Email:support@f-response.com
Website:www.f-response.com
Phone: 1-800-317-5497

YOUR MISSION: USE THE F-RESPONSE UNIVERSAL/NOW CONSOLE COM CONTROL TO AUTOMATE OPERATIONS

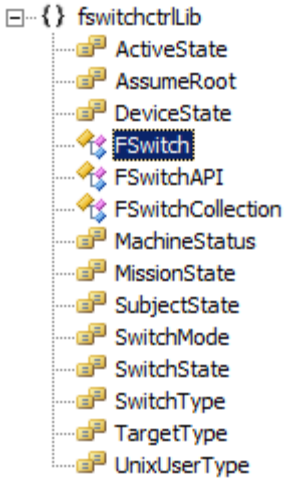
Note: This guide assumes you have downloaded and installed F-Response Universal or F-Response Now, and have an active licensed F-Response Universal or Now appliance.

OVERVIEW

F-Response Universal and Now both leverage an internal COM object, (fswitchctrl) to interact with remote systems and the appliance itself. In developing this document we are providing a mechanism for customers of F-Response Now and Universal to automate the operations of the user console. The COM library provided by F-Response includes both documented and un-documented methods and properties. We caution you against using any of the undocumented methods as they are not presently used and may represent stubs for future enhancements.

UNDERSTANDING THE FSWITCHSTRL OBJECT MODEL

The FSWITCHCTRL control presents a very simple and straightforward object model that can be accessed from a variety of COM aware tools, including the free Microsoft Visual Studio Express edition.



Let's take a moment to provide a quick overview of the individual objects and collections in the FSWITCHCTRL.

FSwitch

The FSwitch object represents a remote F-Response Universal or Now appliance. The FSwitch name stems from a prior branding exercise and can be ignored. The object possess all methods and properties necessary to interact with the remote appliance, authenticate, mount and unmount devices, etc.

FSwitchAPI

The FSwitchAPI object forms the root of the object hierarchy and contains a single method, FSwitchCollection. This method returns the FSwitchCollection object.

FSwitchCollection

The FSwitchCollection object represents all the currently configured F-Response Now and Universal appliances in the console. If no appliances are currently configured this object will be empty.

USING THE FSWITCHCTRL OBJECT IN POWERSHELL

Microsoft's PowerShell is a very powerful COM aware scripting language and is a logical first environment to detail using the FSWITCHCTRL in. We have included two PowerShell scripts below:

Email:support@f-response.com
Website:www.f-response.com
Phone: 1-800-317-5497

SIMPLE CONNECTION TO F-RESPONSE UNIVERSAL/NOW, LISTING SUBJECTS AND MOUNTING TARGETS

```
#initialize the com library
$fswitchapi = New-Object -comObject fswitchctrl.FSwitchAPI

#get a collection of our existing configured F-Response Universal Appliances, if we have none
then we should add one
$fswitchcollection = $fswitchapi.FSwitches

#select the appliance from the collection, you can use ordinal number (0,1,2,etc) or appliance
name
$fswitch = $fswitchcollection.Item("APPLIANCE")

#connect to the appliance
$connstatus = $fswitch.Connect()

#appliance connection status options include Connected = 0, NotConnected = 1, ErrorClockSkew = 2,
ErrorVersionLevelMismatch = 3, ErrorAuthenticationFailed = 4,       ErrorExpired = 5
# ErrorClockSkew means the appliance and examiner's system clocks are too far out of sequence
# ErrorVersionLevelMismatch means the appliance and the examiner are using different protocol
versions, get the appropriate installer from the appliance and re-install on your examiner
machine, http://appliancenameorip/univdl
# ErrorAuthenticationFailed means the provided username and password was invalid
# ErrorExpired means the license on the appliance has expired and must be renewed, please contact
support
#are we connected?
if($connstatus -eq 0){
        $hostname = $fswitch.FSwitchHostname
        write "Connected to $hostname"

        Sleep 5
        $subjects = $fswitch.SubjectsFiltered($machine)
        if (!$subjects){
                write "No Subjects Found?"
        }
        else{
                write "Subjects Detected"
                foreach ($subject in $subjects)
                {
                        $subjectname = $fswitch.SubjectsName($subject);
                        #The following allows us to filter the returned subject targets by name,
for instance if we only wanted DiscoveryShare targets we would use "DiscoveryShare*"
                                $targets = $fswitch.SubjectTargets($subject,"DiscoveryShare*");
                                write "Subject: $subjectname"
                                foreach ($target in $targets)
                                {
                                        $targetname = $fswitch.SubjectTargetName($subject,$target)
                                        write $targetname
                                        $mounted = $fswitch.MountDevice($subject,$target)
                                        #it can take some time to mount a target, so let's check it
every 10 seconds or so until we get a valid mount point
                                        while ($mounted -like "\\.\:\"){
                                                Sleep 10
                                                $mounted =
$fswitch.TargetMountPoint($subject,$target)
                                        }
                                        $mounted =
[string]$fswitch.TargetMountPointShort($subject,$target)
                                        $pathstr = $mounted + ":\\"
                                        write "Mounted $targetname as $pathstr"
```

