

Using Scripts with ArchiveOne

<https://campus.barracuda.com/doc/60358664/>

This article refers to the Barracuda ArchiveOne version 7.0 and higher.

This article provides detailed information for the ArchiveOne scripting API.

Prerequisites

The ArchiveOne scripting API is built using version 4.5 of .NET and as such we recommend you interact with it using a modern .NET compatible scripting technology such as PowerShell. Legacy scripting engines such as VBScript are not suitable for use with ArchiveOne due to incompatibilities with type handling.

This guide assumes you are using PowerShell as your primary scripting technology. By default, PowerShell uses version 2.0 of .NET and so it will need to be configured to use version 4.5. You can confirm which version of .NET your PowerShell is using by examining the contents of the \$PSVersionTable variable by typing the following at a command prompt:

PS C:\Users\paulj> \$PSVersionTable	
Name	Value
----	-----
PSVersion	3.0
WSManStackVersion	3.0
SerializationVersion	1.1.0.1
CLRVersion	4.0.30319.42000
BuildVersion	6.2.9200.16481
PSCompatibleVersions	{1.0, 2.0, 3.0}
PSRemotingProtocolVersion	2.2

Confirm the **CLRVersion** is set to 4.0.30319.42000 or later. If it is not set to this version (or later), your PowerShell is not using .NET 4.5. You can configure PowerShell to use 4.5 as follows:

From the PowerShell command prompt, type the following:

```
PS C:\Users\paulj> notepad $PSHome\PowerShell.exe.config
```

If the file doesn't exist, create it. Otherwise edit the config file and ensure it looks like the following:

PowerShell.exe.config
<pre><?xml version="1.0"?> <configuration> <startup useLegacyV2RuntimeActivationPolicy="true"> <supportedRuntime version="v4.0.30319"/> <supportedRuntime version="v2.0.50727"/> </startup> </configuration></pre>

Save the file and close your PowerShell window. Open a new PowerShell window and examine the \$PSVersionTable variable again, you should now see that the CLR version is 4.0 or later.

API Usage

The API is exposed to the scripting environment via a DLL called C2CSYSTEMS.ArchiveOne.Policy.dll. This DLL contains data types and methods that make up the complete ArchiveOne API. As such, whenever you want to access the API you will need to load this DLL into the PowerShell environment as shown below.

To access the API from the command line, type the following

<pre>PS C:\Users\paulj> Add-Type -Path "C:\Program Files (x86)\Barracuda\ArchiveOne\SDK\C2CSYSTEMS.ArchiveOne.Policy.dll"</pre>
--

To access the API from within a PowerShell script file add the following to the top of your script

<pre>Add-Type -Path "C:\Program Files (x86)\Barracuda\ArchiveOne\SDK\C2CSYSTEMS.ArchiveOne.Policy.dll" function MyFunction() { .. Do something with the API .. }</pre>
--

API Reference

The DLL exposes a set of primary .NET objects that contain methods that are used to perform various activities on the ArchiveOne software, these are listed below.

- ArchiveManager
- MailboxManager
- PolicyManager
- RepositoryManager
- SearchManager

Each .NET object is now described below in detail

ArchiveManager

This manager object contains the following methods:

- GetPhase2Status
- GetCurrentStatusText
- StartArchive
- StopArchive

GetPhase2Status

Description: Get status of phase 2 (archiving)

Return: Boolean - True indicates phase 2 is running, False indicates phase 2 is not running.

Example:

TestGetPhase2Status.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSys...ArchiveOne.Policy.dll";
function TestGetPhase2Status
{
    $archiveManager = new-object
    "C2CSys...ArchiveOne.Policy.Archiv...Manager";
    $archiveManager.GetPhase2Status()
}
```

```
PS C:\Users\paulj> TestGetPhase2Status
False
```

GetCurrentStatusText

Description: Get current status text (as displayed in the Administration console)

Return: String - text description of the current status

Example:

TestGetCurrentStatusText.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";
function TestGetCurrentStatusText
{
    $archiveManager = new-object
    "C2CSystems.ArchiveOne.Policy.ArchiveManager";
    $archiveManager.GetCurrentStatusText()
}
```

```
PS C:\Users\paulj> TestGetCurrentStatusText
Phase two processing not yet started
Status last updated on 17 Nov 2016 at 11:29:16
```

StartArchive

Description: Starts the phase 2 archiving for the given repository name. Any items that were queued for archiving for this repository will be processed when this call is made.

Arguments: RepositoryName - Display name of the repository to process

Return: Boolean - True if the policy was started, False if it was not started

Example:

TestStartArchive.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";
function TestStartArchive($repositoryName)
{
    $archiveManager = new-object
    "C2CSystems.ArchiveOne.Policy.ArchiveManager";
    $archiveManager.StartArchive($repositoryName)
}
```

```
PS C:\Users\paulj> TestStartArchive "Repository 1";
True
```

```
PS C:\Users\paulj> TestGetCurrentStatusText
0 messages queued for archiving on Monday, November 21, 2016 at 11:17:49
Processing repository: Repository 1
    Pending 0 messages
    Processing 0 messages
    Completed 0 messages
Total time taken 0 hours, 0 minutes, 0 seconds
0 messages are queued for other repositories
Status last updated on 21 Nov 2016 at 11:17:50
```

StopArchive

Description: Stops any currently running phase 2 activity immediately without committing.

Return: True if the call was successful, otherwise False

Example:

TestStopArchive.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";
function TestStopArchive
{
    $archiveManager = new-object
    "C2CSystems.ArchiveOne.Policy.ArchiveManager";
    $archiveManager.StopArchive()
}
```

```
PS C:\Users\paulj> TestStopArchive
True
```

MailboxManager

This manager object contains the following methods

- AddNewMailbox
- AddNewMailboxes
- CreateMailbox
- GetEWSUrl
- GetID
- GetMailbox
- GetMailboxes
- GetPublicFolders
- RemoveMailbox
- UpdateMailbox

AddNewMailbox

Description: Adds a new mailbox to the mailbox manager database using default values for the repository and search user etc.. The mailbox is specified using its DN.

