

# SSH Command - User Guide

## SSH Command Assertion

- [Overview](#)
- [Installation](#)
  - [Pre-Requisites](#)
  - [How-To Install](#)
    - [9.0 and above](#)
    - [Prior to 9.0](#)
- [Configuration](#)
- [Usage](#)
- [Limitations](#)
- [Legal Documentation](#)

## Overview

This document is a guide for the installation and usage of the SSH Command assertion.



### Security Considerations

Use this assertion with caution. This assertion connects APIs with system level access to a remote server. The gateway has no control or insight into any commands run and should be used with care.

## Installation

## Pre-Requisites

The assertion installation process involves copying the required libraries onto the SecureSpan gateway, configuring the files, and then restarting the Gateway. You will require 'root'-level access to the Gateway appliance to complete the installation.

## How-To Install

### 9.0 and above

1. In Policy Manager, go to **Tasks > Manage Server Module Files**
2. Upload the **SshCommandAssertion-[version].sjar**
3. Provide the assertion a name like **SSH** and click OK. The name is used in the Manage Server Files dialog
4. The assertion will be uploaded and loaded onto the Gateway. For more information on managing Server Module Files, see *CA API Gateway - Manage Server Module Files*

### Prior to 9.0

1. Log in as into the Gateway as ssgconfig and open a privileged command shell from the Gateway configuration menu.
2. Copy SshCommandAssertion-sdk80-1.0-23512.jar to the following directory:

```
/opt/SecureSpan/Gateway/runtime/modules/lib
```

3. Run the following commands in the target directory:

```
chmod 444 SshCommandAssertion-sdk80-1.0-2351.jar  
chown layer7:layer7 SshCommandAssertion-sdk80-1.0-2351.jar
```

4. Restart the Gateway:

```
service ssg restart
```

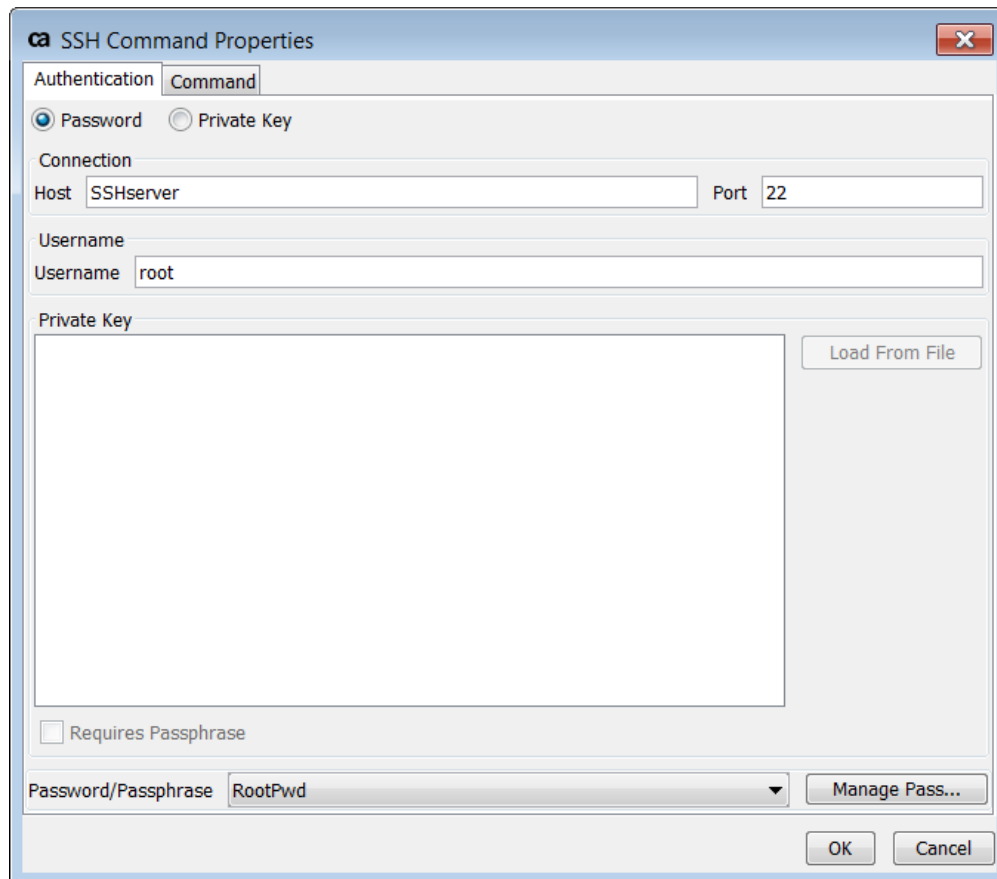
## Configuration

In order to connect to a machine using SSH (Secure SHell), the SSH private/public key must be generated using a tool like ssh-keygen. The public key must reside on the server's .ssh/authorized\_keys file.

## Usage

1. Add SSH Server authentication details.

The login user can be authenticated with a password or a private key. To enter a new password, click on the "Manage Password" button and press the "Add" button. If authenticated with a private key, press the "Load From File" button and select the private key file generated by the ssh-keygen command. If a passphrase is used to generate the key, check the "Requires Passphrase" check box. Then enter the passphrase by clicking on the "Manage Password" button and press the "Add" button. At the end, select the password/passphrase just created in the drop-down box.



