

Creating a Supportconfig Plugin

- [Overview](#)
- [How Supportconfig Plugins Work](#)
- [How are Plugins Organized](#)
- [Resource Functions to Help with Bash Plugins](#)
- [Plugin Integration with the Supportconfig Analysis \(SCA\) Tool](#)
- [Example Plugin](#)

Overview

Supportconfig included in supportutils package has plugin support. The purpose of a plugin is to add custom information gathering support to supportconfig for troubleshooting purposes. You can gather information about your program or feature and have supportconfig include it in the supportconfig tarball. This feature allows you to write your own script or program, have supportconfig run it and include its output as a file with the supportconfig tar ball.

How Supportconfig Plugins Work

Plugins are enabled by default, but if you use the `-p` startup option, they will be disabled.

1. Supportconfig looks for plugins to run in the `/usr/lib/supportconfig/plugins` directory.
2. Each plugin with execute permissions in the plugins directory is executed.
3. Supportconfig creates a plugin text file called `plugin-<plugin_name>.txt`
4. The stdout and stderr are written to the plugin text file.
5. Supportconfig exports the `$LOG` bash environment variable for the location of the supportconfig active directory.
6. If the plugin is a script, it is included in the plugin text file for troubleshooting purposes.

NOTE: Supportconfig does not support dashes "-" in the plugin filename.

How are Plugins Organized

If you are writing a plugin to help troubleshoot an application or service, I have found it helpful to include some basic information. The information you include should help you determine if the application is installed, configured and running properly. I check the system to answer the following basic questions:

Is the service installed?

If it's installed as an RPM package(s), do the RPM packages pass validation?

Is the service configured to start at boot up?

Is the service currently running?

What is the state of its current running processes?

What network ports is the service listening on?

Gather all the configuration files.

Gather important log files.

Anything else that makes sense.

Resource Functions to Help with Bash Plugins

The supportutils package installs two resource files. You can source either one and only one of these files to use the included functions to write a bash script.

```
/usr/lib/supportconfig/resources/scplugin.rc
```

```
/usr/lib/supportconfig/resources/supportconfig.rc
```

Plugin Integration with the Supportconfig Analysis (SCA) Tool

The SCA Tool is used to analyze supportconfigs for known issues, creating a report, and linking to known solutions to those issues. If you use either resource file to write a bash script plugin, the functions automatically include formatting to integrate with the SCA Tool. If you write your plugin in Python, Perl or any other language, make sure you follow these guidelines.

1. Start each section of the file with `##=`
2. The next line starts includes a pound sign followed by the command being executed or the file being gathered, like `# /bin/date`
3. All the lines following are the output of the command or the contents of the file
4. Finally, a double space ends the section

Example Content

```
#[ Command ]=====#
# /bin/date
Thu Mar 25 11:05:12 MDT 2021
```

Example Plugin

Create an example for a plugin called **hostinfo** that collects information about the **hostinfo** package and command for SLE15.

Hostinfo Plugin Source Code

```
sles15sp2-base:/usr/lib/supportconfig/plugins # cat hostinfo
#!/bin/bash

. /usr/lib/supportconfig/resources/supportconfig.rc
OF=plugin-hostinfo-data.txt

if rpm_verify $OF hostinfo; then
    log_cmd $OF "systemctl status hostinfo.service"
    log_cmd $OF "systemctl status hostinfo.timer"
    conf_files $OF /etc/hostinfo.conf
else
    log_entry $OF note "hostinfo not installed"
fi
```

Hostinfo Plugin Output

```
#[ Verification ]=====#
# rpm -V hostinfo-1.0-18.15.noarch
.....T. c /etc/hostinfo.conf
# Verification Status: Differences Found

#[ Command ]=====#
# /bin/systemctl status hostinfo.service
* hostinfo.service - Generate server information file for login session
  Loaded: loaded (/usr/lib/systemd/system/hostinfo.service; enabled; vendor preset: disabled)
  Active: inactive (dead)

Mar 31 09:27:27 sles15sp2 systemd[1]: Starting Generate server information file for login session...
Mar 31 09:28:27 sles15sp2 hostinfo[4794]: SSH keys ignored in /etc/hostinfo.conf
Mar 31 09:28:27 sles15sp2 hostinfo[4794]: Disk information ignored in /etc/hostinfo.conf
Mar 31 09:28:27 sles15sp2 systemd[1]: Started Generate server information file for login session.

#[ Command ]=====#
# /bin/systemctl status hostinfo.timer
* hostinfo.timer - Run hostinfo hourly and on boot
  Loaded: loaded (/usr/lib/systemd/system/hostinfo.timer; disabled; vendor preset: disabled)
  Active: inactive (dead)
  Trigger: n/a

#[ Configuration File ]=====#
# /etc/hostinfo.conf
INCLUDE_MOTD=0
IGNORE_DISK=1
IGNORE_MEM=0
IGNORE_CPULOAD=0
IGNORE_NET=0
IGNORE_UPDATES=0
IGNORE_SSHKEYS=1
```