Return: True if the call was successful, otherwise False. An additional Out parameter is also passed back giving additional status information about the call.

Example:

TestAddNewMailbox.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";
function TestAddNewMailbox($newMailbox)
{
    $status = 0
    $mailboxManager = new-object
    "C2CSystems.ArchiveOne.Policy.MailboxManager";
    $mailboxManager.AddNewMailbox($newMailbox,[ref]$status)
    Write-Host "Status value:" $status
    $mailboxManager.GetMailboxes() | FL
}
```

```
PS C:\Users\paulj> TestAddNewMailbox &quot;/o=Blakes7/ou=Exchange  
Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=Pau  
l1623&quot;  
True  
Status value: ACSS_OK
```

```
RepositoryName      : Repository 1  
MailboxID           : /o=Blakes7/ou=Exchange Administrative Group  
(FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623  
Enabled             : True  
CreateSearchFolder  : True  
EnableOfflineArchive : False  
Email               : Paul1@scutter2000.c2c2000.com  
Name                : Paul1  
Sid                 :  
StoreDn             : /o=Blakes7/ou=Exchange Administrative Group  
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc  
hCasArray.scutter2000.c2c2000.com/cn=Microsoft  
Private MDB
```

AddNewMailboxes

Description: Adds one or more mailboxes using the given array of DN strings to the mailbox manager database. The mailboxes are created using default values for the repository and search user etc..

Return: True if the call was successful, otherwise False. An additional Out parameter is also passed back giving additional status information about the call

Example:

```
TestAddNewMailboxes.ps1
```

```
Add-Type -Path &quot;C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll&quot;;
```

```
function TestAddNewMailboxes
{
$mailboxes = @(&quot;/o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ&quot;;,
&quot;/o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623&quot;);

$status = 0
$mailboxManager = new-object
&quot;C2CSystems.ArchiveOne.Policy.MailboxManager&quot;;
$mailboxManager.AddNewMailboxes($mailboxes,[ref]$status)
Write-Host &quot;Status value:&quot; $status
$mailboxManager.GetMailboxes() | FL
}
```

```
PS C:\Users\paulj&gt; TestAddNewMailboxes
True
Status value: ACSS_OK
```

```
RepositoryName      : Repository 1
MailboxID           : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ
Enabled             : True
CreateSearchFolder  : True
EnableOfflineArchive : False
Email              : PaulJ@scutter2000.c2c2000.com
Name                : Paul Jones
Sid                :
StoreDn             : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc
hCasArray.scutter2000.c2c2000.com/cn=Microsoft Private
MDB

RepositoryName      : Repository 1
MailboxID           : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623
Enabled             : True
CreateSearchFolder  : True
EnableOfflineArchive : False
Email              : Paul1@scutter2000.c2c2000.com
Name                : Paul1
Sid                :
StoreDn             : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc
hCasArray.scutter2000.c2c2000.com/cn=Microsoft Private
MDB
```


CreateMailbox

Description: Creates a mailbox object that can be configured prior to adding it to the mailbox manager database.

Return: True if the call was successful, otherwise False

Example:

TestCreateMailbox.ps1

```
Add-Type -Path &quot;C:\Program Files  
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll&quot;
```

```
function TestCreateMailbox($mailboxDN)  
{
```

```
    $status = 0
```

```
    $mailboxManager = new-object  
&quot;C2CSystems.ArchiveOne.Policy.MailboxManager&quot;  
    $newMailbox = $mailboxManager.CreateMailbox($mailboxDN)
```

```
# Populate newly created mailbox object prior to adding it to the mailbox  
manager database
```

```
$newMailbox.RepositoryName = &quot;Repository 1&quot;  
$newMailbox.CreateSearchFolder = $True  
$newMailbox.AddSearchUser($mailboxDN)  
$newMailbox.Enabled = $True
```

```
# Now add the object to the database  
$mailboxManager.UpdateMailbox($newMailbox)  
Write-Host &quot;Status value:&quot; $status
```

```
$mailboxManager.GetMailboxes() | FL  
}
```

```

PS C:\Users\paulj> TestCreateMailbox "/o=Blakes7/ou=Exchange
Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623"
True
True
Status value: 0

MailboxID      RepositoryName      : Repository 1
                : /o=Blakes7/ou=Exchange Administrative Group
                (FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623
                Enabled      : True
                CreateSearchFolder : True
                EnableOfflineArchive : False
Email          : Paul1@scutter2000.c2c2000.com
                Name         : Paul1
                Sid          :
StoreDn        : /o=Blakes7/ou=Exchange Administrative Group
                (FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc
                hCasArray.scutter2000.c2c2000.com/cn=Microsoft
                Private MDB
  
```

GetEWSUrl

Description: Uses AutoDiscovery to retrieve the URL's for both EWS and OWA. It also returns the major/minor version of the target exchange server hosting EWS.

Return: True if the call was successful, otherwise False

Example:

TestGetEWSUrl.ps1

