Arcserve® Replication and High Availability

PowerShell Commands Guide

r16.5

arcserve®

Pre-release Document, only for reference

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by Arcserve at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Arcserve. This Documentation is confidential and proprietary information of Arcserve and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and Arcserve governing your use of the Arcserve software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and Arcserve.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Arcserve copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Arcserve that all copies and partial copies of the Documentation have been returned to Arcserve or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, ARCSERVE PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL ARCSERVE BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF ARCSERVE IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is Arcserve.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

© 2015 Arcserve, including its affiliates and subsidiaries. All rights reserved. Any third party trademarks or copyrights are the property of their respective owners.

Arcserve Product References

This document references the following Arcserve products:

- Arcserve® Replication
- Arcserve® High Availability (HA)
- Arcserve® Assured Recovery®
- Arcserve® Content Distribution

Contact Arcserve

The Arcserve Support team offers a rich set of resources for resolving your technical issues and provides easy access to important product information.

https://www.arcserve.com/support

With Arcserve Support:

- You can get in direct touch with the same library of information that is shared internally by our Arcserve Support experts. This site provides you with access to our knowledge-base (KB) documents. From here you easily search for and find the product-related KB articles which contain field-tested solutions for many top issues and common problems.
- You can use our Live Chat link to instantly launch a real-time conversation between you and the Arcserve Support team. With Live Chat, you can get immediate answers to your concerns and questions, while still maintaining access to the product.
- You can participate in the Arcserve Global User Community to ask and answer questions, share tips and tricks, discuss best practices and participate in conversations with your peers.
- You can open a support ticket. By opening a support ticket online, you can expect a callback from one of our experts in the product area you are inquiring about.

You can access other helpful resources appropriate for your Arcserve product.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Updated to include user feedback, enhancements, corrections, and other minor changes to help improve the usability and understanding of the product or the documentation itself.

Contents

Chapter 1: Getting Started

This section contains the following topics:

About This Guide

This Guide contains all of the necessary information for running and using Arcserve RHA PowerShell commands. It provides a brief overview of Windows PowerShell, describes each Arcserve RHA PowerShell command, and gives instructions and examples on how to use these commands for controlling, editing and monitoring the DR and HA processes.

Related Documentation

Use this Guide along with the following Guides:

- *Arcserve RHA Installation Guide*
- *Arcserve RHA Administration Guide*

For more information about using Windows PowerShell, refer to the documentation pack that comes with PowerShell installation package, or download it from Microsoft Download Center.

Understanding Arcserve RHA PowerShell Commands

Arcserve RHA PowerShell is offered to users as an alternative or a supplement to managing the replication process using the Arcserve RHA Manager's graphic user interface (GUI). It enlarges and facilities the capabilities of the WS CLI that was provided in previous versions, and it supports both DR and HA operations.

Windows PowerShell™ is a new Windows command-line shell and scripting environment designed especially for system administrators. The shell includes an interactive prompt and a scripting environment that can be used independently or in combination. Unlike most shells, which accept and return text, Windows PowerShell is built on top of the .NET common language runtime (CLR) and the .NET Framework, and accepts and returns .NET objects.

Windows PowerShell™ comes with a large set of built-in commands with a consistent interface. Arcserve RHA PowerShell is based on the standard Windows PowerShell™, while adding to it a number of scenario-related-commands, called snap-ins. These snap-ins, which allow you to configure a replication scenario and control and monitor the replication and switchover processes, are described in this Guide. All the scenarios that are managed by Arcserve RHA PowerShell commands look and operate exactly as the ones that are managed by the Arcserve RHA Manager, and they are automatically saved in the same default location: *INSTALL_DIR/ws_scenarios*

PowerShell Concepts

PowerShell Cmdlets

Windows PowerShell introduces the concept of a cmdlet ("command-let"). A cmdlet is a simple, single-function command-line tool built into the shell, whose aim is to manipulate objects. You can recognize cmdlets by their name format: a verb and noun separated by a dash (-), such as Get-Help, Get-State and Run-Scenario. The verbs express specific actions in Windows PowerShell, while the nouns describe specific types of objects.

In Windows PowerShell, most cmdlets are very simple, and they are designed to be used in combination with other cmdlets. For example, the "get" cmdlets only retrieve data, the "set" cmdlets only establish or change data, the "format" cmdlets only format data, and the "out" cmdlets only direct the output to a specified destination.

PowerShell cmdlets have common parameters, which are not described in this Guide. To get more information about the common parameters, enter:

```
get-help about_commonparameters
```

PowerShell cmdlets can have mandatory and optional parameters. If a mandatory parameter is missing, you will be prompted to enter it. If an optional parameter is missing, PowerShell uses the default value.

Object Pipelines

Windows PowerShell provides a new interactive model that is based on objects, rather than text. One major advantage of using objects is that it makes it much easier to pipeline commands, that is, to pass the output of one command to another command as an input.

The command that receives an object can act directly on its properties and methods without any conversion or manipulation. You can refer to properties and methods of the object by name, rather than calculating the position of the data in the output.

In the following example, the result of a Get-Scenario command is passed to a Get-Hosts command. The pipeline operator (|) sends the result of the command on its left to the command on its right, and the output is sent to a Format-Table command.

```
PS> Get-Scenario "File Server*" | Get-Hosts | FT -AUTO

Scenario     Name        Role   Parent       State   IP          Port

--------     ----        ----   ------       -----   --          ----

File Server 1 192.168.1.152 Master  --          Running 192.168.1.152 25000

File Server 1 192.168.1.153 Replica 192.168.1.152 Running 192.168.1.153 25000

File Server   192.168.1.152 Master  --          Stopped 192.168.1.152 25000

File Server   192.168.1.153 Replica 192.168.1.152 Stopped 192.168.1.153 25000
```

Install Arcserve RHA PowerShell

To use Arcserve RHA PowerShell, you need to install Windows PowerShell and Arcserve RHA snap-ins.

For detailed information about the requirements and installation of Windows PowerShell and Arcserve RHA snap-ins, refer to *Arcserve RHA Installation Guide*.

Important! The Arcserve RHA PowerShell and the Arcserve RHA Control Service to which it is connected must have the same version.

Run Arcserve RHA PowerShell

After the installation of Windows PowerShell and Arcserve RHA snap-ins, you can run Arcserve RHA PowerShell from two places:

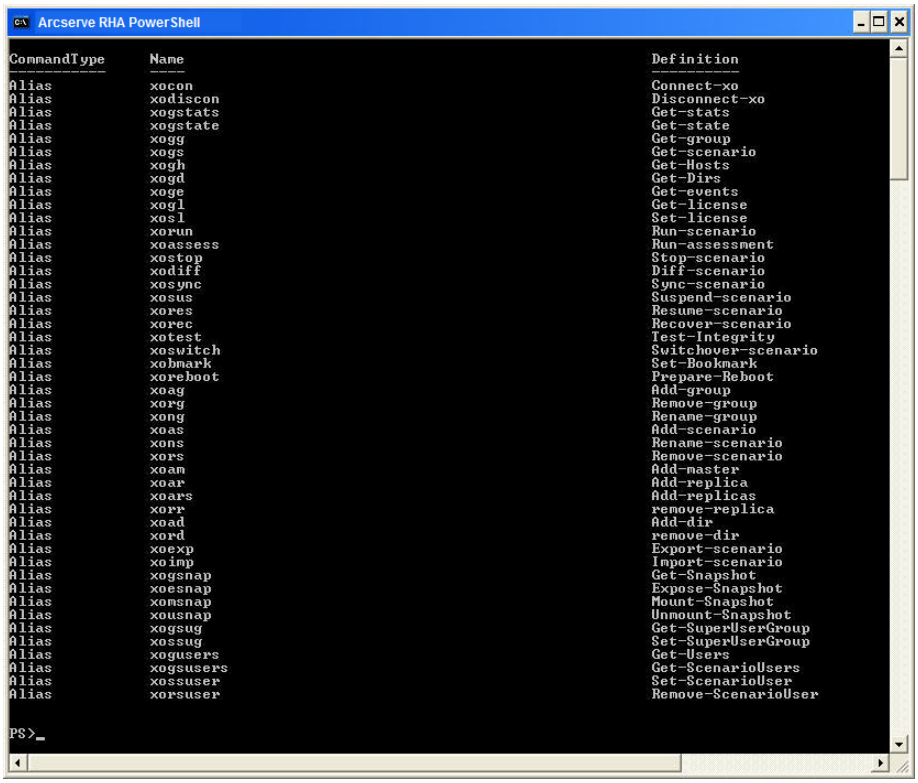
- Arcserve RHA PowerShell shortcut - when using this option, you can immediately start working with Arcserve RHA PowerShell snap-ins.
- Windows PowerShell shortcut - when using this option, you need to manually add Arcserve RHA PowerShell snap-ins to Windows PowerShell. (See below.)

