



Version 7.1.2

# Double-Take Availability for Linux DTCL Scripting Guide

## Notices

Double-Take Availability DTCL Scripting Guide Version 7.1.2, Monday, September 19, 2016

Check your service agreement to determine which updates and new releases you may be eligible for.

If you need maintenance renewal, an upgrade license key, or other sales assistance, contact your reseller/distributor or a Double-Take Software sales representative.

If you need technical assistance, you can contact CustomerCare. All basic configurations outlined in the online documentation will be supported through CustomerCare. Your technical support center is dependent on the reseller or distributor you purchased your product from and is identified on your service agreement. If you do not have access to this agreement, contact CustomerCare and they will direct you to the correct service provider. To contact CustomerCare, you will need your serial number and license key.

Assistance and support for advanced configurations may be referred to a Pre-Sales Systems Engineer or to Professional Services.

Man pages are installed and available on Double-Take Linux servers. These documents are bound by the same Double-Take Software license agreement as the software installation.

This documentation is subject to the following: (1) Change without notice; (2) Furnished pursuant to a license agreement; (3) Proprietary to the respective owner; (4) Not to be copied or reproduced unless authorized pursuant to the license agreement; (5) Provided without any expressed or implied warranties, (6) Does not entitle Licensee, End User or any other party to the source code or source code documentation of anything within the documentation or otherwise provided that is proprietary to Double-Take Software, Inc.; and (7) All Open Source and Third-Party Components ("OSTPC") are provided "AS IS" pursuant to that OSTPC's license agreement and disclaimers of warranties and liability.

Double-Take Software, Inc. and/or its affiliates and subsidiaries in the United States and/or other countries own/hold rights to certain trademarks, registered trademarks, and logos. Hyper-V and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. vSphere is a registered trademark of VMware. All other trademarks are the property of their respective companies. For a complete list of trademarks registered to other companies, please visit that company's website.

© 2016 Double-Take Software, Inc. All rights reserved.

# Contents

<b>Chapter 1 Client and scripts</b>	<b>7</b>
Command Line client	7
Scripts	9
Command syntax conventions	11
Getting help for scripting commands	12
<b>Chapter 2 Replication sets</b>	<b>13</b>
Creating a replication set	13
Modifying a replication set	17
Deleting a replication set	21
Calculating replication set size	23
<b>Chapter 3 Establishing a connection</b>	<b>25</b>
<b>Chapter 4 Simulating a connection</b>	<b>28</b>
<b>Chapter 5 Connections</b>	<b>30</b>
Monitoring connections	30
Queuing Double-Take Availability data	32
Reconnecting automatically	34
Pausing and resuming target processing	36
Disconnecting a connection	38
<b>Chapter 6 Mirroring</b>	<b>39</b>
Stopping, starting, pausing, or resuming mirroring	39
Mirroring automatically	42
Removing orphan files	44
<b>Chapter 7 Replication</b>	<b>47</b>
Starting replication	47
Inserting tasks during replication	48
<b>Chapter 8 Verifying your target data</b>	<b>51</b>
<b>Chapter 9 Data transmission</b>	<b>55</b>
Stopping, starting, pausing, and resuming transmission	55
Scheduling data transmission	58
Limiting transmission bandwidth	63
Compressing data for transmission	64
<b>Chapter 10 Restoring data</b>	<b>66</b>
<b>Chapter 11 Failover</b>	<b>70</b>
Configuring failover monitoring	70
Editing failover monitoring configuration	80
Failing over	82
Stopping failover monitoring	83
Deleting failover monitoring configuration	85
<b>Chapter 12 Failing back</b>	<b>87</b>
<b>Chapter 13 Configuring server settings</b>	<b>90</b>
Server settings	92

<b>Chapter 14 Scripting examples</b>	<b>113</b>
Creating and connecting a replication set	113
Creating and connecting a replication set with failover monitoring	114
Restoring a replication set	114
Using variables to create and connect a replication set and run verification	114
Controlling a mirror using flow control	115
Using variables to pause a target	115
Using variables to resume a target	115
Creating a backup of the target by rotating connections	116
<b>Chapter 15 Scripting commands</b>	<b>117</b>
Compression Disable	120
Compression Enable	121
Compression List	122
Compression Set	123
ConID	124
Connect	125
Connect TDU	127
Device List	128
Disconnect	129
Email Add	130
Email Disable	131
Email Enable	132
Email Filter	133
Email From Address	134
Email Get Email Config	135
Email Mail Server	136
Email Remove	137
Email Set Filter Include	138
Email Subject	139
Email Test	140
Environment	141
Exit	142
Failback	143
Failover	144
Get	145
GetEnvStr	146
Get Local	147
Help	148
Limit Bandwidth	149
Load Source	150
Load Target	151
Login	152
Logout	153
Mirror Pause	154
Mirror Resume	155
Mirror Start	156
Mirror Stop	157
Monitor Clear	158
Monitor Create	159

Monitor Delete .....	160
Monitor Display .....	161
Monitor Get .....	162
Monitor List .....	163
Monitor Move .....	164
Monitor Option .....	165
Monitor Remove .....	167
Monitor Script Add .....	168
Monitor Script Remove .....	169
Monitor Start .....	170
Monitor Stop .....	171
Monitor Use .....	172
NIC List .....	173
Orphans Preview .....	174
Orphans Start .....	175
Orphans Stop .....	176
Pause Target .....	177
Ping .....	178
Queue Task .....	179
Quit .....	181
Replication Start .....	182
Replication Stop .....	183
Repset Calculate .....	184
Repset Create .....	185
Repset Delete .....	186
Repset Display .....	187
Repset List .....	188
Repset Resync .....	189
Repset Rule Add .....	190
Repset Rule Remove .....	191
Repset Save .....	192
Repset Use .....	193
Restore .....	194
Resume Target .....	196
Schedule Clear .....	197
Schedule Disable .....	198
Schedule Enable .....	199
Schedule End .....	200
Schedule Start .....	201
Schedule Window .....	202
Set .....	203
Set Local .....	204
Shutdown .....	205
Source .....	206
StatsLog Start .....	207
StatsLog Status .....	208
StatsLog Stop .....	209
Status .....	210
Target .....	211
Test Connections .....	212

Time Now .....	213
Transmission Pause .....	214
Transmission Resume .....	215
Transmission Start .....	216
Transmission Stop .....	217
Unload Source .....	218
Unload Target .....	219
Verify .....	220
Version .....	221
Wait .....	222
Wait on Mirror .....	223
Wait on Restore .....	224
Wait on Target .....	225
Write .....	226

---

## Chapter 1 Client and scripts

Double-Take Availability has its own scripting language to control some, but not all Double-Take Availability features. This *Scripting Guide* does not explain Double-Take Availability features and functions, only the scripting aspects of Double-Take Availability.

The scripting language can be used in the Double-Take Availability Command Line client or in script files. The client and script files use the same set of commands.

This section includes the following topics.

- [Command Line client](#)
- [Scripts](#)
- [Command syntax conventions](#)
- [Getting help for scripting commands](#)

### Command Line client

There are three different methods of executing commands from the Command Line client.

- **Interactive Entry**—Type the case-sensitive command **DTCL -i**. A **Command** prompt will appear and the scripting commands can be entered from that prompt. Any resulting errors are immediately displayed.

```
[root@server ~]# DTCL -i
Command: login serve root *****
      User access level set to DT_FULL_ACCESS
Command:
```

- **File Entry**—Create a file with all of the scripting commands you want to run. Type the case-sensitive command **DTCL -f filename** where **filename** is the name of the file containing the scripting commands. For example, you might use the following three lines in your script. to log on to a machine and display its replication sets.

```
login server_name root *****;

source server_name;

repset list;
```

```
[root@server ~]#DTCL -f script
User access level set to DT_FULL_ACCESS
- List of rep sets -
  DataFiles          enabled

[root@server ~]#
```

- **Single Line Entry**—Determine all of the scripting commands you want to run, and enter them in a single line (separated by semi-colons) at the command prompt using the case-sensitive command DTCL. For example, you might use the following command to log on to a machine and display its replication sets.

```
[root@server ~]# DTCL login server_name root password; source server_name; repset list
User access level set to DT_FULL_ACCESS
- List of rep sets -
DataFiles          enabled
[root@server ~]#
```



If you are running the Command Line client single line entry in a Linux shell, you must escape the semi-colon and any quotation marks with a backslash to keep the shell from interpreting them as its own command separators. For example, you might enter the following command.

```
DTCL login \"server_name\" root \"password\"; source \"server_name\"; repset list;
```

If you are running the Command Line client single line entry from a Windows client, the Windows command process (cmd.exe) strips two layers of quotation marks during processing. Therefore, any scripting command that you use that requires quotation marks must have three quotation marks around it. For example, if your machine name had a space in it, login "server name" username password would be sufficient for the Command Line client interactive entry or file entry. But for the Command Line client single line entry, you would have to use login """"machine name"""" username password.



# Scripts

You can create script files to execute series and combinations of scripting commands to meet specific needs. When working with scripts, keep in mind the following.

- When creating scripts, each commented line must start and end with the pound (or number) sign (#).
- Scripts must use ANSI coding. Do not use Unicode or other formats.
- Variables can be used in commands to replace items that vary such as machine names. A variable name must start with the dollar sign (\$) character and can contain letters or numbers. Values are assigned to variables using the equals (=) assignment statement. For example, valid variable assignment statements are \$MySource=alpha; and \$MyPassword="pass word";.
- All commands return values which can also be assigned to variables. For example, the connection command returns a connection ID for the connection being created. The statement \$ConnectionID=CONNECT dataset to target; saves the connection ID as a variable. The variable components of the connect command could be replaced as well. For example, \$TheRepset="DataFiles";, \$TheTarget=beta;, and \$ConnectionID=CONNECT \$TheRepset to \$TheTarget; are all valid statements.
- Scripts can take advantage of four flow control programming constructs.
  - **IF conditional**—The IF conditional expression is a comparison between two values or variables. Use the following syntax for an IF conditional.

IF <relational\_expression> THEN <statement\_block> [ ELSE <statement\_block> ] END

Use the following conditions to create the relational expression.

- equal to =
  - not equal to !=
  - less than <
  - less than or equal to <=
  - greater than >
  - greater than or equal to >=
  - statement block—any sequence of valid commands
- **FOR Loop**—The FOR loop sets a variable to a start value, executes each statement in the statement block, and then adds the step value to the variable. If the new value of the variable does not exceed the end value then the statements will be executed again. This continues until the variable exceeds the end value. If a step value is not provided, the default adds one to the variable. Use the following syntax for the FOR loop.

FOR <variable> = <start\_value> TO <end\_value> [STEP <step\_value>] DO <statement\_block> END

- **WHILE Loop**—The WHILE loop evaluates a relational expression and, if it is true, then the statement block is executed. When the statement block has completed execution, the expression will be reevaluated again and, if it is true, the statement block is executed again. This continues until the expression is false. Use the conditions specified in the IF conditional to create the relational expression and the following syntax for the WHILE loop.

WHILE <relational\_expression> DO <statement\_block> END

- **WRITE Command**—The WRITE command writes values to the screen. These values can be variables or constants of type integer\$, string\$, date\$, or time\$. The write command is used in conjunction with either the IF conditional, WHILE loop, or FOR loop.
- Each command and assignment must end with a semi-colon. However, no semicolon is necessary after the END keyword for the IF conditional and the loop statements.
- When using the Command Line client, a script file can be executed using the -f option. Create a one-line batch file using the following command to initiate the Command Line client with the -f option and specify the name of the script to execute.

DTCL -f /directory\_name/script\_name

If you do not specify a path for the script file, Double-Take Availability will look in the directory where the DTCL -f command was executed.

# Command syntax conventions

The following conventions are used for the scripting commands.

- UPPERCASE letters indicate syntax that must be typed exactly as shown.
- *lowercase italic blue* letters are variables such as file names, user names, or machine names.
- Angle brackets, < and >, surround required items that must be supplied with the command.
- Square brackets, [ and ], surround optional items that can be supplied with the command but are not required.
- The pipe character, |, separates items in a list.
- Identifiers that contain a space or non-alphanumeric characters must be enclosed in quotation marks. For example "this is a password" or "machine name." The only exception is IP addresses which do not need to be enclosed in quotation marks.



You can specify an IP address or an IP address:port combination (separated by a colon) in place of a machine name in any DTCL command that allows for the entry of a machine name. For example, all of the following login commands are acceptable.

- login alpha root \*\*\*\*\*
- login 216.234.244.47 root \*\*\*\*\*
- login 216.234.244.47:1205 root \*\*\*\*\*

If you have modified your ports so that your source, target and/or clients are running on different ports and you are using DTCL scripts, then you will need to modify the scripts to include the port values as noted above. This only applies to source/target/client combinations that are communicating with each other on different ports.

---

# Getting help for scripting commands

A listing of the DTCL commands and their syntax is available by typing the [help](#) command.

---

## Command

HELP

## Description

Displays the DTCL commands and their syntax

## Syntax

HELP

## Notes

- Press any key to scroll through the list of commands.
  - Press q to exit the help function.
  - You can also type dtcl help from the directory where the Double-Take Availability program files are installed to display the DTCL commands and their syntax.
-

---

## Chapter 2 Replication sets

Using the scripting commands, you can perform the following functions to manage your replication sets.

- [Creating a replication set](#)
- [Modifying a replication set](#)
- [Deleting a replication set](#)
- [Calculating replication set size](#)

### Creating a replication set

Before you can establish a connection, you must create a replication set.

1. Use the `repset create` command to create a new replication set.

---

#### Command

REPSET CREATE

#### Description

Creates a replication set

#### Syntax

REPSET CREATE <*name*>

#### Options

*name*—Name of the replication set

#### Examples

repset create DataFiles

#### Notes

- The name of the replication set should not be a Double-Take Availability keyword. These are any DTCL command (source, target, and so on.) or any DTCL shortcut command (env, mon, rep, and so on).
  - Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.
-

2. Specify it as the active replication set by using the [repset use](#) command.

---

**Command**

REPSET USE

**Description**

Specifies a replication set as the active replication set

**Syntax**

REPSET USE *<repset>*

**Options**

*repset*—Name of the replication set

**Examples**

repset use DataFiles

**Notes**

Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.

---

3. Define a replication set rule by using the [repset rule add](#) command.

---

**Command**

REPSET RULE ADD

**Description**

Adds a rule to a replication set. A rule is the specification of a path including volume, directories, wild cards, and/or file names.

**Syntax**

REPSET RULE ADD *<path>* [INClude|EXClude] [,  
RECURSive|NONRECURSive] [TO *<repset>*]

**Options**

- *path*—Volume, directory, wild card, and/or file name
- INClude—Include the specified path in the replication set
- EXClude—Exclude the specified path in the replication set
- RECURSive—All subdirectories and files of the specified path are recursively included or excluded
- NONRECURSive—No subdirectories and files of the specified path are included or excluded
- *repset*—Name of the replication set

## Examples

- repset rule add "/data" to DataFiles
- repset rule add "/temp" exc rec to DataFiles

## Notes

- The default settings for this command are include and recursive.
  - The options include/exclude and recursive/nonrecursive can be used in any combination and in any order. The first option does not require a comma, but the second option does require a comma before the option.
  - If you do not specify a replication set name, the current replication set will be used.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

4. Repeat the [repset rule add](#) command to completely define your replication set.
  5. If you need to see the devices available, use the [device list](#) command.
- 

## Command

DEVICE LIST

## Description

Displays the block devices available on a machine

## Syntax

DEVICE LIST <[filter](#)> [ON <[machine](#)>]

## Options

- [filter](#)—Use one of the following options for the device filter
- ALL—Lists all unique devices on the specified machine. If there is no machine specified, the source, if designated, will be tried first. The target, if designated, will be tried second.
- OKSOURCE—List all devices on the specified source that are replication capable. If a target is specified, an error will be returned.
- OKTARGET—List all devices on the specified target that are capable of being used as a target path. If a source is specified, an error will be returned.
- [machine](#)—Name of the machine

## Examples

device list all on alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

- 
6. If you need to remove a rule, use the [repset rule remove](#) command.
- 

**Command**

REPSET RULE REMOVE

**Description**

Removes a rule from a replication set

**Syntax**

REPSET RULE REMove <[path](#)> [FROM <[repset](#)>]

**Options**

- [path](#)—Volume, directory, wild card, and/or file name
- [repset](#)—Name of the replication set

**Examples**

- `repset rule remove "/data" from DataFiles`
- `repset rule rem "/temp"`

**Notes**

- If you do not specify a replication set name, the current replication set will be used.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

7. After you have added all of the rules, save the replication set by using the [repset save](#) command.
- 

**Command**

REPSET SAVE

**Description**

Saves all replication set rules for the currently selected source

**Syntax**

REPSET SAVE

---



# Modifying a replication set

Make modifications to a replication set when you want to change the data you wish to protect.

1. View the replication sets associated with the active source machine by using the [repset list](#) command.

---

**Command**

REPSET LIST

**Description**

Lists all replication set names for the currently selected source

**Syntax**

REPSET LIST

---

2. Identify a replication set as active by using the [repset use](#) command.

---

**Command**

REPSET USE

**Description**

Specifies a replication set as the active replication set

**Syntax**

REPSET USE <[repset](#)>

**Options**

[repset](#)—Name of the replication set

**Examples**

repset use DataFiles

**Notes**

Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.

---

3. View the replication set's rules by using the [repset display](#) command.

---

**Command**

REPSET DISPLAY

**Description**

Displays the replication set rules

**Syntax**

REPSET DISPlay [[repset](#)]

**Options**

[repset](#)—Name of the replication set

**Examples**

- repset display DataFiles
- repset disp DataFiles

**Notes**

- If you do not specify a replication set name, the current replication set will be used.
  - Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

4. Remove the existing rule that you need to change using the [repset rule remove](#) command.

---

**Command**

REPSET RULE REMOVE

**Description**

Removes a rule from a replication set

**Syntax**

REPSET RULE REMove <[path](#)> [FROM <[repset](#)>]

**Options**

- [path](#)—Volume, directory, wild card, and/or file name
- [repset](#)—Name of the replication set

**Examples**

- repset rule remove "/data" from DataFiles
- repset rule rem "/temp"

## Notes

- If you do not specify a replication set name, the current replication set will be used.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

5. Add a new rule by using the `repset rule add` command.
- 

## Command

REPSET RULE ADD

## Description

Adds a rule to a replication set. A rule is the specification of a path including volume, directories, wild cards, and/or file names.

## Syntax

```
REPSET RULE ADD <path> [INClude|EXClude] [,  
RECURSive|NONRECURSive] [TO <repset>]
```

## Options

- **path**—Volume, directory, wild card, and/or file name
- **INClude**—Include the specified path in the replication set
- **EXClude**—Exclude the specified path in the replication set
- **RECURSive**—All subdirectories and files of the specified path are recursively included or excluded
- **NONRECURSive**—No subdirectories and files of the specified path are included or excluded
- **repset**—Name of the replication set

## Examples

- `repset rule add "/data" to DataFiles`
- `repset rule add "/temp" exc rec to DataFiles`

## Notes

- The default settings for this command are include and recursive.
  - The options include/exclude and recursive/nonrecursive can be used in any combination and in any order. The first option does not require a comma, but the second option does require a comma before the option.
  - If you do not specify a replication set name, the current replication set will be used.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
-

6. If you decide that you do not like the changes you have just made, you can use the [repset resync](#) command to restore the previously saved settings, undoing your modifications.

---

**Command**

REPSET RESYNC

**Description**

Retrieves the last saved replication set settings, clearing any unsaved changes

**Syntax**

REPSET RESYNC

---

7. Save the replication set by using the [repset save](#) command.

---

**Command**

REPSET SAVE

**Description**

Saves all replication set rules for the currently selected source

**Syntax**

REPSET SAVE

---

## Deleting a replication set

1. View the replication sets associated with the active source machine by using the [repset list](#) command.

---

**Command**

REPSET LIST

**Description**

Lists all replication set names for the currently selected source

**Syntax**

REPSET LIST

---

2. Delete the replication set by using the [repset delete](#) command.

---

**Command**

REPSET DELETE

**Description**

Deletes the specified replication set

**Syntax**

REPSET DELETE <[repset](#)>

**Options**

[repset](#)—Name of the replication set

**Examples**

- repset delete DataFiles
- repset del DataFiles

**Notes**

Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.

---

3. After deleting the replication set, use the [repset save](#) command so that the deletion will be registered by other Double-Take Availability clients.