```

Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll"

function TestGetEWSUrl($emailAddress)
{
    $urlEWS = ""
    $urlOWA = ""
    $majorVersion = 0
    $minorVersion = 0

    $mailboxManager = new-object "C2CSystems.ArchiveOne.Policy.MailboxManager"
    $mailboxManager.GetEWSUrl($emailAddress, [ref]$urlEWS, [ref]$urlOWA, [ref]$majorVersion, [ref]$minorVersion)

    Write-Host "Email Address : " $emailAddress
    Write-Host "EWS Url      : " $urlEWS
    Write-Host "OWA Url      : " $urlOWA
    Write-Host "Major Version : " $majorVersion
    Write-Host "Minor Version : " $minorVersion
}
  
```

```
PS C:\Users\paulj> TestGetEWSUrl paulj@scutter2000.c2c2000.com
True
Email Address : paulj@scutter2000.c2c2000.com
EWS Url       :
https://exchcasarray.scutter2000.c2c2000.com/EWS/Exchange.asmx
OWA Url       : https://ex2010-cas1.scutter2000.c2c2000.com/owa/
Major Version : 14
Minor Version : 3
```

GetID

Description: Attempts to get the ID (mailbox DN) of the given email address or display name. The ID is the primary identifier for mailboxes within the ArchiveOne scripting API.

Return: Mailbox DN if the call was success, * if the given name was ambiguous or blank if no ID could be found

Example:

TestGetID.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestGetID($emailAddress)
{
    $mailboxManager = new-object
    "C2CSystems.ArchiveOne.Policy.MailboxManager";
    $id = $mailboxManager.GetID($emailAddress)
    Write-Host "ID :" $id
}
```

```
PS C:\Users\paulj> TestGetID paulj@scutter2000.c2c2000.com
ID : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ
PS C:\Users\paulj> TestGetID "Paul Jones";
ID : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ
PS C:\Users\paulj> TestGetID "Paul";
ID : *
PS C:\Users\paulj> TestGetID asdfasdfasdf
ID :
```

GetMailbox

Description: Retrieves a mailbox object from the database using the given ID.

Return: A mailbox object if successfully otherwise null

Example:

TestGetMailbox.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestGetMailbox($mailboxDN)
{
    $mailboxManager = new-object
    "C2CSystems.ArchiveOne.Policy.MailboxManager";
    $mailbox = $mailboxManager.GetMailbox($mailboxDN)
    $mailbox | fl
}
```

```
PS C:\Users\paulj> TestGetMailbox "/o=Blakes7/ou=Exchange
Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623";
```

```
RepositoryName      : Repository 1
MailboxID            : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623
Enabled              : True
CreateSearchFolder   : True
EnableOfflineArchive : False
Email                : Paul1@scutter2000.c2c2000.com
Name                 : Paul1
Sid                  :
StoreDn              : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc
hCasArray.scutter2000.c2c2000.com/cn=Microsoft
Private MDB
```

GetMailboxes

Description: Retrieves the entire mailbox manager database as an array of Mailbox objects.

Return: Array of mailbox objects if successful, otherwise null

Example:

TestGetMailboxes.ps1

```

Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll"

function TestGetMailboxes
{
    $mailboxManager = new-object
    "C2CSystems.ArchiveOne.Policy.MailboxManager"
    $mailboxes = $mailboxManager.GetMailboxes()
    $mailboxes | fl
}

```

PS C:\Users\paulj> TestGetMailboxes

```

RepositoryName      : Repository 1
MailboxID            : /o=Blakes7/ou=Exchange Administrative Group
                      (FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ
                      Enabled       : True
                      CreateSearchFolder : True
                      EnableOfflineArchive : False
Email                : PaulJ@scutter2000.c2c2000.com
                      Name          : Paul Jones
                      Sid           :
StoreDn              : /o=Blakes7/ou=Exchange Administrative Group
                      (FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc
                      hCasArray.scutter2000.c2c2000.com/cn=Microsoft
                      Private MDB

RepositoryName      : Repository 1
MailboxID            : /o=Blakes7/ou=Exchange Administrative Group
                      (FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623
                      Enabled       : True
                      CreateSearchFolder : True
                      EnableOfflineArchive : False
Email                : Paul1@scutter2000.c2c2000.com
                      Name          : Paul1
                      Sid           :
StoreDn              : /o=Blakes7/ou=Exchange Administrative Group
                      (FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc
                      hCasArray.scutter2000.c2c2000.com/cn=Microsoft
                      Private MDB

```

GetPublicFolders

Description: Retrieves the current list of Public Folders that have been configured for archiving.

Return: Array of Public Folder objects if successful, otherwise null

Example:

TestGetPublicFolders.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestGetPublicFolders
{
    $mailboxManager = new-object
    "C2CSystems.ArchiveOne.Policy.MailboxManager";
    $publicFolders = $mailboxManager.GetPublicFolders()
    $publicFolders | fl
}
```

PS C:\Users\paulj> TestGetPublicFolders

```
RepositoryName      : Repository 1
MailboxID            : \PaulJ
Enabled              : True
CreateSearchFolder   : False
EnableOfflineArchive : False
Email                :
Name                 : \PaulJ
Sid                  :
StoreDn              :
```

RemoveMailbox

Description: Removes the mailbox from the mailbox manager database using the given ID.