Important! When running some commands in PowerShell, a system error occurs if you set different passwords on the Master, Replica and Control Service hosts and some operations may fail to run or complete. Use the same administrator password on all hosts to avoid this issue.

To run Arcserve RHA PowerShell from Arcserve RHA PowerShell shortcut:

1. Open Arcserve RHA PowerShell by selecting **Start, Programs, arcserve RHA**, **PowerShell**.

   Once you open Arcserve RHA PowerShell, the following window is displayed, listing all Arcserve RHA PowerShell snap-ins:



   Now, you need to connect to the Control Service that manages your Arcserve RHA operations. To perform this, use the Connect-XO command (see page 18).

To run Arcserve RHA PowerShell from Windows PowerShell shortcut:

1. Open Windows PowerShell by selecting **Start**, **Programs**, **Windows PowerShell 1.0**, **Windows PowerShell**.

   The Windows PowerShell window is displayed.

2. Enter the following command to change the working directory to your Arcserve RHA PowerShell Snapin INSTALLDIR:

   ```
   CD 'INSTALLDIR\Powershell Snapin'
   ```

   The directory changes.

3. Enter the following command to install Arcserve RHA PowerShell snap-ins:

   ```
   .\xo.ps1
   ```

   The Arcserve RHA PowerShell snap-ins are installed, and you can start using them to connect to the Control Service that manages your Arcserve RHA operations.

Using Help

There are several ways to get help and additional information in PowerShell:

Help for a specific command

- The Help parameter - when you specify the **-?** parameter to any command, the command is not executed. Instead, Windows PowerShell displays help for the command. The syntax is:

  <command_name> -?

- To display the type and syntax of a command, enter:

  get-command <command_name>

- Each command has a detailed help file. To access the help file, enter:

  get-help <command_name> -detailed

  The detailed view of the command help file includes a description of the command, the command syntax, descriptions of the parameters, and example that demonstrate the use of the command.

- To display help for a parameter in a command, after the parameter prompt enter **!?**:

  <parameter_name>: !?

List of available commands

- To display a list of available Windows PowerShell commands, enter:

  get-command

- To display a list of available Arcserve RHA PowerShell snap-in commands, enter:

  get-command | where {$_.DLL -match "XO"} | format-table

- To display a list of all aliases defined for XO commands, type:

  alias xo*

Arcserve RHA PowerShell commands verification

- To verify the installation of Arcserve RHA PowerShell snap-ins, enter the following command and look for Arcserve RHA PowerShell snap-ins:

  get-pssnapin

Formatting Command Output

In Windows PowerShell, there are several commands that enable you to change the output view:

- Format-List
- Format-Custom
- Format-Table
- Format-Wide

To change the format of the output from any command, use the pipeline operator (|) to send the output of the command to a Format command.

For example, the following command sends the output of a Get-Scenario command to the Format-Table command. As a result, the data is formatted as a table:

```
PS>get-scenario |Format-table

ID         Group     Name          Type       Master       State   Sync  AR

--         -----     ----          ----       ------       -----   ----  --

1123633468 Scenarios File Server 1   FileServer 192.168.1.152 Running File  False

1123633497 Scenarios Exchange Server Exchange   192.168.1.152 Running Block True

1123633852 Scenarios File Server 3   FileServer              Unknown File  False

3848963840 Scenarios File Server     FileServer 192.168.1.152 Stopped File  False

3848982942 Scenarios File System 1   FileServer QA99-W2K3-EX8 Running File  False
```

For more details, use the following commands to read the help for the Format commands:

```
get-help format-list

get-help format-table

get-help format-wide

get-help format-custom
```

Chapter 2: Using Arcserve RHA PowerShell Commands

This chapter describes in detail how to use Arcserve RHA PowerShell commands to control, edit and monitor the Replication and HA processes. The commands are displayed in alphabetical order and they are divided into four groups: Connecting and Registration, Controlling, Editing and Monitoring.

This section contains the following topics:

Connecting and Registration Commands

This section describes how to connect to the Control Service, how to disconnect from it, and how to enter your license key for Arcserve RHA registration.

Connect-XO - Connect PowerShell to a Control Service

In order to work with Arcserve RHA scenarios using PowerShell, the first thing you need to do is connect to the Control Service that acts as the point-of-control of the Arcserve RHA operation. The **Connect-XO** command enables you to connect PowerShell to a specific Control Service.

Note: Once you finished working with Arcserve RHA PowerShell, do not forget to disconnect from the Control Service using the Disconnect-XO command. (see page 19) Closing PowerShell window will also cause PowerShell to disconnect from the Control Service.

Syntax

```
Connect-XO [-Host] <String> [-Credentials] <PSCredential> [[-Protocol] [<String>]] [[-Port] [<String>]]
```

Parameters

Host

The IP address or hostname of the machine where the Control service is running.

Credentials\PSCredentials

The Domain\User Name for the Control Service. These credentials must belong to a user who has Admin rights on the Control Service. After you enter the credentials, a **Windows PowerShell Credential Request** dialog appears, prompting you to enter your password.

Note: To avoid the need to manually enter your credentials into the **PSCredentials** dialog, refer to Connect PowerShell to a Control Service using a Script.

Protocol

The protocol that is used for connecting to the Control Service. Enter one of the following: **http** or **https**.

Port (optional)

The TCP/IP port that is used for connecting to the Control Service. For **http** the default value is **8088**; For **https** the default value is **443**.

Example: Connect to a Control Service

```
connect-xo 192.168.1.151 qa88-w3k3\administrator https
```

Outcome

A **Windows PowerShell Credential Request** dialog appears, prompting you to enter your password. Then, the following appears:

```
Connecting...
192.168.1.151 connected!
```

Connect PowerShell to a Control Service using a Script

You can avoid the need to manually enter your credentials into the **PSCredentials** dialog by encrypting your password and running it as an object.

To encrypt your password and run it as an object

Enter the following commands, using your password where indicated, and run it once:

```
read-host -assecurestring | convertfrom-securestring | out-file C:\ securestring.txt <password>

$pass = cat C:\securestring.txt | convertto-securestring

$mycred = new-object -typename System.Management.Automation.PSCredential -argumentlist <domain\user_name>, $pass

Connect -XO [-Host] <String> $mycred [[-Protocol][<String>]] [[-Port] [<String>]]
```

The outcome is the same as if you used a standard connection:

```
Connecting...

<IP Address> connected!
```

For more information, refer to the PowerShell documentation or search the internet.

Disconnect-XO - Disconnect from a Running Control Service

After you finished working with Arcserve RHA PowerShell, you need to disconnect from the running Control Service. The **Disconnect-XO** command enables you to disconnect PowerShell from the running Control Service.

Note: Closing the PowerShell window also causes PowerShell to disconnect from the Control Service.

Syntax

```
Disconnect-XO
```

Note: This command does not have parameters. It automatically disconnects the running Control Service.

Example: Disconnect from a Control Service

```
disconnect-xo
```

Outcome:

```
192.168.1.151 disconnected!
```

Get-License - Display your Arcserve RHA License

The **Get-License** command enables you to display your Arcserve RHA license details.

Syntax

get-license

Example: Display your Arcserve RHA license details

get-license

Outcome:

Key: TVC2LF24FTU7G3WJ2QAFMCLGXA5KLPCCYIXTJTWX2MOZFU5GL7EJ3OYZQND7V3G123456

Company:

License expires on: 11 2009

Maintenance till: 11 2009

Number of Assured Recovery nodes:240

Number of CDP Repository nodes:240

Product list:

- Application Server, Windows Cluster edition, 30 instances HA

- File server, Windows Enterprise edition, 130 instances HA

- Application Server, Windows Enterprise edition, 130 instances DR

- File server, Windows Enterprise edition, 30 instances DR

- Application Server, Virtual Machine, 100 instances DR

- Application Server, Virtual Machine, 100 instances HA

Set-License - Register Arcserve RHA

The **Set-License** command enables you to register Arcserve RHA using a license key. You need to have a valid registration key before using this command.

Syntax

set-license

Parameters

Key

A valid license key.

Example: Register Arcserve RHA using a license key

set-license
TVC2LF24FTU7G3WJ2QAFMCLGXA5KLPCCYIXTJTWX2MOZFU5GL7EJ3OYZQND7V3G123456

Outcome:

Key registered successfully

xo-import-credential

This command reads all credential records in the given XML file and adds them to the connected Control Service, via the command Add-Credential.

Syntax

xo-import-credential

Parameters

Specify the XML file name.

Input

None. No object is piped to xo-import-credential.

xo-convertto-securefile

This command converts a plain text CVS file to a secured XML file.

Syntax

xo-convertto-securefile

Parameters

Source file name and destination are specified.