---

**Command**

REPSET SAVE

**Description**

Saves all replication set rules for the currently selected source

**Syntax**

REPSET SAVE

---

# Calculating replication set size

1. Use the [repset calculate](#) command to manually calculate the size of a replication set.

---

## Command

REPSET CALCULATE

## Description

Calculates the size of a replication set

## Syntax

REPSET CALCulate [[repset](#)]

## Options

[repset](#)—Name of the replication set

## Examples

- repset calculate DataFiles
- repset calc DataFiles

## Notes

- If a replication set name is not specified, the active replication set will be used.
  - The results of the calculation are logged to the Double-Take Availability log file.
  - Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

2. Use the [CalculateOnConnect](#) option with the [get](#) and [set](#) commands to calculate the size of the replication set automatically upon connection.

---

## Command

GET

## Description

Requests the value of a Double-Take Availability program setting from the specified server

## Syntax

GET <[setting](#)> [[machine](#)]

## Options

- [setting](#)—See [Server settings](#) for a complete list of the Double-Take Availability program settings

<b>Examples</b>	<ul style="list-style-type: none"> <li>• <i>machine</i>—Name of the machine</li> </ul>
	<ul style="list-style-type: none"> <li>• get AutoRemirror</li> <li>• get MoveOrphansDir</li> </ul>
<b>Notes</b>	<ul style="list-style-type: none"> <li>• If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.</li> <li>• Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.</li> </ul>
<hr/>	
<b>Command</b>	SET
<b>Description</b>	Modifies the value of a Double-Take Availability program setting for the specified server
<b>Syntax</b>	SET < <i>setting</i> >=< <i>value</i> > [ <i>machine</i> ]
<b>Options</b>	<ul style="list-style-type: none"> <li>• <i>setting</i>—See <a href="#">Server settings</a> for a complete list of the Double-Take Availability program settings</li> <li>• <i>value</i>—See <a href="#">Server settings</a> for a complete list of the values for each Double-Take Availability program setting</li> <li>• <i>machine</i>—Name of the machine</li> </ul>
<b>Examples</b>	<ul style="list-style-type: none"> <li>• set AutoRemirror=3</li> <li>• set MoveOrphansDir="/OrphanFiles"</li> </ul>
<b>Notes</b>	<ul style="list-style-type: none"> <li>• Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.</li> <li>• If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.</li> <li>• Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.</li> </ul>



---

## Chapter 3 Establishing a connection

After you have created a replication set, you can establish a connection by connecting the replication set to a target.

1. If you do not know which replication set you will be connecting to the target machine, the [repset list](#) command will display the available replication sets for that source.

---

### Command

REPSET LIST

### Description

Lists all replication set names for the currently selected source

### Syntax

REPSET LIST

---

2. Connect the replication set to the target by using the [connect](#) command.

---

### Command

CONNECT

### Description

Establishes a connection between a replication set and a target machine

### Syntax

CONnect <[repset](#)> TO <[target](#)> MAP EXACT | MAP BASE <[target\\_path](#)> |  
MAP <[source\\_path](#)> TO <[target\\_path](#)> [...] [MIRror | NOMIRror] [, REPlicate  
| NOREPlicate] [, MONitor | NOMONitor] [, ORPHANS | NOORPHANS] [,  
COMPRESSion <[level](#)>] [CLEARRESTOREREQUIRED] [ROUTE=<[target\\_](#)  
[IP](#)>]

### Options

- [repset](#)—Name of the replication set
- [target](#)—Name of the target or an IP address on the target
- MAP EXACT—Specifies that the replication set data will be sent to the same logical volume on the target (/data and /files is copied to /data and /files, respectively)
- MAP BASE [target\\_path](#)—Substitute a complete path, including the volume, for [target\\_path](#) and the data will be replicated to [target\\_path](#)\SrcVolName on the target machine

- MAP *source\_path* TO *target\_path*—Custom location that specifies each directory on the source and where that data will be copied to on the target machine
- ...—Indicates that the source\_path TO target\_path option can be used more than once for each source directory in the replication set
- MIRror—Automatically initiates a mirror when the connection is established
- NOMIRror—Does not initiate a mirror when the connection is established
- REPligate—Automatically initiates replication when the connection is established
- NOREPligate—Does not initiate replication when the connection is established
- MONitor—Specifies that the target is going to monitor the specified source machine for failover. The source machine must have already been defined as a monitor machine.
- NOMONitor—Specifies that the target is not going to monitor the source machine for failover
- ORPHANS—Moves or deletes orphan files on the target. Orphan files will not be immediately processed when you create the connection. This setting is for processes that are run after a connection is already established (remirror, auto-remirror, verification, and so on).
- NOORPHANS—Does not move or delete orphan files on the target
- COMPRESSion *level*—Enables compression of data being sent to the target at the level specified. Valid levels are 1 (minimum), 2 (moderate), or 3 (maximum).
- CLEARRESTOREREQUIRED—Clears the restore required flag and initiates the connection
- ROUTE=*target\_ip*—Specifies the IP address on the target that will receive the incoming Double-Take Availability data

## Examples

- connect DataFiles to beta map exact
- connect UserData to beta map base d:\UserData\
- connect UserFiles to beta map exact orphans, compression 2
- con DataFiles to beta map exact mir, compress 1

## Notes

- The default settings for this command are mirror, replicate, nomonitor, and noorphans.
- The options (no)mirror, (no)replicate, (no)monitor, (no)orphans, and compression can be used in any combination and in any order. The first option does not require a comma, but the second and remaining options do require a comma before the option.
- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- If you are establishing a connection within a NAT or firewall environment, you will need to specify the target using the IP address and port number

(separated by a colon) of the router. For example, connect DataFiles to 10.10.1.57:1105 map exact.

- When scripting with this command, if a successful connection is established, the command will return a positive number, which is the connection ID assigned to that connection.
-

---

## Chapter 4 Simulating a connection

After you have created a replication set, you can simulate a connection by connecting the replication set to the TDU.

1. If you do not know which replication set you will be connecting to the target machine, the [reset list](#) command will display the available replication sets for that source.

---

### Command

RESET LIST

### Description

Lists all replication set names for the currently selected source

### Syntax

RESET LIST

---

2. To initiate a connection to the TDU, use the [connect TDU](#) command. This logs the connection statistics to the file specified. The remaining connection flags are identical to the standard connect command.

---

### Command

CONNECT TDU

### Description

Establishes a simulation connection between a replication set and the Throughput Diagnostics Utility. This connection imitate a normal connection without transmitting any data across the network.

### Syntax

CONNECT <[reset](#)> TO TDU <[filename](#)> [[connection\\_flags](#)]

### Options

- [reset](#)—Name of the replication set
- [filename](#)—Name of the file to store the connection statistics generated by the TDU
- [connection\\_flags](#)—The same options available in the standard [connect](#) command

### Examples

- connect DataFiles to TDU
- connect UserData to TDU map /userdata to /backup/userdata

## Notes

- The statistic file that the TDU creates can be viewed using DTStat. By default, the file is called statistic.sts. To view the statistic file, type DTStat -f <filename>.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
-

---

## Chapter 5 Connections

Using the scripting commands, you can perform the following functions to manage your connections.

- [Monitoring connections](#)
- [Queuing Double-Take Availability data](#)
- [Reconnecting automatically](#)
- [Pausing and resuming target processing](#)
- [Disconnecting a connection](#)

### Monitoring connections

1. Identify the connection ID that you want to monitor by using the `conID` command.

---

#### Command

CONID

#### Description

- Assigns the value of a connection ID to a variable
- Lists the target and replication set for all connection IDs for a source

#### Syntax

- `<variable>=CONID <repset> TO <target>`
- `CONID LIST [source]`

#### Options

- **variable**—Name of the variable that you want to use to store the connection ID
- **repset**—Replication set that was used to establish the connection
- **target**—Name of the target or an IP address on the target
- **source**—Name of the source or an IP address on the source

#### Examples

- `$con_id=conid DataFiles to beta`
- `$ConnectionNumber=conid UserData to beta`
- `conid list alpha conid list alpha`

#### Notes

- The `conid list` and `variable=conid` commands are two separate commands.
- Make sure there are no spaces before or after the equal sign when using the `variable=conid` command.
- If no machine name is specified in the `conid list` command, the active source will be used.

- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

2. Use the `status` command to display statistical and informational data.
- 

**Command**

STATUS

**Description**

Requests connection and statistical information

**Syntax**

STATUS CONnect <*con\_id*> | MIRror <*con\_id*> | REPlicate <*con\_id*> |  
TRANsmitt <*con\_id*> | TARget <*target*>

**Options**

- CONnect—Displays connection information for the connection ID specified
- *con\_id*—Connection ID assigned to the source/target connection
- MIRror—Displays mirroring information for the connection ID specified
- REPlicate—Displays replication information for the connection ID specified
- TRANsmitt—Displays transmission information for the connection ID specified
- TARget—Displays target state information for the target machine specified
- *target*—Name of the target or an IP address on the target

**Examples**

- status connect 1
- status rep 1
- status tar beta

**Notes**

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Queuing Double-Take Availability data

Use the [get](#) and [set](#) commands with the following options to configure how Double-Take Availability queues data.

- [QJournalDir](#)
- [QJournalFileSize](#)
- [QJournalFreeSpaceMin](#)
- [QJournalSpaceMax](#)
- [QmemoryBufferMax](#)
- [QueueSizeAlertThreshold](#)

---

## Command

GET

## Description

Requests the value of a Double-Take Availability program setting from the specified server

## Syntax

GET <*setting*> [*machine*]

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *machine*—Name of the machine

## Examples

- get AutoRemirror
- get MoveOrphansDir

## Notes

- If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.
- Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

## Command

SET



## Description

Modifies the value of a Double-Take Availability program setting for the specified server

## Syntax

SET <*setting*>=<*value*> [*machine*]

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *value*—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting
- *machine*—Name of the machine

## Examples

- set AutoRemirror=3
- set MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Reconnecting automatically

Use the [AutoReconnect](#) option with the [get](#) and [set](#) commands to configure Double-Take Availability to reconnect connections automatically.

---

## Command

GET

## Description

Requests the value of a Double-Take Availability program setting from the specified server

## Syntax

GET <[setting](#)> [[machine](#)]

## Options

- [setting](#)—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- [machine](#)—Name of the machine

## Examples

- get AutoRemirror
- get MoveOrphansDir

## Notes

- If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

---

## Command

SET

## Description

Modifies the value of a Double-Take Availability program setting for the specified server

## Syntax

SET <[setting](#)>=<[value](#)> [[machine](#)]

## Options

- **setting**—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- **value**—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting
- **machine**—Name of the machine

## Examples

- set AutoRemirror=3
- set MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Pausing and resuming target processing

1. Pause the execution of operations on the target by using the [pausetarget](#) command on a target that you are logged in to.

---

## Command

PAUSE TARGET

## Description

Allows you to pause the execution of Double-Take Availability operations on the target

## Syntax

PAUSETARGET <[target](#)> [FROM <[source](#)>]

## Options

- [target](#)—Name of the target or an IP address on the target
- [source](#)—Name of the source or an IP address on the source

## Examples

pausetarget beta

## Notes

- You must be logged on to the target machine for this command to work.
  - If the target machine has not been identified using the target command, you must specify the target name in the command.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

2. Resume the execution of operations on the target by using the [resumetarget](#) command on a target that you are logged in to.

---

## Command

RESUME TARGET

## Description

Allows you to resume the execution of Double-Take Availability operations on the target

## Syntax

RESUMETARGET <[target](#)> [FROM <[source](#)>]

## Options

- **target**—Name of the target or an IP address on the target
- **source**—Name of the source or an IP address on the source

## Examples

resumetarget beta

## Notes

- You must be logged on to the target machine for this command to work.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Disconnecting a connection

Use the [disconnect](#) command to disconnect the source/target connection.

---

## Command

DISCONNECT

## Description

Disconnects a specified source/target connection for the currently selected source

## Syntax

DISCONnect <[con\\_id](#) | \*>

## Options

- [con\\_id](#)—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

## Examples

- disconnect 1
  - disconnect \*
-

---

## Chapter 6 Mirroring

Using the scripting commands, you can perform the following functions to manage mirroring.

- [Stopping, starting, pausing, or resuming mirroring](#)
- [Mirroring automatically](#)
- [Removing orphan files](#)

### Stopping, starting, pausing, or resuming mirroring

You can stop, start, pause, or resume a mirror.

---

#### Command

MIRROR STOP

#### Description

Stops a mirror

#### Syntax

MIRror STOP <*con\_id* | \*>

#### Options

- *con\_id*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

#### Examples

- mirror stop 1
  - mir stop \*
- 

---

#### Command

MIRROR START

#### Description

Initiates the mirror process

#### Syntax

MIRror START <*con\_id*> [DIFFERENT [,NEWER] ,CHECKSUM | NOCHECKSUM]  
[ORPHANS | NOORPHANS] [CALCulate | NOCALCulate]  
[CLEARRESTOREREQUIRED]

## Options

- **con\_id**—Connection ID assigned to the source/target connection
- **DIFFERENT**—Mirrors only those files that are different based on the file date, time, and/or size
- **NEWER**—Mirrors only those files that are newer on the source than on the target
- **CHECKSUM**—Mirrors only those blocks that are different based on block checksum comparisons
- **NOCHECKSUM**—Does not perform a checksum comparison when mirroring files
- **ORPHANS**—Moves or deletes orphan files on the target
- **NOORPHANS**—Does not move or delete orphan files on the target
- **CALCulate**—Calculate the size of the replication set prior to mirroring
- **NOCALCulate**—Does not calculate the size of the replication set prior to mirroring
- **CLEARRESTOREREQUIRED**—Clears the restore required flag and initiates the mirror

## Examples

- mirror start 1 different, newer
- mir start 2 different, checksum orphans calc

## Notes

The default settings for this command are noorphans and calculate.

---

---

## Command

MIRROR PAUSE

## Description

Pauses a mirror that is in progress

## Syntax

MIRror PAUSE <**con\_id** | \*>

## Options

- **con\_id**—Connection ID assigned to the source/target connection
- **\***—Specifies all connection IDs

## Notes

- mirror pause 1
  - mir pause \*
-



---

**Command**

MIRROR RESUME

**Description**

Resumes a paused mirror

**Syntax**

MIRror RESUME <*con\_id* | \*>

**Options**

- *con\_id*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

**Notes**

- mirror resume 1
  - mir resume \*
-

# Mirroring automatically

Use the [AutoRemirror](#) option with the [get](#) and [set](#) commands to configure if a mirror is initiated automatically when a connection is automatically reconnected.

---

## Command

GET

## Description

Requests the value of a Double-Take Availability program setting from the specified server

## Syntax

GET <[setting](#)> [[machine](#)]

## Options

- [setting](#)—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- [machine](#)—Name of the machine

## Examples

- get AutoRemirror
- get MoveOrphansDir

## Notes

- If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

---

## Command

SET

## Description

Modifies the value of a Double-Take Availability program setting for the specified server

## Syntax

SET <[setting](#)>=<[value](#)> [[machine](#)]

## Options

- **setting**—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- **value**—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting
- **machine**—Name of the machine

## Examples

- set AutoRemirror=3
- set MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Removing orphan files

1. Use the [MoveOrphanedFiles](#), [MoveOprhansDir](#), [RemoveAllOrphans](#), [RemoveOrphansTime](#), and [LogAllOrphans](#) options with the `get` and `set` commands for orphan file configuration.

---

## Command

GET

## Description

Requests the value of a Double-Take Availability program setting from the specified server

## Syntax

GET <*setting*> [*machine*]

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *machine*—Name of the machine

## Examples

- `get AutoRemirror`
- `get MoveOrphansDir`

## Notes

- If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

---

## Command

SET

## Description

Modifies the value of a Double-Take Availability program setting for the specified server

## Syntax

SET <*setting*>=<*value*> [*machine*]

## Options

- **setting**—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- **value**—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting
- **machine**—Name of the machine

## Examples

- set AutoRemirror=3
- set MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

2. If you want to preview which files are identified as orphan files, use the [orphans preview](#) command. Check the log file on the target for the list of orphaned files.
- 

## Command

ORPHANS PREVIEW

## Description

Previews which files are orphan files on the target

## Syntax

ORPHANS PREVIEW *con\_id*

## Options

*con\_id*—Connection ID assigned to the source/target connection

## Examples

orphans preview 1

---

3. To remove orphan files manually, use the [orphans start](#) command.

---

**Command**

ORPHANS START

**Description**

Manual operation to remove any orphan files on the target

**Syntax**

ORPHANS START *con\_id* [CLEARRESTOREREQUIRED]

**Options**

- *con\_id*—Connection ID assigned to the source/target connection
- CLEARRESTOREREQUIRED—Clears the restore required flag and initiates the orphan operation

**Examples**

orphans start 1

---

4. If you want to stop removing orphan files after it has been started, use the [orphans stop](#) command.

---

**Command**

ORPHANS STOP

**Description**

Stops the process to remove orphan files on the target

**Syntax**

ORPHANS STOP *con\_id*

**Options**

*con\_id*—Connection ID assigned to the source/target connection

**Examples**

orphans stop 1

---

---

## Chapter 7 Replication

Using the scripting commands, you can perform the following functions to manage replication.

- [Starting replication](#)
- [Inserting tasks during replication](#)

### Starting replication

Start replication by using the [replication start](#) command.

---

#### Command

REPLICATION START

#### Description

Initiates the replication process

#### Syntax

REplication START <*conid* | \*> [CLEARRESTOREREQUIRED]

#### Options

- *conid*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs
- CLEARRESTOREREQUIRED—Clears the restore required flag and initiates replication

#### Examples

- replication start 1
  - rep start \*
-

# Inserting tasks during replication

Task command processing is a Double-Take Availability feature that allows you to insert and run tasks at various points during the replication of data. Because the tasks are user-defined, you can achieve a wide variety of goals with this feature. For example, you might insert a task to create a snapshot or run a backup on the target after a certain segment of data from the source has been applied on the target. This allows you to coordinate a point-in-time backup with real-time replication.

Task command processing can be enabled from the Replication Console, but it can only be initiated through a scripting command.

If you disable this option on a source server, you can still submit tasks to be processed on a target, although task command processing must be enabled on the target.

Because Double-Take Availability replication follows the same write sequence within and across multiple files, it provides complete data integrity at all times. At any given moment, the target represents a single point in time from the source, which makes the target crash consistent. But for some applications, crash consistency may not be adequate. You may require that the source data be in a quiescent (latent) state, similar to an application checkpoint. You need to be able to identify when the application is stable, which is usually when all of the data has been written to disk. This can be triggered by stopping the service. With task command processing, you can stop the source service just long enough to identify that stopped point in time as a stable state, insert a task at that point into the Double-Take Availability replication queue to trigger a backup or snapshot on the target, and then restart the service. Here is how the process would work.

1. Double-Take Availability and an application are both running on the source. Only Double-Take Availability is running on the target.
2. The application data is changing on the source and Double-Take Availability is capturing those data changes and transmitting them to the target.
3. A script is launched (either manually or perhaps by a scheduler program) that stops the application service on the source, pauses to give the service time to shutdown and write the data to disk, initiates a Double-Take Availability task command, and then restarts the application service on the source.
4. The Double-Take Availability task command is transmitted, inline with the source replication data, to the target.
5. The data is applied on the target as it is received. Since the task command was inserted inline, the replication data from the source is applied to the target first. When the target gets to the Double-Take Availability task command, the target data will be in the exact same state as the source data when the source application service was stopped. Since this was a stable point on the source, it is also a stable point on the target.
6. The target processes the Double-Take Availability task command and completes whatever task is defined, perhaps a snapshot or backup. Since the Double-Take Availability task command is user-defined, you can insert any valid executable or batch file.

Double-Take Availability task command processing must be enabled, and there must be an active Double-Take Availability connection for task command processing to function properly. To insert a task command, you would use the `queuetask` command.