Return: True if the call was successful, otherwise False

Example:

TestRemoveMailbox.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestRemoveMailbox($mailboxDN)
{
    $mailboxManager = new-object
    "C2CSystems.ArchiveOne.Policy.MailboxManager";
    $mailboxManager.RemoveMailbox($mailboxDN)
}
```

```
PS C:\Users\paulj> TestRemoveMailbox "/o=Blakes7/ou=Exchange
Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623";
True
```

UpdateMailbox

Description: Writes the details of the given mailbox into the mailbox manager database (creating it if it doesn't already exist).

Return: True if the call was successful, otherwise False

Example:

TestUpdateMailbox.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestUpdateMailbox($mailboxDN)
{
    $status = 0
    $mailboxManager = new-object
    "C2CSystems.ArchiveOne.Policy.MailboxManager";
    $mailbox = $mailboxManager.GetMailbox($mailboxDN)

    $mailbox.RepositoryName = "Repository 2";

    $mailboxManager.UpdateMailbox($mailbox)
    $mailboxManager.GetMailboxes() | FL
}
```

```
PS C:\Users\paulj> TestUpdateMailbox "/o=Blakes7/ou=Exchange
Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623";
True
```

```
RepositoryName      : Repository 2
MailboxID            : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=Paul1623
Enabled              : True
CreateSearchFolder   : True
EnableOfflineArchive : False
Email                : Paul1@scutter2000.c2c2000.com
Name                 : Paul1
Sid                  :
StoreDn              : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=Exc
hCasArray.scutter2000.c2c2000.com/cn=Microsoft
Private MDB
```

PolicyObject

The PolicyManager (described in next section) exposes a Policy object. This object represents an ArchiveOne policy and can be used to manipulate policies. The policy object contains the following attributes and methods

- **MailboxRange:** Int - 0 = None, 1 = Selected, 2 = All
- **ServerMailboxes:** Boolean - True = Process mailboxes, False = don't process mailboxes
- **CoupledPSTs:** Boolean - True = Process coupled PSTs, False = don't process coupled psts
- **UncoupledPSTs:** Boolean - True = Process uncoupled PSTs, False = don't process uncoupled psts
- **ShowInAdmin:** Boolean - True = Show policy in Administration console, False = don't show in Administration console
- **PolicyType:** Int - 0 = Mailbox Policy, 1 = Journal Policy, 2 = PST Policy
- **PSTWhereToSearch:** Int - 0 = Search in PSTs associated with selected Mailboxes, 1 = Search in list of selected PSTs, 2 = Search in all mailboxes and PST lists

Methods exposed by the policy object

- **AddMailbox**(mailboxDN) - Adds a mailbox (referenced by its DN) to the policy mailbox list
- **AddPst**(machine,file) - Adds the given PST file to the policy pst list
- **AddSearchCriterion**(property, comparison, value) - Adds the search criterion to the policy
- **ClearMailboxList**() - Clears the current list of mailboxes from the Policy object
- **SetPolicyActionArchive**(repository) - Set the policy action to 'Archive' and optionally set the

repository override (blank string to ignore)

- **SetPolicyActionList()** - Set the policy action to 'List'

Once you have access to a Policy object you can then call any of these methods or access its attributes. As soon as you make a change to this object it is immediately written back ArchiveOne. See the 'Policy Manager' section for examples of how to use this particular object.

PolicyManager

This manager object contains the following methods

- CreatePolicy
- GetCurrentStatusText
- GetPhase1Status
- GetPolicy
- GetRunningStatus
- StartPolicy
- StopPolicy

CreatePolicy

Description: Creates a new policy object in ArchiveOne. The returned object can be access via the attributes and methods mentioned previously.

Return: True if the call was successful, otherwise False

Example:

TestCreatePolicy.ps1

```
Add-Type -Path &quot;C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll&quot;

function TestCreatePolicy($policyName)
{
    $policyManager = new-object
&quot;C2CSystems.ArchiveOne.Policy.PolicyManager&quot;;
    $newPolicy = $policyManager.CreatePolicy($policyName)
    $newPolicy | FL
}
```

```
PS C:\Users\paulj> TestCreatePolicy "Test Policy";
```

```
Summary          : Search in all messages; Click button to define
                   which mailboxes or public folders to process;
                   Mailbox Processing
PolicyActionType  : POLACT_NONE
PolicyType        : MailboxPolicy
MailboxRange      :
ServerMailboxes   :
CoupledPSTs       :
UncoupledPSTs     :
ShowInAdmin       :
MatchIfAnyCriteriaMatch :
ArchiveMailboxes  :
TaskID            :
ObjectIdentifier  :
PstWhereToSearch  : All
LicencedModule    :
DateIndicatingNotSet : 01/01/1980 00:00:00
```

GetCurrentStatusText

Description: Retrieve the current status text of all running policies (also includes status of second phase). You can optionally just retrieve the phase 1 status by passing a boolean value of \$True.