The source should be a CSV file which has the following format:

hostname,	username,	password
host1,	user1,	pwd1
host2,	user2,	pwd2

Input

None. No object is piped.

xo-credential - Convert a String to a PSCredential Object

The xo-credential command lets you convert a username and password string to a secured credential object, for use within other commands that take PSCredential objects as command arguments.

Syntax

xo-credential <username> <password>

Example

xo-credential johnsmith mypword2

Add-Credential - Add Credentials to a Host

The Add-Credential command lets you add credentials to host.

Syntax

Add-Credential [-Credentials] <PSCredential> [-Host] <String> [[-Port] <UInt32>]]

Parameters

Credentials

The PowerShell credential object of the host. You can create this credential object using xo-credential cmdlet.

Host

The host name or IP address of the host on which you want to apply the credentials to.

Port

The port number of the host.

Default: 25000.

Example:

```
$c = xo-credential "administrator" "Password";

Add-Credential $c "9.182.102.229" 25000
```

Set-HostUserCredential - Set the User Credentials Property for a Host

The Set-HostUserCredential command lets you set the user credentials property of a host.

Syntax

Set-HostUserCredential [-Name] <String> [-Host] <String> [-Credentials] <PSCredential>

Parameters

Name

The scenario name.

Host

The host name or IP address of the host for which you want to set the credentials.

Credentials

The PowerShell credential object of the specific host. You can set this credential object using xo-credential cmdlet.

Example:

```
$c = xo-credential "administrator" "Password";

Set-HostUserCredential -name "scenario 1" -host 9.182.102.229 -credential $c
```

Output:

Property updated successfully.

Set-ScenarioUserCredential - Set the User Credentials Property of a Scenario

The Set-ScenarioUserCredential command lets you set the user credentials property of a scenario.

Syntax

```
Set-ScenarioUserCredential [-Name] <String> [-Credentials] <PSCredential>
```

Parameters

Name

The scenario name.

Credentials

The PowerShell credential object of the host. You can create this credential object using xo-credential cmdlet.

Example:

```
$c = xo-credential "administrator" "Password";

Set-ScenarioUserCredential -name "scenario 1" -credential $c
```

Output:

Property updated successfully

Controlling Commands

This section describes Arcserve RHA PowerShell commands that enable you to control the Replication and HA processes.

Diff-Scenario - Generate a Difference Report

The **Diff-Scenario** command enables you to generate a Difference Report for a given scenario.

Important! We do not recommend initiating a Difference Report when data is being updated on the Master, since all updates that are not yet applied to the Replica will be shown as different.

Syntax

Diff-Scenario [-Name] <String> [-Mode] <String> [-Ignore] <Boolean>

Parameters

Name

The name of the scenario for which you want to generate the report. You can enter several scenario names by using the Get-Scenario command. (see page 73)

Mode

The synchronization mode. Enter one of the following:

B=Binary

F=File

Ignore

Ignore files of the same name and size during the data comparison. Enter one of the following:

1=Yes

0= No

Note: To view the Difference Report after its generation, open the Report Center from the Overview Page, and select the required report.

Example: Generate a Difference Report

diff-scenario "File Server 1" F 1

Outcome:

Differences report is running for scenario File Server 1...
Done!

Export-Scenario - Export a Scenario to a Specified Location

The **Export-Scenario** command enables you to export scenarios to other locations in order to reuse them. The scenario is exported as an XMC file, and you can specify its location.

Syntax

Export-Scenario [-Name] <String> [[-File] [<String>]]

Parameters

Name

The scenario name.

File (optional)

The full path of the exported file. If you do not specify a path, the file is exported to the current directory and carries the name of the scenario with an .xmc extension.

Example: Export a scenario to a specified location

export-scenario "File Server 1" C:\Scenarios\Scenario_exp_file_1

Outcome:

Scenario File Server 1 exported successfully to C:\Scenarios\Scenario_exp_file_1

Expose-Snapshot - Expose a Snapshot

The **Expose-Snapshot** command enables you to expose a snapshot. You can either expose the snapshot as a local read-only folder by mounting it on an unused folder, or expose it as a local read-only volume by mounting it on an unused drive letter.

Notes:

- An exposed snapshot remains exposed through subsequent boots. Dismounting an exposed snapshot releases it without losing the snapshot itself.
- The Expose and Mount actions produce the same result - mounting a snapshot to a certain path. The difference between them is that when you want to mount a snapshot for the first time, you cannot use the Mount action directly; you must use the Expose action. The Expose action both exposes and mounts the snapshot. Then, you can use the Unmount and Mount actions.

Syntax

```
Expose-Snapshot [-Name] <String> [-Index] <Int32> [-Path] <String> [-Port] <String>
```

Parameters

Name

The name of the host whose snapshot you want to expose.

Index

The index no. of the snapshot, as returned by the Get-Snapshot command (see page 74).

Path

The path under which you want to expose the snapshot. The path can be either a drive letter or a full folder path.

Port (Optional)

The port that is used for connecting to the given host. The default port is **25000**.

Example: Expose a snapshot as a local read-only volume

```
Expose-Snapshot 192.168.1.153 0 E: 25000
```

Outcome:

```
Snapshot {97127d0b-f1c9-4db5-943d-96c39b712fe6} mounted as E:
```

Import-Scenario - Import a Scenario to the Manager

The **Import-Scenario** command enables you to import a scenario, in the form of an XMC file, from a specified location. Use this option if you want to relocate scenarios from one Control Service to another, or if you want to use older scenarios that were kept on your system.

Syntax

```
Import-Scenario [-File] <String>
```

Parameters

File

The full path of the imported scenario file.

Notes:

- If a scenario with the same name already exists, the imported scenario will be renamed.
- All imported scenarios are stored in the default **Scenarios** group.

Example: Import a scenario from a specified location to your Manager

```
import-scenario c:\scenarios
```

Outcome:

```
Scenario File Server 2 imported successfully from c:\scenarios
```

Mount-Snapshot - Mount a Snapshot

The **Mount-Snapshot** command enables you to mount an exposed snapshot. You can either mount the snapshot as a local read-only folder on an unused folder, or mount it as a local read-only volume on an unused drive letter.

Syntax

Mount-Snapshot [-Name] <String> [[-Index] [<Int32>]] [[-Path] [<String>]] [[-Port] [<String>]]

Parameters

Name

The name of the host whose snapshot you want to mount.

Index

The index number of the snapshot, as returned by the Get-Snapshot command (see page 74).

Path

The path under which you want to expose the snapshot. The path can be either a drive letter or a full folder path.

Port (Optional)

The port that is used for connecting to the given host. The default port is **25000**.

Example: Mount a snapshot as a local read-only volume

mount-snapshot 192.168.1.153 0 F:

Outcome:

Snapshot {745d6ce9-d880-40bf-a0cb-d4f0114bb0f8} mounted as F:

Prepare-Reboot - Prepare a Host for Maintenance

The **Prepare-Reboot** command enables you to perform maintenance procedures, such as rebooting a host or moving groups between Microsoft Cluster nodes, without performing re-synchronization once these processes are completed.

The hosts that can be prepared for maintenance need to participate in running scenarios. The preparation is done on one host at a time, but this host can participate in multiple scenarios. In these scenarios, the host can function as both the Master and the Replica. When a host participates in a scenario that is not running, the preparation that relates to this scenario does not occur.

After you receive the message informing you that the host is preparing for reboot, you can reboot your host or switch groups between cluster nodes. Once you completed your maintenance procedures, the replication process automatically resumes, without performing re-synchronization.

Note: If after preparing the host for maintenance, you decided not to reboot it and continue running its scenarios, you need to stop the scenarios and re-run them.

Syntax

```
Prepare-Reboot [-Name] <String>
```

Parameters

Name

The host name.

Example: Prepare a Replica host for reboot

```
Prepare-Reboot QA95-W2K3-EX2
```

Outcome:

```
Host QA95-W2K3-EX2 Preparing for reboot
```

Recover-Scenario - Recover Lost Data from the Replica to the Master

The **Recover-Scenario** command enables you to recover data that was lost on the Master by transferring it from any of the Replica hosts that participate in a scenario. This is done by activating a synchronization process in the reverse direction: from a Replica to the Master. When you activate the **Recover-Scenario** command, you need to define from which Replica host you want to recover the data, and whether to delete data already on the Master but not on the Replica during the recovery process.

Important! You must stop replication in order to initiate recovery.

To verify that the recovery process is completed, use the Get-Events command (see page 69). After you get a message informing you that the "Recovery process has finished", you can restart the replication process from the Master to the Replica by using the Run-Scenario command (see page 35).

Syntax

Recover-Scenario [-Name] <String> [-Host] <String> [-Mode] <String> [-Ignore] <Boolean> [-RemoveMasterFiles] <Boolean> [-RecoveryMode] <String> [-RebootAfterRecovery] <Boolean>

Parameters

Name

The scenario name.

Host

The Replica host from which you want to recover data.

