

1015 - ESE: Command-line options

 support.redstor.com/hc/en-gb/articles/115002725853-1015-Backup-Pro-ESE-Linux-command-line-options

Once an Enterprise Server Edition (ESE) Agent is installed, a command-line tool called `backuppercli` can be used to perform basic configuration. This works on both Linux and Windows.

Options

The following options are available:

Option	Purpose
<code>about</code>	Display the Agent version, brand ID and Java version.
<code>backup</code>	Initiate a backup from the command-line (from v18.9).
<code>full_system_backup</code>	Configure the Full System Backup feature (from v20.5).
<code>check_update</code>	Check for updates.
<code>create_account</code>	Create a new Account. (See example below .)
<code>show_account</code>	Display current Account details.
<code>remote_tunnel</code>	Configure remote access. (See example below .)
<code>show_remote_tunnel</code>	Show current remote access configuration.
<code>reconnect_account</code>	Reconnect the current Agent to an existing Account. From v18.9, also restores the workspace (See example below .)
<code>restore_workspace</code>	Set to "true" will restore, set to "false" will skip the restore.
<code>-r file</code>	JSON file containing responses to prompts. (See example below .)
<code>-q</code>	Quiet option. Used in conjunction with <code>-r</code> when you want to fail on missing values instead of prompting the user. Useful when invoking <code>cli</code> from a non-interactive script.
<code>-h</code>	Show additional command-line arguments.
<code>-v</code>	Increase verbosity

Responses

By default, the user is prompted for required information. If you want to provide some or all of this information without prompting the user, specify a response file.

If the quiet option is specified and a required value is not provided in the response file, the script fails with an error. Use this if you're invoking the cli from a non-interactive script.

If the quiet option is not specified, the user is prompted for any values that are not specified in the response file. This can be used, for example, if you want to provide the Storage Platform address, Group and create key yourself, but prompt the user for the Account name and password.

The response file is a JSON file containing keys and values. Options not required by the current command are ignored, so you can use the same response file when invoking, for example, `create_account` and `remote_tunnel`.

Command	Key	Description
<code>create_account</code>	<code>storage_platform_address</code>	
	<code>group</code>	
	<code>create_key</code> ¹	
	<code>account_name</code>	
	<code>account_password</code> ¹	
	<code>encryption_key</code> ¹	
	<code>gen_account_password</code> ²	JSON boolean, It accepts 1, "t", "T", "TRUE", "True", true, 0, "f", "F", "FALSE", "False", false. If set to true, generate a random secure account password.
	<code>gen_encryption_key</code> ²	JSON boolean, It accepts 1, "t", "T", "TRUE", "True", true, 0, "f", "F", "FALSE", "False", false. If set to true, generate a random secure account encryption key.

reconnect_account	storage_platform_address	
	group	
	account_name	
	account_password ¹	
	encryption_key ¹	
remote_tunnel	remote_tunnel_enabled	JSON boolean, integer 0 or 1, or string "y", "yes", "t", "true", "n", "no", "f", "false"
	remote_tunnel_access_type	"platform" or "password"
	remote_tunnel_password ¹	only required if access type is set to "password"
import	import_backup_timestamp	Timestamp in RFC3339 format. eg. 2018-10-09T12:30:01+02:00 or 2018-10-09T10:30:01Z (where Z means Zulu, ie. UTC)
	import_backup_root	Root of directory tree to import. On Linux, this is treated as "/", on Windows it must contain directories for each drive letter, eg. C_, D_

¹May be specified as either a plaintext string, or the prefix "{base64}" followed by a base64 encoded string if you wish to obfuscate the password.

²Use Golang's crypto/rand to generate secure random numbers - convert these to printable ASCII characters in the range between 32 and 127 (excluding space and DEL).

Examples

A few typical command-line usages are illustrated below:

Creating an Account

```
backupprocli create_account
```

The user will be prompted for the following information:

No.	Value	Description
1	Storage Platform address	Use the IP or DNS of your Storage Platform.
2	Backup Group	Must be the full path, e.g.: COLLECTION\Group.
3	Account Create Key	As per the Group's setting in the Storage Platform Console.
4	Backup Account Name	Defaults to the host name of the machine - can be changed.
5	Backup Account Password	(Hidden when entered. Re-type to confirm.)
6	Account Encryption Key	(Hidden when entered. Re-type to confirm.)

Reconnecting to an Account

`backupprocli reconnect_account`

The user will be prompted for the following information:

No.	Value	Description
1	Storage Platform address	Use the IP or DNS of your Storage Platform.
2	Backup Group	Must be the full path, e.g.: COLLECTION\Group.
3	Backup Account Name	Must be the name of the existing Account.
4	Backup Account Password	The Account's password. (Hidden when entered.)
6	Account Encryption Key	The Account's encryption key. (Hidden when entered.)

Allowing remote connections

When enabled, this allows remote connections to the current Agent from the Storage Platform Console, even from a Windows machine.

`backupprocli remote_tunnel`

You will be prompted to enable or disable the "remote tunnel" with a choice of access restriction:

- "platform": Requires Storage Platform authentication
- "password": Configures a password which will be required when connecting from a remote Agent

Using a response file

By default, the user is prompted for required information. If you want to provide some or all of this information without prompting the user, specify a response file.

The response file is a JSON file containing keys and values. Options not required by the current command are ignored, so you can use the same response file, for example, when invoking `create_account` and `remote_tunnel` .

```
backupprocli -r response.json create_account
```

A sample JSON response file with all values populated would look as follows:

```
{
  "storage_platform_address": "192.168.21.87",
  "group": "Dev",
  "create_key": "secret",
  "account_name": "myserver",
  "account_password": "password",
  "encryption_key": "password123",
  "remote_tunnel_enabled": true,
  "remote_tunnel_access_type": "password",
  "remote_tunnel_password": "stay_out"
}
```

The following file would set the Storage Platform and Group, but prompt the user for the Account details (see note below):

```
{
  "storage_platform_address": "192.168.21.87",
  "group": "Dev",
  "create_key": "secret",
}
```

Note: If `-q` (quiet) option is specified, the command will fail if there are missing keys in the response file, as per the response file above:

```
backupprocli -r -q response.json create_account
```

To prompt the user for the missing information, do not include the `-q` option.