Return: String containing the status text

Example:

TestPolicyGetCurrentStatusText.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSysms.ArchiveOne.Policy.dll";

function TestPolicyGetCurrentStatus($policyName)
{
    $policyManager = new-object
    "C2CSysms.ArchiveOne.Policy.PolicyManager";

    $policyManager.GetCurrentStatusText()
    Write-Host "-----";
    $policyManager.GetCurrentStatusText($True)
}
```

```
PS C:\Users\paulj> TestPolicyGetCurrentStatus
No process started
Status last updated on 16 Nov 2016 at 16:43:35

0 messages queued for archiving on Monday, November 21, 2016 at 11:17:52

Completed repository: Repository 1
Pending 0 messages
Processing 0 messages
Completed 0 messages
Total time taken 0 hours, 0 minutes, 3 seconds

0 messages are queued for other repositories

Status last updated on 22 Nov 2016 at 12:07:29
-----
No process started
Status last updated on 16 Nov 2016 at 16:43:35
```

GetPhase1Status

Description: Gets the current running state of phase 1.

Return: True if phase 1 is running, otherwise False

Example:

TestGetPhase1Status.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll"

function TestGetPhase1Status
{
    $policyManager = new-object
    "C2CSystems.ArchiveOne.Policy.PolicyManager"
    $policyManager.GetPhase1Status()
}
```

```
PS C:\Users\paulj> TestGetPhase1Status
False
```

GetPolicy

Description: Retrieves the policy object given its display name.

Return: Policy object if the object was found, otherwise null

Example:

TestGetPolicy.ps1	
<pre>Add-Type -Path "C:\Program Files (x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll" function TestGetPolicy(\$policyName) { \$policyManager = new-object "C2CSystems.ArchiveOne.Policy.PolicyManager" \$newPolicy = \$policyManager.GetPolicy(\$policyName) \$newPolicy FL }</pre>	
<pre>PS C:\Users\paulj> TestGetPolicy "Test Policy"</pre>	
Summary	<pre>: Search in all messages; Click button to define which mailboxes or public folders to process; Mailbox Processing PolicyActionType : POLACT_NONE PolicyType : MailboxPolicy MailboxRange : ServerMailboxes : CoupledPSTs : UncoupledPSTs : ShowInAdmin : MatchIfAnyCriteriaMatch : ArchiveMailboxes : TaskID : ObjectIdentifier : PstWhereToSearch : All LicencedModule : DateIndicatingNotSet : 01/01/1980 00:00:00</pre>

StartPolicy

Description: Starts the given policy

Return: True if the call was successful, otherwise False

Example:

TestStartPolicy.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll"

function TestStartPolicy($policyName)
{
    $policyManager = new-object
    "C2CSystems.ArchiveOne.Policy.PolicyManager"
    $policyManager.StartPolicy($policyName)
}
```

```
PS C:\Users\paulj> TestStartPolicy "Test Policy"
True
```

StopPolicy

Description: Stops any currently running phase 1 activity immediately.

Return: True if the call was successful, otherwise False

Example:

TestStopPolicy.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll"

function TestStopPolicy
{
    $policyManager = new-object
    "C2CSystems.ArchiveOne.Policy.PolicyManager"
    $policyManager.StopPolicy()
}
```

```
PS C:\Users\paulj> TestStopPolicy
True
```

RepositoryObject

The RepositoryManager (described in next section) exposes a Repository object. This object represents an ArchiveOne repository and can be used to manipulate characteristics of it. The Repository object contains the following attributes

- **ArchiveDir:** String - directory of the archive files
- **IndexDir:** String - directory of the index files
- **TempDir:** String - directory of the repository temporary files
- **ShowInAdmin:** Boolean - True to show this repository in the administration console, otherwise false to hide it
- **RepositoryType:** Int - 0 = normal mailbox policy, 1 = journal policy, 2 = litigation repository

RepositoryManager

This manager object contains the following methods

- CreateRepository
- DeleteRepository
- GetRepository

CreateRepository

Description: Creates a new repository in ArchiveOne

Return: If successful a new Repository object is returned, otherwise null

Example:

TestCreateRepository.ps1

```
Add-Type -Path &quot;C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll&quot;;

function TestCreateRepository($repositoryName)
{
    $repositoryManager = new-object
&quot;C2CSystems.ArchiveOne.Policy.RepositoryManager&quot;;
    $repository = $repositoryManager.CreateRepository($repositoryName)

    $repository | FL
}
```

```
PS C:\Users\paulj> TestCreateRepository &quot;Test Repository&quot;
```

```

      Name                : Test Repository
Summary                  : DirArchives=This value set to enable
      IntegrationFramework repository creation; DirIndex=This
      value set to enable IntegrationFramework repository
      creation; DirTemp=
      RepositoryType      : 0
      RepositoryStoreType : Standard
ArchiveDir               : This value set to enable IntegrationFramework
      repository creation
IndexDir                 : This value set to enable IntegrationFramework
      repository creation
      TempDir             :
      ShowInAdmin          : False
      IsTemporary          : False
      Description          :
      MaxArchiveSize       : 0
      UpdatesDir           :
      Owner                : 0
      Permission           : Public
EffectiveDate            : 01/01/1980 00:00:00
      IsLinked             : False
      DisplayName          : Test Repository
      Size                 : 0
      IndexSize            : 0
      Messages             : 0
      RetentionType        : Undefined
      RetentionCountOfUnits : 3
      HideFromBrowseArchive : False
      ActivityId           : 0
DateIndicatingNotSet     : 01/01/1980 00:00:00

```

DeleteRepository

Description: Deletes the given repository from ArchiveOne.