---

## Command

QUEUETASK

## Description

Queues tasks inline with replication data

## Syntax

```
QueueTASK <job_name> TO <target> ONQueue = <task> [args] | ONTRANSMit =  
<task> [args] | ONRECeive = <task> [args] | ONEXECute = <task> [args]  
[TIMEOUT = <timeout>] [INTERACT | NOINTERACT]
```

## Options

- **job\_name**—Unique job name assigned to this task. This will be the identifier you see in the log files.
- **target**—Name of the target or an IP address on the target. The target is required even if you are only queuing a task to be executed on the source.
- **ONQueue**—Execute the specified task on the source machine as soon as the source receives and queues the task. During heavy replication, there may be a delay while the task is queued inline with the replication operations.
- **ONTRANSMit**—Execute the specified task on the source machine just before the source transmits the task to the target.
- **ONRECeive**—Execute the specified task on the target machine as soon as the target receives and queues the task.
- **ONEXECute**—Execute the specified task on the target when the target processes the task from the queue. Since the task is not executed until it is processed, if the target is paused, the task will be held in queue.
- **task**—The path and filename of the task to run relative to the machine it will be run on. Tasks include any valid executable or batch file. The executables or files must exist in the specified location on the machine where they will be executed
- **args**—Arguments or options which need to be supplied with the task. Multiple arguments can be supplied in a space-separated list enclosed in quotation marks.
- **TIMEOUT timeout**—Valid number followed by an optional time indicator indicating the length of time of pause while waiting for the task to complete. The valid time indicators include seconds, minutes, hours, and days. If you do not specify a time indicator, it will default to seconds. The number zero (0) indicates there is no timeout delay and the next operation is immediately processed. The keyword **FOREVER** indicates that the next operation is not processed until the task has completed execution. If you do not specify this option, the timeout will default to forever.
- **INTERACT**—Tasks interact with the desktop and, therefore, display on screen and run in the foreground
- **NOINTERACT**—Tasks do not interact with the desktop

## Examples

- `queuetask backup to beta onreceive=PauseAndBackup.bat onexecute=Resume.bat`
- `qtask backup to beta onrec=PauseAndBackup.bat onexec=resume.bat`

## Notes

- The default setting for this command is `nointeract`.
  - Any combination of one or more execution points can be used with the same `queuetask` command.
  - All script processing messages, including errors, can be viewed in the Double-Take Availability log.
  - Onqueue will still execute as soon as the task is placed on the queue even if transmission is stopped (manually stopped or paused, unmet scheduled transmission criteria, etc.). Any other option will not execute until transmission is restarted.
  - If your source is in a restore required state, any task placed on the queue will be executed immediately. Use caution when submitting tasks while in this state so that the target does not get inadvertently updated.
  - A task may be discarded if all connections to a target are manually disconnected, replication is stopped for all connections to a target, or an auto-disconnect occurs.
  - If a task is submitted after replication is stopped, the task will be executed immediately.
  - If you disable task command processing while tasks are in queue, those tasks will not be executed.
  - The user submitting the task command must be a member of the **Double-Take Admin** security group on both the source and target and the Double-Take service must have proper privileges to access the files or run the commands specified in the task.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

---

## Chapter 8 Verifying your target data

With scripting commands, verification can only be initiated after a connection is established.

1. Log on to the source using the `login` command.

---

### Command

LOGIN

### Description

Log on to a Double-Take Availability machine

### Syntax

LOGIN <*machine*> <*username*> <*password*>

### Options

- *machine*—Name of the machine
- *username*—Name of the user. The username is limited to 100 characters.
- *password*—Password associated with the user name. The password is limited to 100 characters.

### Examples

login alpha root \*\*\*\*\*

### Notes

- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - The password cannot be a Double-Take Availability keyword. These are any DTCL command (source, target, and so on.) or any DTCL shortcut command (env, mon, rep, and so on).
  - When scripting with this command, it will return one of three positive numbers: 0 (no access granted), 1 (monitor access granted), or 2 (full access granted).
-

2. Identify the source that you want to initiate verification for by using the [source](#) command.

---

**Command**

SOURCE

**Description**

Identifies a machine as the active source machine

**Syntax**

SOURce <[source](#)>

**Options**

[source](#)—Name of the source or an IP address on the source

**Examples**

- source alpha
- sou alpha

**Notes**

Source names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

3. Use the [VerifyLogName](#), [VerifyLogAppend](#), and [VerifyLogLimit](#) options with the [get](#) and [set](#) commands to configure the verification log.

---

**Command**

GET

**Description**

Requests the value of a Double-Take Availability program setting from the specified server

**Syntax**

GET <[setting](#)> [[machine](#)]

**Options**

- [setting](#)—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- [machine](#)—Name of the machine

**Examples**

- get AutoRemirror
- get MoveOrphansDir

## Notes

- If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

## Command

SET

## Description

Modifies the value of a Double-Take Availability program setting for the specified server

## Syntax

SET <*setting*>=<*value*> [*machine*]

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *value*—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting
- *machine*—Name of the machine

## Examples

- set AutoRemirror=3
- set MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

4. Initiate the verification process by using the [verify](#) command.
- 

## Command

VERIFY

## Description

Verifies the integrity of the data between the source and target machines

## Syntax

```
VERIFY <con_id> [ SYNC [,NEWER] | NOSYNC] [,CHECKSUM |  
NOCHECKSUM] [STARTTIME=<mm/dd/yy> [hh:mm]] [EVERY <number>  
<time_units>] [ORPHANS | NOORPHANS] [CLEARRESTOREREQUIRED]
```

## Options

- **con\_id**—Connection ID assigned to the source/target connection
- **SYNC**—Synchronizes any data that is different
- **NEWER**—Synchronizes only those files that are newer on the source than on the target
- **NOSYNC**—Do not synchronize any data that is different
- **CHECKSUM**—Compares and/or synchronizes those blocks that are different on the source than on the target based on checksum comparisons
- **NOCHECKSUM**—Does not perform a checksum comparison when comparing and/or synchronizing files
- **STARTTIME**—Starts the verification process at the time specified
- **mm/dd/yy**—Date in month/day/year format when the verification process will begin
- **hh:mm**—Time in hour:minute format using the 24-hour clock when the verification process will begin
- **EVERY**—Repeat the verification process at the frequency specified
- **number**—Length of time to repeat the verification process
- **time\_units**—Minutes (min), hours (hr), or days (day)
- **ORPHANS**—Moves or deletes orphan files on the target
- **NOORPHANS**—Does not move or delete orphan files on the target
- **CLEARRESTOREREQUIRED**—Clears the restore required flag and initiates the verification

## Examples

- verify 1
- verify 2 sync, newer
- verify 2 every 2 hr

## Notes

The default verification settings are sync, checksum, and noorphans.

---

---

## Chapter 9 Data transmission

Using the scripting commands, you can perform the following functions to manage your Double-Take Availability transmissions.

- [Stopping, starting, pausing, and resuming transmission](#)
- [Scheduling data transmission](#)
- [Limiting transmission bandwidth](#)
- [Compressing data for transmission](#)

### Stopping, starting, pausing, and resuming transmission

You can stop, start, pause, or resume transmission.

1. To start the transmission process, use the [transmission start](#) command.

---

#### Command

TRANSMISSION START

#### Description

Initiates the transmission pause

#### Syntax

TRANSmision START <*target*>

#### Options

*target*—Name of the target or an IP address on the target.

#### Examples

- transmission start beta
- trans start beta

#### Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

2. To stop the transmission process, use the [transmission stop](#) command.

---

**Command**

TRANSMISSION STOP

**Description**

Stops the transmission pause

**Syntax**

TRANSmision STOP <*target*>

**Options**

*target*—Name of the target or an IP address on the target.

**Examples**

- transmission stop beta
- trans stop beta

**Notes**

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

3. To pause the transmission process, use the [transmission pause](#) command.

---

**Command**

TRANSMISSION PAUSE

**Description**

Pauses the transmission pause

**Syntax**

TRANSmision PAUSE <*target*>

**Options**

*target*—Name of the target or an IP address on the target.

**Examples**

- transmission pause beta
- trans pause beta

**Notes**

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---



- 
4. To resume the transmission process, use the [transmission resume](#) command.
- 

**Command**

TRANSMISSION RESUME

**Description**

Resumes a paused transmission pause

**Syntax**

TRANSmision RESUME <*target*>

**Options**

*target*—Name of the target or an IP address on the target.

**Examples**

- transmission resume beta
- trans resume beta

**Notes**

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Scheduling data transmission

You can clear an existing schedule, disable or enable the use of a schedule, and set start, stop, and window criteria.

1. To clear all existing transmission options, use the [schedule clear](#) command.



All transmission options are stored on the source machine until they are either cleared or the option is updated. At this time, there is not a command to display the existing transmission options.

---

---

## Command

SCHEDULE CLEAR

## Description

Clears the existing transmission schedule for the specified target

## Syntax

SCHEDULE <*target*> CLEAR

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- schedule beta clear
- sched beta clear

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

2. Enable transmission limiting when you want to apply any transmission options that have been configured. You can also disable the transmission options without losing your settings. Use the [schedule enable](#) or [schedule disable](#) commands to enable or disable transmission limiting.

---

## Command

SCHEDULE ENABLE

## Description

Enables the transmission schedule

## Syntax

SCHEDULE <*target*> ENABLE

---

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- schedule beta enable
- sched beta enable

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

## Command

SCHEDULE DISABLE

## Description

Disables the transmission schedule without clearing the schedule data

## Syntax

SCHEDULE *<target>* DISABLE

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- schedule beta disable
- sched beta disable

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

3. You can specify a start time, a repeat interval, and queue usage criteria using the [schedule start](#) command.
- 

## Command

SCHEDULE START

## Description

Sets criteria to start the transmission of data from the source to the target

## Syntax

```
SCHEDule <target> START [STARTTIME = <mm/dd/yy> <hh:mm>]  
[MEMLIMIT = <percent>] [QUEUESIZE = <bytes>] [EVERY <number>  
<time_units>]
```

## Options

- **target**—Name of the target or an IP address on the target.
- **mm/dd/yy**—Date in month/day/year format indicating when the transmission will begin
- **hh:mm**—Time in hour:minute format using the 24-hour clock indicating when the transmission will begin
- **percent**—Any number between 0 and 100 indicating the percentage of system memory that must be in use to initiate the transmission process
- **bytes**—Number of bytes that must be in the source disk queue to initiate the transmission process
- **number**—Any number indicating how often the transmission process will be repeated
- **time\_units**—Minutes (min), hours (hr), or days (day)

## Examples

- schedule beta start starttime=03/11/07 03:30, queuesize=10000, every 6 hr
- sched beta start queuesize=100000000

## Notes

- The start option EVERY cannot be used by itself and cannot be the first option in a string of options.
  - If you use more than one start option, the transmission will begin when the first start option value is met. Additionally, each option after the first must be separated by a comma, as illustrated in the Examples.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

4. You can specify a transmission duration and a maximum number of bytes using the [schedule end](#) command.
- 

## Command

```
SCHEDULE END
```

## Description

Sets criteria to end the transmission of data from the source to the target

## Syntax

```
SCHEDule <target> END [ DURATION = <number> <time_units> ] [  
BYTES = <bytes> ]
```

## Options

- **target**—Name of the target or an IP address on the target.
- **number**—Any number indicating the length of time before the transmission ends
- **time\_units**—Minutes (min), hours (hr), or days (day)
- **bytes**—Number of bytes transmitted before the transmission ends

## Examples

- schedule beta end duration=3 hr bytes=1500000
- sched beta end duration=6 hr

## Notes

- If you use both of the end options, duration and bytes, the transmission will be stopped when the first end option value is met.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

5. You can specify a transmission window using the **schedule window** command.
- 

## Command

SCHEDULE WINDOW

## Description

Sets criteria to only allow transmissions during a certain period of time

## Syntax

SCHEDule <**target**> WINDOW <**hh:mm**> TO <**hh:mm**>

## Options

- **target**—Name of the target or an IP address on the target.
- **hh:mm**—Time in hour:minute format using the 24-hour clock. The first time is when the transmission will begin and the second time is when the transmission will end.

## Examples

- schedule beta window 23:00 to 06:00
- sche beta window 20:00 to 4:00

## Notes

- Establishing a transmission window by itself is not sufficient to start a transmission. You will need to specify a start criteria.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

## ***Transmission schedule examples***

This section shows examples of how the different schedule commands could be used together. The numbers and times were randomly selected for these examples. Be sure to use values that work for your environment.

- **Example 1**—In this example, transmission is set to begin on January 12, 2009, at 10:30 p.m. and to end after 6 hours. This schedule is also set to repeat every day.

```
schedule alpha start starttime=1/12/09 22:30, every 1 day
```

```
schedule alpha end duration=6 hr
```

```
schedule alpha enable
```

If all the data is not transmitted within the 6-hour duration, the remaining data will remain in the queue and will be transmitted during the next scheduled transmission.

- **Example 2**—In this example, transmission is set to begin after the source queue contains approximately 40 MB of data, and transmission is set to end after approximately 50 MB of data have been sent from the source to the target.

```
schedule alpha start queuesize=40000000
```

```
schedule alpha end bytes=50000000
```

```
schedule alpha enable
```

If there is data remaining in the source queue after the transmission ends, the data will be sent when the source queue again reaches 40000000 bytes of data.

- **Example 3**—In this example, transmission is set to begin after the source queue contains approximately 50 MB of data, and transmission is set to end after 60 MB of data have been sent from the source to the target. However, transmission can only occur if the start criteria is met within the defined 6-hour window.

```
schedule alpha start queuesize=50000000
```

```
schedule alpha end bytes=60000000
```

```
schedule alpha window 22:00 to 04:00
```

```
schedule alpha enable
```

If additional data remains in the source queue after the transmission ends, that data will be sent when the start criteria is again met within the defined 6-hour window. However, if the start criteria is not met within the defined 6-hour window, data remains in the queue until the start criteria is met within the defined window.

At any time, transmission can be manually started, stopped, paused, or resumed regardless of scheduled transmission criteria.

# Limiting transmission bandwidth

With a fixed bandwidth limit, data will be transmitted at all times according to the user-specified bandwidth configuration. If you want to set a fixed bandwidth limit, use the [limit bandwidth](#) command.

---

## Command

LIMIT BANDWIDTH

## Description

Sets a fixed bandwidth limitation for transmitting data from the source to the target

## Syntax

LIMIT BANDWIDTH *<bytes>*, *<seconds>* TO *<target>*

## Options

- *bytes*—Number of bytes to be transmitted
- *seconds*—Maximum number of seconds to wait before transmitting again
- *target*—Name of the target or an IP address on the target.

## Examples

limit bandwidth 19300, 5 to beta

## Notes

- This command transmits in bursts, not bytes per seconds. The time identifies how long to wait before transmitting again. For example, if 5 seconds are specified and it only takes 2 seconds to send the specified bytes, Double-Take Availability will wait an additional 3 seconds before transmitting again.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

## Compressing data for transmission

1. To determine the current compression setting, use the [compression list](#) command. By default, compression is disabled.

---

**Command**

COMPRESSION LIST

**Description**

Identifies the compression level and if compression is enabled

**Syntax**

COMPRESSion LIST

---

2. To set the level of compression, use the [compression set](#) command.

---

**Command**

COMPRESSION SET

**Description**

Sets the compression level

**Syntax**

COMPRESSion SET <[con\\_id](#)> <[level](#)>

**Options**

- [con\\_id](#)—Connection ID assigned to the source/target connection
- [level](#)—Any whole number from 1 to 3 where 1 is minimum compression and 3 is maximum compression

**Examples**

compression set 1 2

**Notes**

This command only sets the level of compression. It does not initiate compression.

---



3. To initiate compression, use the [compression enable](#) command.

---

**Command**

COMPRESSION ENABLE

**Description**

Enables compression

**Syntax**

COMPRESSION enable <[con\\_id](#)>

**Options**

[con\\_id](#)—Connection ID assigned to the source/target connection

**Examples**

compression enable 1

---

4. If you need to stop compression, use the [compression disable](#) command.

---

**Command**

COMPRESSION DISABLE

**Description**

Disables compression

**Syntax**

COMPRESSION disable <[con\\_id](#)>

**Options**

[con\\_id](#)—Connection ID assigned to the source/target connection

**Examples**

compression disable 1

---

---

## Chapter 10 Restoring data

1. Log on to your target using the `login` command.

---

### Command

LOGIN

### Description

Log on to a Double-Take Availability machine

### Syntax

LOGIN <*machine*> <*username*> <*password*>

### Options

- *machine*—Name of the machine
- *username*—Name of the user. The username is limited to 100 characters.
- *password*—Password associated with the user name. The password is limited to 100 characters.

### Examples

login alpha root \*\*\*\*\*

### Notes

- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - The password cannot be a Double-Take Availability keyword. These are any DTCL command (source, target, and so on.) or any DTCL shortcut command (env, mon, rep, and so on).
  - When scripting with this command, it will return one of three positive numbers: 0 (no access granted), 1 (monitor access granted), or 2 (full access granted).
-

2. Identify your source that you will be restoring to by using the [source](#) command.

---

**Command**

SOURCE

**Description**

Identifies a machine as the active source machine

**Syntax**

SOURCE <[source](#)>

**Options**

[source](#)—Name of the source or an IP address on the source

**Examples**

- source alpha
- sou alpha

**Notes**

Source names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

3. Restore Double-Take Availability data from the target to the source by using the [restore](#) command.

---

**Command**

RESTORE

**Description**

Initiates the restoration process

**Syntax**

RESTORE <[repset](#)> FROM <[target](#)> ORIGINAL <[original\\_source](#)> [,  
OVERWRITE | NOOVERWRITE] [, OVERWRITENEWER |  
NOOVERWRITENEWER] [, USETARGETDB | NOUSETARGETDB] [,  
RESTOREDBTOO | NORESTOREDBTOO] [, CHECKSUM |  
NOCHECKSUM] [, ORPHANS | NOORPHANS]

**Options**

- [repset](#)—Name of the replication set
- [target](#)—Name of the target or an IP address on the target
- [original\\_source](#)—Name of the original source
- OVERWRITE—Overwrites files on the source
- NOOVERWRITE—Does not overwrite files on the source

- **OVERWRITENEWER**—Overwrites files on the source even if the source file is newer than on the target
- **NOOVERWRITENEWER**—Does not overwrite files on the source that are newer on the source than on the target
- **USETARGETDB**—Uses the replication set from the target machine
- **NOUSETARGETDB**—Uses the replication set from the source machine
- **RESTOREDBTOO**—Restores the replication set database from the target to the source
- **NORESTOREDBTOO**—Does not restore the replication set database from the target to the source
- **CHECKSUM**—Performs a block checksum comparison and only restores those blocks that are different
- **NOCHECKSUM**—Does not perform a block checksum comparison and restores those files that are different
- **ORPHANS**—Moves or deletes orphan files on the source
- **NOORPHANS**—Does not move or delete orphan files on the source

### Examples

restore DataFiles from beta ,overwritenewer ,usetargetdb

### Notes

- The default settings for this command are overwrite, overwritenewer, usetargetdb, restoredbtoo, and noorphans.
  - The options can be used in any combination and in any order. The first option does not require a comma, but the second and remaining options do require a comma before the option.
  - The source command is required before each use of the restore command.
  - This command requires the original source option.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - When scripting with this command, if a successful restoration connection is established, the command will return a positive number, which is the connection ID assigned to that connection.
-

4. After the restoration is complete, the connection will remain connected and continue replicating data changes from the target to the source until the restoration connection is manually disconnected on the target using the [disconnect](#) command.

---

**Command**

DISCONNECT

**Description**

Disconnects a specified source/target connection for the currently selected source

**Syntax**

DISCONnect <[con\\_id](#) | \*>

**Options**

- [con\\_id](#)—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

**Examples**

- disconnect 1
  - disconnect \*
-

---

## Chapter 11 Failover

Using the scripting commands, you can perform the following functions to manage failover.

- [Configuring failover monitoring](#)
- [Editing failover monitoring configuration](#)
- [Failing over](#)
- [Stopping failover monitoring](#)
- [Deleting failover monitoring configuration](#)

### Configuring failover monitoring

1. Specify a target machine by using the `target` command.

---

#### Command

TARGET

#### Description

Identifies a machine as the active target machine

#### Syntax

TARget <*target*>

#### Options

*target*—Name of the target or an IP address on the target

#### Examples

- target beta
- tar beta

#### Notes

- You must be logged into a machine using the login command before using the target command.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

2. Determine what NICs are available on the specified target by using the [niclist](#) command.

---

**Command**

NICLIST

**Description**

Displays the NICs available on the specified target machine. Each NIC is assigned an integer value and this value is used in the monitor move command.

**Syntax**

NICLIST [[target](#)]

**Options**

[target](#)—Name of the target or an IP address on the target

**Examples**

niclist beta

**Notes**

- If you do not specify a machine name, the value from the current target will be returned. If you have not identified a target, no data will be returned.
  - If you have not logged into the target machine, no data will be displayed.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

3. Establish a monitor to use with the remaining monitor commands by using the [monitor create](#) command.

---

**Command**

MONITOR CREATE

**Description**

Establishes a source as a failover monitor. This is the machine that will be monitored by a target in case it should experience a failure.

**Syntax**

MONitor CREATE <[source](#)>

**Options**

[source](#)—Name of the source or an IP address on the source

**Examples**

- monitor create alpha
- mon create alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

4. Specify that you want to use the monitor that was just created by using the [monitor use](#) command.
- 

## Command

MONITOR USE

## Description

Specifies the source machine designated as the monitor that will be used in subsequent monitor commands

## Syntax

MONitor USE <[monitor](#)>

## Options

[monitor](#)—Name of the source machine designated as the monitor .

## Examples

- monitor use alpha
- mon use alpha

## Notes

Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---



5. Configure the monitor settings (including the IP address to monitor, the target NIC that will assume the IP address when it fails, the monitor interval and missed packets) by using the **monitor move** command.

