CA XOsoft™ PowerShell Commands

Operation Guide

Contact CA

Contact Technical Support

For your convenience, CA provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA products. At http://ca.com/support, you can access:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA product documentation, please complete our short customer survey, which is also available on the CA support website, found at http://ca.com/support.

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 CA XOsoft PowerShell commands. It provides a brief overview of Windows PowerShell, describes each CA XOsoft 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:

- *CA XOsoft Installation Guide*
- *CA XOsoft User 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 CA XOsoft PowerShell Commands

CA XOsoft PowerShell is offered to users as an alternative or a supplement to managing the replication process using the CA XOsoft 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. CA XOsoft 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 CA XOsoft PowerShell commands look and operate exactly as the ones that are managed by the CA XOsoft 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 prompt to enter it. If an optional parameter is missing, PowerShell will use 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 command, 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
```

Installing CA XOsoft PowerShell

To use CA XOsoft PowerShell, you need to install Windows PowerShell and CA XOsoft snap-ins.

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

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

Running CA XOsoft PowerShell

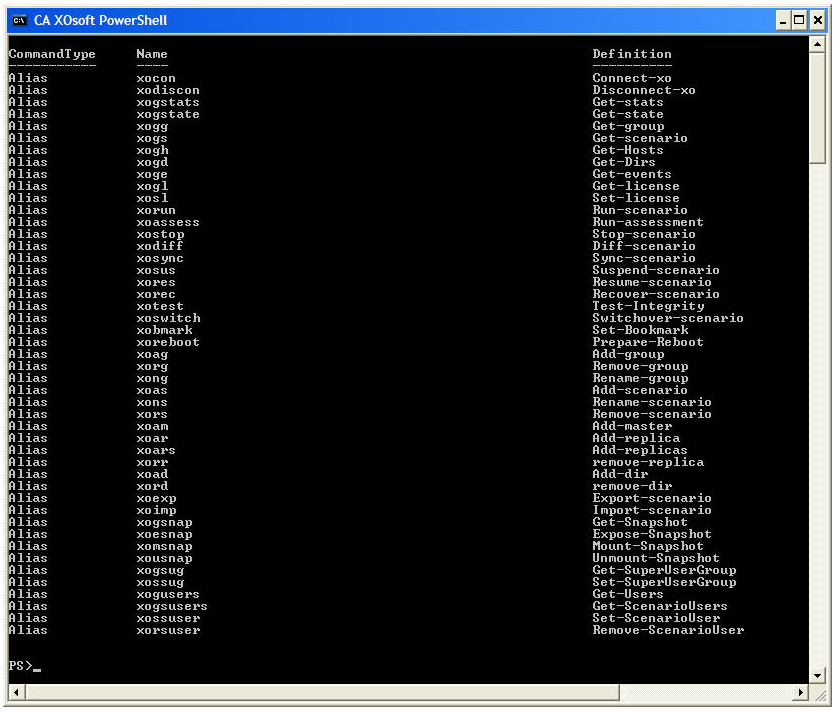
After the installation of Windows PowerShell and CA XOsoft snap-ins, you can run CA XOsoft PowerShell from two places:

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

To run CA XOsoft PowerShell from CA XOsoft PowerShell shortcut:

1. Open CA XOsoft PowerShell by selecting **Start, Programs, CA**, **XOsoft**, **PowerShell**.

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



   Now, you need to connect to the Control Service that manages your CA XOsoft operations. To perform this, use the Connect-XO command (see page 16).

To run CA XOsoft 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 CA XOsoft PowerShell Snapin INSTALLDIR:

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

   The directory changes.

3. Enter the following command to install CA XOsoft PowerShell snap-ins:

   ```
   .\xo.ps1
   ```

   The CA XOsoft PowerShell snap-ins are installed, and you can start using them to connect to the Control Service (see page 17) that manages your CA XOsoft 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 CA XOsoft 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*
  ```

CA XOsoft PowerShell commands verification

- To verify the installation of CA XOsoft PowerShell snap-ins, enter the following command and look for CA XOsoft 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 CA XOsoft PowerShell Commands

This chapter describes in details how to use CA XOsoft PowerShell commands to control, edit and monitor the DR and HA processes. The commands are displayed in alphabetical order and they are divided into 4 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 CA XOsoft registration.

Connect-XO - Connect PowerShell to a Control Service

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

Note: Once you finished working with CA XOsoft PowerShell, do not forget to disconnect from the Control Service using the the Disconnect-XO command. (see page 18) 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 (see page 17).

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:

1. Enter your password in the following command and run it once:

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

   (This command encrypts the password you entered, and saves it in the specified file.)