The image shows a dialog box titled "SSH Command Properties". It has two tabs: "Authentication" and "Command". The "Authentication" tab is selected. Inside this tab, there are two radio buttons: "Password" (which is selected) and "Private Key". Below these, there is a "Connection" section with a "Host" field containing "SSHserver" and a "Port" field containing "22". There is also a "Username" field containing "root". Below the username field is a "Private Key" section with a large empty text area and a "Load From File" button. At the bottom of the "Authentication" tab, there is a "Requires Passphrase" checkbox (which is unchecked) and a "Password/Passphrase" field containing "RootPwd". To the right of the password field is a "Manage Pass..." button. At the very bottom of the dialog box are "OK" and "Cancel" buttons.

Figure 1: Enter authentication details to connect to SSH Server.

## 2. Configure the command to execute.

Enter the command in the "Command" textbox. Enter the arguments in the "Arguments" textbox below by pressing the "Add" button. Make sure each parameter is on its own line. The output has a choice of Request, Response, or Context variable. If the Context variable is chosen, enter the variable name in the textbox below. Context variable expansion is supported for the username, host, port, private key, command name and the command arguments. All command arguments are automatically single quote escaped and wrapped with single quotes.

To run the script: hello.sh for the user: root where the file is located in /root, enter `./hello.sh` in the command textbox.

- The exit status of the command is available in the context variable **ssh.exitStatus**.
- The error stream output from the command is available in the context variable **ssh.errorMessage**.
- The output of the command is put into the body of the target message with the content type **text/plain**. To visually inspect the ssh output, add the "Return Template Response to Requestor" assertion and if the target output is the response message, then output the response: `${Response.mainpart}`.
- The assertion will wait for up to 30 seconds for the command to finish executing.
- There is a checkbox option to fail the assertion if the exit status of the script/command returned is not 0.

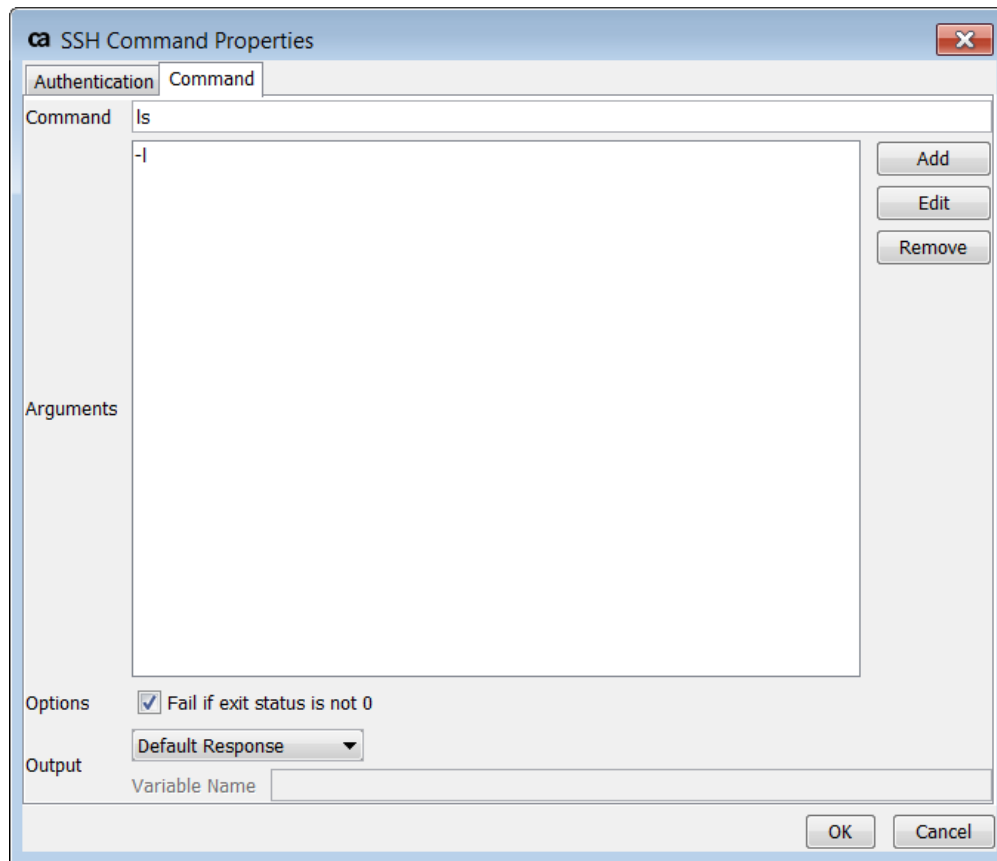


Figure 2: Enter command to execute on SSH Server.

## Limitations

The SSH command assertion has the following limitations:

- If there are several passphrases/passwords stored in the system and the SSH Command Assertion UI is reopened after it has been previously saved, you must reselect the proper passphrase/password from the drop-down list (if you wish to save the assertion again).
- Only one command can be run per connection (although that command can be running a script).
- Environment variables are not supported.
- Shell features such as input/output redirection are not supported.

## Legal Documentation

This product includes GANYMED-SSH-2 BUILD251 BETA1 which is distributed in accordance with the following license agreement:

Copyright (c) 2006 - 2010 Christian Plattner. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- a.) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- b.) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- c.) Neither the name of Christian Plattner nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software includes work that was released under the following license:  
Copyright (c) 2005 - 2006 Swiss Federal Institute of Technology (ETH Zurich),  
Department of Computer Science (<http://www.inf.ethz.ch>),  
Christian Plattner. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- a.) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- b.) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- c.) Neither the name of ETH Zurich nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Java implementations of the AES, Blowfish and 3DES ciphers have been taken (and slightly modified) from the cryptography package released by "The Legion Of The Bouncy Castle".

Their license states the following:

Copyright (c) 2000 - 2004 The Legion Of The Bouncy Castle  
(<http://www.bouncycastle.org>)