---

**Command**

MONITOR MOVE

**Description**

Designates the IP address that will be failed over to the specified target NIC

**Syntax**

MONitor MOVE <*IP\_address*> TO NIC <*target\_NIC*> INTERVAL <*interval*>  
TIMEOUT <*timeout*> | <NOTEST> [*monitor*]

**Options**

- *IP\_address*—The IP address which should be moved during failover
- *target\_NIC*—The integer value of the target NIC obtained from the niclist command
- INTERVAL *interval*—The frequency, in seconds, of the monitor requests sent to the source machine to see if it is online and active
- TIMEOUT *timeout*—The number of seconds before failover will occur. This number is reset to its maximum each time the source sends a response to the monitor request.
- NOTEST—Allows you to failover an IP address without sending monitor requests or expecting responses from the source. This option should only be used if you are monitoring multiple IP addresses but do not want to send monitor requests to each address.
- *monitor*—Name of the source machine designated as the monitor

**Examples**

- monitor move 205.31.2.57 to nic 1 interval 5 timeout 25
- monitor move 205.31.2.68 to nic 2 notest

**Notes**

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

6. To remove an IP address from an established monitor, use the [monitor remove](#) command.

---

**Command**

MONITOR REMOVE

**Description**

Removes an IP address that is currently being monitored

**Syntax**

MONitor REMove <[IP\\_address](#)> [[monitor](#)]

**Options**

- [IP\\_address](#)—The currently monitored IP address that should be removed
- [monitor](#)—Name of the source machine designated as the monitor

**Examples**

- monitor remove 205.31.2.57 alpha
- mon rem 205.31.2.68

**Notes**

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

7. Configure the failover settings by using the [monitor option](#) command.

---

**Command**

MONITOR OPTION

**Description**

Configures the settings to determine how failover will be performed

**Syntax**

MONitor OPTION [, MOVEADDRESSES | NOMOVEADDRESSES] [, FAILONE | FAILALL] [, FODELAY | NOFODELAY] [, FBDELAY | NOFBDELAY] [, APPLY | DISCARD | REVERT] [, INTERVENTION | NOINTERVENTION] [[monitor](#)]

**Options**

- MOVEADDRESSES—Moves the IP address(es) during failover
- NOMOVEADDRESSES—Does not move the IP address(es) during failover, however this option is currently not functioning and will move the addresses

- FAILONE—When multiple IP addresses exist on a monitor machine, only the failed address is failed over to the target machine
- FAILALL—When multiple IP addresses exist on a monitor machine, all of the addresses will fail over to the target machine even if only one address fails
- FODELAY—Guarantees that the pre-failover script has completed before failing over
- NOFODELAY—Does not guarantee that the pre-failover script has completed before failing over
- FBDELAY—Guarantees that the pre-failback script has completed before failing back
- NOFBDELAY—Does not guarantee that the pre-failback script has completed before failing back
- APPLY—When failover is triggered, apply the data that is in the target queue before beginning failover
- DISCARD—When failover is triggered, discard the data that is in the target queue and begin failover immediately
- REVERT—When failover is triggered, revert the target to the last known good Double-Take Availability state
- INTERVENTION—Specifies that network administrator intervention is required before failover begins
- NOINTERVENTION—Specifies that network administrator intervention is not required before failover begins
- *monitor*—Name of the source machine designated as the monitor

### Examples

- monitor option alpha
- mon option alpha
- monitor option failone, nointervention

### Notes

- The default settings are moveaddress, failall, fodelay, fbdelay, apply, and intervention.
  - The options can be used in any combination and in any order. The first option does not require a comma, but the second and remaining options do require a comma before the option.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

8. Specify any scripts by using the `monitor script add` command.

---

**Command**

MONITOR SCRIPT ADD

**Description**

Specifies the scripts that should be run during the failover and failback processes

**Syntax**

MONitor SCRIPT ADD <*type*> <*script\_name*> [ARGS=<*arguments*>]  
[*monitor*]

**Options**

- *type*—Any of the following script types
- PREFAILOVER—The file is a pre-failover script to be run on the target before failover
- POSTFAILOVER—The file is a post-failover script to be run on the target after failover
- PREFAILBACK—The file is a pre-failback script to be run on the target before failback
- POSTFAILBACK—The file is a post-failback script to be run on the target after failback
- SRCPOSTFAILBACK—The file is a post-failback script to be run on the source after failback
- *script\_name*—Full path and name of the script file
- *arguments*—Comma-separated list of valid arguments required to execute the script
- *monitor*—Name of the source machine designated as the monitor

**Examples**

- `monitor script add prefailback "/user/shared/prefailback"`
- `mon script add postfailback "/user/shared/postfailback"`

**Notes**

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

9. Remove any scripts by using the [monitor script remove](#) command.

---

**Command**

MONITOR SCRIPT REMOVE

**Description**

Specifies the scripts that should not be run during the failover and failback processes

**Syntax**

MONitor SCRIPT REMove <*type*> [*monitor*]

**Options**

- *type*—Any of the following script types
- PREFAILOVER—The file is a pre-failover script to be run on the target before failover
- POSTFAILOVER—The file is a post-failover script to be run on the target after failover
- PREFAILBACK—The file is a pre-failback script to be run on the target before failback
- POSTFAILBACK—The file is a post-failback script to be run on the target after failback
- SRCPOSTFAILBACK—The file is a post-failback script to be run on the source after failback
- *monitor*—Name of the source machine designated as the monitor

**Examples**

- monitor script remove prefailback
- mon script rem postfailover

**Notes**

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

10. Review your failover settings by using the [monitor display](#) command.

---

**Command**

MONITOR DISPLAY

**Description**

Displays the monitoring and failover configuration settings for the specified monitor machine

**Syntax**

MONitor DISPlay <[monitor](#)>

**Options**

[monitor](#)—Name of the source machine designated as the monitor

**Examples**

- monitor display alpha
- mon disp alpha

**Notes**

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Depending on your sequence of commands, you may need to use the [monitor get](#) command to specify an active monitor before using monitor display.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

11. Start failover monitoring by using the [monitor start](#) command.

---

**Command**

MONITOR START

**Description**

Initiates failover monitoring

**Syntax**

MONitor START <[monitor](#)> [ON <[target](#)>]

**Options**

- [monitor](#)—Name of the source machine designated as the monitor
- [target](#)—Name of the target or an IP address on the target

## Examples

- *monitor*—Name of the source machine designated as the monitor
- *target*—Name of the target or an IP address on the target

## Notes

Option names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Editing failover monitoring configuration

1. Retrieve your monitor information by using the [monitor get](#) command.

---

**Command**

MONITOR GET

**Description**

Identifies a machine as the active monitor machine

**Syntax**

MONitor GET <[target](#)>

**Options**

[target](#)—Name of the target or an IP address on the target

**Examples**

- monitor get beta
- mon get beta

**Notes**

Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

2. Review the current failover settings by using the [monitor display](#) command.

---

**Command**

MONITOR DISPLAY

**Description**

Displays the monitoring and failover configuration settings for the specified monitor machine

**Syntax**

MONitor DISPlay <[monitor](#)>

**Options**

[monitor](#)—Name of the source machine designated as the monitor

**Examples**

- monitor display alpha
- mon disp alpha



## Notes

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Depending on your sequence of commands, you may need to use the [monitor get](#) command to specify an active monitor before using monitor display.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

3. If you need to edit any of the failover settings, use the [failover configuration](#) commands used to establish the failover configuration.

# Failing over

Verify that the source machine is offline and not connected to the network to avoid IP address conflicts. Trigger failover using the [failover](#) command.

---

## Command

FAILOVER

## Description

Manually initiates the failover process for the specified monitor machine

## Syntax

FAILOVER <[monitor](#)> [ON <[target](#)>] [APPLY | DISCARD | REVERT]

## Options

- [monitor](#)—Name of the source machine designated as the monitor
- [target](#)—Name of the target or an IP address on the target
- APPLY—Apply the data that is in the target queue before beginning failover
- DISCARD—Discard the data that is in the target queue and begin failover immediately
- REVERT—Revert the target to the last known good Double-Take Availability state

## Examples

failover alpha on beta apply

## Notes

If failover is configured for manual intervention, you must open the Failover Control Center to access the intervention prompt.

---

# Stopping failover monitoring

1. Retrieve your monitor information by using the [monitor get](#) command.

---

## Command

MONITOR GET

## Description

Identifies a machine as the active monitor machine

## Syntax

MONitor GET <[target](#)>

## Options

[target](#)—Name of the target or an IP address on the target

## Examples

- monitor get beta
- mon get beta

## Notes

Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

2. To stop failover monitoring, use the [monitor stop](#) command.

---

## Command

MONITOR STOP

## Description

Stops monitoring the source machine for failure

## Syntax

MONitor STOP <[monitor](#)> [ON <[target](#)>]

## Options

- [monitor](#)—Name of the source machine designated as the monitor
- [target](#)—Name of the target or an IP address on the target

## Examples

- monitor stop alpha on beta
- mon stop alpha

**Notes**

Option names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Deleting failover monitoring configuration

1. Retrieve your monitor information by using the [monitor get](#) command.

---

## Command

MONITOR GET

## Description

Identifies a machine as the active monitor machine

## Syntax

MONitor GET <[target](#)>

## Options

[target](#)—Name of the target or an IP address on the target

## Examples

- monitor get beta
- mon get beta

## Notes

Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

2. To delete the monitor, use the [monitor delete](#) command.

---

## Command

MONITOR DELETE

## Description

Deletes the specified failover monitor and all of its parameters

## Syntax

MONitor DELete <[monitor](#)>

## Options

[monitor](#)—Name of the source machine designated as the monitor

## Examples

- monitor delete alpha
- mon del alpha

## Notes

- In order to successfully delete a monitor, the monitor must not be running on the server. Use the monitor stop command to ensure the monitor is not running.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

---

## Chapter 12 Failing back

1. If you are performing failback before restoring, verify that your source machine is not connected to the network and that the issue(s) that caused the failure is resolved. If you have already performed a restoration, your source should already be connected to the network using a unique identity.
2. Login to the target machine that is currently standing in for the failed source by using the [login](#) command.

---

### Command

LOGIN

### Description

Log on to a Double-Take Availability machine

### Syntax

LOGIN <*machine*> <*username*> <*password*>

### Options

- *machine*—Name of the machine
- *username*—Name of the user. The username is limited to 100 characters.
- *password*—Password associated with the user name. The password is limited to 100 characters.

### Examples

login alpha root \*\*\*\*\*

### Notes

- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - The password cannot be a Double-Take Availability keyword. These are any DTCL command (source, target, and so on.) or any DTCL shortcut command (env, mon, rep, and so on).
  - When scripting with this command, it will return one of three positive numbers: 0 (no access granted), 1 (monitor access granted), or 2 (full access granted).
-

3. Identify the machine you just logged into as the target by using the [target](#) command.

---

**Command**

TARGET

**Description**

Identifies a machine as the active target machine

**Syntax**

TARget <[target](#)>

**Options**

[target](#)—Name of the target or an IP address on the target

**Examples**

- target beta
- tar beta

**Notes**

- You must be logged into a machine using the login command before using the target command.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
- 

4. Retrieve your monitor information by using the [monitor get](#) command.

---

**Command**

MONITOR GET

**Description**

Identifies a machine as the active monitor machine

**Syntax**

MONitor GET <[target](#)>

**Options**

[target](#)—Name of the target or an IP address on the target

**Examples**

- monitor get beta
- mon get beta



## Notes

Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

5. Initiate failback by using the **failback** command. If you specified a pre-failback script in your failover configuration, that script will be executed at this time.
- 

## Command

FAILBACK

## Description

Initiates the failback process for the specified monitor machine

## Syntax

FAILBACK <*monitor*> [ON <*target*>] REMONITOR|NOREMONITOR

## Options

- *monitor*—Name of the source machine designated as the monitor
- *target*—Name of the target or an IP address on the target
- REMONITOR—Automatically continues monitoring the source machine after failback
- NOREMONITOR—Automatically discontinues monitoring the source machine after failback.

## Examples

failback alpha on beta remonitor

## Notes

- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - The source machine must be online and Double-Take Availability must be running to ensure that the source post-failback script can be started. If the source has not completed its boot process, the command to start the script may be lost and the script will not be initiated.
- 



The source machine must be online and Double-Take Availability must be running to ensure that the source post-failback script can be started. If the source has not completed its boot process, the command to start the script may be lost and the script will not be initiated.

---

---

## Chapter 13 Configuring server settings

While only a subset of the server settings are available through the Replication Console, all of the Double-Take Availability server settings are accessible through the `get` and `set` commands. See [Server settings](#) for a complete list of the server settings.

- **Retrieving setting values**—To retrieve the current value of a setting, use the `get` command. This command will return the value of the specified setting from the specified server.

---

### Command

GET

### Description

Requests the value of a Double-Take Availability program setting from the specified server

### Syntax

GET <*setting*> [*machine*]

### Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *machine*—Name of the machine

### Examples

- `get AutoRemirror`
- `get MoveOrphansDir`

### Notes

- If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.
- Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

- 
- **Updating setting values**—To update the value of a setting, use the `set` command. This command will modify the value of the specified setting on the specified server.

---

### Command

SET

### Description

Modifies the value of a Double-Take Availability program setting for the specified server

## Syntax

SET <*setting*>=<*value*> [*machine*]

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *value*—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting
- *machine*—Name of the machine

## Examples

- set AutoRemirror=3
- set MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Server settings

The following table lists all of the server settings, in decimal value.

---

## ActivationCode

**Description**—24-character Double-Take Availability activation code

**Values**—Unique value for each customer

**Default**—N/A

**GUI Setting**—Server Properties, Licensing tab, Activation Code

## AdapterFlags

**Description**—Specifies the adapter to use when establishing a connection. This option should not be changed.

**Values**—2 Encryption, 4 Network Data Representation

**Default**—4

**GUI Setting**—None

## Advertisement

This setting is no longer used.

## AllFailover

**Description**—Specifies which IP addresses to failover

**Values**—0 Failover only monitored IP addresses, 1 Failover all IP addresses

**Default**—1

**GUI Setting**—Failover Control Center, Monitor Settings, Items to Failover, IP Address(es)

## AllMustFail

**Description**—Specifies whether or not all IP addresses must fail for failover to take place

**Values**—0 any IP address can fail, 1 All IP addresses must fail

**Default**—1

**GUI Setting**—Failover Control Center, Monitor Settings, Failover Trigger, All Monitored IP Addresses Fail

## AutoReconnect

**Description**—Specifies whether to reinstate the target connection(s) when the source machine is brought online after a source machine failure

**Values**—0 Do not reconnect, 1 Reconnect

**Default**—1

**GUI Setting**—Server Properties, Setup tab, Source Module Startup Options, Automatically Reconnect During Source Initialization

### **AutoRemirror**

**Description**—Specifies whether to remirror when a source is brought online after an auto-disconnect

**Values**—0 Do not remirror, 1 Perform a file differences checksum mirror, 2 Perform a full mirror, 3 Perform a file differences mirror, 4 Perform a date comparison mirror and send data only if the source data is newer than the target data.

**Default**—1

**GUI Setting**—Server Properties, Setup tab, Source Module Startup Options, Perform Remirror After Auto-Reconnect

### **AutoRemirrorRetry**

**Description**—Specifies how often, in seconds, the source should check for connections that have been reconnected but still need to be remirrored

**Values**—any integer

**Default**—30

**GUI Setting**—None

### **AutoRetransmit**

**Description**—Determines whether or not a source that has lost its connection with a target will attempt to reconnect to the target

**Values**—0 Do not attempt to reconnect, 1 Attempt to reconnect

**Default**—1

**GUI Setting**—None

### **BackupDir**

**Description**—Location on the target of the backup of the replication sets

**Values**—any valid path

**Default**—the location where the Double-Take Availability files were installed

**GUI Setting**—None

### **CalculateOnConnect**

**Description**—Specifies whether or not the replication set size should be calculated on connection

**Values**—0 Do not calculate on connection, 1 Calculate on connection

**Default**—1

**GUI Setting**—Connection Manager, Mirroring tab, Calculate Replication Set size on connection

### **CaseSensitiveRepSetQueries**

**Description**—This entry is no longer used.

### **ChecksumAll**

**Description**—Setting to allow for the difference checksum option on mirror, verify, or restore to ignore the date, time, and size of the file and perform a checksum calculation on all files

**Values**—0 Checksum using date, time, size comparison, 1 Checksum all files regardless of the date, time, or file size

**Default**—1

**GUI Setting**—Server Properties, Source tab, Mirroring or Verify, Block Checksum All Files on a Difference Mirror

### **Cleaner**

**Description**—Specifies if a clean mirror will delete files on the target before mirroring

**Values**—0 Do not delete files before mirroring, 1 Delete files before mirroring

**Default**—0

**GUI Setting**—None

**Notes**—This option is only valid if you have this option enabled and use the clean option with the DTCL mirror command.

### **ClientLog**

This setting is no longer used.

### **ClientLogName**

This setting is no longer used.

### **ConnectionFile**

**Description**—Name of the database file containing connection information

**Values**—any valid file name

**Default**—connect.sts

**GUI Setting**—Server Properties, Database tab, Database Files, Connection

### **DataPath**

**Description**—The location of the Double-Take Availability file attribute, replication set, connection, and schedule database files

**Values**—any valid path

**Default**—the location where the Double-Take Availability files were installed

**GUI Setting**—Server Properties, Database tab, Database Files, Folder

## **DefaultProtocol**

**Description**—The default protocol

**Values**—1 IP protocol

**Default**—1

**GUI Setting**—Server Properties, Network tab, Interface, Default Protocol

**Notes**—TCP/IP is the only protocol currently supported

## **DirUNetPort**

**Description**—Port used for directed UDP communications

**Values**—1025 - 65535

**Default**—1505

**GUI Setting**—Server Properties, Network tab, Interface, Network, Status Listen Port and Replication Console, File, Options, Network, Status Transmit Port

**Notes**—If you change this value, the source service must be stopped and restarted to implement the change.

## **DisableAttributeReplication**

**Description**—Specifies whether or not attributes (user, group, or other permissions) are replicated to the target

**Values**—0 Enable attribute replication, 1 Disable attribute replication

**Default**—0

**GUI Setting**—None

## **EnablePerformanceTracking**

**Description**—This entry will be used in the future.

## **EnableTaskCmdProcessing**

**Description**—Queues tasks inline with replication data

**Values**—0 Disable task command processing, 1 Enable task command processing

**Default**—0

**GUI Setting**—Server Properties, Setup tab, Setup Options, Enable Task Command Processing

## ExtendedAttributes

**Description**—Specifies whether or not extended attributes are replicated to the target

**Values**—0 Extended attributes are not mirrored or replicated, 1 Extended attributes are mirrored and replicated

**Default**—0

**GUI Setting**—None

## ExtensionNumber

**Description**—Used by the Double-Take Availability log files.

## FileQueueSize

**Description**—When a mirror is started, one thread reads from the disk and builds the file queue. Another set of threads reads files off of the queue and sends them to the target. This setting is the maximum size of the queue in entries. If you had 100 files to be mirrored and this was set to 16 (the default value), the first thread would fill the queue to a maximum of 16 entries.

**Values**—1 - 65535

**Default**—16

**GUI Setting**—None

**Notes**—This value must be set prior to starting the mirror process. The higher the number, the more memory that is used.

## HBExternalRate

**Description**—Number of seconds between heartbeats

**Values**—0 - 65535

**Default**—3

**GUI Setting**—Server Properties, Network tab, Heartbeat, Time between Heartbeats

**Notes**—Double-Take Software recommends a value that is less than 10 (see HBTTL). Zero (0) turns the heartbeats off.

## HBInternalRate

**Description**—This entry is no longer used

## HBLoopback

**Description**—This entry is no longer used.

## HBTrace

**Description**—Specifies whether heartbeat debugging information is generated

**Values**—0 not generated, 1 Generated



**Default**—0

**GUI Setting**—None

## HBTTL

**Description**—Number of seconds without receiving a heartbeat before a remote machine is considered unavailable

**Values**—0 - 65535

**Default**—10

**GUI Setting**—None

## HPQueueRatio

**Description**—Ratio of replication packets to one mirror packet

**Values**—0 - 65535

**Default**—5

**GUI Setting**—Server Properties, Source tab, Queue Ratio, Replication Packets to 1 Mirror Packet

**Notes**—An HPQueueRatio of 5 indicates 5 replication packets to 1 mirror packet. If you change this value, the source service must be stopped and restarted to implement the change.

