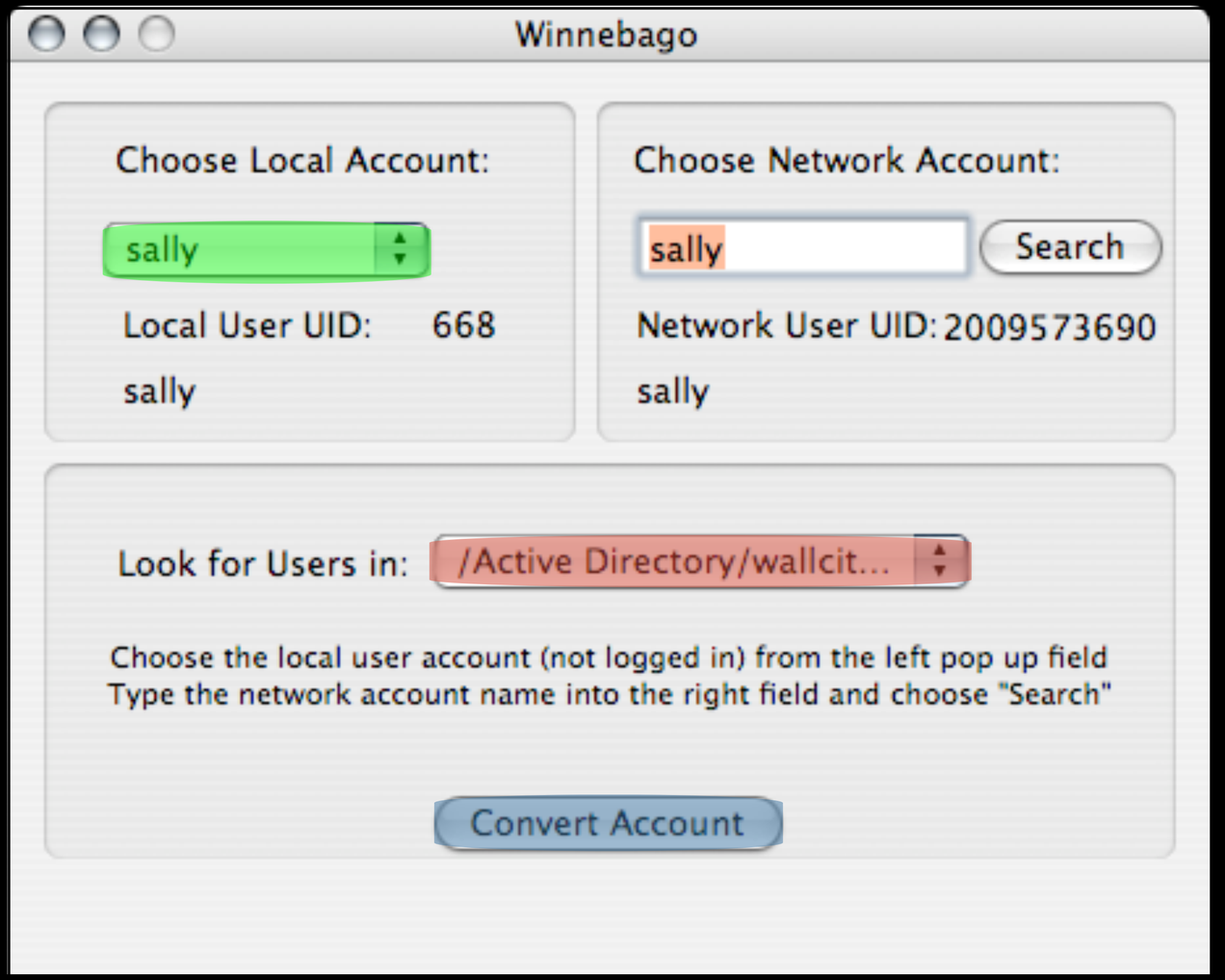
“Directory Services... we don’t need no Directory Services”



