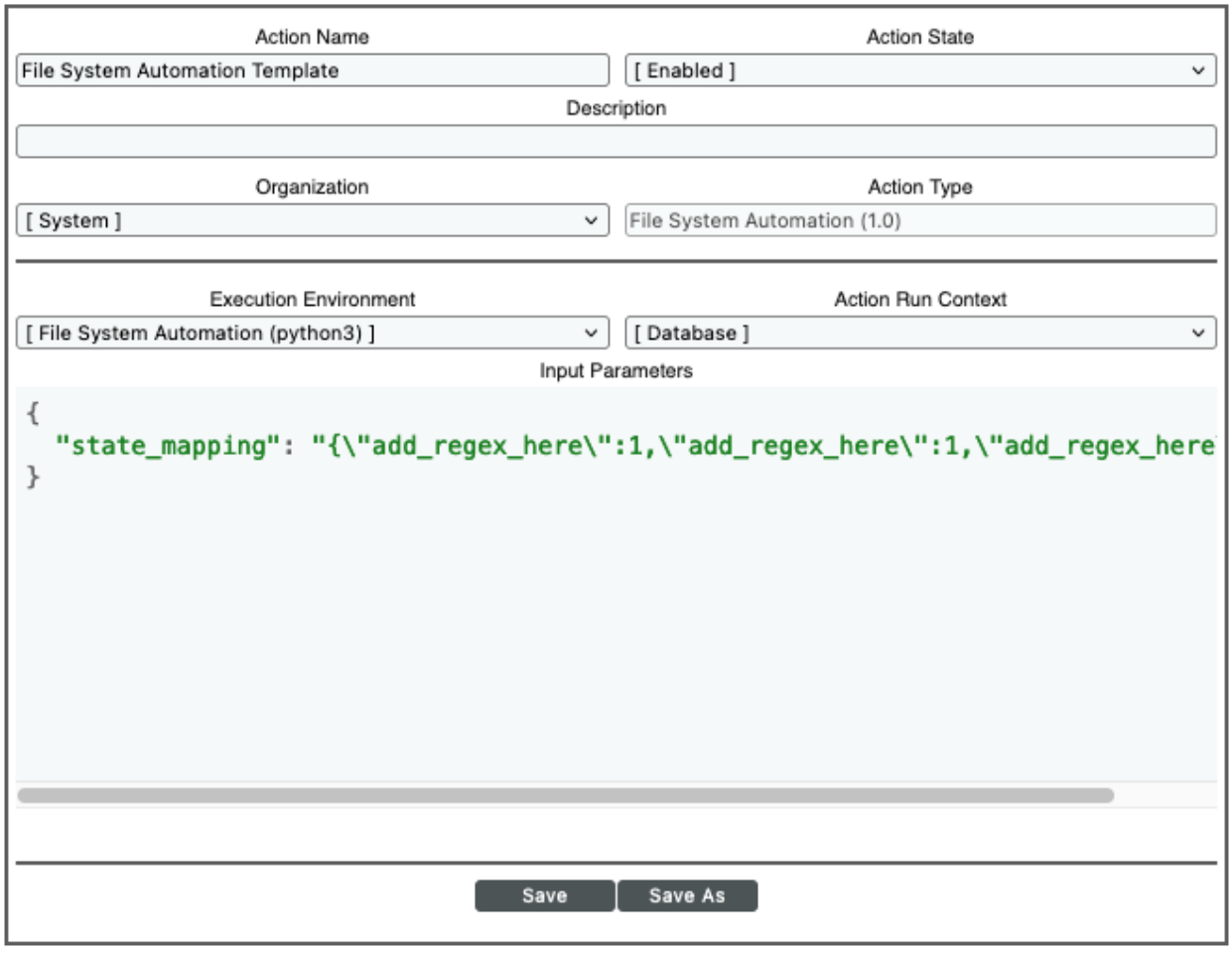
Rackspace File System Automation PowerPack

The Rackspace File System Automation PowerPack contains a run book action type that can be used to create automations that change the monitoring state (also known as the hidden flag) of file systems based on the name.

Configuring an Automation Action

To configure an automation action using the action type in the PowerPack:

1. Go to Registry > Run Book > Actions.
2. Search for "File System Automation Template" and click its wrench icon.
3. Supply a new name in the "Action Name" field
4. In the Input Parameters field, update the state_mapping parameter. See the following section for a description of this field.
5. Click the "Save As" button



ScienceLogic

Configuring State Mappings

The action type includes one input parameter, state_mapping. This parameter specifies a list of key/value pairs. Each key must be a regular expression; each value must be either a 0 or 1.

When the automation runs for an event associated with a device, all file systems associated with that device are evaluated by matching the regular expressions with the mount point name (shown as "Description" in the Device Hardware page). If a match is found, the "Hidden" setting is updated to match the value for that regular expression. For example, if the value is 0, the file system hidden setting is changed to "No".

The configured regular expressions are used with the python regex "search" function; if a state_mapping key contains only text and no regex special characters, that regex will match if the text appears anywhere in the file system name.

The state_mapping parameter is interpreted as an escaped JSON string. The template action has the correct format/escape character usage:

```
{
  "state_mapping": "{\"add_regex_here\":1,\"add_regex_here\":1,\"add_regex_here\":0}"
}
```

The following example enables monitoring (Hide = No) on the "/" file system (using the starts with/ends with special characters) and disables monitoring (Hide = Yes) on all file systems that contain the string "var" (e.g. /var, /var/lib, /var/log, etc):

```
{
  "state_mapping": "{\"^/$\":0,\"var\":1}"
}
```

Output and Logging

The automation action result will include one of the following:

- A single message indicating that the whole automation failed, e.g. if the state_mapping parameter is invalid.
- A list of message for each file system that matched one of the regular expressions. These messages will indicate either a change was made or that no change was made (typically because the file system already has that setting applied).

Example output:

```
Automation Policy Test Filesystem Automation action File System Automation Example ran Successfully
Message:CustomActionType (57) executed without incident
Result:{u'Message': u'
File system matched but value not changed to 0 for mount point / on device 2 for regex ^/$. Query: update master_dev.device_hardware set hide=0 where did = 2 and inv_id=1
Changed hide to 1 for mount point /var on device 2 based on regex var
Changed hide to 1 for mount point /var/lib on device 2 based on regex var
Changed hide to 1 for mount point /var/log on device 2 based on regex var
Changed hide to 1 for mount point /var/log/audit on device 2 based on regex var
Changed hide to 1 for mount point /var/tmp on device 2 based on regex var
'}
```

The debug log for this automation is /var/log/sl1/file_system_automation.log. The automation executes on the primary Database Server.