## IgnoreDeleteOps

**Description**—Specifies if file and directory delete operations will be replicated to the target

**Values**—0 Delete operations are replicated to the target, 1 Delete operations are not replicated to the target

**Default**—0

**GUI Setting**—Server Properties, Source tab, Replication, Ignore Delete Operations

## LoadSourceTarget

**Description**—Specifies the functionality of the Double-Take

**Values**—0 Neither the source nor target modules are loaded, 1 Only the source module is loaded, 2 Only the target module is loaded, 3 Both the source and target modules are loaded

**Default**—3

**GUI Setting**—None

**Notes**—This setting will not take effect until the Double-Take service has been restarted.

## LogAllOrphans

**Description**—Specifies whether orphan files are logged to the Double-Take Availability log on the target

**Values**—0 Do not log the orphan files to the Double-Take Availability log on the target, 1 Log the orphan files to the Double-Take Availability log on the target

**Default**—0

**GUI Setting**—Connection Manager/Restoration Manager, Orphans tab, Log Orphaned Files to Target List

## LogDir

**Description**—The location of the Double-Take Availability messages/alerts, verification, and statistics log files

**Values**—any valid path

**Default**—the location where the Double-Take Availability files were installed

**GUI Setting**—Server Properties, Logging tab, Folder

## LogFile

**Description**—The name of the Double-Take Availability messages/alerts log file

**Values**—any valid file name

**Default**—DTLog

**GUI Setting**—None

## LogMessageLevel

**Description**—Specifies the types of messages logged to the .dtl files

**Values**—0 No messages will be logged, 1 Only alert messages will be logged, 2 Alert and release messages will be logged, 3 Alert, release, and debug messages will be logged

**Default**—2

**GUI Setting**—None

## MaxChecksumBlocks

**Description**—Specifies the number of checksum values retrieved from the target

**Values**—any integer

**Default**—32

**GUI Setting**—None

## MaxConnections

**Description**—Number of network requests that can be processed simultaneously.

**Values**—0 - 65535

**Default**—5

**GUI Setting**—None

**Notes**—Double-Take Software recommends that you not change this value.

### **MaxLogFileSize**

**Description**—Maximum size, in bytes, of any .dtl log file

**Values**—limited by available disk space

**Default**—1048576

**GUI Setting**—Server Properties, Logging tab, Messages & Alerts, Maximum Length

### **MaxNumberOfLogFiles**

**Description**—Maximum number of .dtl log files that can exist at one time. When Double-Take Availability creates a new .dtl file, if this number is exceeded, the oldest .dtl file is deleted.

**Values**—1 - 999

**Default**—5

**GUI Setting**—Server Properties, Logging tab, Messages & Alerts, Maximum Files

### **MaxRemoveOrphansOpSize**

**Description**—Determines whether or not Double-Take Availability will send over multiple remove orphan operations. Double-Take Availability will send over the operations if a directory has more files than this number.

**Values**—0 - 131072

**Default**—1000

**GUI Setting**—None

### **MaxRetry**

**Description**—A generic, application wide setting specifying the number of retry attempts for processes such as creating sockets, starting the service, and so on

**Values**—any integer

**Default**—5

**GUI Setting**—None

### **MaxWriteChunkSize**

**Description**—Maximum merged op size (in bytes) used during replication

**Values**—1 - 131072

**Default**—65536

**GUI Setting**—None

### **MemoryQueueToDiskThreshold**

**Description**—A percentage of QmemoryBufferMax that will trigger queuing to disk.

**Values**—any valid percentage

**Default**—100

**GUI Setting**—None

### **MinCompressionFileSize**

**Description**—The minimum file size, in bytes, that will be compressed. Files smaller than this size will not be compressed.

**Values**—any file size

**Default**—1024

**GUI Setting**—None

### **MirrorChunkSize**

**Description**—Block size, in bytes, used in the mirroring process

**Values**—1 - 1048576

**Default**—65536

**GUI Setting**—Server Properties, Source tab, Mirror Queue, Size of Mirror Packets

**Notes**—A higher block size value gives you better throughput, but only to a certain point, then it starts using more memory (this has to do with the way memory is allocated and deallocated). A lower block size value produces slower throughput, but uses memory efficiently.

### **MirrorOverwrite**

**Description**—Determines if the mirror process overwrites existing files

**Values**—0 never overwrite, 1 always overwrite, 2 overwrite if older

**Default**—1

**GUI Setting**—None

### **MirrorPrompting**

**Description**—This entry is no longer used.

### **MirrorQueueLimit**

**Description**—Maximum number of mirror operations that can be queued on the source machine

**Values**—1 - 65535

**Default**—1000

**GUI Setting**—Server Properties, Source tab, Mirror Queue, Maximum Pending Mirror Operations

### **MirrorZeroKFiles**

**Description**—Specifies whether or not empty files, zero byte files, are included in a mirror

**Values**—0 Zero byte files are skipped and not mirrored to the target, 1 All files are mirrored to the target

**Default**—1

**GUI Setting**—None

**Notes**—If MirrorZeroKFiles is enabled (0), zero byte files are skipped during a full mirror, file differences mirror, and a verification with synchronization.

### **MissedPackets**

**Description**—Specifies the number of requests sent by the target that go unanswered by the source before failover occurs, when using network responses to monitor for failover

**Values**—1 - 65535

**Default**—5

**GUI Setting**—Failover Control Center, Monitor Settings, Missed Packets

### **MoveOrphanedFiles**

**Description**—Specifies if orphaned files are deleted or moved to the directory specified by MoveOrphansDir

**Values**—1 Move, 0 Delete

**Default**—0

**GUI Setting**—Connection/Restoration Manager, Orphans tab, Move/Delete Orphan Files

### **MoveOrphansDir**

**Description**—Indicates the name of the directory where orphaned files will be moved if MoveOrphanedFiles=1

**Values**—any valid path

**Default**—the location where the Double-Take Availability files were installed

**GUI Setting**—Connection/Restoration Manager, Orphans tab, Move Orphaned Files to following location

### **NetPort**

**Description**—Port connection for TCP communications

**Values**—1025 - 65535

**Default**—1500

**GUI Setting**—Server Properties, Network tab, Interface, Service Listen Port

**Notes**—If you change this value, the source service must be stopped and restarted to implement the change.

## **NetworkRetry**

**Description**—Specifies the interval, in seconds, at which Double-Take Availability will attempt to reconnect to the target

**Values**—any positive number

**Default**—10

**GUI Setting**—None

## **NetworkStatusInterval**

**Description**—An internal setting for network communications. Do not modify this setting.

## **NetworkTimeout**

**Description**—The maximum length of time, in seconds, to wait on a network connection. If data is not received over a network connection within the specified time limit, the connection is closed. During idle periods, Double-Take Availability sends small amounts of keep-alive data at an interval 1/6 of the NetworkTimeout value to keep the socket from being inadvertently closed.

**Values**—any integer

**Default**—120

**GUI Setting**—None

## **NodeLockedLicenseKey**

**Description**—24-character activation code for a node-locked license

**Values**—Unique value for each customer

**Default**—N/A

**GUI Setting**—Server Properties, Licensing tab, Additional Codes

## **OpBufferSize**

**Description**—Specifies the number of operations that can be stored in the memory queue prior to queuing to disk.

**Values**—0 There is no limit to the number of operations that can be stored in the memory queue, 1 or any larger integer

**Default**—0

**GUI Setting**—None

## PingFrequency

**Description**—Specifies, in seconds, how often a ping is sent to the source from a monitoring target

**Values**—1 - 65535

**Default**—5

**GUI Setting**—Failover Control Center, Monitor Settings, Monitor Interval

## PreFailbackWait

**Description**—Specifies whether or not to wait for the target pre-failback script to complete before finishing a failback

**Values**—0 Do not wait, 1 Wait

**Default**—0

**GUI Setting**—Failover Control Center, Monitor Settings, Scripts, Target, Pre-Failback, Delay failback until script completes

## PreFailoverWait

**Description**—Specifies whether or not to wait for the target pre-failover script to complete before finishing a failover

**Values**—0 Do not wait, 1 Wait

**Default**—0

**GUI Setting**—Failover Control Center, Monitor Settings, Scripts, Target, Pre-Failover, Delay failover until script completes

## QJournalDir

**Description**—The location where the queue is stored.

**Values**—any valid path

**Default**—the location specified during the installation

**GUI Setting**—Server Properties, Queue tab, Location, Folder

**Notes**—For best results and reliability, you should select a dedicated, non-boot volume.

## QJournalFileSize

**Description**—The size, in MB, of each queuing transaction log file.

**Values**—any valid file size, up to 4095 MB

**Default**—5

**GUI Setting**—None

## **QJournalFreeSpaceMin**

**Description**—The minimum amount of disk space, in MB, in the specified QJournalDir that must be available at all times.

**Values**—dependent on the amount of physical disk space available

**Default**—50

**GUI Setting**—Server Properties, Queue tab, Memory and Disk Usage, Minimum Free Space

**Notes**—The QJournalFreeSpaceMin should be less than the amount of physical disk space minus QJournalSpaceMax.

## **QJournalPreload**

**Description**—The number of operations being pulled from the disk queue at one time. Do not modify this setting.

## **QJournalSpaceMax**

**Description**—The maximum amount of disk space, in MB, in the specified QJournalDir that can be used for Double-Take Availability queuing. When this limit is reached, Double-Take Availability will automatically begin the auto-disconnect process.

**Values**—dependent on the amount of physical disk space available

**Default**—Unlimited

**GUI Setting**—Server Properties, Queue tab, Memory and Disk Usage, Maximum disk space for queue

**Notes**—The unlimited setting allows the disk queue usage to automatically expand whenever the available disk space expands. Setting this option to zero (0) disables disk queuing. Even if you are using the Unlimited option, Double-Take Availability will only store 16,384 log files. If you are using the default 5MB file size, this is approximately 80GB of data. If you anticipate needing to be able to queue more data than this, you should increase the size of the log files.

## **QLogWriteThrough**

**Description**—Specifies if the disk queues are write-through mode

**Values**—0 Disk queues are not write-through mode, 1 Disk queues are write-through mode

**Default**—0

**GUI Setting**—None

**Notes**—While write-through mode may decrease the frequency of auto-disconnects, it may also decrease the performance of the source server.



## QMemoryBufferMax

**Description**—The amount of system memory, in MB, that, when exceeded, will trigger queuing to disk.

**Values**—dependent on the amount of physical memory available; minimum of 32 MB

**Default**—128 or 512 MB, depending on your operating system

**GUI Setting**—Server Properties, Queue tab, Memory and Disk Usage, Maximum system memory for queue

## QueueSizeAlertThreshold

**Description**—The percentage of the queue that must be in use to trigger an alert message in the Double-Take Availability log

**Values**—any valid percentage

**Default**—50

**GUI Setting**—Server Properties, Queue tab, Queue Usage Alert Threshold, Alert at following queue usage percentage

## RemapLink

**Description**—Specifies how Double-Take Availability handles a soft link

**Values**—0 If a soft link exists in a replication set and points to a file or directory inside the replication set, the path contained in the link will retain its original mapping, 1 If a soft link exists in a replication set and points to a file or directory inside the replication set, Double-Take Availability will remap the path contained in that link based on the Double-Take Availability target path

**Default**—1

**GUI Setting**—None

## RemoveAllOrphans

**Description**—Specifies if all orphan files will be removed or only those based on RemoveOrphanTime

**Values**—0 Remove orphans based on the entry RemoveOrphansTime, 1 Remove all orphans

**Default**—1

**GUI Setting**—Connection/Restoration Manager, Orphans tab, Remove All Orphans

## RemoveOrphansTime

**Description**—Specifies the amount of time, in minutes, that must be expired before an orphan file is removed

**Values**—1 - 131072

**Default**—60

**GUI Setting**—Connection/Restoration Manager, Orphans tab, Remove Orphans not modified within the following time period

### **ReplaceTarget**

**Description**—Specifies whether or not to replace the target identity with the source identity during a failover

**Values**—0 Do not replace, 1 Replace

**Default**—0

**GUI Setting**—Failover Control Center, Monitor Settings, Failover Method

### **RepSetDBName**

**Description**—Name of the database that contains replication set information

**Values**—any valid file name

**Default**—DbITake.db

**GUI Setting**—Server Properties, Database tab, Database Files, Replication Set

### **RestoreOverwrite**

**Description**—Determines if the restoration process overwrites existing files

**Values**—0 never overwrite, 1 always overwrite, 2 overwrite if older

**Default**—2

**GUI Setting**—Restoration Manager, Servers tab, Overwrite existing files during restore

### **RestorePrompting**

**Description**—This entry is no longer used.

### **RestoreSpecialExecutableHandling**

**Description**—Specifies if an alternate file is created and updated during a restoration for executables that are in use

**Values**—0 Do not use alternate files for executables that are in use, 1 Use alternate files for executables that are in use

**Default**—1

**GUI Setting**—Restoration Wizard, Restoration Options, Use alternate target files for executables that may be in use

### **SaveStatFile**

**Description**—Determines if the statistic.sts (statistics logging) file is saved or overwritten

**Values**—0 overwrite, 1 saved as statistic-old.sts

**Default**—1

**GUI Setting**—None

### ScheduleFile

**Description**—Name of the database file that contains transmission scheduling information

**Values**—any valid file name

**Default**—Schedule.sts

**GUI Setting**—Server Properties, Database tab, Database Files, Schedule

**Notes**—If you change the name of the database file, the source service must be stopped and restarted to start logging to the new database file.

### ScheduleInterval

**Description**—The number of seconds to wait before checking the transmission schedules to see if transmission should be started or stopped

**Values**—1 - 3600

**Default**—1

**GUI Setting**—None

### ShareUpdateInterval

**Description**—Specifies how often, in minutes, the share file will be sent to the target

**Values**—1 - 65535

**Default**—60

**GUI Setting**—None

### SkipCompressionFileExt

**Description**—A period delimited list of file types that are not compressed, even if compression is enabled.

**Values**—any period delimited list of file types

**Default**—mp3.exe.wmv.wma.qt.mpg.mpeg.zip.jpg.jpeg.tiff.tar.rar.cab

**GUI Setting**—None

### SourcePendingAcks

**Description**—The number of operations received by the target queue in which the source is waiting for a response

**Values**—100 - 20,000

**Default**—2000

**GUI Setting**—None

### StatsFileName

**Description**—Default file for logging statistics

**Values**—any valid file name

**Default**—statistic.sts

**GUI Setting**—Server Properties, Logging tab, Statistics, Filename

### StatsLoggingOn

**Description**—Specifies if Double-Take Availability logs statistics at startup

**Values**—0 Stats logging does not start when Double-Take Availability starts, 1 Stats logging starts when Double-Take Availability starts

**Default**—1

**GUI Setting**—Server Properties, Setup tab, Setup Options, Log Statistics Automatically

### StatsMaxFileSize

**Description**—Maximum size, in MB, for the statistic.sts file

**Values**—limited by available disk space

**Default**—10485760

**GUI Setting**—Server Properties, Logging tab, Statistics, Maximum Length

### StatsMaxObjects

**Description**—This entry is no longer used.

### StatsPort

**Description**—Port used by DTStat to gather Double-Take Availability statistics

**Values**—1025 - 65535

**Default**—1506

**GUI Setting**—None

### StatsShmSize

**Description**—This entry is no longer used.

### StatsWriteInterval

**Description**—Interval, in minutes, in which statistics are written to the statistic.sts file

**Values**—0 - 65535

**Default**—5

**GUI Setting**—Server Properties, Logging tab, Statistics, Write Interval

## SystemMemoryLimit

**Description**—Set by the Double-Take, each time it is started, to record the amount of available memory.

## TargetPaused

**Description**—Internal setting that indicates if the target machine is paused. Do not modify this setting.

## TargetPausedVirtual

**Description**—Internal setting that indicates which target machines are paused. Do not modify this setting.

## TCPBufferSize

**Description**—Size of the TCP/IP buffer in bytes.

**Values**—4096-7500000

**Default**—375000

**GUI Setting**—Server Properties, Network tab, TCP Window Size

**Notes**—This is an operating system buffer, not a Double-Take Availability buffer. If this option is set to zero (0), Linux kernel versions 2.6.7 or later can automatically tune this buffer setting for best server performance. Therefore, the recommended setting is 0 for automatic tuning, if you are using a version 2.6.7 or later Linux kernel. If you want to reduce or control network traffic, you can configure this option to a static size. The default is 375000 for a 1 GB network. Modifications should be relative to that speed using the calculation  $37500 * \text{network\_speed\_in\_bits\_per\_second} / 100$  Mbit.

## TGCloseDelay

**Description**—The length of time, in milliseconds, a file is held open on the target

**Values**—0 - 2000

**Default**—1000

**GUI Setting**—None

**Notes**—If disk caching on the target is disabled either manually or by default, the target system may be slow during a mirror. If so, decreasing this setting to 100, 10, and 0 will result in incremental improvements, with 0 returning the system performance to normal.

## TGExecutionRetryLimit

**Description**—The number of times an unfinished operation will be retried on the target before it is discarded. If this value is set to zero (0), an operation will never be discarded and will be retried on the target until it is applied.

**Values**—0 - 65536

**Default**—0

**GUI Setting**—None

### **TGMirrorCapacityHigh**

**Description**—Maximum percentage of system memory that can contain mirror data before the target signals the source to pause the sending of mirror operations.

**Values**—2-75

**Default**—20

**GUI Setting**—Server Properties, Target tab, Files, Target Mirror Capacity High Percentage

### **TGMirrorCapacityLow**

**Description**—Minimum percentage of system memory that can contain mirror data before the target signals the source to resume the sending of mirror operations.

**Values**—1-75

**Default**—15

**GUI Setting**—Server Properties, Target tab, Files, Target Mirror Capacity Low Percentage

**Notes**—The maximum value for TGMirrorCapacityLow is either 75 or TGMirrorCapacityHigh, whichever is lower.

### **TGRetryLocked**

**Description**—Minimum number of seconds to wait before retrying a failed operation on a target

**Values**—0-65536

**Default**—3

**GUI Setting**—Server Properties, Target tab, Queues, Retry Delay for Incomplete Operations

### **TGThreadCount**

**Description**—This setting is no longer used

### **TGWriteCache**

**Description**—Specifies whether or not Double-Take Availability uses the intermediate cache

**Values**—0 Bypass the intermediate cache and write directly to disk, 1 Do not bypass the intermediate cache

**Default**—1

**GUI Setting**—None

## UNetPort

**Description**—Port connection for UDP communications

**Values**—1025 - 65535

**Default**—1500

**GUI Setting**—Server Properties, Network tab, Interface, Heartbeat Transmit Port and Replication Console, File, Options, Configuration tab, Automatic Service Discovery, Heartbeat Advertisement, Port

**Notes**—If you change this value, the source service must be stopped and restarted to implement the change

## UpdateInterval

**Description**—Interval, in seconds, at which the Failover Control Center updates the monitored machines display

**Values**—1 - 9999

**Default**—1

**GUI Setting**—Failover Control Center, Settings, Refresh Rate

## UserIntervention

**Description**—Specifies whether or not user intervention is required to initiate a failover

**Values**—0 User intervention is not required, 1 User intervention is required

**Default**—1

**GUI Setting**—Failover Control Center, Monitor Settings, Manual Intervention

## UseShareFile

**Description**—Specifies whether to create and use a share file or to use the shares that are currently stored in the target memory

**Values**—0 Use the shares that are currently stored in the target memory, 1 Create and use a file containing the share information

**Default**—1

**GUI Setting**—Failover Control Center, Monitor Settings, Use .SHR Share Mapping File

## VerifyLogAppend

**Description**—Specifies whether the DTVerify.log file will be appended to or overwritten

**Values**—0 Overwrite 1 Append

**Default**—1

**GUI Setting**—Server Properties, Logging tab, Verification, Append

### **VerifyLogLimit**

**Description**—Maximum size of the DTVerify.log file in bytes

**Values**—limited by available hard drive space, up to 4 GB

**Default**—1048576

**GUI Setting**—Server Properties, Logging tab, Verification, Maximum Length

### **VerifyLogName**

**Description**—Name of the verification log file

**Values**—any valid file name

**Default**—DTVerify.log

**GUI Setting**—Server Properties, Logging tab, Verification, Filename

### **VerifyRetryInterval**

**Description**—The time, in minutes, between when one verification fails and a retry is scheduled to begin.

**Values**—any valid number

**Default**—3

**GUI Setting**—None

### **VerifyRetryLimit**

**Description**—The number of time a verification will be retried.

**Values**—any valid number

**Default**—5

**GUI Setting**—None

### **WarningPings**

**Description**—This entry is no longer used.

---



---

## Chapter 14 Scripting examples

Below are links to sample Double-Take Availability scripts. Most of the sample scripts must be modified. They cannot be used as-is. Modify them to fit your environment. If you need basic assistance with script modifications, contact Technical Support. Assistance with advanced scripting will be referred to Professional Services.