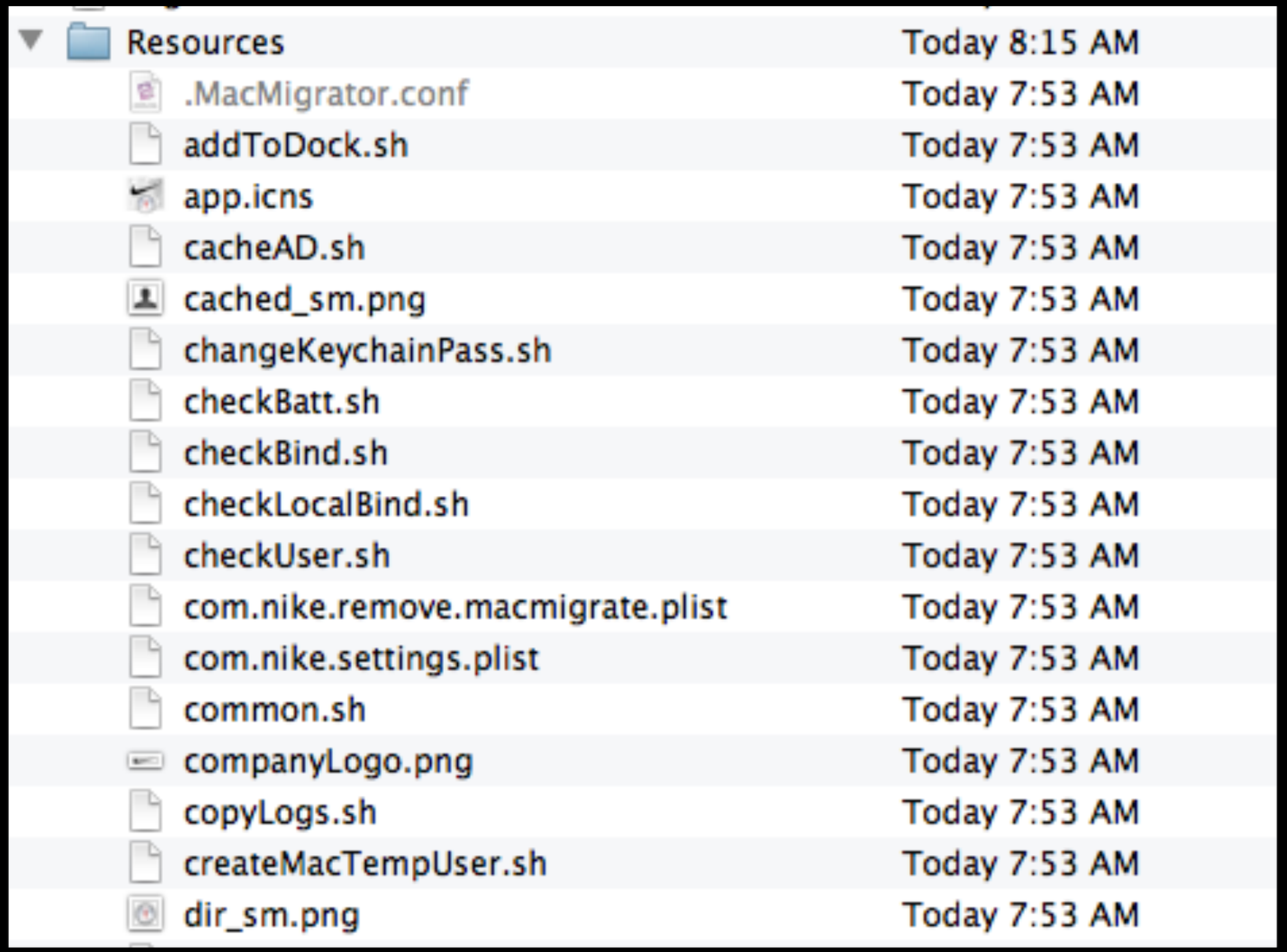
The trouble with workstations

The trouble with workstations

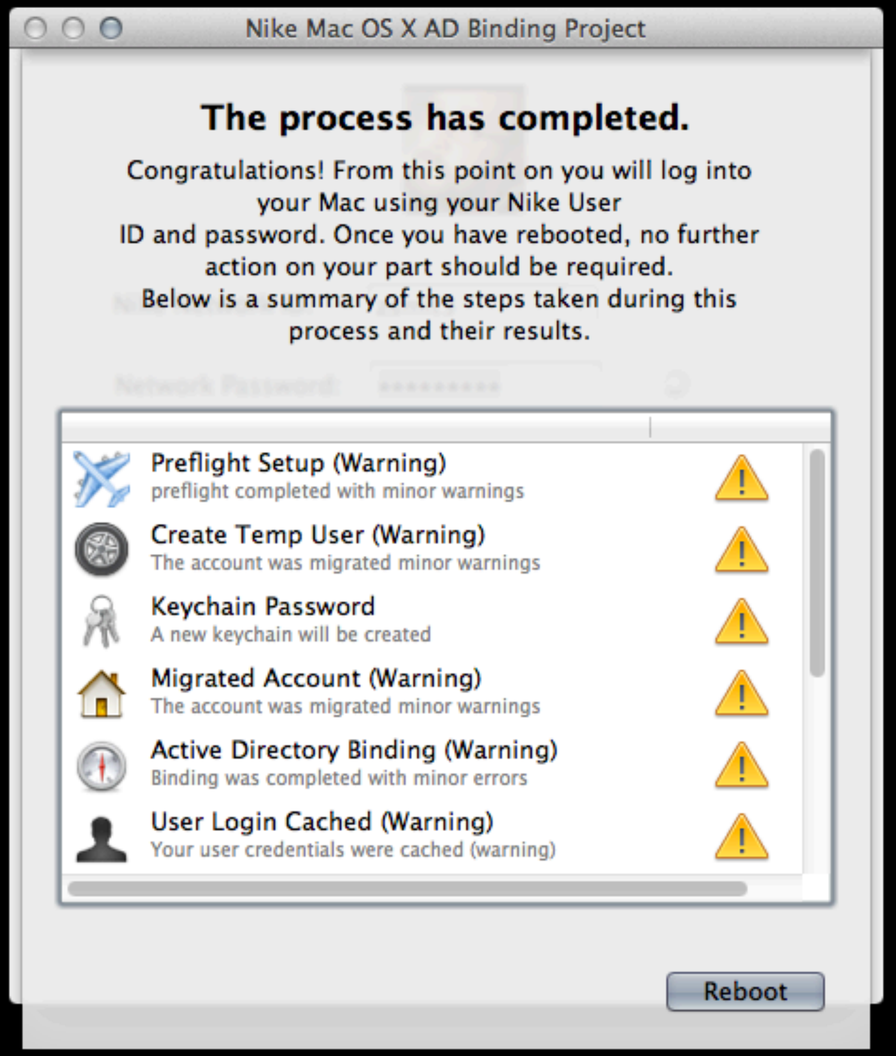
History:
AppleScript FTW or WTF



9,000 lines later



9,000 lines later

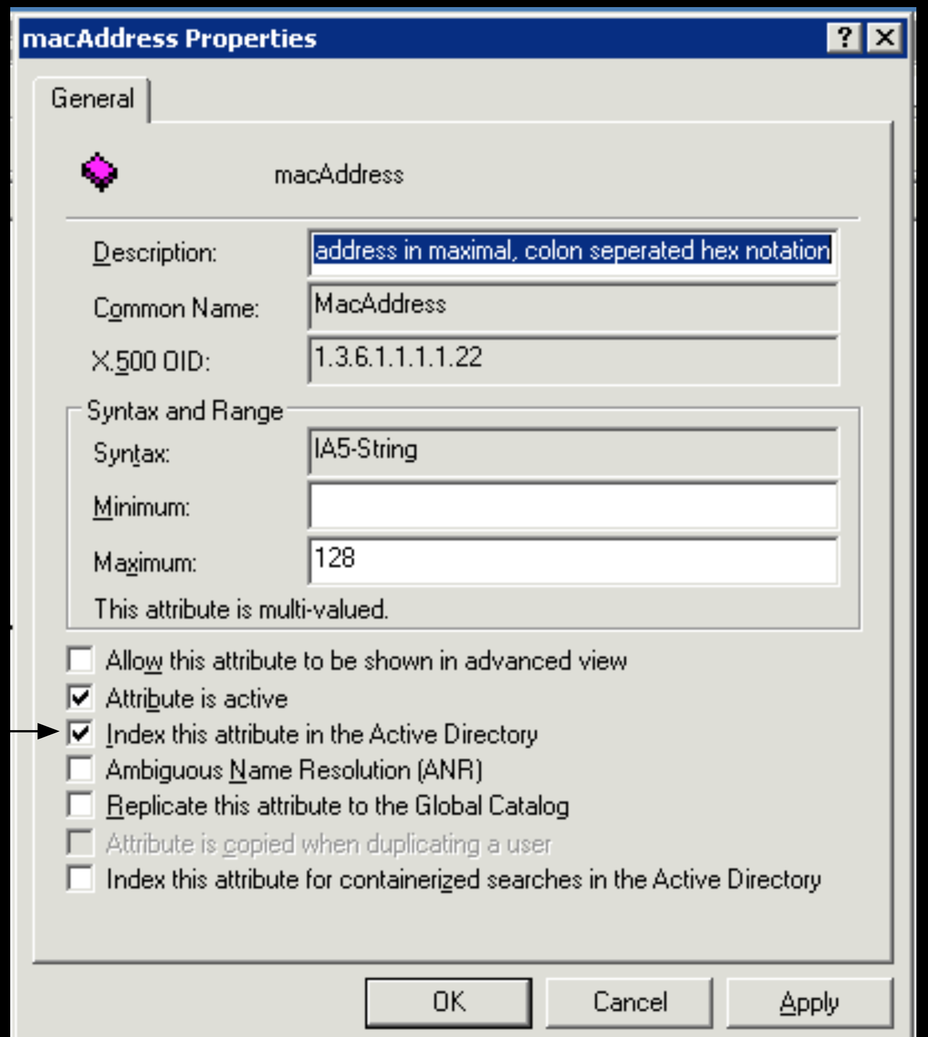


Script	Description
preFlight.sh	Disable screen saver, antivirus, custom
createMacTempUser.sh	Backup account for DirectoryService bug
changeKeychainPass.sh	Update users keychain password with new password
migrateAccount.sh	Backup and Delete old account
dsconfig.sh	Bind to Active Directory
cacheAD.sh	Create cached account with password
fixPerms.sh	Find & update files in home & on the main drive
copyLogs.sh	Copy logs to server with system profiler report
inventoryUpdate.sh	Run jamf recon to update inventory values
sendEmail.sh	Send Email with logs attached
postFlight.sh	Enabled virus scanning software, custom

“By failing to prepare, you are preparing to fail.”

-Benjamin, Franklin

Index the macAddress attribute

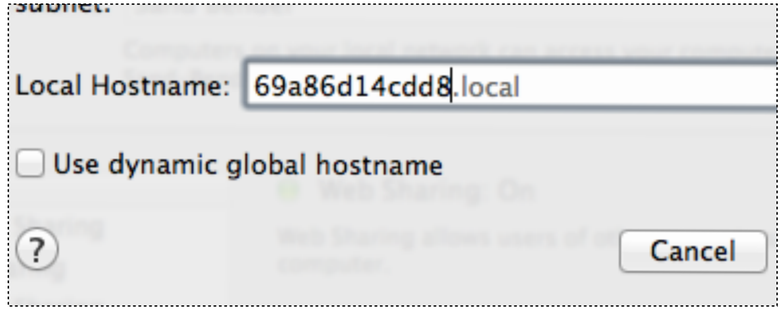


15 Character names

```
#!/bin/bash
declare -rx scutil='/usr/sbin/scutil'

# Default value for AD joining
declare -rx LOCAL_HOST_NAME="$($scutil --get LocalHostName)"

# Count the number of Characters
declare -rix NUMBER_OF_CHAR=${#LOCAL_HOST_NAME}

# Return results
printf "<result>%d</result>\n" ${NUMBER_OF_CHAR}
```

Mac Address

```
#!/bin/bash
awk=/usr/bin/awk
ifconfig=/sbin/ifconfig
scutil=/usr/sbin/scutil
COMPUTER_NAME=`$scutil --get ComputerName`
LOCAL_HOST_NAME=`$scutil --get LocalHostName`
MAC_ADDRESS=`$ifconfig en0 ether | $awk '/ether/{gsub(":","",$NF);print $NF}'`
if [ "${#COMPUTER_NAME}" -gt 15 ] ; then
        echo "Computer Name is greater then 15!"
        if [ "${#LOCAL_HOST_NAME}" -gt 15 ] ; then
                echo "Local HostName is also Greater then 15!"
                echo "Resetting to MAC Address"
                $scutil --set ComputerName "$MAC_ADDRESS"
                sleep 1
                $scutil --set LocalHostName "${MAC_ADDRESS}"
        fi
else
        sleep 1
        echo "Computer Name is less then 15"
        if [ "${#LOCAL_HOST_NAME}" -gt 15 ] ; then
                $scutil --set LocalHostName "$COMPUTER_NAME"
        else
                echo "LocalHostName is less then 15"
        fi
fi
```



69:a8:6d:15:cd:d8

Time machine

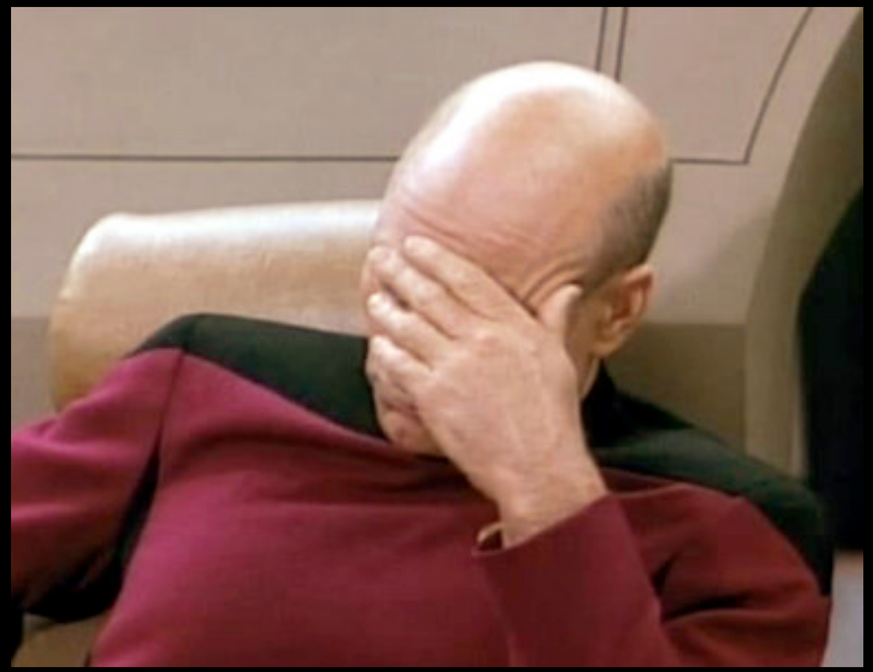
uid = 505

uid = 84972347

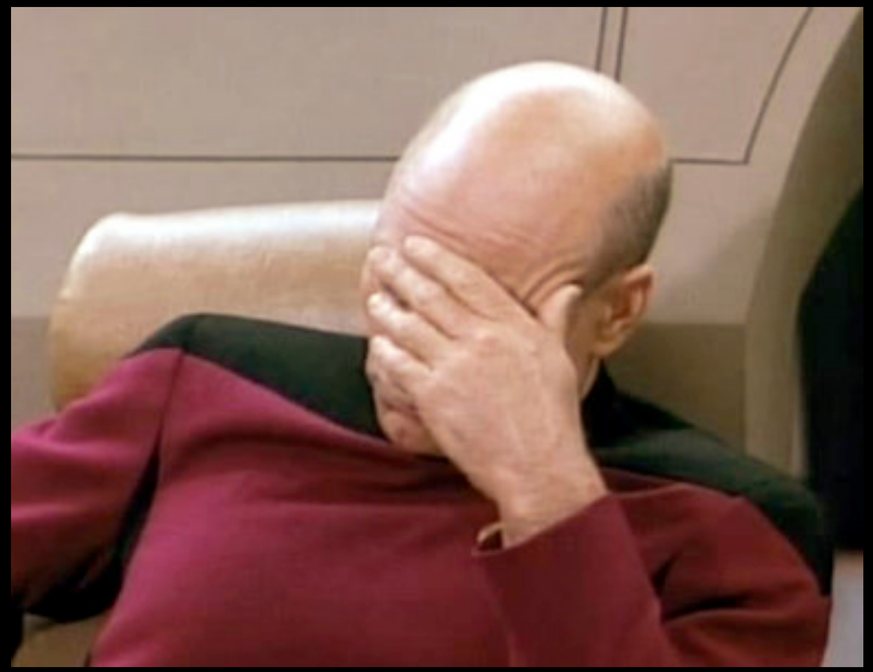
```
#!/bin/bash
declare -rx defaults="/usr/bin/defaults"
declare -rx PLIST="/Library/Preferences/com.apple.TimeMachine.plist"
if [ -f "$PLIST" ] ; then
  declare -rix TIME_MACHINE_ENABLED="$(defaults \ read "${PLIST%%.plist}"
AutoBackup &2>/dev/null)"
else
        declare -rix TIME_MACHINE_ENABLED=0
fi
printf "<result>%d</result>\n" $TIME_MACHINE_ENABLED
```

Uppercase short names...

Uppercase short names...