Return: void

Example:

```
TestDeleteRepository.ps1
```

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestDeleteRepository($repositoryName)
{
    $repositoryManager = new-object
    "C2CSystems.ArchiveOne.Policy.RepositoryManager";
    $repositoryManager.DeleteRepository($repositoryName)
}
```

```
PS C:\Users\paulj> $rm.DeleteRepository("Test Repository");
PS C:\Users\paulj>
```

GetRepository

Description: Retrieves the named repository object from ArchiveOne

Return: A Repository object if the call was successful, otherwise null

Example:

TestGetRepository.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestGetRepository($repositoryName)
{
    $repositoryManager = new-object
    "C2CSystems.ArchiveOne.Policy.RepositoryManager";
    $repository = $repositoryManager.GetRepository($repositoryName)

    $repository | FL
}
```



```
PS C:\Users\paulj> TestGetRepository &quot;Test Repository&quot;
```

```

      Name                : Test Repository
Summary                  : DirArchives=This value set to enable
      IntegrationFramework repository creation; DirIndex=This
      value set to enable IntegrationFramework repository
      creation; DirTemp=
      RepositoryType      : 0
      RepositoryStoreType : Standard
ArchiveDir               : This value set to enable IntegrationFramework
      repository creation
IndexDir                 : This value set to enable IntegrationFramework
      repository creation
      TempDir             :
      ShowInAdmin         : False
      IsTemporary         : False
      Description         :
      MaxArchiveSize      : 0
      UpdatesDir          :
      Owner               : 0
      Permission          : Public
EffectiveDate            : 01/01/1980 00:00:00
      IsLinked            : False
      DisplayName         : Test Repository
      Size                : 0
      IndexSize           : 0
      Messages            : 0
      RetentionType       : Undefined
      RetentionCountOfUnits : 3
      HideFromBrowseArchive : False
      ActivityId          : 0
DateIndicatingNotSet    : 01/01/1980 00:00:00

```

Search Object

The SearchManager (described below) exposes a Search object through the API. This object represents a named ArchiveOne search that you would normally see in the Administration console. The search object contains the following attributes and methods (any changes you make will take immediate effect on the ArchiveOne search object)

- **Words:** String - a list of words that will be matched during the search
- **NotWords:** String - a list of words that must NOT be present in the result set
- **SearchExpired:** Int - 0 = Normal Search, 1 = Normal and Expired, 2 = Expired only
- **SearchDeleted:** Int - 0 = Normal Search, 1 = Normal and Deleted, 2 = Deleted only, 3 = On Hold, 4 = Recycle bin

- **SampleRate:** Int - Rate at which result items will be returned
- **ShowInAdmin:** Boolean - True to show in the administration console, otherwise false to hide it
- **TagBehaviour:** Int - 0 = OR, 1 = AND, 2 = NOT_ANY

Methods exposed by the Search object

- **AddSearchCriterion**(property, comparison, value) - Adds a criterion to the search object
- **AddRepository**(repositoryName) - Adds a repository that the search will use when it executes
- **AddMailbox**(mailboxDN) - Adds to the list of mailboxes that ArchiveOne will limit the search to
- **ClearMailboxList**() - Clears the current list of mailboxes from the Search object
- **AddTag**(tag) - Adds a Tag definition to the current search. Create the TagDefinition object and set its Category/Name properties prior to using this method.
- **SetPolicyActionList**() - Set the policy action to 'List'
- **Summary**() - Retrieve a text summary description of the Search object

See the 'Search Manager' section below for examples of how to use this particular object.

SearchQuery Object

The SearchManager (described below) exposes a SearchQuery object. This object can be used to construct dynamic search requests against ArchiveOne that don't require the construction of a full named search object as mentioned above. The SearchQuery object contains the following attributes and methods

- **Words:** String - a list of words that will be matched during the search
- **NotWords:** String - a list of words that must NOT be present in the result set
- **Folders:** String - the original folder path of the message you would like to search (this can be the partial path or full path)
- **FromUser:** String - set the 'From' user criteria for this search. You can use either SMTP, DN or Display Name. For more accurate search we would recommend either the SMTP or DN
- **ToUser:** String - set the 'To' user criteria for this search. You can use either the SMTP, DN or Display name. For more accurate search we would recommend either the SMTP or DN
- **BeforeDate:** DateTime - matches all messages Before this date
- **AfterDate:** DateTime - matches all messages After this date
- **MailboxOrPublicFolder:** String - Indicate which mailbox or public folder to search. Use * to search all

SearchResult Object

The search methods return an array of SearchResult objects. This object contains a detailed set of attributes describing each result item. Each SearchResult object contains the following attributes

- ZipFilename
- MessageFilenameInZip
- From
- To
- CC
- BCC
- Sent
- SentInLocationTimeZone
- Subject
- StartOfBody
- ArchiveDate
- ArchiveDateInLocalTimeZone
- MessageGuid
- Mailbox
- Folder
- MessageRecordDeleteStatus
- MessageRecordStatus
- MessageCreationTime
- MessageCreationTimeInLocalTimeZone
- Importance
- Sensitivity
- MessageClass
- Size
- Repository
- RepositoryInternalName
- Categories
- AttachmentFilename
- RetrievalURL
- ArchiveProduct

SearchManager

Searching can be performed in two ways. You can dynamically construct a query and have that executed immediately by ArchiveOne. Alternatively you can construct a Search object (see above)

and then run this as a named search within ArchiveOne. The following methods are available

- CreateSearch
- DeleteSearch
- GetSearch
- RunNamedSearch
- GetNamedSearchResultsBatch
- ExecuteQuery

CreateSearch

Description: Creates a new named Search object in ArchiveOne.