- [Creating and connecting a replication set](#)
- [Creating and connecting a replication set with failover monitoring](#)
- [Restoring a replication set](#)
- [Using variables to create and connect a replication set and run verification](#)
- [Controlling a mirror using flow control](#)
- [Using variables to pause a target](#)
- [Using variables to resume a target](#)
- [Creating a backup of the target by rotating connections](#)

### Creating and connecting a replication set

The following script will start a Double-Take Availability connection by creating a replication set called DataFiles on the source alpha and connecting it to the target beta.

```
source alpha;
repset create DataFiles;
repset rule add /files include, recursive;
repset rule add /files/users exclude;
repset rule add /data include, recursive;
repset save;
connect DataFiles to beta map exact;
```

## Creating and connecting a replication set with failover monitoring

The following script will start a Double-Take Availability connection by creating a replication set called DataFiles on the source alpha and connecting it to the target beta. This script will also configure and start failover monitoring.

```
source alpha;
repset create DataFiles;
repset rule add /files include, recursive;
repset rule add /files/users exclude;
repset rule add /data include, recursive;
repset save;
connect DataFiles to beta map exact;
target beta;
monitor create alpha;
monitor move "205.31.4.193" to nic 3 interval 5 timeout 25;
monitor start alpha;
```

## Restoring a replication set

The following script will restore the data in the DataFiles replication set from the target machine beta to the original source machine.

```
source alpha;
restore DataFiles to beta nooverwritenewer, usertargetdb;
```

## Using variables to create and connect a replication set and run verification

The following script uses variables to identify the source, target, and replication set. These variables are then used in the scripting commands. The \$connectionID variable will contain the connection ID for the connection established. This variable can then be used to establish a verification schedule.

```
$TheTarget = "beta";
$TheSource = "alpha";
$TheRepset = "DataFiles";
source $TheSource;
repset create $TheRepset;
repset rule add /files include, recursive;
repset rule add /files\users exclude;
repset rule add /data include, recursive;
repset save;
$ConnectionID = connect $TheRepset to $TheTarget map exact;
verify $ConnectionID sync, newer, checksum, every 12 hr;
```

## Controlling a mirror using flow control

The following script uses the IF conditional, FOR loop, and WHILE loop commands. In the following examples the FOR loop will start a mirror for all connections between 1 and 10. The IF conditional will return the error command "mirror failed to start" if a value of 0 is not obtained. The WHILE loop will wait for a mirror to end and disconnect as long as the returned value is less than 10. The IF conditional will return the error message "failed to disconnect" if a value of 0 is not obtained.

```
# Start mirror for all connections 1-10. #
# Notify user of any errors. #
SOURCE alpha;
FOR $x = 1 TO 10 DO
    $ret = MIRROR START $x;
    IF $ret != 0 THEN
        WRITE "Mirror failed to start";
        WRITE $x;
        WRITE $ret;
    END
END

# Wait for mirror to end and disconnect all connections 1-10. #
# Notify user of any errors. #
SOURCE alpha;
$x = 1;
WHILE $x <= 10 DO
    waitonmirror $x;
    $ret = disconnect $x;
    IF $ret != 0 THEN
        WRITE "Failed to disconnect";
        WRITE $x;
        WRITE $ret;
    END
    $x = $x + 1;
END
END
```

## Using variables to pause a target

The following script uses variables to identify the target and credentials to login to that target. These variables are then used in the scripting commands to pause the target.

```
$TheTarget = "beta";
$TheUser = "admin";
$ThePassword = "password";
login $TheTarget $TheUser $ThePassword;
target $TheTarget;
pausetarget $TheTarget;
```

## Using variables to resume a target

The following script uses variables to identify the target and credentials to login to that target. These variables are then used in the scripting commands to resume the target.

```
$TheTarget = "beta";
$TheUser = "admin";
$ThePassword = "password";
login $TheTarget $TheUser $ThePassword;
target $TheTarget;
resumetarget $TheTarget;
```

# Creating a backup of the target by rotating connections

The following two scripts create multiple copies of your data on the target, but only one copy is active at a time. This allows you the time and availability to backup the inactive copy of data. This is especially helpful when you have applications with interdependent files.

Backups occur sequentially from the first file to the last file. Therefore, when you are using applications that have interdependent files, such as a database application whose database and log files must be synchronized, Double-Take Availability cannot be actively updating files on the target while the backup is running, or there becomes an opportunity for interdependent files to become mismatched, causing a corrupt application on the backup. For example, suppose the following scenario occurs on a target machine that contains a replica of a database.

1. The backup process which is currently running sequentially through the files on the target, reaches the database log file and starts writing the log file to tape. At the same time, Double-Take Availability receives additional updates to the database. The database file on the target is updated, but since the log file is in use by the backup, the associated log operation is placed on the Double-Take Availability queue on the target.
2. When the log file is finished being backed up, the backup process continues with the next sequential file, which is not necessarily the database that corresponds with that log file.
3. Since the log file is no longer in use, Double-Take Availability applies the log operation from the Double-Take Availability queue.
4. Eventually, the backup process reaches the database file and writes it to tape.

At this point, the database file on the tape backup contains an extra update that the log file on the tape backup does not. The two files do not correspond, so the database on the tape backup will not be time consistent.

The two scripts identify an existing connection, disconnect it, establish a new connection, and initiate a difference mirror. The connections in the two scripts transmit the same data to two different locations, allowing you to backup the idle files on the inactive connection.

```
# Script 1 #
$TheSource = "alpha";
$TheTarget = "beta";
$TheUser = "root";
$ThePassword = "password";
$TheRepSet = "DataFiles";
login $TheSource $TheUser $ThePassword;
login $TheTarget $TheUser $ThePassword;
source $TheSource;
$FirstConnection = conid $TheRepSet to $TheTarget map base /first_location;
disconnect $FirstConnection;
$SecondConnection = connect $TheRepSet to $TheTarget map base /second_location, nomirror;
mirror start $SecondConnection different, checksum;
```

```
# Script 2 #
$TheSource = "alpha";
$TheTarget = "beta";
$TheUser = "root";
$ThePassword = "password";
$TheRepSet = "DataFiles";
login $TheSource $TheUser $ThePassword;
login $TheTarget $TheUser $ThePassword;
source $TheSource;
$SecondConnection = conid $TheRepSet to $TheTarget map base /second_location;
disconnect $SecondConnection;
$FirstConnection = connect $TheRepSet to $TheTarget map base /first_location, nomirror;
mirror start $FirstConnection different, checksum;
```

---

## Chapter 15 Scripting commands

Each scripting command includes the following.

- Command name
- Description of the command
- Exact syntax for using the command
- Options, if any
- Examples, if necessary
- Notes, if any

For details on the conventions used for the command syntax, see [Command syntax conventions](#).

- [Compression Disable](#)
- [Compression Enable](#)
- [Compression List](#)
- [Compression Set](#)
- [ConID](#)
- [Connect](#)
- [Connect TDU](#)
- [Device List](#)
- [Disconnect](#)
- [Email Add](#)
- [Email Disable](#)
- [Email Enable](#)
- [Email Filter](#)
- [Email From Address](#)
- [Email Get Email Config](#)
- [Email Mail Server](#)
- [Email Remove](#)
- [Email Set Filter Include](#)
- [Email Subject](#)
- [Email Test](#)
- [Environment](#)
- [Exit](#)
- [Failback](#)
- [Failover](#)
- [Get](#)
- [GetEnvStr](#)
- [Get Local](#)
- [Help](#)
- [Limit Bandwidth](#)

- [Load Source](#)
- [Load Target](#)
- [Login](#)
- [Logout](#)
- [Mirror Pause](#)
- [Mirror Resume](#)
- [Mirror Start](#)
- [Mirror Stop](#)
- [Monitor Clear](#)
- [Monitor Create](#)
- [Monitor Delete](#)
- [Monitor Display](#)
- [Monitor Get](#)
- [Monitor List](#)
- [Monitor Move](#)
- [Monitor Option](#)
- [Monitor Remove](#)
- [Monitor Script Add](#)
- [Monitor Script Remove](#)
- [Monitor Start](#)
- [Monitor Stop](#)
- [Monitor Use](#)
- [NIC List](#)
- [Orphans Preview](#)
- [Orphans Start](#)
- [Orphans Stop](#)
- [Pause Target](#)
- [Ping](#)
- [Queue Task](#)
- [Quit](#)
- [Replication Start](#)
- [Replication Stop](#)
- [Repset Calculate](#)
- [Repset Create](#)
- [Repset Delete](#)
- [Repset Display](#)
- [Repset List](#)
- [Repset Resync](#)
- [Repset Rule Add](#)
- [Repset Rule Remove](#)
- [Repset Save](#)

- [Repset Use](#)
- [Restore](#)
- [Resume Target](#)
- [Schedule Clear](#)
- [Schedule Disable](#)
- [Schedule Enable](#)
- [Schedule End](#)
- [Schedule Start](#)
- [Schedule Window](#)
- [Set](#)
- [Set Local](#)
- [Shutdown](#)
- [Source](#)
- [StatsLog Start](#)
- [StatsLog Status](#)
- [StatsLog Stop](#)
- [Status](#)
- [Target](#)
- [Test Connections](#)
- [Time Now](#)
- [Transmission Pause](#)
- [Transmission Resume](#)
- [Transmission Start](#)
- [Transmission Stop](#)
- [Unload Source](#)
- [Unload Target](#)
- [Verify](#)
- [Version](#)
- [Wait](#)
- [Wait on Mirror](#)
- [Wait on Restore](#)
- [Wait on Target](#)
- [Write](#)

# Compression Disable

---

## Command

COMPRESSION DISABLE

## Description

Disables compression

## Syntax

COMPRESSION DISABLE <*con\_id*>

## Options

*con\_id*—Connection ID assigned to the source/target connection

## Examples

compression disable 1

---



# Compression Enable

---

## Command

COMPRESSION ENABLE

## Description

Enables compression

## Syntax

COMPRESSIon ENABLE <*con\_id*>

## Options

*con\_id*—Connection ID assigned to the source/target connection

## Examples

compression enable 1

---

# Compression List

---

**Command**

COMPRESSION LIST

**Description**

Identifies the compression level and if compression is enabled

**Syntax**

COMPRESSion LIST

---

# Compression Set

---

## Command

COMPRESSION SET

## Description

Sets the compression level

## Syntax

COMPRESSION SET <*con\_id*> <*level*>

## Options

- *con\_id*—Connection ID assigned to the source/target connection
- *level*—Any whole number from 1 to 3 where 1 is minimum compression and 3 is maximum compression

## Examples

compression set 1 2

## Notes

This command only sets the level of compression. It does not initiate compression.

---

# ConID

---

## Command

CONID

## Description

- Assigns the value of a connection ID to a variable
- Lists the target and replication set for all connection IDs for a source

## Syntax

- **<variable>**=CONID **<repset>** TO **<target>**
- CONID LIST [**source**]

## Options

- **variable**—Name of the variable that you want to use to store the connection ID
- **repset**—Replication set that was used to establish the connection
- **target**—Name of the target or an IP address on the target
- **source**—Name of the source or an IP address on the source

## Examples

- \$con\_id=conid DataFiles to beta
- \$ConnectionNumber=conid UserData to beta
- conid list alpha conid list alpha

## Notes

- The conid list and variable=conid commands are two separate commands.
  - Make sure there are no spaces before or after the equal sign when using the variable=conid command.
  - If no machine name is specified in the conid list command, the active source will be used.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Connect

---

## Command

CONNECT

## Description

Establishes a connection between a replication set and a target machine

## Syntax

```
CONnect <repset> TO <target> MAP EXACT | MAP BASE <target_path> | MAP  
<source_path> TO <target_path> [...] [MIRror | NOMIRror] [, REPlicate |  
NOREPlicate] [, MONitor | NOMONitor] [, ORPHANS | NOORPHANS] [,  
COMPRESSion <level>] [CLEARRESTOREREQUIRED] [ROUTE=<target_IP>]
```

## Options

- **repset**—Name of the replication set
- **target**—Name of the target or an IP address on the target
- **MAP EXACT**—Specifies that the replication set data will be sent to the same logical volume on the target (/data and /files is copied to /data and /files, respectively)
- **MAP BASE target\_path**—Substitute a complete path, including the volume, for target\_path and the data will be replicated to target\_path\SrcVolName on the target machine
- **MAP source\_path TO target\_path**—Custom location that specifies each directory on the source and where that data will be copied to on the target machine
- **...**—Indicates that the source\_path TO target\_path option can be used more than once for each source directory in the replication set
- **MIRror**—Automatically initiates a mirror when the connection is established
- **NOMIRror**—Does not initiate a mirror when the connection is established
- **REPlicate**—Automatically initiates replication when the connection is established
- **NOREPlicate**—Does not initiate replication when the connection is established
- **MONitor**—Specifies that the target is going to monitor the specified source machine for failover. The source machine must have already been defined as a monitor machine.
- **NOMONitor**—Specifies that the target is not going to monitor the source machine for failover
- **ORPHANS**—Moves or deletes orphan files on the target. Orphan files will not be immediately processed when you create the connection. This setting is for processes that are run after a connection is already established (remirror, auto-remirror, verification, and so on).
- **NOORPHANS**—Does not move or delete orphan files on the target
- **COMPRESSion level**—Enables compression of data being sent to the target at the level specified. Valid levels are 1 (minimum), 2 (moderate), or 3 (maximum).

- CLEARRESTOREREQUIRED—Clears the restore required flag and initiates the connection
- ROUTE=*target\_ip*—Specifies the IP address on the target that will receive the incoming Double-Take Availability data

## Examples

- connect DataFiles to beta map exact
- connect UserData to beta map base d:\UserData\
- connect UserFiles to beta map exact orphans, compression 2
- con DataFiles to beta map exact mir, compress 1

## Notes

- The default settings for this command are mirror, replicate, nomonitor, and noorphans.
  - The options (no)mirror, (no)replicate, (no)monitor, (no)orphans, and compression can be used in any combination and in any order. The first option does not require a comma, but the second and remaining options do require a comma before the option.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - If you are establishing a connection within a NAT or firewall environment, you will need to specify the target using the IP address and port number (separated by a colon) of the router. For example, connect DataFiles to 10.10.1.57:1105 map exact.
  - When scripting with this command, if a successful connection is established, the command will return a positive number, which is the connection ID assigned to that connection.
-

# Connect TDU

---

## Command

CONNECT TDU

## Description

Establishes a simulation connection between a replication set and the Throughput Diagnostics Utility. This connection imitate a normal connection without transmitting any data across the network.

## Syntax

CONNECT <*repset*> TO TDU <*filename*> [*connection\_flags*]

## Options

- *repset*—Name of the replication set
- *filename*—Name of the file to store the connection statistics generated by the TDU
- *connection\_flags*—The same options available in the standard [connect](#) command

## Examples

- connect DataFiles to TDU
- connect UserData to TDU map /userdata to /backup/userdata

## Notes

- The statistic file that the TDU creates can be viewed using DTStat. By default, the file is called statistic.sts. To view the statistic file, type DTStat -f <filename>.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Device List

---

## Command

DEVICE LIST

## Description

Displays the block devices available on a machine

## Syntax

DEVICE LIST <*filter*> [ON <*machine*>]

## Options

- *filter*—Use one of the following options for the device filter
- ALL—Lists all unique devices on the specified machine. If there is no machine specified, the source, if designated, will be tried first. The target, if designated, will be tried second.
- OKSOURCE—List all devices on the specified source that are replication capable. If a target is specified, an error will be returned.
- OKTARGET—List all devices on the specified target that are capable of being used as a target path. If a source is specified, an error will be returned.
- *machine*—Name of the machine

## Examples

device list all on alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---



# Disconnect

---

## Command

DISCONNECT

## Description

Disconnects a specified source/target connection for the currently selected source

## Syntax

DISCONnect <*con\_id* | \* >

## Options

- *con\_id*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

## Examples

- disconnect 1
  - disconnect \*
-

# Email Add

---

## Command

EMAIL ADD

## Description

Adds an e-mail address to the e-mail notification distribution list

## Syntax

EMAIL ADD <*email\_address*>

## Options

*email\_address*—A valid e-mail address enclosed in quotation marks

## Examples

email add "root@domain.com"

## Notes

- You can repeat this command to add additional addresses to the distribution list.
  - You must have an active source specified for any e-mail command to work properly.
-

# Email Disable

---

## Command

EMAIL DISABLE

## Description

Disables e-mail notification

## Syntax

EMAIL DISABLE

## Notes

You must have an active source specified for any e-mail command to work properly.

---

# Email Enable

---

## Command

EMAIL ENABLE

## Description

Enables e-mail notification

## Syntax

EMAIL ENABLE

## Notes

You must have an active source specified for any e-mail command to work properly.

---

# Email Filter

---

## Command

EMAIL FILTER

## Description

Displays the filter configuration from the email setfilterinclude command

## Syntax

EMAIL FILTER

## Notes

You must have an active source specified for any e-mail command to work properly

---

# Email From Address

---

## Command

EMAIL FROMADDRESS

## Description

Specifies the e-mail address that will appear in the From field of Double-Take Availability generated e-mail messages

## Syntax

EMAIL FROMADDRESS <*email\_address*>

## Options

*email\_address*—A valid e-mail address enclosed in quotation marks

## Examples

email fromaddress "root@domain.com"

## Notes

You must have an active source specified for any e-mail command to work properly.

---

# Email Get Email Config

---

## Command

EMAIL GET EMAILCONFIG

## Description

Displays e-mail notification configuration information

## Syntax

EMAIL GET EMAILCONFIG

## Notes

You must have an active source specified for any e-mail command to work properly.

---

# Email Mail Server

---

## Command

EMAIL MAILSERVER

## Description

Specifies the name of the SMTP mail server for e-mail notification

## Syntax

EMAIL MAILSERVER <*server\_name* | *ip\_address*> [*username*][*password*]

## Options

- *server\_name*—Name of the SMTP mail server
- *ip\_address*—IP address of the SMTP mail server
- *username*—User ID required for SMTP server authentication
- *password*—Password associated with the specified user name

## Examples

email mailserver xchng root \*\*\*\*\*

## Notes

- Your SMTP server must support the LOGIN authentication to supply a username and password. If your server supports a different authentication method or does not support authentication, you may need to add the Double-Take Availability server as an authorized host for relaying e-mail messages. This option is not necessary if you are sending exclusively to e-mail addresses that the SMTP server is responsible for.
  - You must have an active source specified for any e-mail command to work properly.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-



# Email Remove

---

## Command

EMAIL REMOVE

## Description

Removes an e-mail address from the e-mail notification distribution list

## Syntax

EMAIL REMOVE <*email\_address*>

## Options

*email\_address*—An e-mail address, listed in the current distribution list, enclosed in quotation marks

## Examples

email remove "root@domain.com"

## Notes

You must have an active source specified for any e-mail command to work properly.

---

# Email Set Filter Include

---

## Command

EMAIL SETFILTERINCLUDE

## Description

Specifies which Event Viewer messages are sent via e-mail

## Syntax

EMAIL SETFILTERINCLUDE [INFO, WARNING, ERROR] [EXCLUDEIDS "<*ID1*,*ID2-ID4*,...>"]

## Options

- INFO—Information messages will be sent via e-mail
- WARNING—Warning messages will be sent via e-mail
- ERROR—Error messages will be sent via e-mail
- EXCLUDEIDS *ID1*,*ID2-ID4*,...—A comma separated list of IDs or ID ranges. A space should separate the EXCLUDEIDS switch from the list but within the list, there should be no spaces. Ranges are specified with a begin and end number and separated with a dash (-). The entire list should be enclosed in quotation marks.

## Examples

- email setfilterinclude warning, error excludeids "4007, 4012, 4015"
- email setfilterinclude excludeids "4000-4010, 5100"

## Notes

- The default settings for this command are info and warning. No IDs are excluded, by default.
  - The options info, warning, and error can be used in any combination and in any order. The first option does not require a comma, but the second and remaining options do require a comma before the option.
  - When changing the filter options, e-mail notification will automatically be stopped and restarted. If a failure occurs while stopping or restarting e-mail notification, Double-Take Availability will output a related error message
  - You must have an active source specified for any e-mail command to work properly.
-

# Email Subject

---

## Command

EMAIL SUBJECT

## Description

Specifies if additional text will be displayed in the subject of the e-mail message

## Syntax

EMAIL SUBJECT [PREFIX <*prefix*> | NOPREFIX] [DESCRIPTION | NODESCRIPTION]

## Options

- PREFIX *prefix*—Text which will be inserted at the front of the subject line for each Double-Take Availability generated e-mail message. This will help distinguish the Double-Take Availability messages from other messages.
- NOPREFIX—No prefix will be inserted at the front of the subject line for each Double-Take Availability generated e-mail message
- DESCRIPTION—Append the message description to the end of the subject line
- NODESCRIPTION—Do not append the message description to the end of the subject line

## Examples

- email subject prefix "Double-Take Availability Notification" description
- email subject prefix "Double-Take Availability Notification" nodescription

## Notes

- Prefix phrases that contain non-alphanumeric characters must be enclosed in quotation marks.
  - The entire subject line is limited to 150 characters.
  - You must have an active source specified for any e-mail command to work properly.
-

# Email Test

---

## Command

EMAIL TEST

## Description

Sends a test message to demonstrate e-mail notification

## Syntax

EMAIL TEST [SENDTO <*email\_address*[,*email\_address*,...]>] [TEXT <*message\_text*>]

## Options

- *email\_address*—Specifies the e-mail address(es) to send the test message to if you do not want to use the e-mail addresses configured with the EMAIL ADD command. This is a comma separated list of addresses. The entire list should be enclosed in quotation marks. A space should separate the SENDTO switch from the list of addresses but within the list, there should be no spaces.
- *message\_text*—Text to be displayed in the body of the test e-mail message. The test message is limited to 1024 characters and must be enclosed in quotation marks.