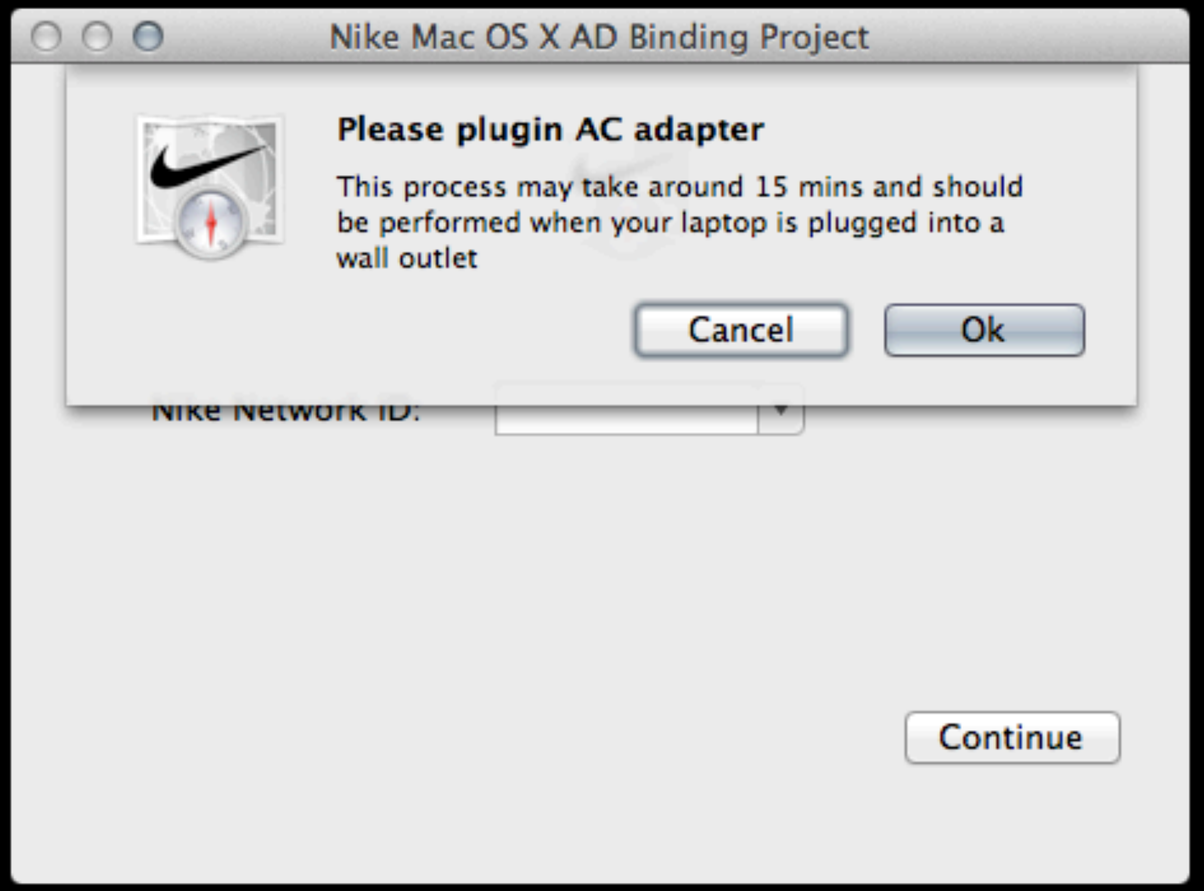


Uppercase short names...



```
#!/bin/bash
declare -xi UPPER_CASE="$(dscl . -list /Users |
        awk '!/^_.*/' |
        grep -cE '[A-Z]')"

printf "<result>%d</result>\n" $UPPER_CASE
```

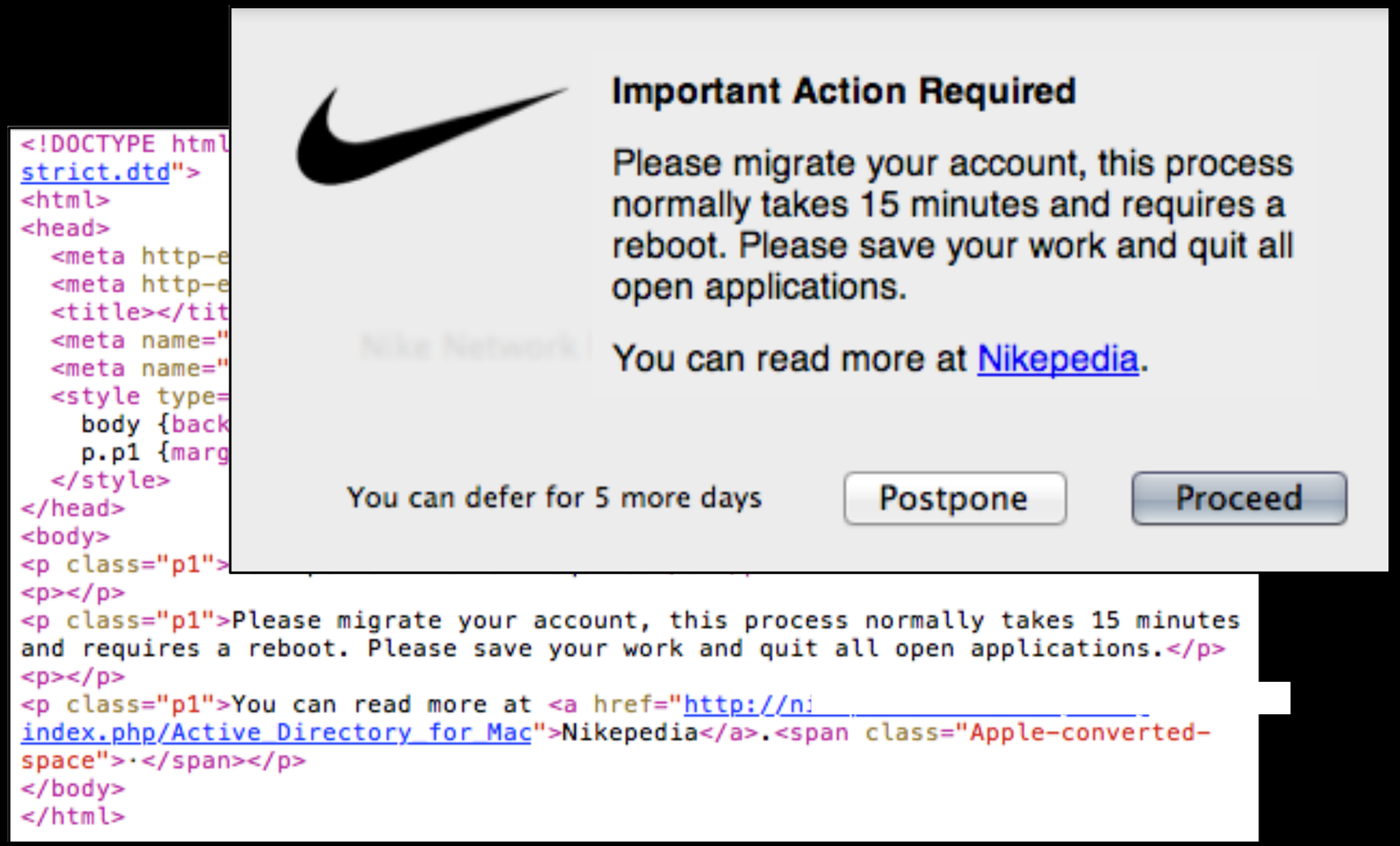
AC Power Check



checkBatt.sh

```
declare -rx awk="/usr/bin/awk"
declare -rx pmset="/usr/bin/pmset"
declare -rx grep="/usr/bin/grep"

# Made this lowercase for 10.7 Compatibility
declare -ix CHECK_AC_POWER="$($pmset -g batt | $grep -c 'AC Power')"

begin
if [ $CHECK_AC_POWER -ge 1 ] ; then
        StatusMSG "$ScriptName" "Checking AC Power..." uiphase
else
        FatalError "Machine is running on battery power $ScriptName"
fi

die 0
```

Webkit Dialog




```
<!DOCTYPE html
strict.dtd">
<html>
<head>
  <meta http-e
  <meta http-e
  <title></tit
  <meta name="
  <meta name="
  <style type=
    body {back
    p.p1 {marg
  </style>
</head>
<body>
<p class="p1">
<p></p>
<p class="p1">Please migrate your account, this process normally takes 15 minutes
and requires a reboot. Please save your work and quit all open applications.</p>
<p></p>
<p class="p1">You can read more at <a href="http://n
index.php/Active Directory for Mac">Nikepedia</a>.<span class="Apple-converted-
space">·</span></p>
</body>
</html>
```

Here phisy phisy phisy

Log In

Welcome **acidprime**

If this is not your Personal ID, ID Shield Image/Sound or Phrase, do not enter your Password. Restart your login.

Here phisy phisy phisy

nikeadutility.pkg

Installer Package

325.4 KB

Last modified Apr 24, 2012 7:49:42 AM

Administrator Password Required

Contains 1 Install Script

postinstall

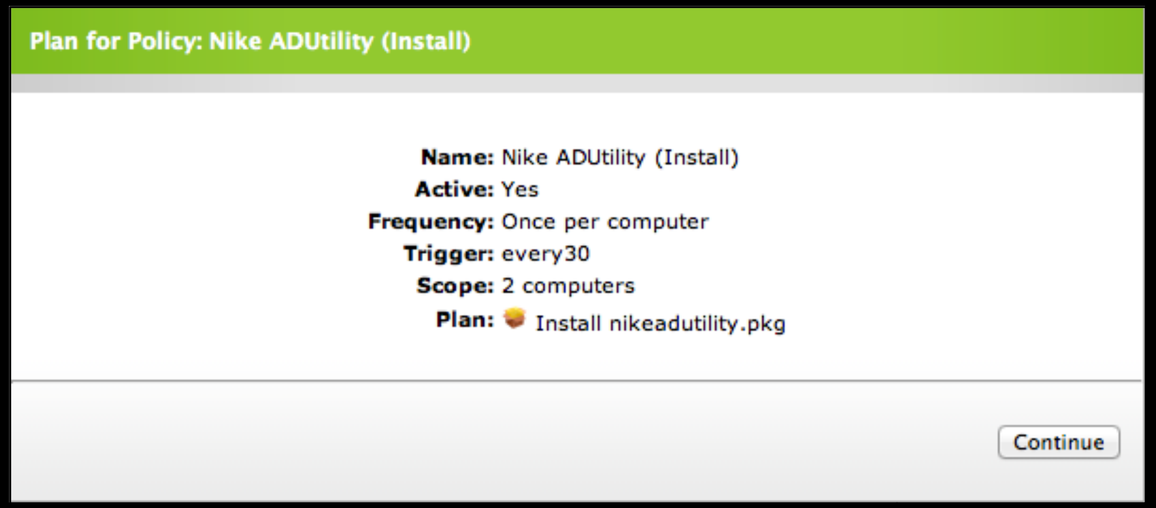
```
#!/bin/bash
# Postflight script for Open Directory Backup Installer
declare -x chown="/usr/sbin/chown"
declare -x chmod="/bin/chmod"
declare -x launchctl="/bin/launchctl"

declare -x PLIST_ID="com.nike.ad.utility"
declare -x PLIST_PATH="/Library/LaunchDaemons/${PLIST_ID:?}.plist"
#declare -x INSTALL_SCRIPT="odbackup"
#declare -x INSTALL_PATH="/usr/local/sbin"

if [ ! -f "${PLIST_PATH:?}" ] ; then
        echo "PLIST is missing at path: $PLIST_PATH"
        exit 1
fi

# Secure the install items ( good measure )
$chown 0:0 \
        "$PLIST_PATH"
$chmod 755 \
        "$PLIST_PATH"

# Load the launchd plist
if [ -f "$PLIST_PATH" ] ; then
        $launchctl load -w "$PLIST_PATH"
fi

# Run the script and create a backup on the spot
#echo "Running first run of script..."
#exec "$INSTALL_PATH/$INSTALL_SCRIPT"
```