Mode

The synchronization mode. Enter one of the following:

B=Binary

F=File

Ignore

Ignore files of the same name and size during the data comparison. Enter one of the following:

1=Yes

0= No

RemoveMasterFiles

Whether to delete files that exist only on the Master during the recovery process. Enter one of the following:

1 = Yes, delete files that exist only on the Master

0 = No, keep files that exist only on the Master

RecoveryMode

The data type to recover. Enter one of the following:

A = Application data

S = System State data (only if the **System State Protection** option is active)

B = Both types of data

The default value is **A**.

RebootAfterRecovery

Whether to reboot the Master host once the recovery process is completed. Enter one of the following:

1 = Yes, reboot the Master

2 = No, do not reboot the Master

Example: Recover lost data

```
Recover-Scenario "File Server 1" 192.168.1.153 F 1 0 A 2
```

Outcome:

```
Recover application data process started
```

Resume-IsAliveCheck - Resume IsAlive Checking of a Running Scenario

The **Resume-IsAliveCheck** command lets you manually resume IsAlive checking for a given running HA scenario.

Syntax

```
Resume-IsAliveCheck [-ScenarioName] <String>
```

Parameters

ScenarioName

The target scenario name.

Example: Resume IsAlive checking of SQLscenario

```
Resume-IsAliveCheck SQLscenario
```

Outcome:

```
Resumes periodic is-alive checking of SQLscenario.
```

Resume-Scenario - Resume Replication on a Suspended Replica

The **Resume-Scenario** command enables you to resume the replication process on a suspended Replica host. After the replication resumes, the accumulated changes are transferred and applied to the Replica without any need to perform a full re-synchronization of the data.

Syntax

Resume-Scenario [-Name] <String> [-Host] <String>

Parameters

Name

The scenario name.

Host

The name of the suspended Replica host that you want to resume.

Example: Resume the replication process on a suspended Replica

resume-scenario "File Server 1" 192.168.1.153

Outcome:

Scenario File Server 1 resumed on 192.168.1.153

Run-Scenario - Start a Scenario

The **Run-Scenario** command enables you to start one or several scenarios.

Syntax

Run-Scenario [-Name] <String> [-Mode] <String> [-Ignore] <Boolean>

Parameters

Name

The scenario name. You can enter several scenario names by using the Get-Scenario command (see page 73).

Mode

The synchronization mode. Enter one of the following:

B=Binary

F=File

V=Volume

Ignore

Ignore files of the same name and size during the data comparison. Enter one of the following:

1=Yes

0= No

Notes:

- To check if the operation is completed successfully, use the Get-Scenario (see page 73) and Get-Events (see page 69) commands.
- To run several scenarios at once, use the Get-Scenario command (see page 73):

  Get-Scenario |Run-Scenario

Example: Start a scenario

run-scenario "File Server 1" F 1

Outcome:

Scenario File Server 1 Starting...

Run-Assessment - Run a Scenario in Assessment Mode

The **Run-Assessment** command enables you to assess the accurate bandwidth usage and compression ratio benchmarking that is needed for replication, without actually replicating data. When you run this command, no replication occurs but statistics are gathered. A report is provided once the assessment process is stopped.

Important! Do not forget to stop the scenario that runs in Assessment Mode after the period you wanted to assess has passed, by using the Stop-Scenario command (see page 38).

Note: To view the Assessment Report after its generation, open the Report Center from the Overview Page, and select the required report.

Syntax

Run-Assessment [-Name] <String>

Parameters

Name

The scenario name.

Example: Run a scenario in Assessment Mode

run-assessment "File Server 1"

Outcome:

Scenario File Server 1 executed successfully

Set-Bookmark - Set a Rewind Bookmark

A bookmark is a checkpoint that is manually set to mark a state that you may want to rewind back to. The **Set-Bookmark** enables you to set a bookmark for a given scenario. Bookmarks are set in real-time, and not for past events. We recommend setting a bookmark just before any activity that may cause data to become unstable.

Notes:

- You can use this option only if you set the Replica Properties list the **Recovery - Data Rewind** option to On.
- You cannot set bookmarks during the synchronization process.

Syntax

```
Set-Bookmark [-Name] <String> [[-Message] <String>]
```

Parameters

Name

The name of the scenario.

Message (Optional)

The name of the bookmark. The default name includes the date and time of the bookmark setting.

Note: We recommend giving a meaningful name to the bookmark that will later help you recognize it.

Example: Set a rewind bookmark

```
set-bookmark "File Server 1" Backup1
```

Outcome:

```
Scenario File Server 1: Rewind bookmark set successfully
```

Stop-Scenario - Stop a Scenario

The **Stop-Scenario** command enables you to stop one or several scenarios.

Note: To check if the operation was completed successfully, use the Get-Scenario (see page 73) and Get-Events (see page 69) commands.

Syntax

```
Stop-Scenario [-Name] <String>
```

Parameters

Name

The name of the scenario you want to stop. You can enter several scenario names by using the Get-Scenario command (see page 73).

Example: Stop a scenario

```
stop-scenario "File Server 1"
```

Outcome:

```
Scenario File Server 1 stopped
```

Suspend-IsAliveCheck - Suspend IsAlive Checking of a Running Scenario

The **Suspend-IsAliveCheck** command lets you manually suspend IsAlive checking for a given running HA scenario.

Syntax

```
Suspend-IsAliveCheck [-ScenarioName] <String>
```

Parameters

ScenarioName

The scenario name.

Example: Suspend IsAlive checking of SQLscenario

```
Suspend-IsAliveCheck SQLscenario
```

Outcome:

```
Suspends periodic is-alive checking of SQLscenario.
```

Suspend-Scenario - Suspend Updates on a Replica

The **Suspend-Scenario** command enables you to temporarily cease delivering changes to a suspended Replica. During the suspension, changes are accumulated in a spool until replication is resumed so that re-synchronization is not required.

Important! It is imperative that during suspension, you do nothing on the Replica that causes the data to change in any way, including starting an application such as Exchange Server, SQL Server, or Oracle. If you need to start programs that will change data on the Replica, you may use the Assured Recovery option (see page 42).

Notes:

- You cannot suspend replication during synchronization. You can suspend replication only temporarily, since changes are accumulated in the spool directory of the Master or upstream Replica. Make sure that sufficient disk space is available for the spool to hold the changes during the time the Replica is suspended.
- To end the suspension, use the Resume-Scenario command (see page 34).

Syntax

```
Suspend-Scenario [-Name] <String> [-Host] <String>
```

Parameters

Name

The scenario name.

Host

The Replica host that you want to suspend.

Example: Suspend updates on a Replica

```
suspend-scenario "File Server 1" 192.168.1.153
```

Outcome:

```
Scenario File Server 1 Suspended on 192.168.1.153
```

Switchover-Scenario - Perform a Switchover

The **Switchover-Scenario** command enables you to start the switchover process for a given HA scenario. To switch back the roles between the Master and the Replica, use the **Switchover-Scenario** command again.

Syntax

Switchover-Scenario [-Name] <String> -WaitJournal [<Boolean>] -run_reverse_scenario <string>

Parameters

Name

The scenario name.

WaitJournal

Whether to wait for journals.

0: Do not wait for journals. You cannot run the reverse scenario if you choose 0.
1: Wait for journals.

Run_reverse_scenario

Specifies whether to automatically run the reverse scenario.

Value 1 or True automatically runs the reverse scenario.

Value 0 or False do not automatically run the reverse scenario.

Example: Perform switchover

Switchover-Scenario "SQL Server 1" 1

Outcome

Scenario SQL Server 1 switching over to <Hostname>.

Done!

Sync-Scenario - Initiate a Synchronization

The **Sync-Scenario** command enables you to synchronize the Master and the Replica of a given scenario. The synchronization process can be manually activated at any time, whether replication is running or not.

Syntax

Sync-Scenario [-Name] <String> [-Mode] <String> [-Ignore] <Boolean>

Parameters

Name

The scenario name. You can enter several scenario names by using the Get-Scenario command (see page 73).

Mode

The synchronization mode. Enter one of the following:

B=Binary

F=File

V=Volume

Ignore

Ignore files of the same name and size during the data comparison. Enter one of the following:

1=Yes

0= No

Example: Initiate a synchronization

sync-scenario "File Server 1" F 1

Outcome:

Synchronization is running for scenario FS 1...

Done!

Test-Integrity - Perform Integrity Test for Assured Recovery

The **Test-Integrity** command enables you to activate an automatic integrity test on a Replica host for assured recovery.

Notes:

- To activate the **Test Integrity** command, it is necessary to use a scenario with the **Integrity Testing for Assured Recovery** option turned to On.
- The Assured Recovery option supports both Replication and HA solutions. However, it is best suited for HA since the Replica server contains the actual database servers on which the test is performed, and not only data. If you are using AR test as a part of Replication scenario, you must verify that the root directories path is the same on the Master and the Replica. In addition, the Replica should have the database application installed, or share files if you test a File Server, and they need to be configured on the Master and the Replica in exactly the same way. Otherwise, the AR test will not produce meaningful results.
- The scenario needs to run before you start the test.