Return: If successful a new Search object is returned, otherwise null

Example:

TestCreateSearch.ps1

```
Add-Type -Path &quot;C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll&quot;;

function TestCreateSearch($searchName)
{
    $searchManager = new-object
    &quot;C2CSystems.ArchiveOne.Policy.SearchManager&quot;;
    $search = $searchManager.CreateSearch($searchName)
    $search.Words = &quot;Test&quot;;
    $search.SearchExpired = 2

    $search | FL
}
```

```
PS C:\Users\paulj> TestCreateSearch &quot;TestSearch&quot;;
```

```
Summary          : Message matches <strong>'Test'</strong>; Search in all archived mailboxes
and public folders; <strong>Only include expired items</strong>; Search all repositories; Show 1 in
1; None; Show all matched results
SampleRate       :
Words            :
NotWords         :
SearchExpired    :
SearchDeleted    :
ShowInAdmin      :
TagBehaviour     :
TaskID           :
ObjectIdentifier :
MatchIfAllCriteriaMatch :
RepositorySelectionGroup :
DateIndicatingNotSet : 01/01/1980 00:00:00
```

DeleteSearch

Description: Deletes a given named search from ArchiveOne

Return: void

Example:

TestDeleteSearch.ps1

```
Add-Type -Path &quot;C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSys...ArchiveOne.Policy.dll&quot;;

function TestDeleteSearch($searchName)
{
    $searchManager = new-object
&quot;C2CSys...ArchiveOne.Policy.SearchManager&quot;;
    $search = $searchManager.DeleteSearch($searchName)
}
```

```
PS C:\Users\paulj&gt; TestDeleteSearch &quot;Test Search&quot;;
PS C:\Users\paulj&gt;
```

RunNamedSearch

Description: Executes the named search in ArchiveOne

Return: void

Example:

TestRunNamedSearch.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestRunNamedSearch($searchName)
{
    $searchManager = new-object
    "C2CSystems.ArchiveOne.Policy.SearchManager";
    $search = $searchManager.CreateSearch($searchName)
    $search.Words = "Test";

    $count = 0
    $searchManager.RunNamedSearch($searchName, "Test comment";,
    [ref]$count)
    Write-Host "Search returned:" $count " items";
}
```

```
PS C:\Users\paulj> TestRunNamedSearch "Test Search";
Search returned: 3528 items
```

GetNamedSearchResultsBatch

Description: Retrieves a batch of results from the previously executed named search. The method requires a batch size to be passed in which limits the number of results it returns for each call. Successive calls to this API will return the next batch of results until all items have been retrieved.

Return: void

Example:

```
TestGetNamedSearchResultsBatch.ps1
```

```
Add-Type -Path &quot;C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll&quot;

function TestGetNamedSearchResultsBatch($searchName)
{
    $searchManager = new-object
    &quot;C2CSystems.ArchiveOne.Policy.SearchManager&quot;;
    $count = 0
    $searchManager.RunNamedSearch($searchName, &quot;Test comment&quot;,
    [ref]$count)
    Write-Host &quot;Search returned:&quot; $count &quot; items&quot;;

    $searchResults = $null
    $searchManager.GetNamedSearchResultsBatch(5000,[ref]$searchResults)

    $searchResults[0] | FL
}
```

PS C:\Users\paulj> TestGetNamedSearchResultsBatch "Test Search"
 Search returned: 3528 items

```

    result                                     : AOnePollLib.AOneSearchMessageStruct
ZipFilename                                : c:\repositories\repo2\data\1\A1P160704121126_PAUL2015_P000000.zip
  MessageFilenameInZip                     : 000029_Message_100_sent_04_07_2016_12_11_00_.msg
    From                                   : Paul Jones
    To                                    : {Paul Jones}
    ToNames                              : Paul Jones
    ToNamesForDisplay                    : Paul Jones
    CC                                   : {}
    CCNames                             : 
    CCNamesForDisplay                   : 
    BCC                                 : {}
    BCCNames                           : 
    BCCNamesForDisplay                  : 
    Sent                               : 04/07/2016 11:11:05
    SentInLocalTimeZone                 : 04/07/2016 12:11:05
  Subject                                : Message 100 sent 04/07/2016 12:11:00
StartOfBody                               : This is test message number 100 sent on 04/07/2016 12:11:00. The following
                                         is
                                         random te (Truncated)
    ArchiveDate                         : 04/07/2016 12:11:26
    ArchiveDateInLocalTimeZone          : 04/07/2016 13:11:26
  MessageGuid                           : 0x0641d07a94c89d49a210607dc1cc3ff04d
Mailbox                                  : /o=Blakes7/ou=Exchange Administrative Group
                                         (FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ
    Folder                             : Paul Jones\Inbox
    MessageRecordDeleteStatus           : 0
    MessageRecordStatus                 : Visible
    MessageCreationTime                 : 04/07/2016 11:11:05
    MessageCreationTimeInLocalTimeZone : 04/07/2016 12:11:05
    Importance                         : 1
    Sensitivity                         : 0
    MessageClass                       : IPM.Note
    Size                               : 1391
    SizeInKB                           : 1
    Repository                         : Repository 2
    RepositoryInternalName              : Repository 2
    Categories                         : {}
    AttachmentFilename                 : {}
    AttachmentFileNames                : 
    AttachmentFileNamesForDisplay      : 
    MailboxName                        : Paul Jones
  RetrievalURL                          : ?ID=7AD04106-C894-499D-
A210-607DC1CC3FF0_Repository%20$1$A1P160704121126_PAUL2015
_P000000$000029_Message_100_sent_04_07_2016_12_11_00_
    ArchiveProduct                     : 1
    RetentionType                      : Default
    RetentionCountOfUnits              : -1
    RetentionDate                      : 01/01/0001 00:00:00
    Tags                              : {}
    TagNames                          : 
    SerializedString                   : XXXX0012Repository
200011000110066Sc:\repositories\repo2\data\1\A1P160704121126_PA
UL2015_P000000.zip0050S000029_Message_100_sent_04_07_2016_12_11_00_.msg0011SPaul
Jones0023AXXX0010Paul JonesYYYY0009AXXXYYYYY0009AXXXYYYYY0020D04/07/2016
11:11:050038SMessage 100 sent 04/07/2016 12:11:00 0101SThis is test message
number 100 sent on 04/07/2016 12:11:00. The following is random te
(Truncated)0020D04/07/2016
12:11:260037S0x0641d07a94c89d49a210607dc1cc3ff04d0085S/o=Blakes7/ou=Exchange
Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ0017SPaul
Jones\Inbox0002I00020D04/07/2016
11:11:050002I10002I00009SIPM.Note0005I13910009AXX
XXYYYY0009AXXXYYYYY0002I00011SPaul
Jones0001S0002I50006SInbox0006SInbox0002I10009AXXXYYYYYYYYY

```


ExecuteQuery

Description: Runs a query immediately without using a named search object. The results are passed back as an array of SearchResult objects. This API is intended for small searches as it is limited to a maximum of 1000 results. If your search is likely to return more than this then we recommend you construct a named search (shown above) and use the batch retrieval API call instead.

Return: void

Example:

TestExecuteQuery.ps1