Email:support@f-response.com
Website:www.f-response.com
Phone: 1-800-317-5497

```
                                        #now that we have the device mounted as a DiscoveryShare,
let's list the root contents using powershell
                                        $discoverysharecontents = Get-ChildItem -Path $pathstr
                                        foreach ($item in $discoverysharecontents)
                                        {
                                                Write $item.Name
                                        }
                                #now let's unmount the device
                                        $result = $fswitch.UnmountDevice($subject,$target)
                                        Sleep 5
                                }
                }
        }

$result = $fswitch.Disconnect()
}
$result = [System.Runtime.Interopservices.Marshal]::ReleaseComObject($fswitchapi)
Remove-Variable fswitchapi
```

Essentially the simple connection and DiscoveryShare script above shows how we can automate connecting to specific subjects and targets and leverage the capabilities of Powershell to locate files and folders of interest. For additional information on methods and properties available see the Object Viewer in Visual Studio or another COM aware application.

AUTOMATING DEPLOYMENT AND COLLECTION OF MEMORY IMAGE FROM POWERSHELL

```
################################################################################
#
# F-Response Universal Script for deploying F-Response Universal to remote machines
# and copying MemoryShare contents to a specified location
#
# Pre-requisites: F-Response Universal Appliance, hostname for remote subject machine, hostname
must match deployment hostname
# Date: January 25, 2015
# Author: M. Shannon
# usage, memoryshare_sample_script.ps1 -appliance <APPLIANCE> -machine <TARGETHOSTNAME> -
deployuser <USERNAME> -deploydomain <DOMAIN> -deploypass <PASSWORD> -dest
<FULLPATHTOPLACEIMAGEFILE>
################################################################################
Param(
        [parameter(Mandatory=$true)]$appliance,
        [parameter(Mandatory=$true)]$machine,
        [parameter(Mandatory=$true)]$deployuser,
        [parameter(Mandatory=$true)]$deploydomain,
        [parameter(Mandatory=$true)]$deploypass,
        [parameter(Mandatory=$true)]$dest
)

#initialize the com library
$fswitchapi = New-Object -comObject fswitchctrl.FSwitchAPI

#get a collection of our existing configured F-Response Universal Appliances, if we have none
then we should add one
$fswitchcollection = $fswitchapi.FSwitches


#select the appliance from the collection, you can use ordinal number (0,1,2,etc) or appliance
name
```