Syntax

Test-Integrity [-Name] <String> [-Host] <String> [-Mode] <String> -WaitJournal [<Boolean>]

Parameters

Name

The name of the scenario.

Host

The IP address or hostname of the Replica host you want to test.

Mode

Specifies whether to run the test integrity mode in the manual or auto mode. Type m, M, manual, or Manual to choose the manual mode. The default is the auto mode.

WaitJournal

Start testing when all journals are applied for P2V scenario.

0: Do not wait for journals

1: Wait for journals

Example: Perform Integrity Test for Assured Recovery

Test-Integrity "Exchange Server 1" 192.168.1.153

Outcome

Integrity testing for assured recovery started on 192.168.1.153

Done!

Integrity testing for assured recovery completed on 192.168.1.153

Unmount-Snapshot - Unmount a Snapshot

The **Unmount-Snapshot** command enables you to release an exposed snapshot without losing the snapshot itself. The snapshot is still exposed but it does not use a mount point.

Syntax

Unmount-Snapshot [-Name] <String> [[-Index] [<Int32>]] [[-Port] [<String>]]

Parameters

Name

The name of the host whose snapshot you want to expose.

Index

The index no. of the snapshot, as returned by the Get-Snapshot command (see page 74).

Port (Optional)

The port that is used for connecting to the given host. The default port is **25000**.

Example: Unmount a snapshot

Unmount-Snapshot {97127d0b-f1c9-4db5-943d-96c39b712fe6} 1

Outcome

Snapshot {97127d0b-f1c9-4db5-943d-96c39b712fe6} unmounted

Editing Commands

This section describes Arcserve RHA PowerShell commands that enable you to edit scenarios and scenario groups.

Add-Appliance - Specify an Appliance for a Full System Scenario

The **Add-Appliance** command enables you to add an appliance for a Full System scenario.

Syntax

Add-Appliance [-Name] <string> [-Host] <string> [-Parent] <string> [-Type] <string> [[-Platform] <string>] [[-Credentials] <PSCredentials>] [[-ResourcePool] <string>] [[-Storage] <string>] [[-Port] <string>] [[-SSL] <Boolean>] [[-Dynamic] <Boolean>]

Parameters

Name

The name of the new scenario.

Host

The name of the replica host.

Parent

The parent host, we support 1:m:n.

Type

The platform type.

- **E or e** - ESX
- **H or h** - Hyper-V
- **X or x** - Xen
- **V or v** - VCenter
- **Others** - warning message

Platform

The IP of the virtual platform. Not applicable in case of Hyper-V.

Credentials (Optional)

The credentials of the virtual platform.

ResourcePool

The resource pool name. If you do not select this field, the default is selected.

Storage

The storage name, if you do not set this field, the default is selected.

Port

The port number of the virtual platform, if you do not set this field, the default is selected.

SSL

The switch of the SSL connection, the default value is true.

Dynamic

The switch of the "with provision" property, the default value is true.

Example: Add an appliance for a Full System scenario

```
Add-Appliance -Name FULL -Parent 9.181.130.110 -Host 9.181.130.64 -Type H
```

Outcome

```
Replica 9.181.130.64 added successfully
```

Add-Dir - Add Root Directories to the Master and Replica Hosts

The **Add-dir** command enables you to add root directories to the Master and Replica hosts. You can define the same root directory path for both the Master and Replica, or you can enter two different paths. If you do not enter a different path for the Replica, by default it is the same as the Master path.

Syntax

Add-Dir [-Name] <String> [-MasterPath] <String> [[-ReplicaPath] [<String>]]

Parameters

Name

The scenario name.

MasterPath

The full path of the root directories on the Master.

For a Full System scenario, enter both the driver letter and mount point. When you enter the driver letter, you can ignore ":", "/" or "\".

For example, "e", "e:", "e:\", "e:/", "e:\\\\\", "e:////" are the same.

Note: Be aware of the following points:

- When you enter a mount point, use "\" or "/" as the separator.
- When there is no root directory in the scenario, the default volume is also added, such as the boot volume and system volume.
- When you enter all, all volumes are added.

ReplicaPath (Optional)

The full path of the root directories on the Replica. If no value is entered, the same path is used for the Master and Replica.

Note: For a Full System scenario, ignore this parameter.

Example: Add the same root directory to the Master and the Replica.

add-dir "File Server 1" C:/Tools

Outcome

Root Directory: C:/Tools added successfully.

Add-Group - Create a Scenario Group

The **Add-Group** command enables you to create a new scenario group.

Note: When no scenario is assigned, empty scenario groups do not appear on the Overview Page.

Syntax

Add-Group [-Name] <String>

Parameters

Name

The name of the new scenario group.

Note: Enter a unique name, since you cannot use the same name for more than one scenario group. If you use an existing name for the new group, the system will change it automatically.

Example: Create a new scenario group

add-group "File Server Scenarios"

Outcome

Group File Server Scenarios added successfully

Add-Master - Add a Master Host to a Scenario

The **Add-Master** command enables you to add a Master host to a given scenario. When defining a Master host, you need to enter its hostname. In addition, you can also enter the Master IP address, but this parameter is not mandatory.

Notes:

- You can enter the IP address as the hostname.
- You can use this command for changing an existing Master as well.

Syntax

```
Add-Master [-Name] <String> [-Host] <String> [[-IP] [<String>]]
```

Parameters

Name

The name of the scenario.

Host

The hostname of the new Master.

IP (optional)

The IP address of the new Master. If no IP address is defined, by default the system seeks it by using the specified hostname, and uses the first IP address it finds. For this reason, if the host has multiple IP addresses, we recommend to enter here the IP address you want to use.

Example: Add a Master host to a scenario

```
add-master "File Server 1" 130.119.185.152
```

Outcome

```
Master 130.119.185.152 added successfully
```

Add-Replica - Add a Replica Host to a Scenario

The **Add-Replica** command enables you to add a Replica host to a given scenario. When defining a Replica host, you need to enter its hostname, and optionally, its IP address as well. Then, you need to enter its parent host, which can be either the Master or another Replica.

Note: You can enter the IP address as the hostname.

When using ACL security delegations, you need to enter the values of three additional parameters: UserName, Password, and DomainName.

Syntax

Add-Replica [-Name] <String> [-Host] <String> [[-IP] [<String>]] [-Parent] <String> [[-UserName] <String>] [[-Password] <String>] [[-DomainName] <String>]

Parameters

Name

The name of the scenario.

Host

The hostname of the new Replica.

IP (optional)

The IP address of the new Replica. If no IP address is defined, by default the system seeks it by using the specified hostname, and uses the first IP address it finds. For this reason, if the host has multiple IP addresses, we recommend to enter here the IP address you want to use.

Parent

The parent host of the new Replica host. The parent can be either the Master or an upstream Replica, and you can use either its hostname or IP address.

UserName; Password;DomainName (ACL only)

The User Name, password and domain of a user, who has the right to add a new Replica host.

Example: Add a Replica host to a scenario

add-replica "File Server 1" 130.119.185.153 -parent 130.119.185.152

Outcome

Replica 130.119.185.153 added successfully

Add Replicas - Add Multiple Replica Hosts to a Scenario

The **Add-Replicas** command enables you to add multiple Replica hosts at once to a given scenario. To add multiple Replica hosts, you need to create a text file that contains the hostnames and IP addresses of the hosts. When you use the command, first define the scenario name and the parent host of all the Replica hosts you want to add. Then, specify the name and path of the file that contains the details of the new hosts.

Syntax

Add-Replicas [-ScenarioName] <String> [-ParentHost] <String> [-FileName] <String>

Parameters

ScenarioName

The name of the scenario.

ParentHost

The parent host of the new Replica host. The parent can be either the Master or an upstream Replica, and you can use either its hostname or IP address.

FileName

A text file that contains the host names and their IP addresses. The text should be formatted as follows:

#host name IP address

QA95-W2K3-SQL1 *130.119.185.155

QA95-W2K3-EX2 *130.119.185.153

UserName; Password;DomainName (ACL only)

The User Name, password and domain of a user, who has the right to add new Replica hosts.

Example: Add multiple Replica hosts to a scenario

add-replicas "Exchange Server" QA95-W2K3-EX1 D:\New_Replica_Hosts.txt

Outcome

130.119.185.151 QA95-W2K3-EX1

130.119.185.152 QA95-W2K3-EX2

2 replicas were added.

Add-Scenario - Create a New Scenario

The **Add-Scenario** command enables you to create a new scenario. When creating a new scenario, you need to define the following:

- The scenario name
- The scenario group to which this scenario will be assigned (optional)
- The type of application or database server to be protected
- The type of data protection solution
- Whether to enable the Integrity Testing option for Assured Recovery

The new scenario is created without hosts and root directories. You define these parameters at a later stage, using the Add-Master (see page 48), Add-Replica (see page 49) and Add-Dir (see page 46) commands.

