# PowerShell Desired State Configuration for Linux

## Resource Reference

*Applies to: v1.0.0-CTP*

### MSFT_nxFileResource (nxFile)

Manage Linux files and directories.

#### Properties

| Property | Description |
|---|---|
| DestinationPath | Indicates the location where you want to ensure the state for a file or directory. |
| SourcePath | Indicates the path from which to copy the file or folder resource. |
| Ensure | Indicates the checksum type to use when determining whether two files are the same. If **Checksum** is not specified, only the file or directory name is used for comparison. Valid values include: md5, mtime, ctime |
| Type | Indicates if the resource being configured is a directory or a file. Set this property to **Directory** to indicate that the resource is a directory. Set it to **File** to indicate that the resource is a file. Set it to **link** to indicate that the resource is a symbolic link. The default value is **File**. |
| Force | |
| Contents | Specifies the contents of a file. |
| Checksum | Indicates the checksum type to use when determining whether two files are the same. If **Checksum** is not specified, only the file or directory name is used for comparison. Valid values include: md5, mtime, ctime |
| Recurse | Indicates if subdirectories are included. Set this property to **$true** to indicate that you want subdirectories to be included. The default is **$false**.<br><br>**Note:** This property is only valid when the Type property is set to Directory. |
| Links | Indicates the desired behavior for symbolic links. Set this property to **follow** to follow symbolic links and act on the links target (e.g. copy the file instead of the link). Set this property to **manage** to act on the link (e.g. copy the link itself). Set this property to **ignore** to ignore symbolic links. |
| Group | The name of the **Group** to own the file or directory |
| Mode | Indicates the desired permissions for the resource, in octal or symbolic notation. (e.g. 777 or rwxrwxrwx). If using symbolic notation, do not provide the first character which indicates directory or file). |
| Owner | The name of the **Owner** to own the file or directory |

### MSFT_nxUserResource (nxUser)

Manage local Linux users.

#### Properties

| Property | Description |
|---|---|
| **UserName** | Indicates the name of the environment variable for which you want to ensure a specific state. |
| **Ensure** | Indicates if a user exists. Set this property to **Present** to create the user if it does not exist or to ensure that its properties match the provided properties. Set it to **Absent** to delete the user if it exists. |
| **FullName** | The full name of the user to manage |
| **Description** | The description for the user to manage |
| **Password** | The password for the user to manage, provided as a salted SHA1 hash. |

| | |
|---|---|
| **Disabled** | If **$true**, the user account will be disabled from login. If **$false,** the user account will be enabled for login. |
| **PasswordChangeRequired** | If true, the user will be required to enter a new password on the next login. If false, the user will not be required to set a new password on next login. |
| **HomeDirectory** | The home directory for the user |
| **GroupID** | The primary group id for the user |

## MSFT_nxGroupResource (nxGroup)

Manage local Linux groups.

### Properties

| **GroupName** | **The name of the group** |
|---|---|
| **Ensure** | Indicates if a group exists. Set this property to **Present** to create the group if it does not exist or to ensure that its properties match the provided properties. Set it to **Absent** to delete the group if it exists. |
| **Members** | Indicates that you want to ensure these members form the group. |
| **MembersToInclude** | Indicates the users who you want ensure are not members of this group. |
| **MembersToExclude** | Indicates the users who you want to ensure are members of the group. |

## MSFT_nxScriptResource (nxScript)

Run scripts on target nodes.

### Properties

| **Property** | **Description** |
|---|---|
| **GetScript** | Provides a block of script that runs when you invoke the Get-DscConfiguration cmdlet |
| **SetScript** | Provides a block of script. When you invoke the Start-DscConfiguration cmdlet, the TestScript block runs first. If the TestScript block returns False, the SetScript block will run. If the TestScript block returns True, the SetScript block will not run. |
| **TestScript** | Provides a block of script. When you invoke the Start-DscConfiguration cmdlet, this block runs. If it returns False, the SetScript block will run. If it returns True, the SetScript block will not run. The TestScript block also runs when you invoke the Test-DscConfiguration cmdlet. However, in this case, the SetScript block will not run, no matter what value the TestScript block returns.The TestScript block must return True if the actual configuration matches the current desired state configuration, and False if it does not match. (The current desired state configuration is the last configuration enacted on the node that is using DSC). |
| **User** | The user to run the script as |
| **Group** | The group to run the script as |

## MSFT_nxServiceResource (nxService)

Manage Linux daemons.

### Properties

| **Property** | **Description** |
|---|---|
| **Name** | The name of the service/daemon to configure |
| **Controller** | The type of service controller to use when configuring the service. Values can be: **init, upstart, or systemd**. |
| **Enabled** | Indicates whether the service starts on boot |
| **State** | Indicates whether the service is running. Set this property to Stopped to ensure that the service is not running. Set it to Running to ensure that the service is running |