

5nine EASY Backup API 1.0

Contents

5nine EASY Backup PowerShell API.....	2
Hyper-V Infrastructure objects	2
Add datacenter.....	2
Add cluster	2
Add host	2
Add storage	3
Set datacenter.....	3
Set cluster.....	3
Set host.....	3
Set Storage	3
Remove datacenter	4
Remove cluster.....	4
Remove host	4
Remove Storage	4
List Datacenters.....	4
List Clusters	4
List Hosts	5
List Storages	5
Get Host	5
List Virtual Machines	5
Operations commands.....	5
Backup.....	5
Restore	6
Results commands	7
Get archives.....	7
Get backup jobs.....	7
Get restore jobs.....	7
Get Scheduled Jobs	7
Examples	7

5nine EASY Backup PowerShell API

To add EASY Backup snapin to console, use command:

```
add-pssnapin easyBackupAPI
```

Hyper-V Infrastructure objects

Add datacenter

```
Add-DC -Name <String> [-Login <String>] [-Password <String>]
```

Add datacenter to objects tree.

Name – (mandatory) the name of the object.

Login – (optional) the username under which credentials the object will be queried by EASY Backup service.

Password – (optional) the password of the username.

Add cluster

```
Add-Cluster [-DCName <String>] [-DC <Datacenter>] -Address <String> [-Login <String>] [-Password <String>]
```

Add cluster to objects tree.

DCName – (optional) the name of the datacenter under which object will be placed.

DC – (optional) the DataCenter object, under which object will be placed. One of the parameters DCName or DC must be specified.

Address – (mandatory) the address of the object. This can be FQDN or IP.

Login – (optional) the username under which credentials the object will be queried by EASY Backup service.

Password – (optional) the password of the username.

Add host

```
Add-Host [-DCName <String>] [-DC <Datacenter>] -Address <String> [-Login <String>] [-Password <String>]
```

Add host to objects tree.



DCName – (optional) the name of the datacenter under which object will be placed.

DC – (optional) the DataCenter object, under which object will be placed. One of the parameters DCName or DC must be specified.

Address – (mandatory) the address of the object. This can be FQDN, IP or computer name.

Login – (optional) the username under which credentials the object will be queried by 5nine service.

Password – (optional) the password of the username.

Add storage

Add-Storage -Path <String>

Add storage.

Path – (mandatory) full path to the folder which will be used for storing archives.

Set datacenter

Set-DC -Datacenter <Datacenter>

Update properties of the existed datacenter.

Datacenter – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Datacenter]

Set cluster

Set-Cluster -Cluster <Cluster>

Update properties of the existed cluster.

Cluster – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Cluster]

Set host

Set-Host -Host <Host>

Update properties of the existed host.

Host – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Host]

Set Storage

Set-Storage -Storage <Storage>

Update properties of the existed storage.



Storage – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Storage]

Remove datacenter

Remove-DC -Datacenter <Datacenter>

Delete the specified datacenter.

Datacenter – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Datacenter]

Remove cluster

Remove-Cluster -Cluster <Cluster>

Delete the specified cluster.

Cluster – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Cluster]

Remove host

Remove-Host -Host <Host>

Delete the specified host.

Host – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Host]

Remove Storage

Remove-Storage -Storage <Storage>

Remove the specified storage.

Storage – the object of type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Storage]

List Datacenters

Get-DCs

List available datacenters.

List Clusters

Get-Clusters [-DC <Datacenter>] [-DCName <String>]

List all clusters of the specified datacenter.

DCName – (optional) the name of the datacenter under which object will be placed.

DC – (optional) the DataCenter object, under which object will be placed. One of the properties DCName or DC must be specified.



List Hosts

Get-Hosts [-DC <Datacenter>] [-DCName <String>]

List all hosts of the specified datacenter.

DCName – (optional) the name of the datacenter under which object will be placed.

DC – (optional) the DataCenter object, under which object will be placed. One of the properties DCName or DC must be specified.

List Storages

Get-Storages

List available storages

Get Host

Get-Host -Path <String>

Get the host object by specified path.

Path – the logical path to the host. The path can be in one of the following form:
dcname\hostaddress, or dcname\clusteraddress\nodename

List Virtual Machines

Get-VMs [-Host <Host>] [-HostPath <String>]

List all virtual machines on the specified host.