Syntax

Add-Scenario [-Name] <String> [[-Group] [<String>]] [[-Application] [<String>]] [[-Type] [<String>]] [[-AR] [<Boolean>] [[-IntOpt] <String> [[-ServerURL] <String>]]

Parameters

Name

The name of the new scenario.

Note: Enter a unique name, since you cannot use the same name for more than one scenario. If you will use an existing name for the new scenario, the system will change it automatically.

Group (optional)

The scenario group name that contains the new scenario.

Notes:

- If you do not enter a group name, the new scenario is assigned to the default **Scenarios** group.
- You can create here a new scenario group, by entering a new group name. You can also create a new scenario group by using the Add-Group command (see page 47).

Application

The type of server whose data will be replicated:

- **EX** - Exchange
- **SQL** - SQL server
- **ORA** - Oracle
- **IIS** - Internet Information Server
- **FS** - File Server

- **P2V** - Full System

Type

The type of solution:

- **DR** - Replication/Disaster Recovery
- **HA** - High Availability

AR

Whether to perform an Assured Recovery test of the recoverability of the data on the Replica server:

- **0** - No
- **1** - Yes

IntOpt

The integration with other Arcserve products:

- **BAB** - Integration with Arcserve Backup
- **D2D** - Integration with Arcserve D2D
- **HBBU** - Integration with Arcserve Central Host-Based VM Backup

Example: Create a new scenario

```
add-scenario "File Server 1" "File Server Scenarios" FS DR 0
```

Outcome

```
Scenario File Server 1 added successfully
```

Create-D2DScenario-Create the D2D Scenario

The **Create-D2DScenario** command enables you to create a D2D scenario.

Syntax

create-D2DScenario [-ServerURL] <String> [-Credentials] <PSCredential> [-ReplicaHostName] <String> [-ScenarioName] <String>

Parameters

ServerURL

Defines the URL of the D2D server.

Credentials

Defines the credentials to log on to the D2D server.

ReplicaHostName

Defines the name or the IP address of the replica.

ScenarioName

Defines the name of the scenario.

Example: Create a D2D integration scenario

Create-D2DScenario -ServerURL http://test01:8014 -Credentials testserv/testpass -ReplicaHostName test01 -ScenarioName testD2D

Create-HBBUScenario - Create the Arcserve Central Applications Integration Scenario

The **Create-HBBUScenario** command lets you create a scenario that connects to the Arcserve Central Applications server to get the policy and related backup destination as the master and auto discover all backup VM files.

Syntax

create-HBBUScenario [-ServerURL] <String> [-Credentials] <PSCredential> [-ScenarioName] <String> [-MasterHostName] <String> [-ReplicaHostName] <String> [[-FilterFileName] <String>] [[-InstanceUUID] <String>]

Parameters

ServerUrl

Defines the URL of the Arcserve Central Host-Based VM Backup server.

Credentials

Defines the credentials to log on to the Arcserve Central Applications server.

ScenarioName

Defines the name of the scenario.

MasterHostName

Defines the name or the IP address of the master server.

ReplicaHostName

Defines the name or the IP address of the replica.

FilterFileName

Defines the filter file name of the virtual machine

InstanceUUID

Defines the UUID of the virtual machine

Example: Create a Central Applications Integration scenario

Create-HBBUScenario -ServerURL http://test01:8015 -ScenarioName testHBBU -MasterHostName test02 -ReplicaHostName test_replica

Get-D2DBackupDestination - Get the D2D Backup Destination

The **Get-D2DBackupDestination** command gets the D2D backup destination details.

Syntax

get-D2DBackupDestination [-ServerUrl] <String> [-Credentials] <PSCredential>

Parameters

ServerURL

Defines the URL of the D2D server.

Credentials

Defines the credentials to log on to the D2D server.

Example: Get the D2D backup destination

Get-D2DBackupDestination -ServerURL http://test01:8014 Credentials admin/testpass

Get-HBBUVM - Get VMs from the Arcserve Central Applications Server

The **Get-HBBUVM** command gets the virtual machines that are assigned to the backup policy from the Arcserve Central Applications server.

Syntax

Get-HBBUVM [-ServerUrl] <String> [-Credentials] <PSCredential>

Parameters

ServerUrl

Defines the URL of the Arcserve Central Host-Based VM Backup server.

Credentials

Defines the credentials to log on to the Arcserve Central Applications server.

Example: Get the virtual machine from the Central Applications server

Get-HBBUVM http://test01:8015

Remove-Dir - Remove Root Directories from the Master and Replica

The **Remove-Dir** command enables you to remove root directories from the Master and Replica hosts.

Note: You cannot remove a root directory only from the Replica using this command. Once you remove the Master root directories, the corresponding Replica root directories are removed as well.

Syntax

Remove-Dir [-Name] <String> [-MasterPath] <String>

Parameters

Name

The name of the scenario.

MasterPath

The root directory path on the Master.

For a P2V scenario, enter both the driver letter and mount point. When you enter the driver letter, you can ignore ":", "/" or "\".

For example, "e", "e:", "e:\", "e:/", "e:\\\\\", "e:////" are the same.

Note: Be aware of the following points:

- When you enter a mount point, use "\" or "/" as the separator.
- Sometimes, the system volume does not have a driver letter or a mount point and you cannot add or remove it. This drive letter or mount point is added automatically. Do not try to remove such a volume.

Example: Remove a root directory from the Master and Replica

remove-dir "File Server 1" C:/Tools

Outcome:

Root Directory: C:/Tools removed

Remove-Group - Delete a Scenario Group

The **Remove-Group** command enables you to delete a given scenario group.

Note: You can only remove an empty scenario group. If you want to remove a group that contains scenarios, first you need to remove the scenarios.

Syntax

Remove-Group [-Name] <String>

Parameters

Name

The name of the scenario group you want to delete.

Example: Delete a scenario group

remove-group "new group 1"

Outcome

Group new group 1 removed

Remove-Replica - Remove a Replica Host from a Scenario

The **Remove-Replica** command enables you to remove a Replica host from a given scenario.

Syntax

Remove-Replica [-Name] <String> [-Host] <String> [-Parent] <String>

Parameters

Name

The name of the scenario.

Host

The name of the Replica host you want to remove.

Parent

The parent of the Replica host you want to remove in the replication tree. It can be either the Master or an upstream Replica.

Example: Remove a Replica host from a scenario

remove-replica "FS 1" 130.119.185.153 -parent 130.119.185.152

Outcome

Replica 130.119.185.153 removed

Remove-Scenario - Delete a Scenario

The **Remove-Scenario** command enables you to delete a given scenario. The command also deletes the related VM resources when the scenario is a Full System scenario.

Note: You cannot delete a running scenario.

Syntax

Remove-Scenario [-Name] <String> [[-DeleteVM] [<Boolean>]]

Parameters

Name

The name of the scenario you want to delete.

Delete_VM [<Boolean>]

Deletes all VM resources of a P2V scenario.

1 - Delete **(Default)**

0 - Do not delete

Example: Remove a scenario

remove-scenario "File Server 2"

Outcome

Scenario File Server 2 removed

Rename-Group - Rename a Scenario Group

The **Rename-Group** command enables you to change the name of a given scenario group.

Syntax

Rename-Group [-Name] <String> [-NewName] <String>

Parameters

Name

The current name of the scenario group.

NewName

The new name for the scenario group.

Note: Enter a unique name, since you cannot use the same name for more than one scenario group. If you use an existing name for the scenario group, the system changes it automatically.

Example: Rename a scenario group

rename-group Server "Exchange Server Scenarios"

Outcome

Group Server renamed!

Rename-Scenario - Change a Scenario Name

The **Rename-Scenario** command enables you to change the name of a given scenario.

Note: You cannot rename a running scenario. To change its name, stop the scenario first.

Syntax

Rename-Scenario [-Name] <String> [-NewName] <String>

Parameters

Name

The current name of the scenario.

New Name

The new name for the scenario.

Example:

rename-scenario "File Server 1" "File Server"

Outcome

Scenario File Server 1 renamed!

Commands for Changing Scenarios While Running

You may now change certain scenario properties while a scenario is running. This section lists the syntax and examples for the commands that apply to editing a scenario while it is running.

- Apply-AllPendingRuntimeChangeableScenarioProperties
- Apply-PendingRuntimeChangeableScenarioProperties
- Discard-AllPendingRuntimeChangeableScenarioProperties
- Discard-RuntimeChangeableScenarioProperties
- Get-AllRuntimeChangeableScenarioProperties
- Get-RuntimeChangeableScenarioProperties
- Get-PendingRuntimeChangeableScenarioProperties
- Get-AllPendingRuntimeChangeableScenarioProperties
- Set-RuntimeChangeableScenarioProperty
- Test-RuntimeChangeableScenarioProperty

Apply-AllPendingRuntmeChangeableScenarioProperties

The Apply-AllPendingRuntimeChangeableScenarioProperties command lets you apply all changes to all changed scenarios at run time.