## Examples

- email test sendto "admin@domain.com, support@domain.com"
- email test sendto "admin@domain.com" text "This is a test message."

## Notes

- The default setting for this command are to use the addresses configured with the email add command. and to appends the word Test to the prefix defined in the email subject prefix command.
  - You must have an active source specified for any e-mail command to work properly.
-

# Environment

---

## Command

ENVIRONMENT

## Description

Displays a list of all Double-Take Availability machines available to the specified machine. Each machine is identified by machine name, IP addresses and whether or not the source and/or target modules are loaded. If no machine is specified, the information is provided for the machine currently specified as the source.

## Syntax

ENVironment [*machine*]

## Options

*machine*—Name of the machine to poll for environment information

## Examples

- environment alpha
  - env alpha
-

# Exit

---

**Command**

EXIT

**Description**

Exits the Command Line Interactive client

**Syntax**

EXIT

---

# Failback

---

## Command

FAILBACK

## Description

Initiates the failback process for the specified monitor machine

## Syntax

FAILBACK <*monitor*> [ON <*target*>] REMONITOR|NOREMONITOR

## Options

- *monitor*—Name of the source machine designated as the monitor
- *target*—Name of the target or an IP address on the target
- REMONITOR—Automatically continues monitoring the source machine after failback
- NOREMONITOR—Automatically discontinues monitoring the source machine after failback.

## Examples

failback alpha on beta remonitor

## Notes

- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - The source machine must be online and Double-Take Availability must be running to ensure that the source post-failback script can be started. If the source has not completed its boot process, the command to start the script may be lost and the script will not be initiated.
-

# Failover

---

## Command

FAILOVER

## Description

Manually initiates the failover process for the specified monitor machine

## Syntax

FAILOVER <*monitor*> [ON <*target*>] [APPLY | DISCARD | REVERT]

## Options

- *monitor*—Name of the source machine designated as the monitor
- *target*—Name of the target or an IP address on the target
- APPLY—Apply the data that is in the target queue before beginning failover
- DISCARD—Discard the data that is in the target queue and begin failover immediately
- REVERT—Revert the target to the last known good Double-Take Availability state

## Examples

failover alpha on beta apply

## Notes

If failover is configured for manual intervention, you must open the Failover Control Center to access the intervention prompt.

---



# Get

---

## Command

GET

## Description

Requests the value of a Double-Take Availability program setting from the specified server

## Syntax

GET <*setting*> [*machine*]

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *machine*—Name of the machine

## Examples

- get AutoRemirror
- get MoveOrphansDir

## Notes

- If you do not specify a machine name, the value from the current source will be returned. If you have not identified an active source, no data will be returned.
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# GetEnvStr

---

## Command

GETENVSTR

## Description

Retrieves an operating system environment variable and stores the value in a Double-Take Availability variable

## Syntax

GETENVSTR <*env\_variable*> \$<*variable\_name*>

## Options

- *env\_variable*—Name of the operating system environment variable that you want to store in the user defined variable
- *variable\_name*—Name of the variable you want to create. This variable will store the operating system environment variable specified.

## Examples

getenvstr computername \$server

## Notes

It is not necessary for either the environment or Double-Take Availability variable to exist when using this command. A non-existent environment variable will store a null string in the Double-Take Availability variable.

---

# Get Local

---

## Command

GETLOCAL

## Description

Requests the value of a Double-Take Availability program setting from the local machine

## Syntax

GETLOCAL <*setting*>

## Options

*setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings

## Examples

- getlocal AutoRemirror
  - getlocal MoveOrphansDir
-

# Help

---

## Command

HELP

## Description

Displays the DTCL commands and their syntax

## Syntax

HELP

## Notes

- Press any key to scroll through the list of commands.
  - Press q to exit the help function.
  - You can also type dtcl help from the directory where the Double-Take Availability program files are installed to display the DTCL commands and their syntax.
-

# Limit Bandwidth

---

## Command

LIMIT BANDWIDTH

## Description

Sets a fixed bandwidth limitation for transmitting data from the source to the target

## Syntax

LIMIT BANDWIDTH <*bytes*>, <*seconds*> TO <*target*>

## Options

- *bytes*—Number of bytes to be transmitted
- *seconds*—Maximum number of seconds to wait before transmitting again
- *target*—Name of the target or an IP address on the target.

## Examples

limit bandwidth 19300, 5 to beta

## Notes

- This command transmits in bursts, not bytes per seconds. The time identifies how long to wait before transmitting again. For example, if 5 seconds are specified and it only takes 2 seconds to send the specified bytes, Double-Take Availability will wait an additional 3 seconds before transmitting again.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Load Source

---

## Command

LOAD SOURCE

## Description

Loads the Double-Take Availability source module

## Syntax

LOAD SOURCE <*machine*>

## Options

*machine*—Name of the machine

## Examples

load source alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters should be enclosed in quotation marks.

---

# Load Target

---

## Command

LOAD TARGET

## Description

Loads the Double-Take Availability target module

## Syntax

LOAD TARGET <*machine*>

## Options

*machine*—Name of the machine

## Examples

load target beta

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters should be enclosed in quotation marks.

---

# Login

---

## Command

LOGIN

## Description

Log on to a Double-Take Availability machine

## Syntax

LOGIN <*machine*> <*username*> <*password*>

## Options

- *machine*—Name of the machine
- *username*—Name of the user. The username is limited to 100 characters.
- *password*—Password associated with the user name. The password is limited to 100 characters.

## Examples

login alpha root \*\*\*\*\*

## Notes

- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - The password cannot be a Double-Take Availability keyword. These are any DTCL command (source, target, and so on.) or any DTCL shortcut command (env, mon, rep, and so on).
  - When scripting with this command, it will return one of three positive numbers: 0 (no access granted), 1 (monitor access granted), or 2 (full access granted).
-



# Logout

---

## Command

LOGOUT

## Description

Logs off of a Double-Take Availability machine

## Syntax

LOGOUT <*machine*>

## Options

*machine*—Name of the machine

## Examples

logout alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Mirror Pause

---

## Command

MIRROR PAUSE

## Description

Pauses a mirror that is in progress

## Syntax

MIRror PAUSE <*con\_id* | \*>

## Options

- *con\_id*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

## Notes

- mirror pause 1
  - mir pause \*
-

# Mirror Resume

---

## Command

MIRROR RESUME

## Description

Resumes a paused mirror

## Syntax

MIRror RESUME <*con\_id* | \*>

## Options

- *con\_id*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

## Notes

- mirror resume 1
  - mir resume \*
-

# Mirror Start

---

## Command

MIRROR START

## Description

Initiates the mirror process

## Syntax

MIRror START <*con\_id*> [DIFFERENT [,NEWER] ,CHECKSUM | NOCHECKSUM]  
[ORPHANS | NOORPHANS] [CALCulate | NOCALCulate]  
[CLEARRESTOREREQUIRED]

## Options

- *con\_id*—Connection ID assigned to the source/target connection
- DIFFERENT—Mirrors only those files that are different based on the file date, time, and/or size
- NEWER—Mirrors only those files that are newer on the source than on the target
- CHECKSUM—Mirrors only those blocks that are different based on block checksum comparisons
- NOCHECKSUM—Does not perform a checksum comparison when mirroring files
- ORPHANS—Moves or deletes orphan files on the target
- NOORPHANS—Does not move or delete orphan files on the target
- CALCulate—Calculate the size of the replication set prior to mirroring
- NOCALCulate—Does not calculate the size of the replication set prior to mirroring
- CLEARRESTOREREQUIRED—Clears the restore required flag and initiates the mirror

## Examples

- mirror start 1 different, newer
- mir start 2 different, checksum orphans calc

## Notes

The default settings for this command are noorphans and calculate.

---

# Mirror Stop

---

## Command

MIRROR STOP

## Description

Stops a mirror

## Syntax

MIRror STOP <*con\_id* | \*>

## Options

- *con\_id*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

## Examples

- mirror stop 1
  - mir stop \*
-

# Monitor Clear

---

## Command

MONITOR CLEAR

## Description

Clears all of the failover configuration and monitoring parameters for the specified monitor machine

## Syntax

MONitor CLEAR [*monitor*]

## Options

*monitor*—Name of the source machine designated as the monitor

## Examples

- monitor clear alpha
- mon clear alpha

## Notes

- If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Monitor Create

---

## Command

MONITOR CREATE

## Description

Establishes a source as a failover monitor. This is the machine that will be monitored by a target in case it should experience a failure.

## Syntax

MONitor CREATE <*source*>

## Options

*source*—Name of the source or an IP address on the source

## Examples

- monitor create alpha
- mon create alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Monitor Delete

---

## Command

MONITOR DELETE

## Description

Deletes the specified failover monitor and all of its parameters

## Syntax

MONitor DELete <*monitor*>

## Options

*monitor*—Name of the source machine designated as the monitor

## Examples

- monitor delete alpha
- mon del alpha

## Notes

- In order to successfully delete a monitor, the monitor must not be running on the server. Use the monitor stop command to ensure the monitor is not running.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-



# Monitor Display

---

## Command

MONITOR DISPLAY

## Description

Displays the monitoring and failover configuration settings for the specified monitor machine

## Syntax

MONitor DISPlay <*monitor*>

## Options

*monitor*—Name of the source machine designated as the monitor

## Examples

- monitor display alpha
- mon disp alpha

## Notes

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Depending on your sequence of commands, you may need to use the [monitor get](#) command to specify an active monitor before using monitor display.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Monitor Get

---

## Command

MONITOR GET

## Description

Identifies a machine as the active monitor machine

## Syntax

MONitor GET <*target*>

## Options

*target*—Name of the target or an IP address on the target

## Examples

- monitor get beta
- mon get beta

## Notes

Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Monitor List

---

## Command

MONITOR LIST

## Description

Displays a list of all failover monitor machines

## Syntax

MONitor LIST

## Options

- monitor list
- mon list

## Notes

You must use the [monitor get](#) command to specify an active monitor before using monitor list.

---

# Monitor Move

---

## Command

MONITOR MOVE

## Description

Designates the IP address that will be failed over to the specified target NIC

## Syntax

MONitor MOVE <*IP\_address*> TO NIC <*target\_NIC*> INTERVAL <*interval*>  
TIMEOUT <*timeout*> | <NOTEST> [*monitor*]

## Options

- *IP\_address*—The IP address which should be moved during failover
- *target\_NIC*—The integer value of the target NIC obtained from the niclist command
- INTERVAL *interval*—The frequency, in seconds, of the monitor requests sent to the source machine to see if it is online and active
- TIMEOUT *timeout*—The number of seconds before failover will occur. This number is reset to its maximum each time the source sends a response to the monitor request.
- NOTEST—Allows you to failover an IP address without sending monitor requests or expecting responses from the source. This option should only be used if you are monitoring multiple IP addresses but do not want to send monitor requests to each address.
- *monitor*—Name of the source machine designated as the monitor

## Examples

- monitor move 205.31.2.57 to nic 1 interval 5 timeout 25
- monitor move 205.31.2.68 to nic 2 notest

## Notes

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Monitor Option

---

## Command

MONITOR OPTION

## Description

Configures the settings to determine how failover will be performed

## Syntax

MONitor OPTION [, MOVEADDRESSES | NOMOVEADDRESSES] [, FAILONE | FAILALL] [, FODELAY | NOFODELAY] [, FBDELAY | NOFBDELAY] [, APPLY | DISCARD | REVERT] [, INTERVENTION | NOINTERVENTION] [*monitor*]

## Options

- MOVEADDRESSES—Moves the IP address(es) during failover
- NOMOVEADDRESSES—Does not move the IP address(es) during failover, however this option is currently not functioning and will move the addresses
- FAILONE—When multiple IP addresses exist on a monitor machine, only the failed address is failed over to the target machine
- FAILALL—When multiple IP addresses exist on a monitor machine, all of the addresses will fail over to the target machine even if only one address fails
- FODELAY—Guarantees that the pre-failover script has completed before failing over
- NOFODELAY—Does not guarantee that the pre-failover script has completed before failing over
- FBDELAY—Guarantees that the pre-failback script has completed before failing back
- NOFBDELAY—Does not guarantee that the pre-failback script has completed before failing back
- APPLY—When failover is triggered, apply the data that is in the target queue before beginning failover
- DISCARD—When failover is triggered, discard the data that is in the target queue and begin failover immediately
- REVERT—When failover is triggered, revert the target to the last known good Double-Take Availability state
- INTERVENTION—Specifies that network administrator intervention is required before failover begins
- NOINTERVENTION—Specifies that network administrator intervention is not required before failover begins
- *monitor*—Name of the source machine designated as the monitor

## Examples

- monitor option alpha
- mon option alpha
- monitor option failone, nointervention

## Notes

- The default settings are moveaddress, failall, fodelay, fbdelay, apply, and intervention.
  - The options can be used in any combination and in any order. The first option does not require a comma, but the second and remaining options do require a comma before the option.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Monitor Remove

---

## Command

MONITOR REMOVE

## Description

Removes an IP address that is currently being monitored

## Syntax

MONitor REMove <*IP\_address*> [*monitor*]

## Options

- *IP\_address*—The currently monitored IP address that should be removed
- *monitor*—Name of the source machine designated as the monitor

## Examples

- monitor remove 205.31.2.57 alpha
- mon rem 205.31.2.68

## Notes

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Monitor Script Add

---

## Command

MONITOR SCRIPT ADD

## Description

Specifies the scripts that should be run during the failover and failback processes

## Syntax

MONitor SCRIPT ADD <*type*> <*script\_name*> [ARGS=<*arguments*>] [*monitor*]

## Options

- *type*—Any of the following script types
- PREFAILOVER—The file is a pre-failover script to be run on the target before failover
- POSTFAILOVER—The file is a post-failover script to be run on the target after failover
- PREFAILBACK—The file is a pre-failback script to be run on the target before failback
- POSTFAILBACK—The file is a post-failback script to be run on the target after failback
- SRCPOSTFAILBACK—The file is a post-failback script to be run on the source after failback
- *script\_name*—Full path and name of the script file
- *arguments*—Comma-separated list of valid arguments required to execute the script
- *monitor*—Name of the source machine designated as the monitor

## Examples

- monitor script add prefailback "/user/shared/prefailback"
- mon script add postfailback "/user/shared/postfailback"

## Notes

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
- Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.



# Monitor Script Remove

---

## Command

MONITOR SCRIPT REMOVE

## Description

Specifies the scripts that should not be run during the failover and failback processes

## Syntax

MONitor SCRIPT REMove <*type*> [*monitor*]

## Options

- *type*—Any of the following script types
- PREFAILOVER—The file is a pre-failover script to be run on the target before failover
- POSTFAILOVER—The file is a post-failover script to be run on the target after failover
- PREFAILBACK—The file is a pre-failback script to be run on the target before failback
- POSTFAILBACK—The file is a post-failback script to be run on the target after failback
- SRCPOSTFAILBACK—The file is a post-failback script to be run on the source after failback
- *monitor*—Name of the source machine designated as the monitor

## Examples

- monitor script remove prefailback
- mon script rem postfailover

## Notes

- If you do not specify a monitor, the current source designated as the monitor will be used. If you have not identified a monitor, you will receive an error message stating that a monitor has not been selected.
  - Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Monitor Start

---

## Command

MONITOR START

## Description

Initiates failover monitoring

## Syntax

MONitor START <*monitor*> [ON <*target*>]

## Options

- *monitor*—Name of the source machine designated as the monitor
- *target*—Name of the target or an IP address on the target

## Examples

- *monitor*—Name of the source machine designated as the monitor
- *target*—Name of the target or an IP address on the target

## Notes

Option names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Monitor Stop

---

## Command

MONITOR STOP

## Description

Stops monitoring the source machine for failure

## Syntax

MONitor STOP <*monitor*> [ON <*target*>]

## Options

- *monitor*—Name of the source machine designated as the monitor
- *target*—Name of the target or an IP address on the target

## Examples

- monitor stop alpha on beta
- mon stop alpha

## Notes

Option names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Monitor Use

---

## Command

MONITOR USE

## Description

Specifies the source machine designated as the monitor that will be used in subsequent monitor commands

## Syntax

MONitor USE <*monitor*>

## Options

*monitor*—Name of the source machine designated as the monitor .

## Examples

- monitor use alpha
- mon use alpha

## Notes

Monitor names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# NIC List

---

## Command

NICLIST

## Description

Displays the NICs available on the specified target machine. Each NIC is assigned an integer value and this value is used in the monitor move command.

## Syntax

NICLIST [*target*]

## Options

*target*—Name of the target or an IP address on the target

## Examples

niclist beta

## Notes

- If you do not specify a machine name, the value from the current target will be returned. If you have not identified a target, no data will be returned.
  - If you have not logged into the target machine, no data will be displayed.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Orphans Preview

---

## Command

ORPHANS PREVIEW

## Description

Previews which files are orphan files on the target

## Syntax

ORPHANS PREVIEW *con\_id*

## Options

*con\_id*—Connection ID assigned to the source/target connection

## Examples

orphans preview 1

---

# Orphans Start

---

## Command

ORPHANS START

## Description

Manual operation to remove any orphan files on the target

## Syntax

ORPHANS START *con\_id* [CLEARRESTOREREQUIRED]

## Options

- *con\_id*—Connection ID assigned to the source/target connection
- CLEARRESTOREREQUIRED—Clears the restore required flag and initiates the orphan operation

## Examples

orphans start 1

---

# Orphans Stop

---

## Command

ORPHANS STOP

## Description

Stops the process to remove orphan files on the target

## Syntax

ORPHANS STOP *con\_id*

## Options

*con\_id*—Connection ID assigned to the source/target connection

## Examples

orphans stop 1

---



# Pause Target

---

## Command

PAUSE TARGET

## Description

Allows you to pause the execution of Double-Take Availability operations on the target

## Syntax

PAUSETARGET <*target*> [FROM <*source*>]

## Options

- *target*—Name of the target or an IP address on the target
- *source*—Name of the source or an IP address on the source

## Examples

pausetarget beta

## Notes

- You must be logged on to the target machine for this command to work.
  - If the target machine has not been identified using the target command, you must specify the target name in the command.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Ping

---

## Command

PING

## Description

Checks a specified machine to see if Double-Take Availability is running

## Syntax

PING <*machine*>

## Options

*machine*—Name of the machine

## Examples

ping alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Queue Task

---

## Command

QUEUETASK

## Description

Queues tasks inline with replication data

## Syntax

```
QueueTASK <job_name> TO <target> ONQueue = <task> [args] | ONTRANSMit =  
<task> [args] | ONRECeive = <task> [args] | ONEXECute = <task> [args]  
[TIMEOUT = <timeout>] [INTERACT | NOINTERACT]
```

## Options

- **job\_name**—Unique job name assigned to this task. This will be the identifier you see in the log files.
- **target**—Name of the target or an IP address on the target. The target is required even if you are only queuing a task to be executed on the source.
- **ONQueue**—Execute the specified task on the source machine as soon as the source receives and queues the task. During heavy replication, there may be a delay while the task is queued inline with the replication operations.
- **ONTRANSMit**—Execute the specified task on the source machine just before the source transmits the task to the target.
- **ONRECeive**—Execute the specified task on the target machine as soon as the target receives and queues the task.
- **ONEXECute**—Execute the specified task on the target when the target processes the task from the queue. Since the task is not executed until it is processed, if the target is paused, the task will be held in queue.
- **task**—The path and filename of the task to run relative to the machine it will be run on. Tasks include any valid executable or batch file. The executables or files must exist in the specified location on the machine where they will be executed
- **args**—Arguments or options which need to be supplied with the task. Multiple arguments can be supplied in a space-separated list enclosed in quotation marks.
- **TIMEOUT timeout**—Valid number followed by an optional time indicator indicating the length of time of pause while waiting for the task to complete. The valid time indicators include seconds, minutes, hours, and days. If you do not specify a time indicator, it will default to seconds. The number zero (0) indicates there is no timeout delay and the next operation is immediately processed. The keyword FOREVER indicates that the next operation is not processed until the task has completed execution. If you do not specify this option, the timeout will default to forever.
- **INTERACT**—Tasks interact with the desktop and, therefore, display on screen and run in the foreground
- **NOINTERACT**—Tasks do not interact with the desktop