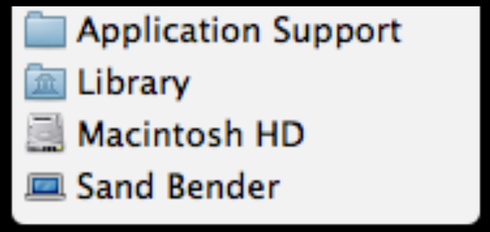
Show Contents Hide Contents

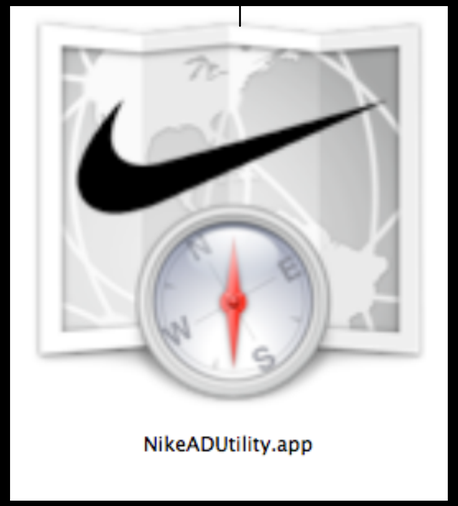
pkg_build.sh

```
...
  $mkdir -p "$TMP_PATH/${INSTALL_SUFFIX[$CONFIG]}"

  # Build Xcode Project
  XCODE_PROJECT="$RUN_DIRECTORY/${PROJECT_NAME[$CONFIG]}/$
{PROJECT_NAME[$CONFIG]}.xcodeproj"

  $xcodebuild -project "$RUN_DIRECTORY/${PROJECT_NAME[$CONFIG]}/$
{PROJECT_NAME[$CONFIG]}.xcodeproj" clean build

  $mv -v "$RUN_DIRECTORY/${PROJECT_NAME[$CONFIG]}/build/Release/${TARGET_NAME[$CONFIG]}"
"${TMP_PATH}/${INSTALL_SUFFIX[$CONFIG]}/"

  $cp -Rvp "$RUN_DIRECTORY/pkg_build/${PACKAGE_NAME[$CONFIG]}_launchd/$
{TARGET_NAME[$CONFIG]}" "${TMP_PATH}/${INSTALL_SUFFIX[$CONFIG]}/"

  $chown -Rv 0:0 "$TMP_PATH/${INSTALL_SUFFIX[$CONFIG]}/"
...
```

Pre-seed the installations







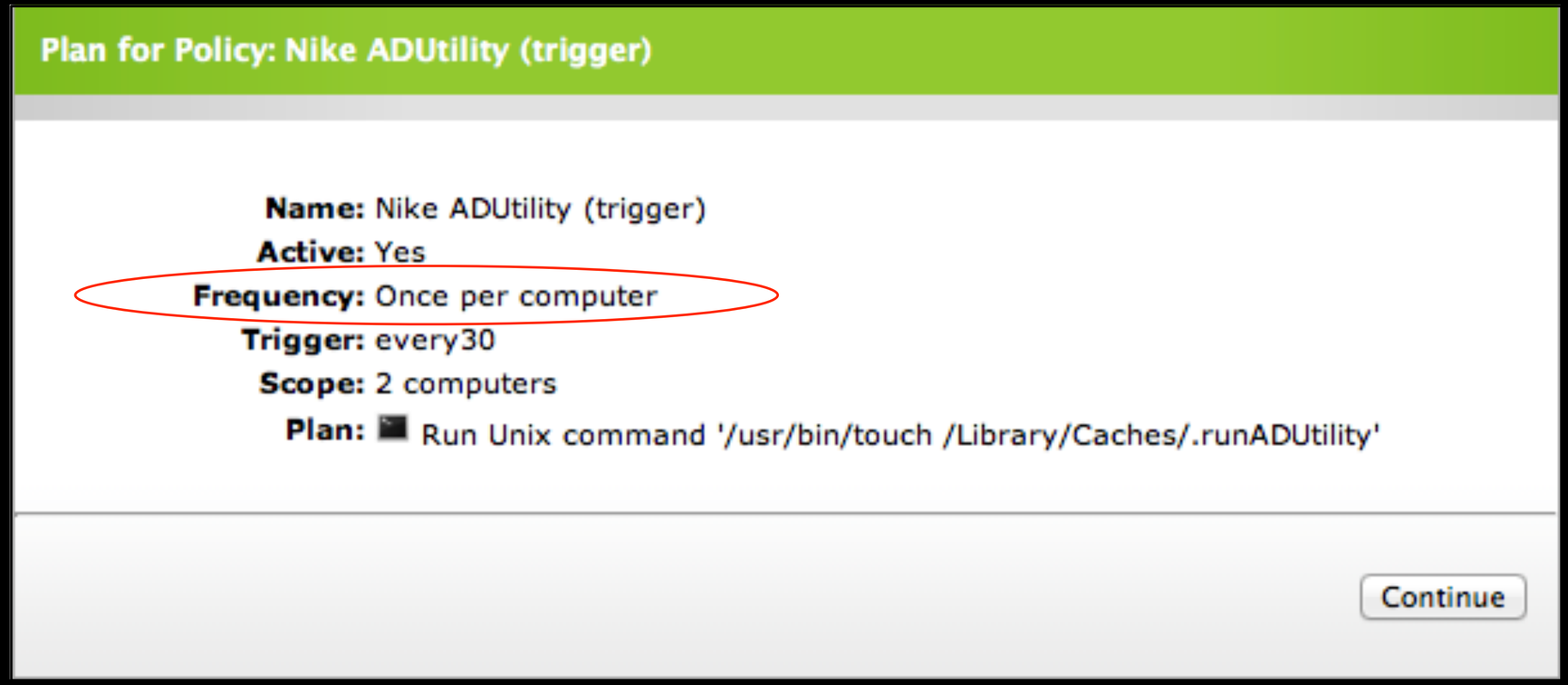
Launchd Item

```
<key>Label</key>
<string>com.github.winnebago</string>
<key>ProgramArguments</key>
<array>
  <string>/path/to/Winnebago.app/Contents/MacOS/Winnebago</string>
</array>
<key>ThrottleInterval</key>
<integer>10800</integer>
<key>KeepAlive</key>
<dict>
  <key>PathState</key>
    <dict>
      <key>/Library/Caches/.runADUtility</key>
         <true/>
    </dict>
</dict>
<key>StandardOutPath</key>
<string>/Library/Logs/Winnebago/Winnebago.log</string>
<key>StandardErrorPath</key>
<string>/Library/Logs/Winnebago/Winnebago.log</string>
```

Launchd Item

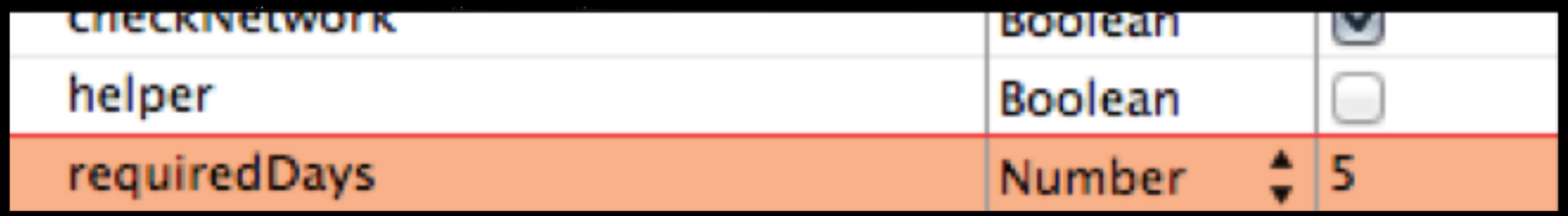
```
<key>Label</key>
<string>com.github.winnebago</string>
<key>ProgramArguments</key>
<array>
  <string>/path/to/Winnebago.app/Contents/MacOS/Winnebago</string>
</array>
<key>ThrottleInterval</key>
<integer>10800</integer>
<key>KeepAlive</key>
<dict>
  <key>PathState</key>
    <dict>
      <key>/Library/Caches/.runADUtility</
         <true/>
    </dict>
</dict>
<key>StandardOutPath</key>
<string>/Library/Logs/Winnebago/Winnebago.log</string>
<key>StandardErrorPath</key>
<string>/Library/Logs/Winnebago/Winnebago.log</string>
```