Syntax

Apply-AllPendingRuntimeChangeableScenarioProperties

Parameters

None

Example

Suppose you changed properties on several scenarios. Those changes are all pending. To apply them immediately use this command.

Apply-PendingRuntimeChangeableScenarioProperties

The **Apply-PendingRuntimeChangeableScenarioProperties** command lets you apply all changes to the specified running scenario.

Syntax

Apply-PendingRuntimeChangeableScenarioProperties [Name]

Parameters

Name - The name of the running scenario you wish to change.

Example

The following command applies changes to the scenario called FileServer:

Apply-PendingRuntimeChangeableScenarioProperties FileServer

Outcome

The result is a list of properties and their values:

error_source : engine_verification_error

error_level: Error

host_index: 2

root_dir_index: 0

property_xpath:
Scenario.ReplicationTree.ReplNode.ReplNode.SpecificReplicaProps.
ReplicaScheduledTask.Suspend.SuspendScript.Path

Discard-AllPendingRuntimeChangeableScenarioProperties

The Discard-AllPendingRuntimeChangeableScenarioProperties command cancels all the changes you configured for all scenarios.

Syntax

```
Discard-AllPendingRuntimeChangeableScenarioProperties
```

Parameters

None

Discard-RuntimeChangeableScenarioProperties

The Discard-RuntimeChangeableScenarioProperties command lets you cancel any changes made on the specified running scenario.

Syntax

```
Discard-RuntimeChangeableScenarioProperties [Name]
```

Parameters

Name - This is the name of the running scenario for which you wish to discard changes.

Example

If you have made numerous changes to a running scenario called FileServer and discover problems, use the following command to set the scenario back to all original values:

```
Discard-RuntimeChangeableScenarioProperties FileServer
```

Outcome

All changes made to the scenario called FileServer are reset back to their original values.

If you wish to set only a specific property back to its original value, use the Set-RuntimeChangeableScenarioProperties command to change only the specified property for the running scenario. Note that when you have discarded changes, all records in the Get-AllPendingRuntimeChangeableScenarioProperties list are deleted. Running Apply-PendingRuntimeChangeableScenarioProperties results in a warning message that the specified scenario was not changed at run time.

Get-AllPendingRuntimeChangeableScenarioProperties

The Get-AllPendingRuntimeChangeableScenarioProperties command lists all the scenarios that were changed while running.

Syntax

```
Get-AllPendingRuntimeChangeableScenarioProperties
```

Parameters

None

Example

The following command lists all the scenarios changed while running.

```
Get-AllPendingRuntimeChangeableScenarioProperties
```

Outcome

The result is a list of scenario names.

Get-AllRuntimeChangeableScenarioProperties

The Get-AllRuntimeChangeableScenarioProperties command enables you to list all the properties that can be changed while any scenario is running.

Syntax

```
get-AllRuntimeChangeableScenarioProperties
```

Parameters

None

Example:

The following code issues the command to list all editable properties and redirect the output to a text file.

```
Get-AllRuntimeChangeableScenarioProperties > d:\1.txt
```

Outcome

The result is a set of the editable properties in Scenario/Master/Replica/PropertyName format.

Get-RuntimeChangeableScenarioProperties

The Get-RuntimeChangeableScenarioProperties command enables you to find all the properties you can edit while the specified scenario is running. The command returns a list of the properties you may change for that scenario. Combine with I/O redirection commands to format output to a text file.

Syntax

```
Get-RuntimeChangeableScenarioProperties [Name]
```

Parameters

Name - the name of the running scenario

Example:

The following command lists all the editable properties for the scenario called File Server Scenario 1:

```
Get-RuntimeChangeableScenarioProperties FileServerScenario1
```

Outcome

Property: False

Host: 10.0.0.0

Value: False

Index: 111

Category: Replica

Name: SpecificReplicaProps.Suspend.SuspendScript

Get-PendingRuntimeChangeableScenarioProperties

The Get-PendingRuntimeChangeableScenarioProperties command lets you list all the changes made to the specified running scenario.

Syntax

```
Get-PendingRuntimeChangeableScenarioProperties [Name]
```

Parameters

Name - The name of the running scenario

Example:

The following example shows the list of all properties changed for the scenario called FileServer:

```
Get-PendingRuntimeChangeableScenarioProperties FileServer
```

Outcome

The result is a list of Properties, their Original Values and their New Values.

Set-RuntimeChangeableScenarioProperty

The Set-RuntimeChangeableScenarioProperty command enables you to update the value of the specified property in the named scenario while it is running.

Syntax

Set-RuntimeChangeableScenarioProperty [Name] [Index] [Value] [Host]

Parameters

Name - The name of the running scenario whose properties you wish to change.

Index - The index or name of the property you want to change. To determine correct names, use the Get-AllRuntimeChangeableScenarioProperties or the Get-RuntimeChangeableScenarioProperties commands. The index value is generated internally; for two scenarios even of the same type, the same property can have a different index. Ensure you obtain the correct indexes or names.

Value - The new setting for the specified property.

Host - (Optional) The host needs to be specified only for Master or Replica property changes. For generic or HA properties, this value is optional.

Example:

The following command updates the argument of the suspend script property based on its name:

Set-RuntimeChangeableScenarioProperty FileServer SpecificReplicaProps.Suspend.SuspendScript 456 -Host 10.0.0.1

Outcome

The result is a table showing the Property, its Original Value and the New Value.

Test-RuntimeChangeableScenarioProperty

The Test-RuntimeChangeableScenarioProperty command lets you confirm whether the specified property can still be edited while the scenario is running.

Syntax

Test-RuntimeChangeableScenarioProperty [Name] [Index] [Host]

Parameters

Name - The name of the running scenario

Index - The index or name of the property you wish to test

Host - Optional

Example

The following command tests the property whose index is 113 for a running scenario called FileServer.

Test-RuntimeChangeableScenarioProperty FileServer 113 -host 10.0.0.3

Outcome

The result is a true or false value for the index or property name specified.

Monitoring commands

This section describes Arcserve RHA PowerShell commands that enable you to monitor the DR and HA processes.

Get-Dirs - List all Root Directories of a Scenario

The **Get-Dirs** command enables you to list all root directories of a given scenario.

Syntax

Get-Dirs [-Name] <String>

Parameters

Name

The scenario name.

Example: List the root directories of a given scenario

get-dirs "File Server 1"

Outcome

ID : 2721474912

Scenario : File Server 1

Master : 192.168.1.152

Path : C:/Tools

DB : False

Get-Events - List all Events of a Scenario

The **Get-Events** command displays a list of replication events of a given scenario. The event list can include informational, warning and error events. The displayed information consists of the event ID, the event date and time, the scenario name, the event severity and the event message.

Syntax

Get-Events [-Name] <String>

Parameters

Name

The name of the scenario whose events you want to view.

Example: List events of a given scenario in an auto-sized format table

get-events "File Server 1" | FT -auto

Outcome

```
ID     Time                 Scenario     Severity    Message

--     ----                 --------     --------    -------

SM00165 10/28/2008 6:02:52 PM File Server 1 Significant Connected to...

SR00014 10/30/2008 7:17:31 PM File Server 1 Significant Starting...

SR00139 10/30/2008 7:17:35 PM File Server 1 Significant Starting File...

IR00119 10/30/2008 7:18:16 PM File Server 1 Info        Root directory...

SR00120 10/30/2008 7:18:16 PM File Server 1 Significant Synchronization...

IM00405 10/30/2008 7:15:06 PM File Server 1 Info        Posting...

SR00202 10/30/2008 7:18:21 PM File Server 1 Significant All modifications...

SR00096 11/3/2008 6:47:40 PM  File Server 1 Significant Stopping scenario...
```

Get-Group - List Groups that carry a Given Name

The **Get-Group** command enables you to list all scenario groups that carry a given name. To display this list, you need to enter the name you are searching for.

In addition, this command enables you to list all existing scenario groups. To list all scenario groups, just enter the command without a scenario name.

Syntax

Get-group [[-GroupName] [<String>]]

Parameters

Name

The name of the scenario group.

Note: You can use the '*" or "?" wildcards as part of the scenario group name.

Example: List all scenario groups that carry a given name

get-group *Server*

Outcome

File Server Scenarios 2

File Server Scenarios 1

Exchange Server Scenarios

File Server Scenarios

Get-Hosts - List all Hosts of a Scenario

The **Get-Hosts** command enables you to list all hosts of a given scenario.

Syntax

Get-Hosts [-Name] <String>

Parameters

Name

The scenario name.