Email:support@f-response.com
Website:www.f-response.com
Phone: 1-800-317-5497

```
$fswitch = $fswitchcollection.Item([string]$appliance)

#connect to the appliance
$connstatus = $fswitch.Connect()

#appliance connection status options include Connected = 0, NotConnected = 1, ErrorClockSkew = 2,
ErrorVersionLevelMismatch = 3, ErrorAuthenticationFailed = 4,       ErrorExpired = 5
# ErrorClockSkew means the appliance and examiner's system clocks are too far out of sequence
# ErrorVersionLevelMismatch means the appliance and the examiner are using different protocol
versions, get the appropriate installer from the appliance and re-install on your examiner
machine, http://appliancenameorip/univdl
# ErrorAuthenticationFailed means the provided username and password was invalid
# ErrorExpired means the license on the appliance has expired and must be renewed, please contact
support

#are we connected?
if($connstatus -eq 0){
        write "Connected to "
        write $fswitch.FSwitchHostname

        #lets setup a user account to use to deploy F-Response Universal software
        $fswitch.DeployViaLanWanAddWindowsCredential($deployuser,$deploydomain,$deploypass)

        #let's get the status of a remote windows machine we'd like to deploy to
        $status = $fswitch.DeployViaLanWanGetStatus($machine)

        #possible status values NotAvailable= 10, Available  = 11, Stopped = 12,    Started = 13,
MachineError = 14
        #NotAvailable means the provided credentials were insufficient to deploy F-Response
Universal to the remote machine
        #Available means the provided credentials were sufficient to authenticate to the machine
there is no F-Response Universal software there
        #Stopped means the remote machine has F-Response Universal software but it is currently
stopped
        #Started means the remote machine has F-Response Universal software installed and running
        #MachineError means the remote machine reported values during deployment that were
inconsistent with what we expected
        if ($status -eq 11){#Available, let's install and get started
                write "F-Response Universal software not deployed to $machine, installing"
                $status = $fswitch.DeployViaLanWanInstallStart($machine)
        }
        if ($status -eq 13){#Started, let's get some details

                write "Let's wait 10 seconds to let the subject become visible in the system"
                write "Waiting for $machine..."
                Sleep 3

                #it's time to get a list of subjects on the appliance, do we want to filter that
list by name? If so, put something in there with * where desired; we are going to use the
wildcard to get all subjects
                $subjects = $fswitch.SubjectsFiltered($machine)
                if (!$subjects){
                        write "No Subjects Found?"
                }
                else{
                        write "Subjects Detected"
                        foreach ($subject in $subjects)
                        {
                                $subjectname = $fswitch.SubjectsName($subject);
```

Email:support@f-response.com
Website:www.f-response.com
Phone: 1-800-317-5497

```
                                #The following allows us to filter the returned subject targets by
name, for instance if we only wanted DiscoveryShare targets we would use "DiscoveryShare*"
                                $targets = $fswitch.SubjectTargets($subject,"MemoryShare*");
                                write "Subject: $subjectname"
                                foreach ($target in $targets)
                                {
                                                        $targetname =
$fswitch.SubjectTargetName($subject,$target)
                                                        write $targetname

                                                        $mounted = $fswitch.MountDevice($subject,$target)
                                                        #it can take some time to mount a target, so let's
check it every 10 seconds or so until we get a valid mount point
                                                        while ($mounted -like "\\.\:\"){
                                                                Sleep 10
                                                                $mounted =
$fswitch.TargetMountPoint($subject,$target)
                                                        }
                                                        $path = $mounted.TrimStart("\\.\")
                                                        $memorypath = "$path\\physical-memory.img"
                                                        $memorydest = "$dest\\$subjectname-$targetname.img"
                                                        write "Copying MemoryImage from $memorypath to
$memorydest..."

                                                        Copy-Item $memorypath $memorydest
                                                        #now let's unmount the device
                                                        write "MemoryImage copy complete, unmounting..."
                                                        $fswitch.UnmountDevice($subject,$target)
                                                        Sleep 5
                                                        }
                                }
                        }
                }
        #we are all done with our system, time to remove it
        write "F-Response Universal software deployed to $machine, uninstalling"
        $status = $fswitch.DeployViaLanWanStopRemove($machine)
        }
$result = $fswitch.Disconnect()

$result = [System.Runtime.Interopservices.Marshal]::ReleaseComObject($fswitchapi)
Remove-Variable fswitchapi
```

The above script essentially provides a command line mechanism for deploying F-Response Universal to a remote machine, then accessing the MemoryShare™ of that machine and copying the live physical memory file to a local folder. Essentially this script could be used with little modifications in an IDS or SEIM system to automate the collection of volatile system data. For additional information on methods and properties available see the Object Viewer in Visual Studio or another COM aware application.

SUMMARY

In summary the F-Response Universal COM control is a highly flexible and extensible interface providing automation capabilities to any COM aware programming environment.