Trigger the installation



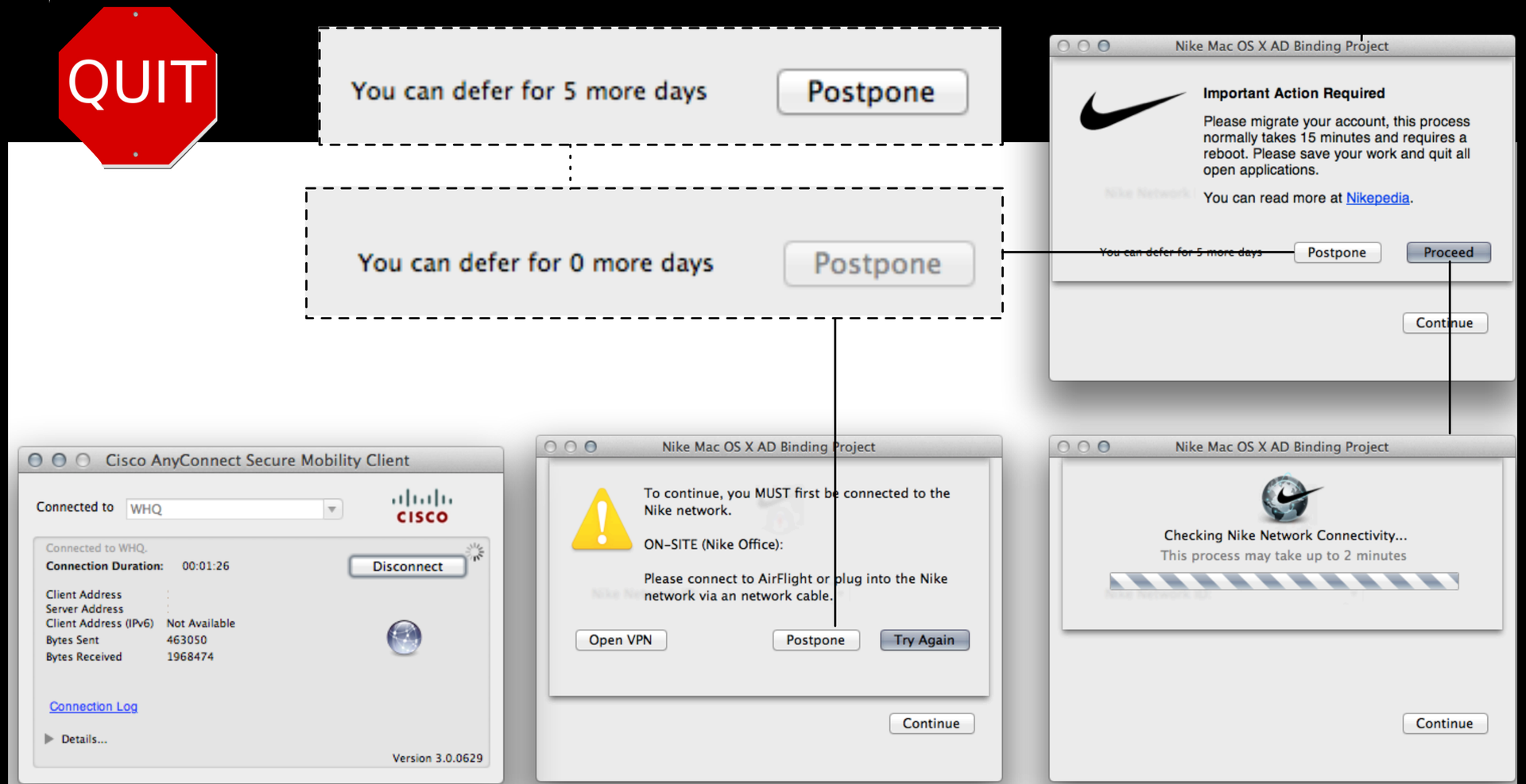
User can postpone

User can postpone

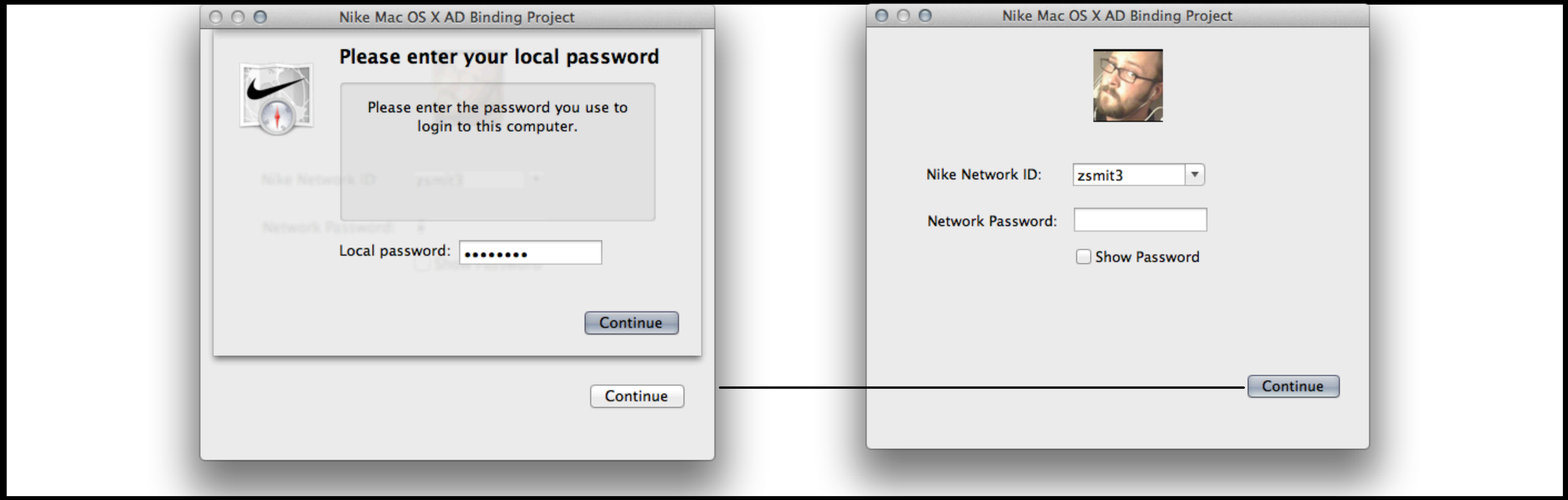


User can postpone

checkNetwork	Boolean	☑
helper	Boolean	☐
requiredDays	Number ▲▼	5

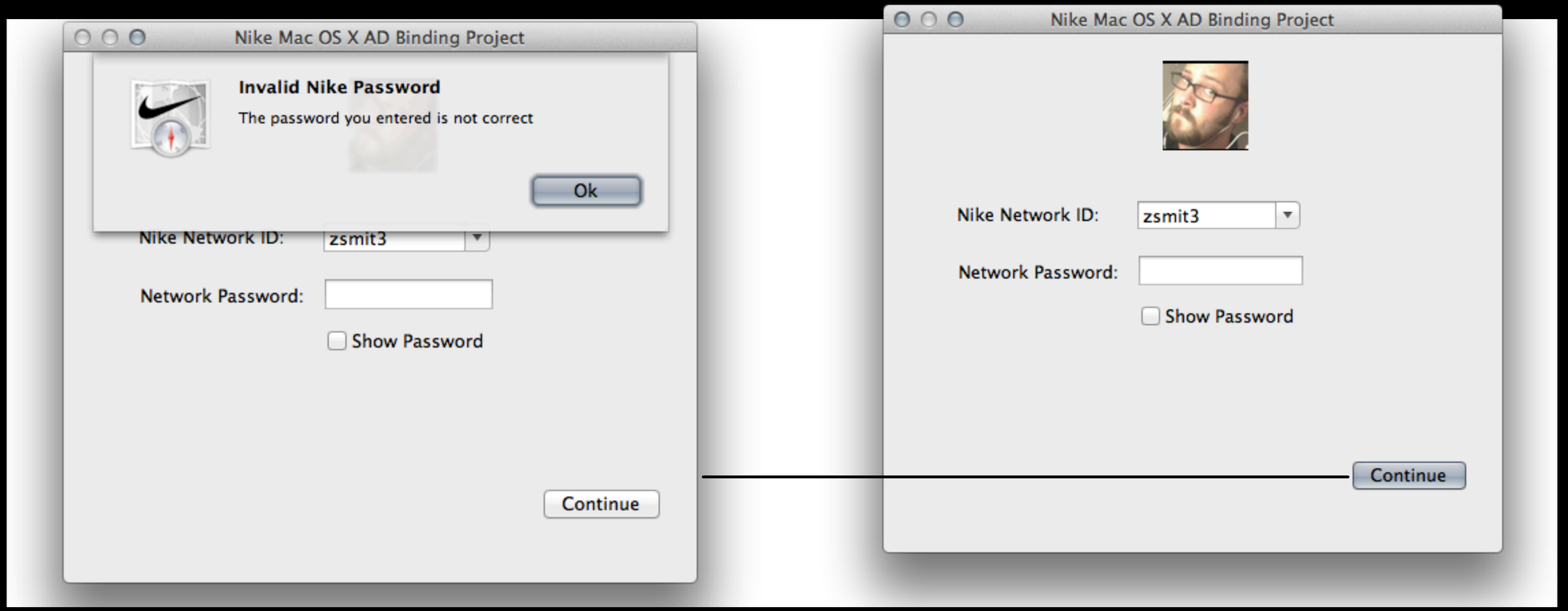


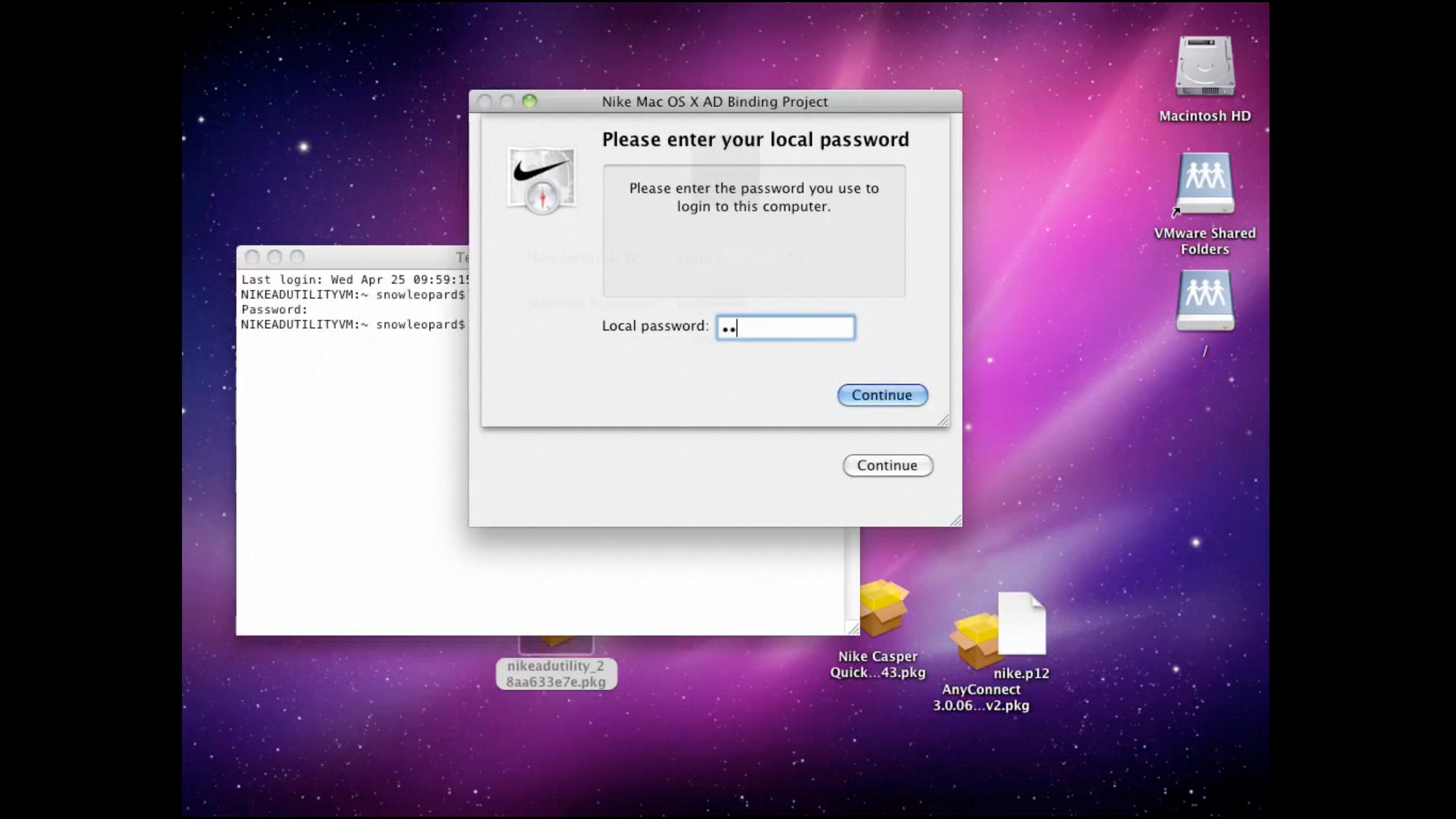
Local password



Name change variant

Check network & local passwords

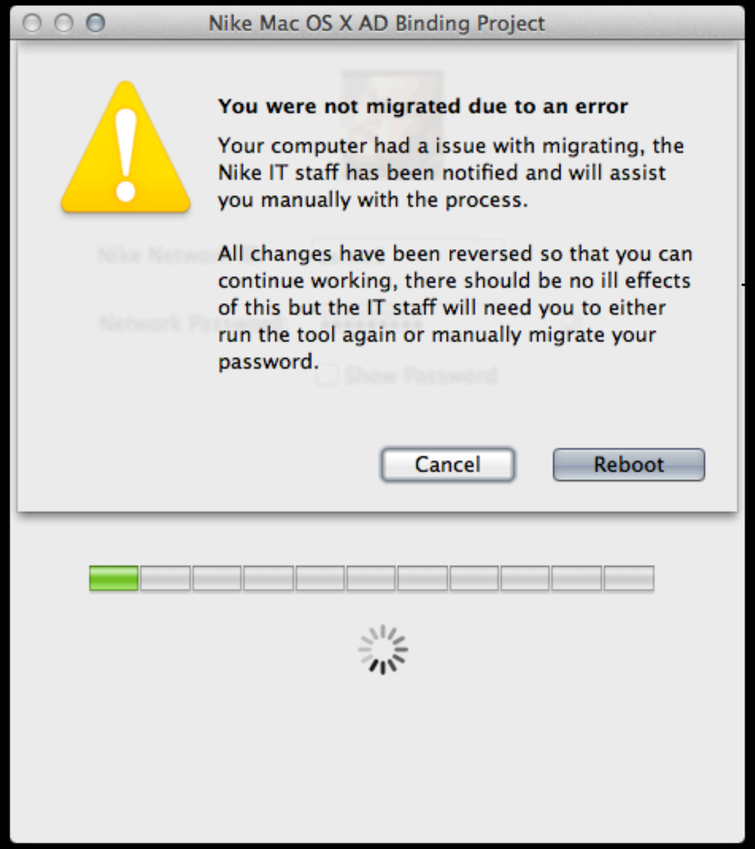


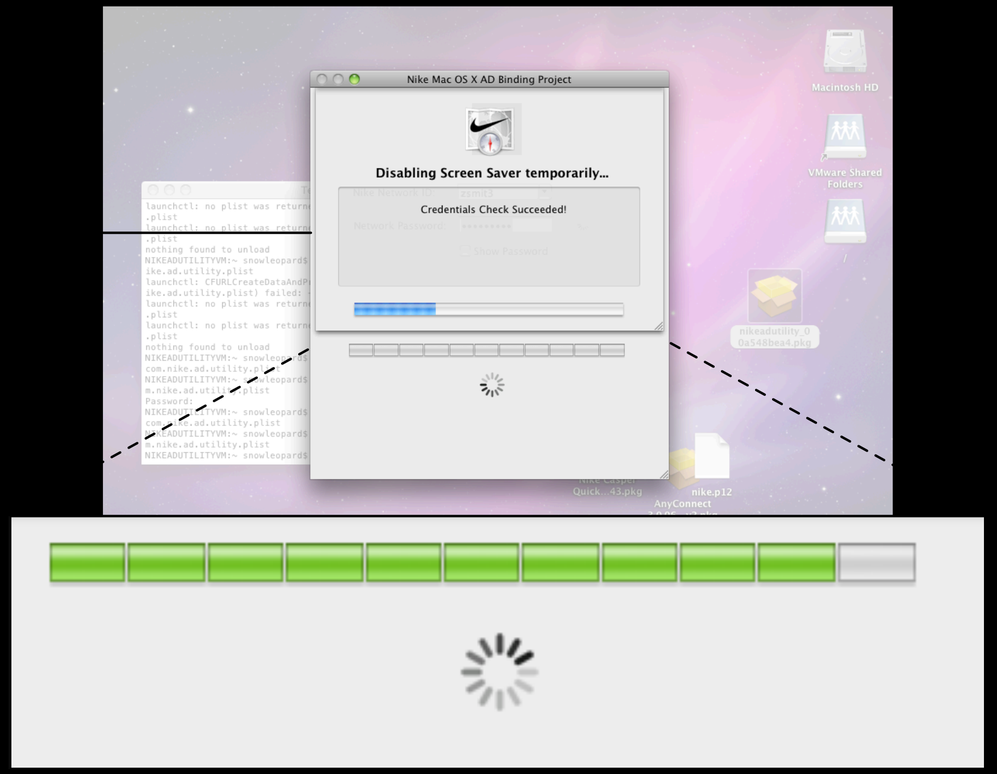

Macintosh HD
VMware Shared Folders
/
Nike Mac OS X AD Binding Project
Please enter your local password
Please enter the password you use to login to this computer.
Local password:
Continue
Continue
Last login: Wed Apr 25 09:59:15
NIKEADUTILITYVM:~ snowleopard$
Password:
NIKEADUTILITYVM:~ snowleopard$
nikeadutility_2 8aa633e7e.pkg
Nike Casper Quick...43.pkg
AnyConnect 3.0.06...v2.pkg
nike.p12

dsimport

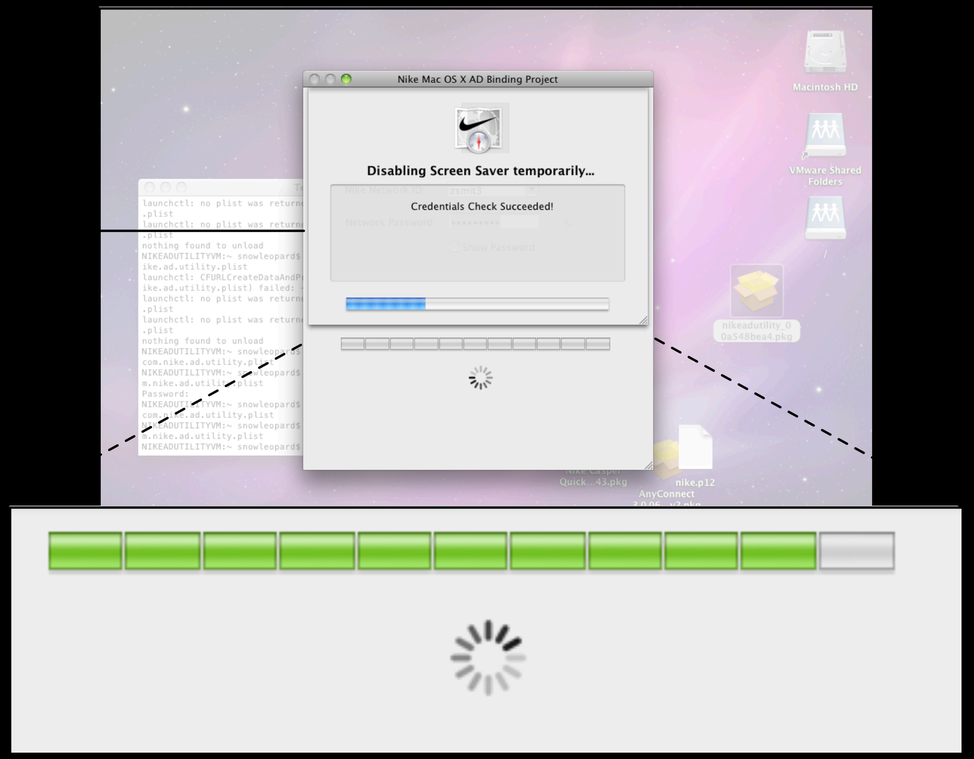
```
StatusMSG $FUNCNAME "Reimporting old user accounts"
if [ "$OsVersion" == "6" ]; then