Example: List the hosts of a given scenario in an auto-sized format table

Get-Hosts "File Server 1" |FT -auto

Outcome

Scenario Name Role Parent State IP Port

-------- ---- ---- ------ ----- -- ----

File Server 1 192.168.1.152 Master -- Running 192.168.1.152 25000

File Server 1 192.168.1.153 Replica 1192.168.1.152 Running 192.168.1.153 25000

Get-NetworkAdapters - Get Network Adapters of the Specified Host

The Get-NetworkAdapters command enables you to list all network adapters of a specified host.

Syntax

Get-NetworkAdapters [-Name] <string> [-Host] <string>

Parameters

Name

The scenario name.

Host

The host name in a Full System scenario.

Example: List all network adapters of a specified host of a Full System scenario

Get-NetworkAdapters FULL 9.181.130.64

Outcome: All network adapters are displayed.

Intel(R) 82579LM Gigabit Network Connection - Virtual Switch

Arcserve RHA internal for AR

Get-Scenario - List Scenarios that carry a Given Name

The **Get-Scenario** command enables you to list all scenarios that carry a given name. To display this list, enter the name you are searching for.

In addition, this command enables you to list all existing scenarios. To list all scenarios, enter the command without a scenario name.

Syntax

Get-Scenario [[-Name] [<String>]]

Parameters

Name

The scenario name.

Note: You can use the '*" or "?" wildcards as part of the scenario name.

Example: List all scenarios that carry a given name in an auto-sized format table.

get-scenario File* |FT -auto

Outcome

```
ID         Group     Name          Type       Master         State   Sync AR
--         -----     ----          ----       ------         -----   ---- --
1123633852 Scenarios FileServer    FileServer                Unknown File False
1123633468 Scenarios File Server 1 FileServer 192.168.1.153  Stopped File False
```

Get-Snapshot - Display VSS Snapshots of a Replica Host

The **Get-Snapshot** command enables you to display all VSS snapshots of a given Replica Host.

Syntax

Get-Snapshot [-Name] <String> [[-Port] <String>]

Parameters

Name

The name of the host as it appears in the scenario.

Port (optional)

The connection port to the given host. The default port number is **25000**.

Example: Display all VSS snapshots of a given Replica host in an auto-sized format table

Get-Snapshot 130.119.173.7 |FT -auto

Outcome

Index Snapshot Created Exposed Mounted Drive Scenario

----- -------- ------- ------- ------- ----- --------

0 {4f2bb053-5f2d} 11/18/2008 4:03:09 PM False Not Mounted C:/ FileServer

1 {bcbdda2b-6165} 11/18/2008 4:06:00 PM False Not Mounted C:/ FileServer

2 {c1f206be-2ad0} 11/18/2008 4:07:17 PM False Not Mounted C:/ FileServer

Get-State - List all Scenarios defined for a Given Host

The **Get-State** command enables you to list all the scenarios that are defined for a given host, along with their details and states.

Syntax

Get-State [-Name] <String>

Parameters

Name

The name of the host.

Example:

get-state 130.119.185.152

Outcome

ID : 2505374864
Group : FS Scenarios
Name : FS 1
Type : FileServer
Master : 130.119.185.152
State : Running
Sync : File
AR : False

ID : 2721467841
Group : File Server Scenarios
Name : File Server 1
Type : FileServer
Master : 130.119.185.152
State : Stopped
Sync : File
AR : False

Get-Stats - Display Replication Statistics of a Scenario

The **Get-Stats** command enables you to display scenario statistic per host during a run.

Syntax

Get-Stats [-Name] <String>

Parameters

Name

The name of the scenario.

Example: Display replication statistics of a given scenario during a run

get-stats "File Server 1"

Outcome

Scenario : File Server 1

Name : 192.168.1.152

Role : Master

Spool_Size : 0

Sync_Files : 345

Sync_MBytes : 86

Rep_MBytes : 0

Scenario : File Server 1

Name : 192.168.1.153

Role : Replica

Spool_Size : 0

Sync_Files : 345

Sync_MBytes : 86

Rep_MBytes : 0

User Management Commands

This section describes Arcserve RHA PowerShell commands that enable you to monitor and manage User Groups and Users for ACL-based Control Service.

Note: A special license is needed for using the ACL commands.

Clean-VMResource – Clean VM Resource on a Virtual Platform

The **Clean-VMResource** command enables you to clean virtual machine resources on a virtual platform. If you enter a replica host, the VM resources of the replica are cleaned. When you do not enter the Host parameter, all VM resources in a virtual platform are cleaned.

Syntax

Clean-VMResource – Clean vm resource on virtual platform

Parameters

Name

The name of the scenario.

Host

The replica host in the scenario.

Example: Clean all resources

Clean-VMResource FULL 9.181.130.64

Outcome

VM Resources deleted successfully.

Edit-NetworkMapping – Map Network Adapters to the Master and Replica

The Edit-NetworkMapping command lets you map the network adapters between the master and replica of a Full System scenario.

Syntax

Edit-NetworkMapping [-Name] <string> [-Host] <string> [-SourceAdapter] <string> [-TargetAdapter] <string> [[IP] <string[]>] [[DNS] <string[]>] [[gateway] <string[]>] [[PrimaryWins] <string>] [[SecondaryWins] <string>] [[-Type] <string>]

Parameters

Name

The name of the scenario. Only the HA and P2V scenarios are supported.

Host

The host name or the IP address.

SourceAdapter

The name of the source adapter.

TargetAdapter

The name of the target adapter. User Null to specify an empty value.

IP

The array of the IP address. Use a ":" as a separator. For example, "192.168.1.1:255.255.252.0". 255.255.252.0 is a subnet mask.

DNS

The array of the DNS address.

Gateway

The array of the gateway address.

PrimaryWins

Primary WINS (Windows Internet Name Service) address.

SecondaryWins

Secondary WINS (Windows Internet Name Service) address.

Type

The AR or SW network. The default is the SW network. Type "ar" to set the AR network.

Example

Edit-NetworkMapping -Name FULL -Host 9.181.130.61 -SourceAdapter "Microsoft Network Adapter Multiplexor Driver" -TargetAdapter "Arcserve RHA internal for AR" -Type AR -IP 9.181.130.140:255.255.0.0

Outcome

Successfully edited the network mapping.

Get-SuperUserGroup - Display the Super User Group Name

The **Get-SuperUserGroup** command enables you to display the name of the Super User Group of the connected Control Service.

Syntax

Get-SuperUserGroup

Note: This command does not have parameters.

Example: Display the name of the Super User Group

Get-SuperUserGroup

Outcome

QA95-W2K3-SQL\\<supergroup user name>

Set-SuperUserGroup - Change the Super User Group

The **Set-SuperUserGroup** command enables you to change the Super User Group.

Syntax

Set-SuperUserGroup [-GroupName] <String>

Parameters

GoupName

The name of the new Super User group.

Example: Change the Super User Group

Set-SuperUserGroup Administrators

Outcome

Set super user group successfully

Get-Users - List all Users of the Super User Group

The **Get-Users** command enables you to list all users that belong to the Super User Group.

Syntax

get-users

Note: This command does not have parameters.

Example: List all users that belong to the Super User Group

get-users

Outcome

QA95-W2K3\Administrator

QA95-W2K3-SQL\User2

QA95-W2K3-SQL\User1

Get-ScenarioUsers - List all Users with Rights on a Scenario

The **Get-ScenarioUsers** command enables you to list all users that have rights on a certain scenario.

Syntax

get-ScenarioUsers [-ScenarioName] <String>

Parameters

ScenarioName

The name of the scenario.

Example: List all users that have rights on a given scenario

Get-ScenarioUsers "File Server"

Outcome

name

QA95-W2K3-SQL\User2

QA95-W2K3-SQL\User1

Set-ScenarioUser - Assign User Rights on a Scenario

The **Set-ScenarioUser** command enables you to assign to a user certain rights on a specific scenario.

Syntax

Set-ScenarioUser [-Name] <String> [-User] <String> [-Right] <String>

Parameters

Name

The name of the scenario.

User

The full name of the user.

Right

The type of rights the user will have on the scenario. Enter one of the following:

A = Admin

C = Control

V = View Only

Example: Assign to a user Control rights on a given scenario

Set-ScenarioUser "File Server" QA95-W2K3-SQL\User2 C

Outcome

Set user right successfully

Remove-ScenarioUser - Cancel User Rights on a Scenario

The **Remove-ScenarioUser** command enables you to cancel the rights of a user on a certain scenario.

Syntax

```
Remove-ScenarioUser [-Name] <String> [-User] <String>
```

Parameters

ScenarioName

The name of the scenario.

UserName

The name of the user.

Example: Cancel the rights of a user on a given scenario

```
Remove-ScenarioUser "File Server" QA95-W2K3-SQL\User2
```

Outcome

```
User removed successfully.
```

Index

L

M

O

P

R

S

T

U