## Examples

- `queuetask backup to beta onreceive=PauseAndBackup.bat onexecute=Resume.bat`
- `qtask backup to beta onrec=PauseAndBackup.bat onexec=resume.bat`

## Notes

- The default setting for this command is `nointeract`.
  - Any combination of one or more execution points can be used with the same `queuetask` command.
  - All script processing messages, including errors, can be viewed in the Double-Take Availability log.
  - Onqueue will still execute as soon as the task is placed on the queue even if transmission is stopped (manually stopped or paused, unmet scheduled transmission criteria, etc.). Any other option will not execute until transmission is restarted.
  - If your source is in a restore required state, any task placed on the queue will be executed immediately. Use caution when submitting tasks while in this state so that the target does not get inadvertently updated.
  - A task may be discarded if all connections to a target are manually disconnected, replication is stopped for all connections to a target, or an auto-disconnect occurs.
  - If a task is submitted after replication is stopped, the task will be executed immediately.
  - If you disable task command processing while tasks are in queue, those tasks will not be executed.
  - The user submitting the task command must be a member of the **Double-Take Admin** security group on both the source and target and the Double-Take service must have proper privileges to access the files or run the commands specified in the task.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Quit

---

**Command**

QUIT

**Description**

Quits the Command Line Interactive client

**Syntax**

QUIT

---

# Replication Start

---

## Command

REPLICATION START

## Description

Initiates the replication process

## Syntax

REplication START <*conid* | \*> [CLEARRESTOREREQUIRED]

## Options

- *conid*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs
- CLEARRESTOREREQUIRED—Clears the restore required flag and initiates replication

## Examples

- replication start 1
  - rep start \*
-

# Replication Stop

---

## Command

REPLICATION STOP

## Description

Stops the replication process

## Syntax

REPLlication STOP <*conid* | \*>

## Options

- *conid*—Connection ID assigned to the source/target connection
- \*—Specifies all connection IDs

## Examples

- replication stop 1
  - rep stop \*
-

# Repset Calculate

---

## Command

REPSET CALCULATE

## Description

Calculates the size of a replication set

## Syntax

REPSET CALCulate [*repset*]

## Options

*repset*—Name of the replication set

## Examples

- repset calculate DataFiles
- repset calc DataFiles

## Notes

- If a replication set name is not specified, the active replication set will be used.
  - The results of the calculation are logged to the Double-Take Availability log file.
  - Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.
-



# Repset Create

---

## Command

REPSET CREATE

## Description

Creates a replication set

## Syntax

REPSET CREATE <*name*>

## Options

*name*—Name of the replication set

## Examples

repset create DataFiles

## Notes

- The name of the replication set should not be a Double-Take Availability keyword. These are any DTCL command (source, target, and so on.) or any DTCL shortcut command (env, mon, rep, and so on).
  - Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Repset Delete

---

## Command

REPSET DELETE

## Description

Deletes the specified replication set

## Syntax

REPSET DELeTe <*repset*>

## Options

*repset*—Name of the replication set

## Examples

- repset delete DataFiles
- repset del DataFiles

## Notes

Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Repset Display

---

## Command

REPSET DISPLAY

## Description

Displays the replication set rules

## Syntax

REPSET DISPlay [*repset*]

## Options

*repset*—Name of the replication set

## Examples

- repset display DataFiles
- repset disp DataFiles

## Notes

- If you do not specify a replication set name, the current replication set will be used.
  - Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Repset List

---

## Command

REPSET LIST

## Description

Lists all replication set names for the currently selected source

## Syntax

REPSET LIST

---

# Repset Resync

---

## Command

REPSET RESYNC

## Description

Retrieves the last saved replication set settings, clearing any unsaved changes

## Syntax

REPSET RESYNC

---

# Repset Rule Add

---

## Command

REPSET RULE ADD

## Description

Adds a rule to a replication set. A rule is the specification of a path including volume, directories, wild cards, and/or file names.

## Syntax

REPSET RULE ADD <*path*> [INClude|EXClude] [, RECURSive|NONRECURSive] [TO <*repset*>]

## Options

- *path*—Volume, directory, wild card, and/or file name
- INClude—Include the specified path in the replication set
- EXClude—Exclude the specified path in the replication set
- RECURSive—All subdirectories and files of the specified path are recursively included or excluded
- NONRECURSive—No subdirectories and files of the specified path are included or excluded
- *repset*—Name of the replication set

## Examples

- repset rule add "/data" to DataFiles
- repset rule add "/temp" exc rec to DataFiles

## Notes

- The default settings for this command are include and recursive.
  - The options include/exclude and recursive/nonrecursive can be used in any combination and in any order. The first option does not require a comma, but the second option does require a comma before the option.
  - If you do not specify a replication set name, the current replication set will be used.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Repset Rule Remove

---

## Command

REPSET RULE REMOVE

## Description

Removes a rule from a replication set

## Syntax

REPSET RULE REMove <path> [FROM <repset>]

## Options

- **path**—Volume, directory, wild card, and/or file name
- **repset**—Name of the replication set

## Examples

- repset rule remove "/data" from DataFiles
- repset rule rem "/temp"

## Notes

- If you do not specify a replication set name, the current replication set will be used.
  - Options that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Repset Save

---

## Command

REPSET SAVE

## Description

Saves all replication set rules for the currently selected source

## Syntax

REPSET SAVE

---



# Repset Use

---

## Command

REPSET USE

## Description

Specifies a replication set as the active replication set

## Syntax

REPSET USE <*repset*>

## Options

*repset*—Name of the replication set

## Examples

repset use DataFiles

## Notes

Replication set names that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Restore

---

## Command

RESTORE

## Description

Initiates the restoration process

## Syntax

```
RESTORE <repset> FROM <target> ORIGINAL <original_source> [,  
OVERWRITE | NOOVERWRITE] [, OVERWRITENEWER |  
NOOVERWRITENEWER] [, USETARGETDB | NOUSETARGETDB] [,  
RESTOREDBTOO | NORESTOREDBTOO] [, CHECKSUM | NOCHECKSUM][,  
ORPHANS | NOORPHANS]
```

## Options

- **repset**—Name of the replication set
- **target**—Name of the target or an IP address on the target
- **original\_source**—Name of the original source
- **OVERWRITE**—Overwrites files on the source
- **NOOVERWRITE**—Does not overwrite files on the source
- **OVERWRITENEWER**—Overwrites files on the source even if the source file is newer than on the target
- **NOOVERWRITENEWER**—Does not overwrite files on the source that are newer on the source than on the target
- **USETARGETDB**—Uses the replication set from the target machine
- **NOUSETARGETDB**—Uses the replication set from the source machine
- **RESTOREDBTOO**—Restores the replication set database from the target to the source
- **NORESTOREDBTOO**—Does not restore the replication set database from the target to the source
- **CHECKSUM**—Performs a block checksum comparison and only restores those blocks that are different
- **NOCHECKSUM**—Does not perform a block checksum comparison and restores those files that are different
- **ORPHANS**—Moves or deletes orphan files on the source
- **NOORPHANS**—Does not move or delete orphan files on the source

## Examples

restore DataFiles from beta ,overwritenewer ,usetargetdb

## Notes

- The default settings for this command are `overwrite`, `overwritenewer`, `usetargetdb`, `restoredbtoo`, and `noorphans`.
  - The options can be used in any combination and in any order. The first option does not require a comma, but the second and remaining options do require a comma before the option.
  - The source command is required before each use of the restore command.
  - This command requires the original source option.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
  - When scripting with this command, if a successful restoration connection is established, the command will return a positive number, which is the connection ID assigned to that connection.
-

# Resume Target

---

## Command

RESUME TARGET

## Description

Allows you to resume the execution of Double-Take Availability operations on the target

## Syntax

RESUMETARGET <*target*> [FROM <*source*>]

## Options

- *target*—Name of the target or an IP address on the target
- *source*—Name of the source or an IP address on the source

## Examples

resumetarget beta

## Notes

- You must be logged on to the target machine for this command to work.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Schedule Clear

---

## Command

SCHEDULE CLEAR

## Description

Clears the existing transmission schedule for the specified target

## Syntax

SCHEDule <*target*> CLEAR

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- schedule beta clear
- sched beta clear

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Schedule Disable

---

## Command

SCHEDULE DISABLE

## Description

Disables the transmission schedule without clearing the schedule data

## Syntax

SCHEDule <*target*> DISABLE

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- schedule beta disable
- sched beta disable

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Schedule Enable

---

## Command

SCHEDULE ENABLE

## Description

Enables the transmission schedule

## Syntax

SCHEDule <*target*> ENABLE

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- schedule beta enable
- sched beta enable

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Schedule End

---

## Command

SCHEDULE END

## Description

Sets criteria to end the transmission of data from the source to the target

## Syntax

SCHEDULE <*target*> END [ DURATION = <*number*> <*time\_units*> ] [ BYTES = <*bytes*> ]

## Options

- *target*—Name of the target or an IP address on the target.
- *number*—Any number indicating the length of time before the transmission ends
- *time\_units*—Minutes (min), hours (hr), or days (day)
- *bytes*—Number of bytes transmitted before the transmission ends

## Examples

- schedule beta end duration=3 hr bytes=1500000
- sched beta end duration=6 hr

## Notes

- If you use both of the end options, duration and bytes, the transmission will be stopped when the first end option value is met.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-



# Schedule Start

---

## Command

SCHEDULE START

## Description

Sets criteria to start the transmission of data from the source to the target

## Syntax

SCHEDULE <*target*> START [STARTTIME = <*mm/dd/yy*> <*hh:mm*>] [MEMLIMIT = <*percent*>] [QUEUESIZE = <*bytes*>] [EVERY <*number*> <*time\_units*>]

## Options

- *target*—Name of the target or an IP address on the target.
- *mm/dd/yy*—Date in month/day/year format indicating when the transmission will begin
- *hh:mm*—Time in hour:minute format using the 24-hour clock indicating when the transmission will begin
- *percent*—Any number between 0 and 100 indicating the percentage of system memory that must be in use to initiate the transmission process
- *bytes*—Number of bytes that must be in the source disk queue to initiate the transmission process
- *number*—Any number indicating how often the transmission process will be repeated
- *time\_units*—Minutes (min), hours (hr), or days (day)

## Examples

- schedule beta start starttime=03/11/07 03:30, queuesize=10000, every 6 hr
- sched beta start queuesize=100000000

## Notes

- The start option EVERY cannot be used by itself and cannot be the first option in a string of options.
  - If you use more than one start option, the transmission will begin when the first start option value is met. Additionally, each option after the first must be separated by a comma, as illustrated in the Examples.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Schedule Window

---

## Command

SCHEDULE WINDOW

## Description

Sets criteria to only allow transmissions during a certain period of time

## Syntax

SCHEDule <*target*> WINDOW <*hh:mm*> TO <*hh:mm*>

## Options

- *target*—Name of the target or an IP address on the target.
- *hh:mm*—Time in hour:minute format using the 24-hour clock. The first time is when the transmission will begin and the second time is when the transmission will end.

## Examples

- schedule beta window 23:00 to 06:00
- sche beta window 20:00 to 4:00

## Notes

- Establishing a transmission window by itself is not sufficient to start a transmission. You will need to specify a start criteria.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Set

---

## Command

SET

## Description

Modifies the value of a Double-Take Availability program setting for the specified server

## Syntax

SET <*setting*>=<*value*> [*machine*]

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *value*—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting
- *machine*—Name of the machine

## Examples

- set AutoRemirror=3
- set MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - If you do not specify a machine name, the value from the current source will be updated. If you have not identified an active source, no changes will be made.
  - Options, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Set Local

---

## Command

SET LOCAL

## Description

Modifies the value of a Double-Take Availability program setting for the local machine

## Syntax

SETLOCAL <*setting*>=<*value*>

## Options

- *setting*—See [Server settings](#) for a complete list of the Double-Take Availability program settings
- *value*—See [Server settings](#) for a complete list of the values for each Double-Take Availability program setting

## Examples

- setlocal AutoRemirror=3
- setlocal MoveOrphansDir="/OrphanFiles"

## Notes

- Some settings, although immediately applied to Double-Take Availability, will not take effect until the service is restarted.
  - Program setting values that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Shutdown

---

## Command

SHUTDOWN

## Description

Stops the Double-Take service .

## Syntax

SHUTDOWN <*machine*>

## Options

*machine*—Name of the machine

## Examples

shutdown alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Source

---

## Command

SOURCE

## Description

Identifies a machine as the active source machine

## Syntax

SOUrce <*source*>

## Options

*source*—Name of the source or an IP address on the source

## Examples

- source alpha
- sou alpha

## Notes

Source names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# StatsLog Start

---

## Command

STATSLOG START

## Description

Starts the DTStats statistics logging utility

## Syntax

STATSLOG START <*machine*> [TO <*filename*>] [EVERY <*minute*>] [MAXSIZE <*kilobytes*>]

## Options

- *machine*—Name of the machine
- *filename*—Any valid path and filename to which the statistical information will be logged
- *minutes*—The number of minutes to write to the log file
- *kilobytes*—The maximum file size in kilobytes

## Examples

statslog start alpha to "dtstat.sts" every 20 maxsize 200000

## Notes

- The default settings are the file name statistic.sts, a five (5) minute interval, and a file size of 10485760 bytes (10 MB).
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# StatsLog Status

---

## Command

STATSLOG STATUS

## Description

Identifies whether the DTStats statistics logging utility is currently running

## Syntax

STATSLOG STATUS <*machine*>

## Options

*machine*—Name of the machine

## Examples

statslog status alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---



# StatsLog Stop

---

## Command

STATSLOG STOP

## Description

Stops the DTStats statistics logging utility

## Syntax

STATSLOG STOP <*machine*>

## Options

*machine*—Name of the machine

## Examples

statslog stop alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Status

---

## Command

STATUS

## Description

Requests connection and statistical information

## Syntax

STATUS CONnect <*con\_id*> | MIRror <*con\_id*> | REPlicate <*con\_id*> | TRANsmit <*con\_id*> | TARget <*target*>

## Options

- CONnect—Displays connection information for the connection ID specified
- *con\_id*—Connection ID assigned to the source/target connection
- MIRror—Displays mirroring information for the connection ID specified
- REPlicate—Displays replication information for the connection ID specified
- TRANsmit—Displays transmission information for the connection ID specified
- TARget—Displays target state information for the target machine specified
- *target*—Name of the target or an IP address on the target

## Examples

- status connect 1
- status rep 1
- status tar beta

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Target

---

## Command

TARGET

## Description

Identifies a machine as the active target machine

## Syntax

TARget <*target*>

## Options

*target*—Name of the target or an IP address on the target

## Examples

- target beta
- tar beta

## Notes

- You must be logged into a machine using the login command before using the target command.
  - Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Test Connections

---

## Command

TEST CONNECTIONS

## Description

Sends a test command to determine if Double-Take Availability is running. If there are active connections, the replication set(s), IP address(es), and connection ID(s) will be included in the testcon log file.

## Syntax

TESTCONnections <*machine*> [ *filename* [OVERWRITE]]

## Options

- *machine*—The name of the machine
- *filename*—The name of the log file. The file extension .dts is appended to any file name supplied.
- OVERWRITE—Indicates that existing data in the file will be overwritten

## Examples

- testconnections alpha connectionstat
- testcon alpha

## Notes

- The default log file is status.dts and is located in the same directory as the other Double-Take Availability log files. By default, the log file is not overwritten.
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Time Now

---

## Command

TIME NOW

## Description

Prints the current date and time. This command is useful for scripting to determine process start and stop times.

## Syntax

TIMEnow

## Examples

- timenow
  - time
-

# Transmission Pause

---

## Command

TRANSMISSION PAUSE

## Description

Pauses the transmission pause

## Syntax

TRANSmision PAUSE <*target*>

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- transmission pause beta
- trans pause beta

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Transmission Resume

---

## Command

TRANSMISSION RESUME

## Description

Resumes a paused transmission pause

## Syntax

TRANSmision RESUME <*target*>

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- transmission resume beta
- trans resume beta

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Transmission Start

---

## Command

TRANSMISSION START

## Description

Initiates the transmission pause

## Syntax

TRANSmision START <*target*>

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- transmission start beta
- trans start beta

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---



# Transmission Stop

---

## Command

TRANSMISSION STOP

## Description

Stops the transmission pause

## Syntax

TRANSmision STOP <*target*>

## Options

*target*—Name of the target or an IP address on the target.

## Examples

- transmission stop beta
- trans stop beta

## Notes

Target names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Unload Source

---

## Command

UNLOAD SOURCE

## Description

Unloads the Double-Take Availability source module

## Syntax

UNLOAD SOURCE <*machine*>

## Options

*machine*—Name of the machine

## Examples

unload source alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Unload Target

---

## Command

UNLOAD TARGET

## Description

Unloads the Double-Take Availability target module

## Syntax

UNLOAD TARGET <*machine*>

## Options

*machine*—Name of the machine

## Examples

unload target alpha

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Verify

---

## Command

VERIFY

## Description

Verifies the integrity of the data between the source and target machines

## Syntax

```
VERIFY <con_id> [ SYNC [,NEWER] | NOSYNC] [,CHECKSUM | NOCHECKSUM]  
[STARTTIME=<mm/dd/yy> [hh:mm]] [EVERY <number> <time_units>]  
[ORPHANS | NOORPHANS] [CLEARRESTOREREQUIRED]
```

## Options

- **con\_id**—Connection ID assigned to the source/target connection
- **SYNC**—Synchronizes any data that is different
- **NEWER**—Synchronizes only those files that are newer on the source than on the target
- **NOSYNC**—Do not synchronize any data that is different
- **CHECKSUM**—Compares and/or synchronizes those blocks that are different on the source than on the target based on checksum comparisons
- **NOCHECKSUM**—Does not perform a checksum comparison when comparing and/or synchronizing files
- **STARTTIME**—Starts the verification process at the time specified
- **mm/dd/yy**—Date in month/day/year format when the verification process will begin
- **hh:mm**—Time in hour:minute format using the 24-hour clock when the verification process will begin
- **EVERY**—Repeat the verification process at the frequency specified
- **number**—Length of time to repeat the verification process
- **time\_units**—Minutes (min), hours (hr), or days (day)
- **ORPHANS**—Moves or deletes orphan files on the target
- **NOORPHANS**—Does not move or delete orphan files on the target
- **CLEARRESTOREREQUIRED**—Clears the restore required flag and initiates the verification

## Examples

- verify 1
- verify 2 sync, newer
- verify 2 every 2 hr

## Notes

The default verification settings are sync, checksum, and noorphans.

# Version

---

## Command

VERSION

## Description

Displays the Double-Take Availability version currently installed

## Syntax

VERSION [*machine*]

## Options

*machine*—Name of the machine

## Examples

version alpha

## Notes

- If no machine name is specified, the version of the active source will be displayed. If there is no active source, no version information will be displayed.
  - Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.
-

# Wait

---

## Command

WAIT

## Description

This command is used in scripts to force the script to stop executing until the specified number of milliseconds has elapsed.

## Syntax

WAIT <*ms*>

## Options

*ms*—Length of time in milliseconds

## Examples

wait 5000

## Notes

This example waits for 5 seconds.

---

# Wait on Mirror

---

## Command

WAIT ON MIRROR

## Description

This command is used in scripts to force the script to stop executing until the connection has finished mirroring or verifying

## Syntax

WAITONMIRROR <*con\_id*>

## Options

*con\_id*—Connection ID assigned to the source/target connection

## Examples

waitonmirror 1

---

# Wait on Restore

---

## Command

WAIT ON RESTORE

## Description

This command is used in scripts to force the script to stop executing until the connection has finished restoring

## Syntax

WAITONRESTORE <*rest\_id*>

## Options

*rest\_id*—Restoration ID assigned to the source/target connection

## Examples

waitonrestore 1

## Notes

This command requires using the source command immediately before the waitonrestore command.

---



# Wait on Target

---

## Command

WAIT ON TARGET

## Description

This command is used in scripts to force the script to stop executing until the target queue has been empty for the time specified

## Syntax

WAITONTARGET <*target*> [FROM <*source*>] <*time*>

## Options

- *target*—Name of the target or an IP address on the target
- *source*—Name of the source or an IP address on the source
- *time*—Number of seconds that the target queue needs to be empty before the command will return

## Examples

waitontarget beta from alpha 30

## Notes

Machine names, except for IP addresses, that contain non-alphanumeric characters must be enclosed in quotation marks.

---

# Write

---

## Command

WRITE

## Description

Displays the value of a Double-Take Availability variable

## Syntax

WRITE \$<*variable\_name*>

## Options

*variable\_name*—The name of the variable that you have established and want to display

## Examples

- write \$thetarget
  - write \$dbrepset
-