  $dsimport /Library/Caches/$LocalUser.dsimport  /Local/Default O

elif [ "$OsVersion" == "5" ];then

    $dsimport -g "/Library/Caches/$LocalUser.dsimport"  /Local/Default O -u
"$TMP_USER" -p "$LocalPassword"

fi
```

Warning - Successful revert





Fatal (Critical) Failures



Exit codes

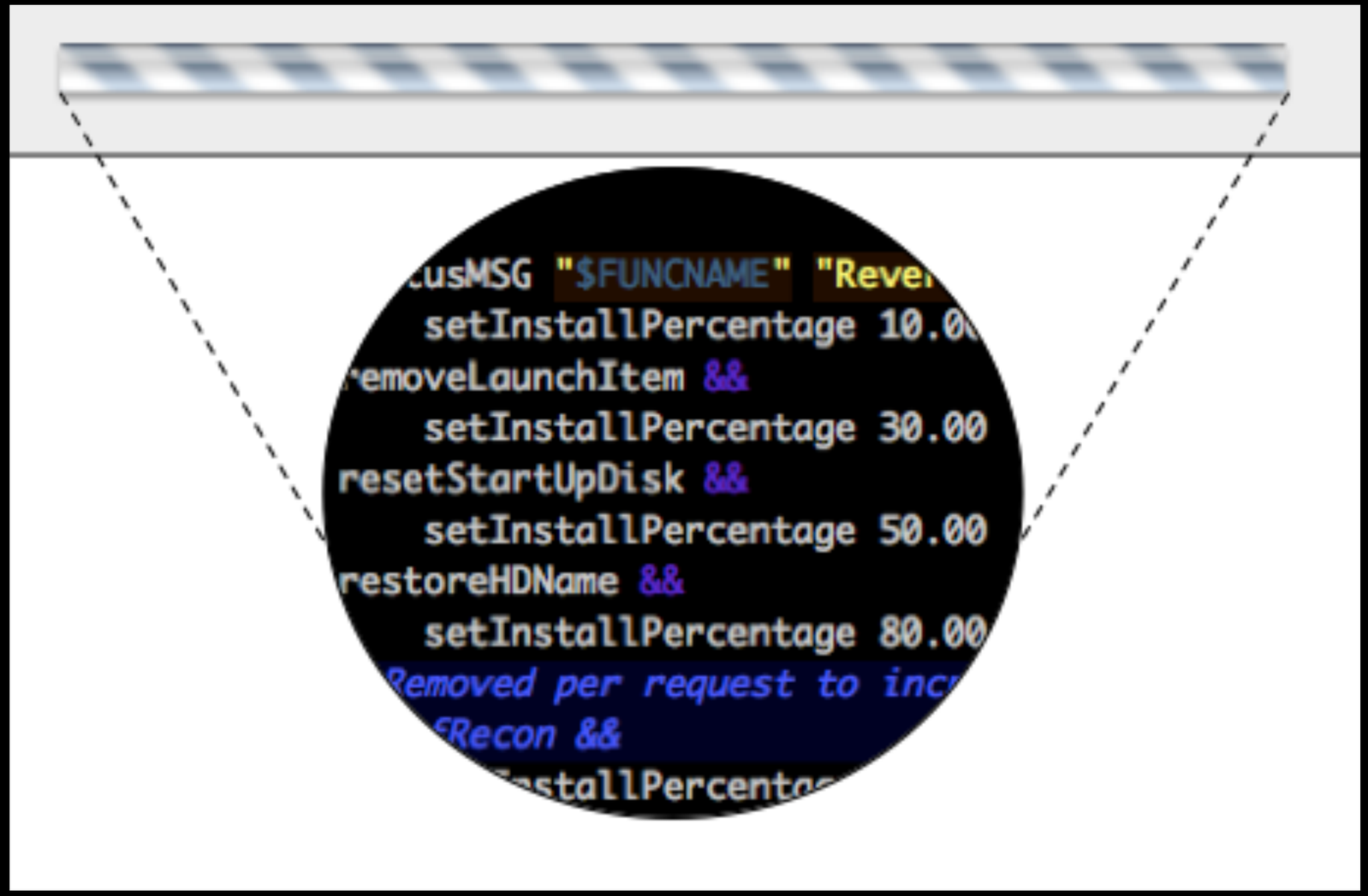
Exit Status	Meaning
0	Script exited successfully continue run
1	Script exited with warning continue run
200	Script exited with Fatal Error attempt to revert account

▼ scriptPlugins	Dictionary	(1 item)
▼ mainRunLoopScripts	Dictionary	(1 item)
▼ com.nike.adutility	Dictionary	(2 items)
headerTitle	String	Nike AD Utility Scripts
▼ itemScripts	Array	(11 items)
▶ Item 0	Dictionary	(13 items)
▶ Item 1	Dictionary	(13 items)
▶ Item 2	Dictionary	(13 items)
▶ Item 3	Dictionary	(13 items)
▶ Item 4	Dictionary	(13 items)
▶ Item 5	Dictionary	(13 items)
▶ Item 6	Dictionary	(13 items)
▶ Item 7	Dictionary	(13 items)
▶ Item 8	Dictionary	(13 items)
▶ Item 9	Dictionary	(13 items)
▶ Item 10	Dictionary	(13 items)