2. Create an object for your password by entering the following command:

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

   (This command reads the password file, and keeps it in the $pass object.)

3. Create an object for the **PSCredential** dialog by entering the following command:
   ```
   $mycred = new-object -typename System.Management.Automation.PSCredential -
   argumentlist <user_name> $pass
   ```

   (This command creates a PSCredential object using the specified user name and password.)

4. Connect to the Control Service by using the following syntax:
   ```
   Connect-XO [-Host] <String> $mycred [[-Protocol] [<String>]] [[-Port]
   [<String>]]
   ```

   The outcome is the same as in a standard connection:
   ```
   Connecting...

   192.168.1.151 connected!
   ```

Note: This information is taken from http://geekswithblogs.net/Lance/archive/2007/02/16/106518.aspx

Disconnect-XO - Disconnect from a Running Control Service

After you finished working with CA XOsoft 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 PowerShell window will also cause 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 CA XOsoft License

The **Get-License** command enables you to display your CA XOsoft license details.

Syntax

```
get-license
```

Example: Display your CA XOsoft license detailes

```
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 CA XOsoft

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

Syntax

```
get-license
```

Parameters

Key

A valid license key.

Example: Register CA XOsoft using a license key

```
set-license TVC2LF24FTU7G3WJ2QAFMCLGXA5KLPCCYIXTJTWX2MOZFU5GL7EJ3OYZQND7V3G123456
```

Outcome:

```
Key registered successfully
```

Controlling Commands

This section describes CA XOsoft PowerShell commands that enable you to control the DR and HA processes.

Diff-Scenario - Generate a Difference Report

The **Diff-Scenario** command enables you to generates 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 difference.

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 51)

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 will be exported to the current directory and will carry the name of the scenario with .xmc extension.

Example: Export a scenario to a specified location

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

Outcome:

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

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 and you need to 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 52).

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 XMC file, from a specified location to your Manager. 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 in 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 no. of the snapshot, as returned by the Get-Snapshot command (see page 52).

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:
```

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 that exist 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 48). 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 28).

Syntax

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

Example: Recover lost data

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

Outcome:

```
Recover application data process started
```

Resume-Scenario - Resume Replication on a Suspended Replica

The **Resume-Scenario** command enables you to resume the replication process on a suspended Replica host. Once the replication is resumed, 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 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 51).

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

Notes:

- To check if the operation is completed successfully, use the Get-Scenario (see page 51) and Get-Events (see page 48) commands.
- To run several scenarios at once, use the Get-Scenario command (see page 51):

  ```
  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 31).

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 in the Replica Properties list the **Recovery - Data Rewind** option to On.
- You can not 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 to give a meaningful name to that will later help you recognize the required bookmark.

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 stops one or several scenarios.

Note: To check if the operation was completed successfully, use the Get-Scenario (see page 51) and Get-Events (see page 48) 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 51).

Example: Stop a scenario

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

Outcome:

```
Scenario File Server 1 stopped
```

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, 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 35).

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 27).

Syntax

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

Parameters

Name

The scenario name.

Host

The Replica host 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>
```

Parameters

Name

The scenario name.

Example: Perform a switchover

```
Switchover-Scenario "SQL Server 1"
```

Outcome:

```
Scenario SQL Server 1 switching over to 192.168.1.153

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 51).

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

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 I**ntegrity Testing for Assured Recovery** option turned to On.
- The Assured Recovery option supports both DR and HA solutions. However, it is best suited for HA since in this case the Replica server necessarily 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 DR scenario, you must verify that the root directories path is the same on the Master and the Replica. In addition, the Replica should have 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>
```

Parameters

Name

The name of the scenario.

Host

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

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 52).

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 CA XOsoft PowerShell commands that enable you to edit scenarios and scenario groups.

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 dir 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 will be 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.

ReplicaPath (Optional)

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

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 will 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 will 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.

Syntax

```
Add-Replica [-Name] <String> [-Host] <String> [[-IP] [<String>]] [-Parent]
<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.

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-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 39), Add-Replica (see page 40) and Add-Dir (see page 37) commands.

Syntax

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

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 will not enter a group name, the new scenario will be 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 38).

Application

The type of server whose data will be replicated:

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

Type

The type of solution:

- **DR** - 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

Example: Create a new scenario

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

Outcome:

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

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 will be 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.

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.

Note: You cannot delete a running scenario.

Syntax

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

Parameters

Name

The name of the scenario you want to 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 will use an existing name for the scenario group, the system will change 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, you need to stop it 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!
```

Monitoring commands

This section describes CA XOsoft 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-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, you need to enter the name you are searching for.

In addition, this command enables you to list all existing scenarios. To list all scenarios, just 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 no. 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
```

Index