```
Add-Type -Path "C:\Program Files
(x86)\Barracuda\ArchiveOne\SDK\C2CSystems.ArchiveOne.Policy.dll";

function TestExecuteQuery
{
    $searchManager = new-object
    "C2CSystems.ArchiveOne.Policy.SearchManager";

    $searchQuery = new-object
    "C2CSystems.ArchiveOne.Policy.SearchQuery";
    $searchQuery.Words = "Test";
    $searchQuery.MailboxOrPublicFolder = "*";
    $searchQuery.Repository = "Repository 1";

    $searchResults = $searchManager.ExecuteQuery($searchQuery)
    Write-Host "There are " $searchResults.count " result
    items";
    Write-Host "Item 0 = :"
    $searchResults[0] | FL
}
```

```
PS C:\Users\paulj> TestExecuteQuery
There are 1000 result items
Item 0 = :

result : AOnePollLib.AOneSearchMessageStruct
ZipFilename : c:\repositories\repo1\data\1\A1P160907152700_PAUL2015_P000007.zip
MessageFilenameInZip : 000132_Message_2_sent_07_09_2016_15_10_16_.msg
From : Paul Jones
To : {Paul Jones}
ToNames : Paul Jones
ToNamesForDisplay : Paul Jones
CC : {}
CCNames : {}
CCNamesForDisplay : {}
BCC : {}
BCCNames : {}
BCCNamesForDisplay : {}
Sent : 07/09/2016 14:10:17
SentInLocalTimeZone : 07/09/2016 15:10:17
Subject : Message 2 sent 07/09/2016 15:10:16
StartOfBody : This is test message number 2 sent on 07/09/2016 15:10:16. The following is
random text (Truncated)
ArchiveDate : 07/09/2016 15:27:01
ArchiveDateInLocalTimeZone : 07/09/2016 16:27:01
MessageGuid : 0x9a982e6440aa9b4a887f2f98d2dbd6454d
Mailbox : /o=Blakes7/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ
Folder : Paul Jones\Inbox
MessageRecordDeleteStatus : 0
MessageRecordStatus : Visible
MessageCreationTime : 07/09/2016 14:10:17
MessageCreationTimeInLocalTimeZone : 07/09/2016 15:10:17
Importance : 1
Sensitivity : 0
MessageClass : IPM.Note
Size : 1399
SizeInKB : 1
Repository : Repository 1
RepositoryInternalName : Repository 1
Categories : {}
AttachmentFilename : {}
AttachmentFileNames : {}
AttachmentFileNamesForDisplay : {}
MailboxName : Paul Jones
RetrievalURL : ?ID=642E989A-AA40-4A9B-887F-2F98D2DBD645_Repository%201$1$A1P160907152700_PAUL2015_P000007$000132_Message_2_sent_07_09_2016_15_10_16_
ArchiveProduct : 1
RetentionType : Default
RetentionCountOfUnits : -1
RetentionDate : 01/01/0001 00:00:00
Tags : {}
TagNames : {}
SerializedString : XXXX0012Repository 100011000110066Sc:\repositories\repo1\data\1\A1P160907152700_PA
UL2015_P000007.zip000485000132_Message_2_sent_07_09_2016_15_10_16_.msg0011SPaul
Jones0023AXXX0010Paul JonesYYYY0009AXXXYYYY0009AXXXYYYY0020D07/09/2016
14:10:1700365Message 2 sent 07/09/2016 15:10:16 01015This is test message number
2 sent on 07/09/2016 15:10:16. The following is random text
(Truncated)0020D07/09/2016
15:27:01003750x9a982e6440aa9b4a887f2f98d2dbd6454d0085S/o=Blakes7/ou=Exchange
Administrative Group (FYDIBOHF23SPDLT)/cn=Recipients/cn=PaulJ0017SPaul
Jones\Inbox0002I00020D07/09/2016 14:10:170002I10002I000095IPM.Note0005I13990009AXX
XXXX0009AXXXYYYY0002I00011SPaul
Jones000150002I50006SInbox0006SInbox0002I10009AXXXYYYYYYYY
```

© Barracuda Networks Inc., 2024 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.