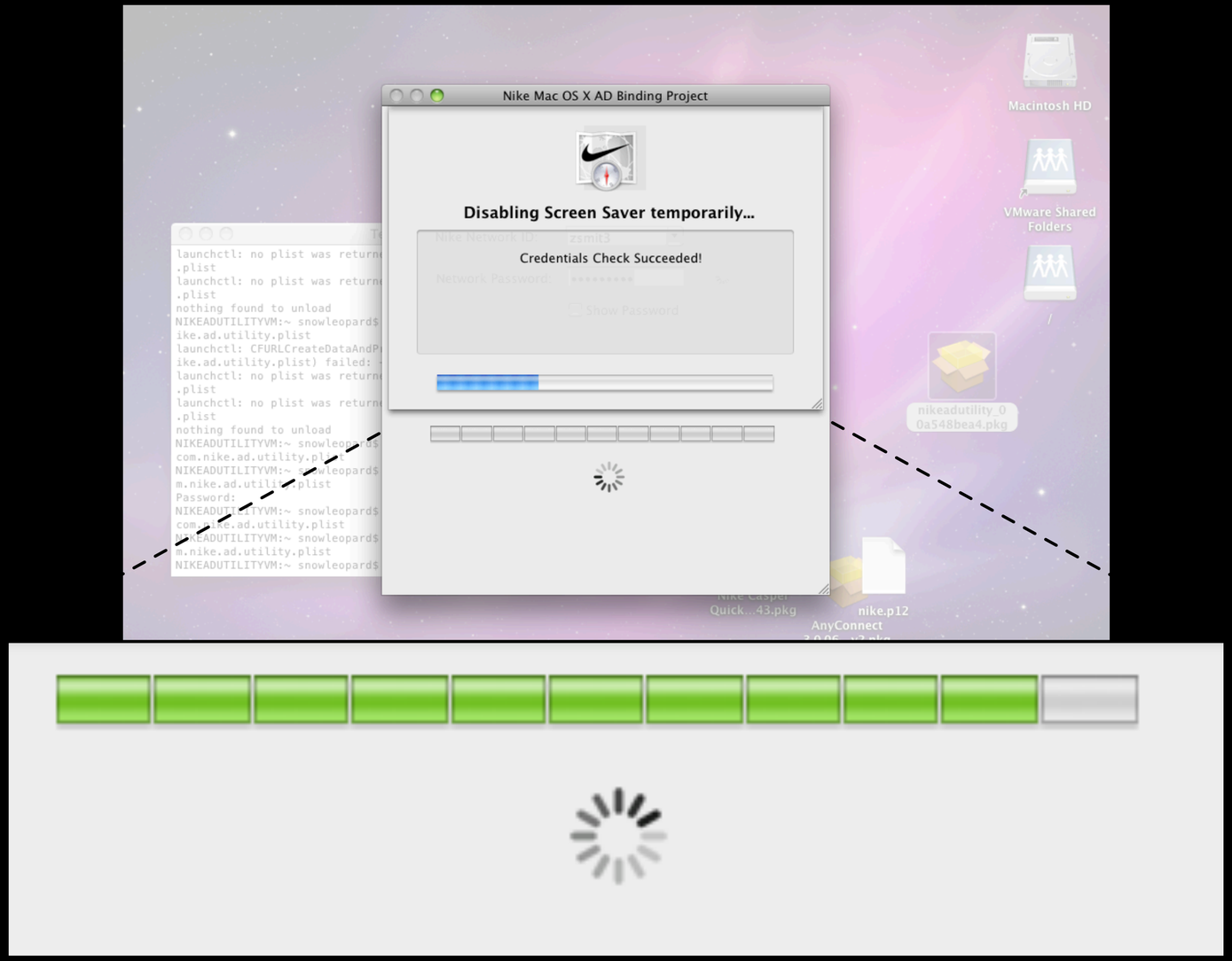
I appreciate your feedback

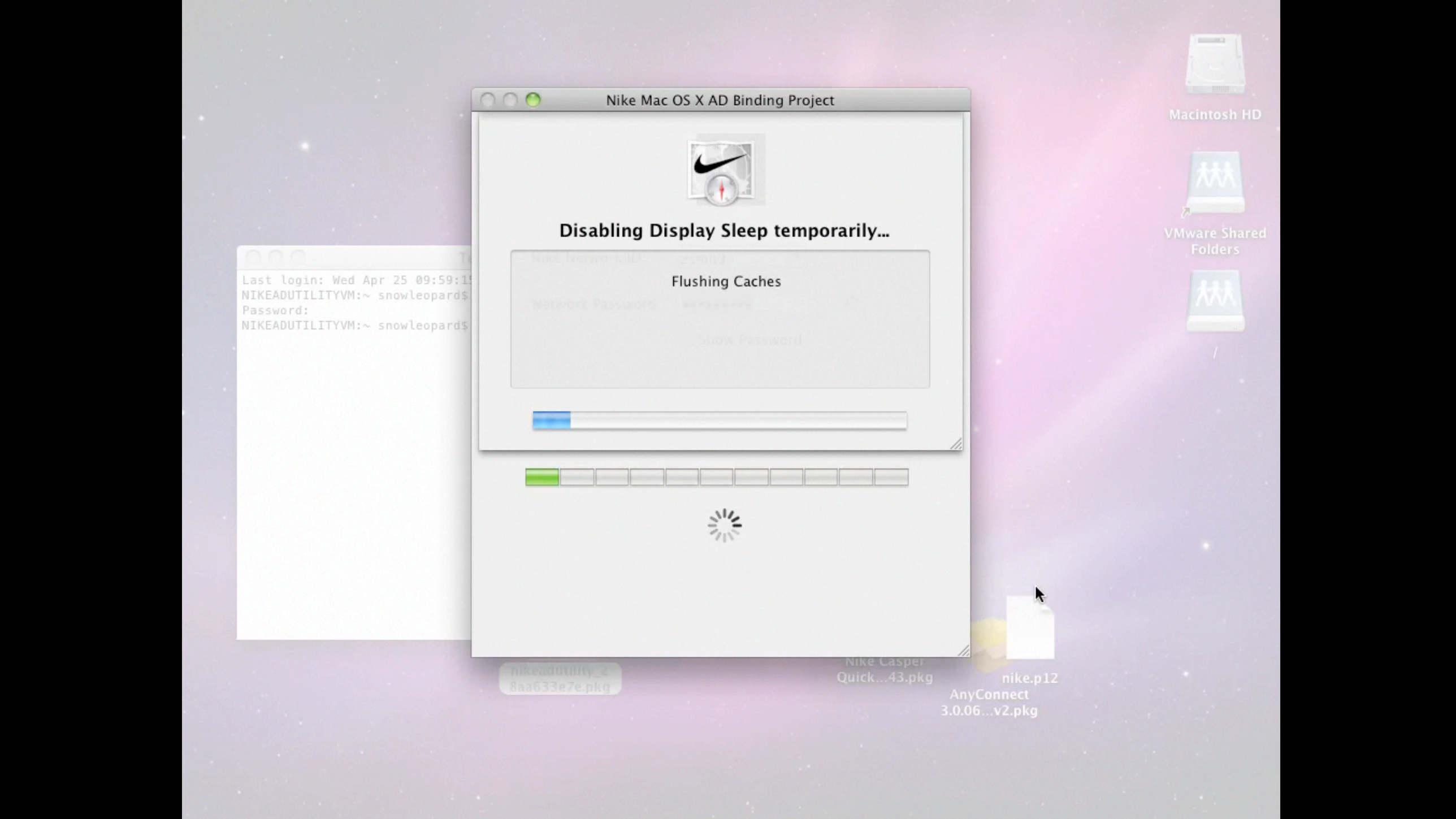
```
StatusMSG $ScriptName "Migrating Account..." uiphase

StatusMSG $ScriptName "Step Complete" uistatus
```