HostPath – (optional) the logical path to the host. The path can be in one of the following form:
dcname\hostaddress, or dcname\clusteraddress\nodename

Host – (optional) the Host object. One of the properties Host or HostPath must be specified.

Operations commands

Backup

Start-Backup [-Label <String>] [-ObjectsPath <String>] [-Archives <Archive[>]>] [-Storage <Storage>] [-StoragePath <String>] [-Scheduler <Scheduler>]

Create backup job with specified parameters.

Label – the name of backup job.

ObjectsPath – the path to objects that should be backed up in this job. Different objects must be comma separated. Path to objects can end with * - that means include all descendant objects. For example this parameter can be such: `-ObjectsPath dcname\hostname1*, dcname\hostname2\vm1`

Archives – for advanced setting you can manually prepare the array of archives (type [FiveNine.EasyBackup.Interfaces.ArchiveManager.Datacenter]) and specify there hosts and vms you want to include in backup. One of the parameters *ObjectsPath* or *Archives* must be specified.

StoragePath – the path to the folder of storage. If storage with such path is not existed then new Storage object and corresponding folder will be automatically created.

Storage – the Storage object where backup will be placed. One of the parameters *StoragePath* or *Storage* must be specified.

Scheduler – you can specify object of type [FiveNine.EasyBackup.Interfaces.TasksSheduler] to run job delayed or recurrent. See class description below:

```
public class Scheduler
{
    public enum SchedulerType
    {
        Immediatly = 0,
        Delayed,
        Daily,
        Weekly,
        Monthly
    }
    /// <summary>Scheduler type</summary>
    public SchedulerType Type;
    /// <summary> First start time. Also interpreted as time of start for recurrent
job.</summary>
    public DateTime Datetime;
    /// <summary>
    /// Recurrence: recur day, recur month day (32 is last day) or recur week day (Sunday
is 0)
    /// </summary>
    public int EveryCount;
}
```

Restore

Start-Restore [-Host <Host>] [-HostPath <String>] [-VMIds <Guid[]>] [-VMNames <String[]>] [-Archive <Archive>] [-ArchiveLabel <String>]

Initiate restore process.

HostPath – the logical path to host where VMs will be restored.

Host – the host object where VMs will be restored. One of the parameters *HostPath* or *Host* must be specified.

VMIds – the array of identifiers of virtual machines that must be restored.



VMNames – the array of names of virtual machines that must be restored. One of the parameters VMIDs or VMNames must be specified.

Archive – the archive object from which the VMs should be restored.

ArchiveLabel – the name of archive from which the VMs should be restored.

Results commands

Get archives

Get-Archives

Get all archives with backed up VMs.

Get backup jobs

Get-BackupJobs

Get all backup jobs.

Get restore jobs

Get-RestoreJobs

Get all restore jobs.

Get Scheduled Jobs

Get-ScheduledJobs

Get active scheduled jobs.

Note: For other operations with scheduled tasks you can use directly TaskScheduler PowerShell module from Microsoft *PowerShell Pack*.

Examples

Add DC object: ***add-dc -name testdc***

Check it appeared in list: ***get-dcs***

Create host object under specified dc: ***add-host -dcname testdc -address ohdev03***

Check host appeared: ***get-hosts -dcname testdc***

Get list of VMs under host: ***get-vms -hostpath testdc\ohdev03***

Create backup with label 'testcorebackup' for host 'testdc\ohdev03' for several vms and store it in



specified folder: ***start-backup -label testcorebackup -objectspath testdc\ohdev03\VM1, testdc\ohdev03\VM2 -storagepath e:\temp***

Get list of currently running backup jobs: ***get-backupjobs | where {\$_.Status -eq 0}***

Get list of finished backup jobs: ***get-backupjobs | where {\$_.Status -eq 1}***

Get list of failed backup jobs: ***get-backupjobs | where {\$_.Status -eq 2}***

Get archives by label: ***get-archives | where {\$_.Label -eq "testcorebackup"}***

Start restore for specified vms from archive with label 'testcorebackup' on specified host 'testdc\ohdev03': ***start-restore -archivelabel testcorebackup -hostpath testdc\ohdev03 -vmnames VM1, VM2***

Get list of currently running restore jobs: ***get-restorejobs | where {\$_.Status -eq 0}***