We are making progress

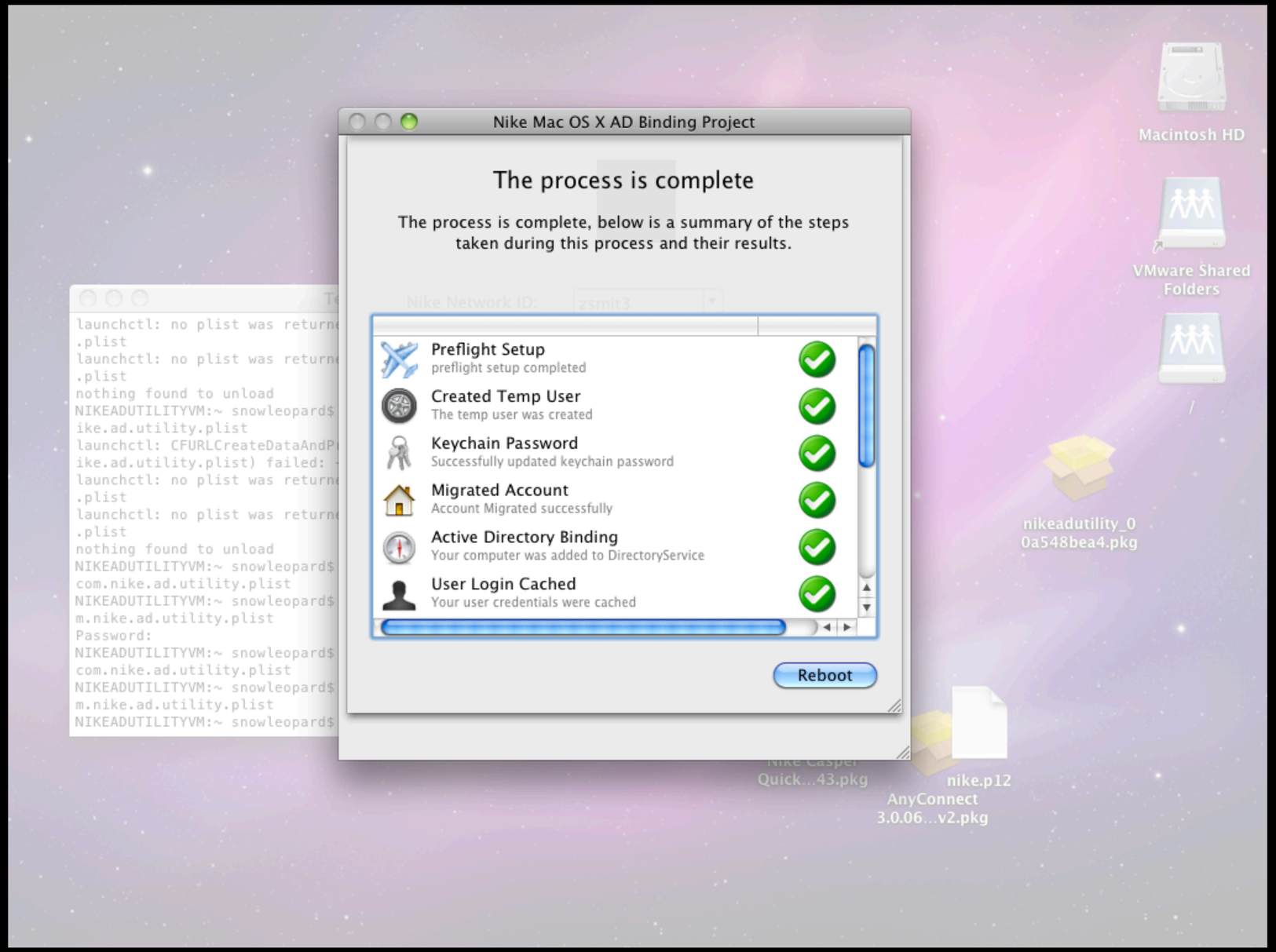


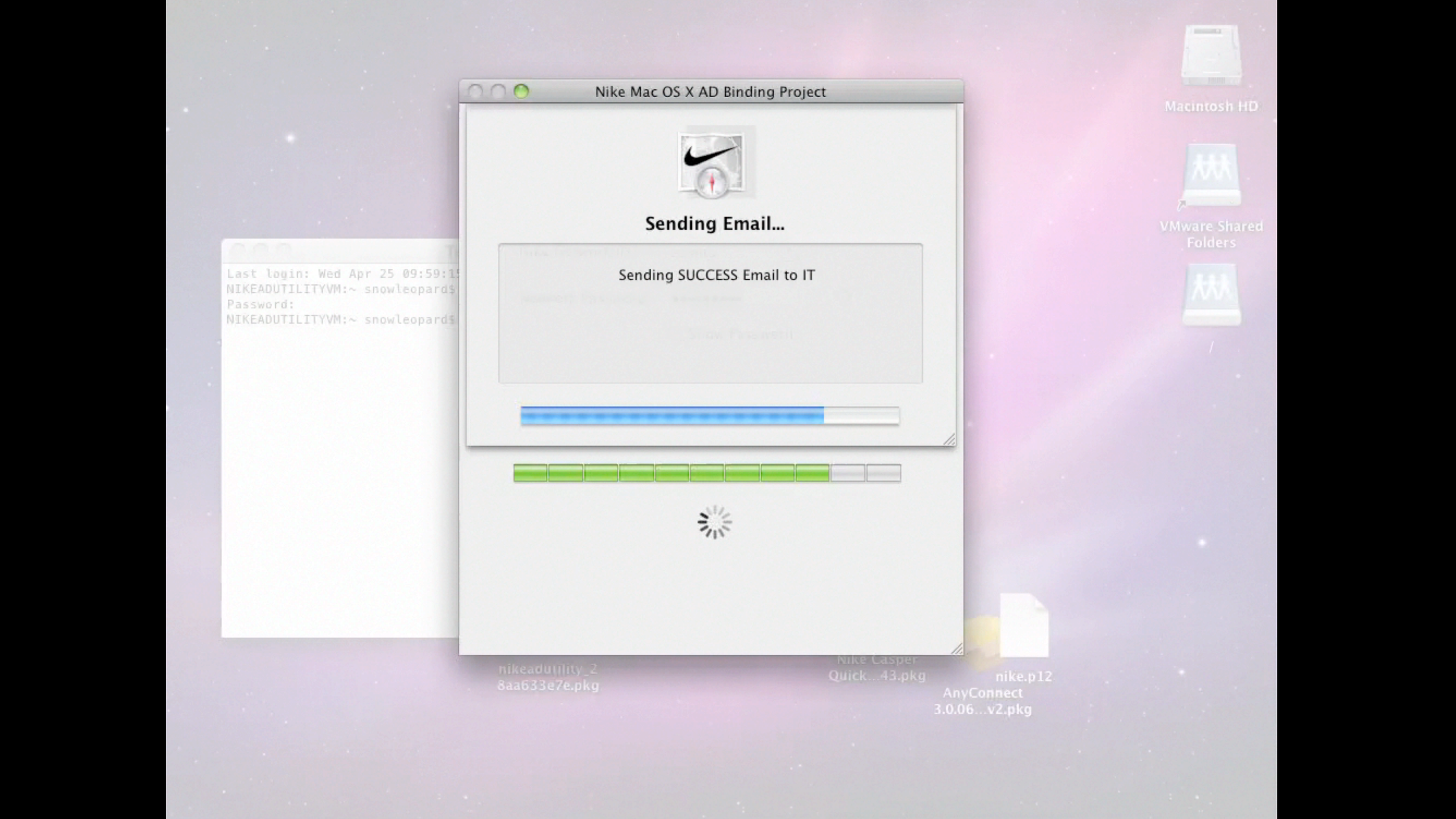
Level Indicator



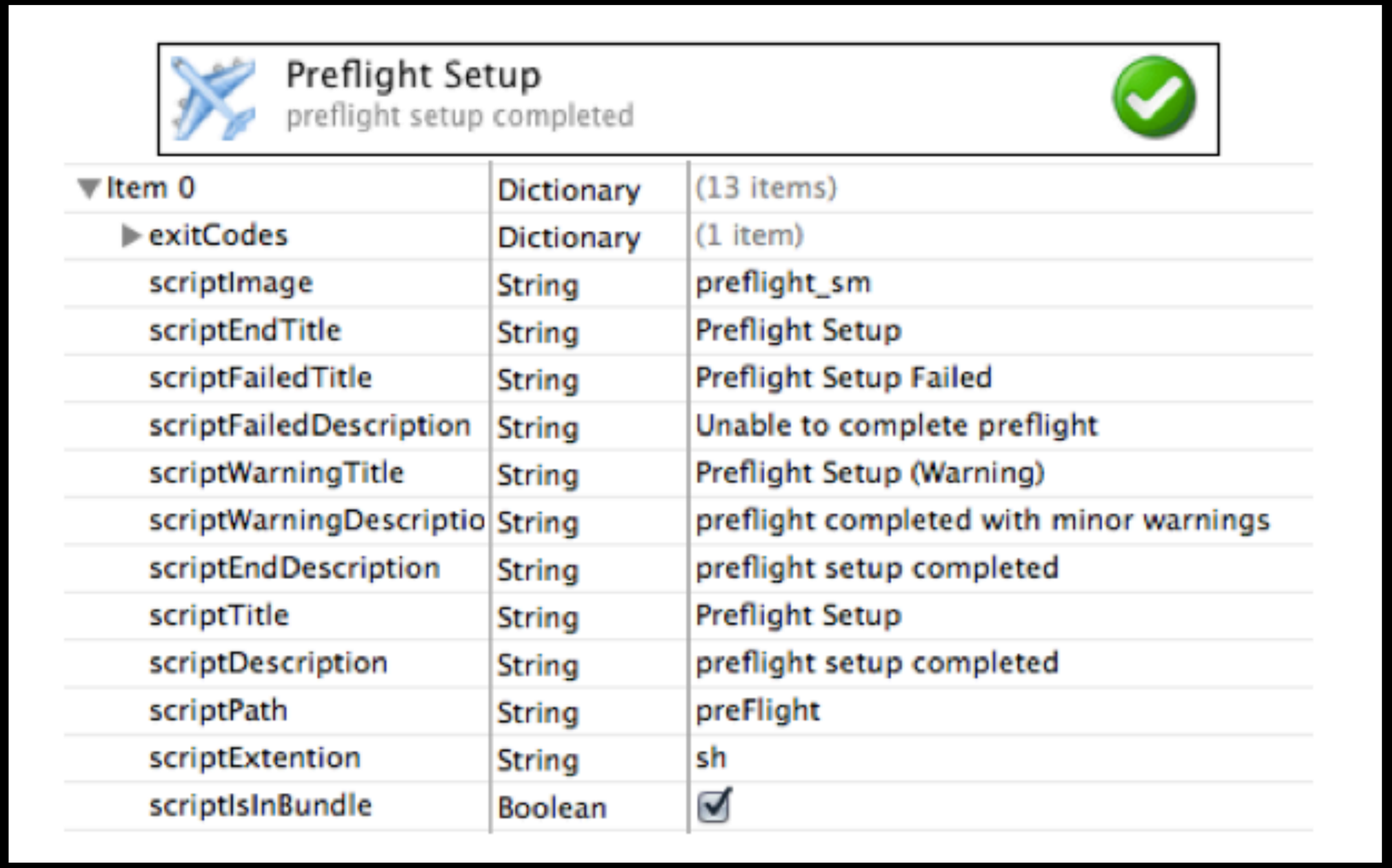

Nike Mac OS X AD Binding Project
Disabling Display Sleep temporarily...
Flushing Caches
Last login: Wed Apr 25 09:59:1
NIKEADUTILITYVM:~ snowleopard$
Password:
NIKEADUTILITYVM:~ snowleopard$
Macintosh HD
VMware Shared Folders
nike.p12
AnyConnect 3.0.06...v2.pkg

Process Complete




Nike Mac OS X AD Binding Project
Sending Email...
Sending SUCCESS Email to IT
Macintosh HD
VMware Shared Folders
Last login: Wed Apr 25 09:59:15
NIKEADUTILITYVM:~ snowleopard$
Password:
NIKEADUTILITYVM:~ snowleopard$
nikeadutility_2 8aa633e7e.pkg
Nike Casper Quick...43.pkg
nike.p12
AnyConnect 3.0.06...v2.pkg

Granular warnings



▼ Item 0	Dictionary	(13 items)
▶ exitCodes	Dictionary	(1 item)
scriptImage	String	preflight_sm
scriptEndTitle	String	Preflight Setup
scriptFailedTitle	String	Preflight Setup Failed
scriptFailedDescription	String	Unable to complete preflight
scriptWarningTitle	String	Preflight Setup (Warning)
scriptWarningDescriptio	String	preflight completed with minor warnings
scriptEndDescription	String	preflight setup completed
scriptTitle	String	Preflight Setup
scriptDescription	String	preflight setup completed
scriptPath	String	preFlight
scriptExtention	String	sh
scriptIsInBundle	Boolean	☑

Migration Logs

- dumped to share point
- email sent on failures
- logged all locally
- launchd logs any cocoa issues

Migration Logs

- dumped to share point
- email sent on failures
- logged all locally
- launchd logs any cocoa issues

```
<key>StandardOutPath</key>
<string>/Library/Logs/Winnebago/Winnebago.log</string>
<key>StandardErrorPath</key>
<string>/Library/Logs/Winnebago/Winnebago.log</string>
```

To Do:

To Do:

- Move all dialogs to WebKit rather then plist text keys.

To Do:

- Move all dialogs to WebKit rather then plist text keys.
- Consolidate configuration files (source file & plist)

To Do:

- Move all dialogs to WebKit rather then plist text keys.
- Consolidate configuration files (source file & plist)
- Convert all NSTasks/Scripts to use Environmental variables rather then passed parameters.

To Do:

- Move all dialogs to WebKit rather then plist text keys.
- Consolidate configuration files (source file & plist)
- Convert all NSTasks/Scripts to use Environmental variables rather then passed parameters.
- Allow custom images to be passed from shell functions.

To Do:

- Move all dialogs to WebKit rather then plist text keys.
- Consolidate configuration files (source file & plist)
- Convert all NSTasks/Scripts to use Environmental variables rather then passed parameters.
- Allow custom images to be passed from shell functions.
- Change uibridge to NSConnections

To Do:

- Move all dialogs to WebKit rather then plist text keys.
- Consolidate configuration files (source file & plist)
- Convert all NSTasks/Scripts to use Environmental variables rather then passed parameters.
- Allow custom images to be passed from shell functions.
- Change uibridge to NSConnections
- Convert BASH binding script to python, ruby...

Pull requests accepted

http://github.com/dev318/winnebago