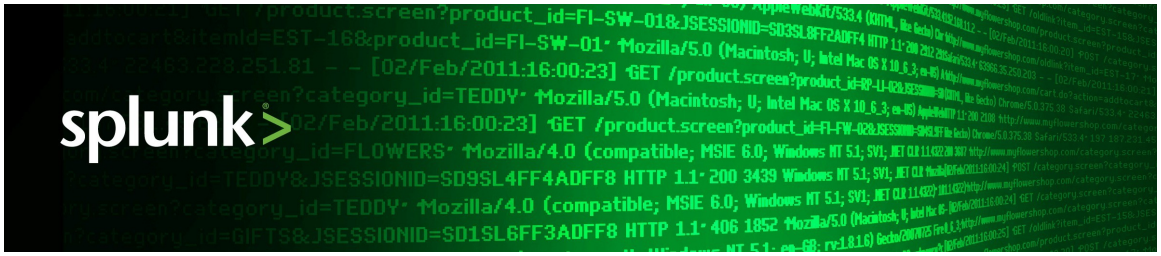

Splunk® Universal Forwarder Forwarder Manual 6.5.0

Configure forwarding with outputs.conf

Generated: 10/19/2016 9:55 pm

Configure forwarding with outputs.conf

The outputs.conf file defines how forwarders send data to receivers. You can specify some output configurations at installation time (Windows universal forwarders only) or the CLI, but most advanced configuration settings require that you edit `outputs.conf`.

The topics that describe various forwarding topologies, such as load balancing and intermediate forwarding, provide detailed examples on configuring `outputs.conf` to support those topologies.

Although `outputs.conf` is a required file for configuring forwarders, it addresses only the outputs from the forwarder, where you want the forwarder to send the data it collects. To specify the data that you want to collect from the forwarder, you must separately configure the inputs, as you would for any Splunk instance. See Add data and configure inputs in *Getting Data In*.

Edit outputs.conf to configure forwarding

This procedure details the steps you must take to edit the default `outputs.conf` which is in `$SPLUNK_HOME/etc/system/local`. You might have to edit the file in other places, as sections in this topic explain. For an example of what an `outputs.conf` file looks like, see "Examples of outputs.conf" later in this topic.

1. On the host that forwards that data that you want to collect, open a shell or command prompt or PowerShell window.
2. Go to the configuration directory for the forwarder.

Unix	Windows
`cd $SPLUNK_HOME/etc/system/local`	`cd %SPLUNK_HOME%\etc\system\local`

3. Open `outputs.conf` for editing with a text editor.

Unix	Windows
`vi outputs.conf`	`notepad outputs.conf`

4. Edit outputs.conf. Add a minimum of at least one forwarding target group or a single receiving host.
5. Save the `outputs.conf` file and close it.
6. Restart the universal forwarder to complete your changes.

Unix	Windows
`cd $SPLUNK_HOME/bin` `./splunk restart`	`cd %SPLUNK_HOME%\bin` `.\splunk restart`

Types of outputs.conf files

A single forwarder can have multiple `outputs.conf` files. For example, one can be located in an apps directory and another in `$SPLUNK_HOME/etc/system/local`. No matter how many `outputs.conf` files the forwarder has and where they reside, the forwarder combines all their settings, using the rules of configuration file precedence. The forwarder contains both default and custom `outputs.conf` files.

Default versions of outputs.conf

The universal forwarder ships with these default versions of `outputs.conf`:

- One in `$SPLUNK_HOME/etc/system/default`.
- Another in `$SPLUNK_HOME/etc/apps/SplunkUniversalForwarder/default`.

The default version in the `SplunkUniversalForwarder` app has precedence over the version under `/etc/system/default`.

Do not edit default versions of any configuration files. See About configuration files.

Custom versions of outputs.conf

When you configure forwarding behavior, those changes get saved in custom versions of `outputs.conf`. There are several ways you can specify forwarding behavior:

- While installing the forwarder (on the Windows universal forwarder only.)
- By running CLI commands.
- By directly editing an `outputs.conf` file.

The forwarder automatically creates or edits custom versions of `outputs.conf` in response to the first three methods. The locations of those versions vary, depending on the type of forwarder and other factors.

- If you use the CLI to make changes to universal forwarder output behavior, the CLI creates or edits a copy of `outputs.conf` in `$SPLUNK_HOME/etc/system/local`.
- The Windows installation process writes configuration changes to an `outputs.conf` file located in the `MSICreated` app.

In addition to any `outputs.conf` files that you create and edit indirectly (for example, through the CLI), you can also create or edit an `outputs.conf` file

directly with a text editor. You should work with a single copy of the file, which you place in `$SPLUNK_HOME/etc/system/local/`. If a copy of the file already exists in that directory, because of configuration changes made through the CLI, edit that copy. For purposes of distribution and management simplicity, you can combine settings from all non-default versions into a single custom `outputs.conf` file.

The universal forwarder must be restarted after you make changes to `outputs.conf`.

For information on `outputs.conf`, see the outputs.conf spec file.

Configuration levels for outputs.conf

There are two types of output processors for forwarding data: `tcpout` and `syslog`. The universal forwarder only has the `tcpout` processor, which uses the `[tcpout]` header in `outputs.conf`.

You can configure the `tcpout` processor at three levels of stanzas:

- **Global.** (Optional) At the global level, you specify any attributes that you want to apply globally, as well as certain attributes only configurable at the system-wide level for the output processor.
- **Target group.** A target group defines settings for one or more receiving indexers. There can be multiple target groups per output processor. Most configuration settings can be specified at the target group level.
- **Single server.** (Optional) You can specify configuration values for single servers (receivers) within a target group.

Configurations at the more specific levels take precedence over the global level. For example, if you specify `compressed=true` for a target group, the forwarder sends the hosts in that target group compressed data, even if you set the `compressed` attribute to "false" for the global level.

Outputs.conf global stanza

The global stanza in `outputs.conf` lets you set any attributes that you want to apply globally. While this stanza is optional, there are several attributes that you can set only at the global level, including `defaultGroup`.

The `[tcpout]` header specifies the global stanza for the `tcpout` processor. Following is an example of a global tcpout stanza.

```
[tcpout]
```

```
defaultGroup=indexer1
compressed=true
```

This global stanza includes two attribute/value pairs:

- **defaultGroup=indexer1** This tells the forwarder to send all data to the "indexer1" target group. See "Default target groups".
- **compressed=true** This tells the forwarder to compress the data before it forwards the data to receiving indexers in the target groups. If you set `compressed` to "false", the forwarder sends raw data.

Set default target groups in outputs.conf

The `defaultGroup` attribute lets you set default groups for automatic forwarding at the global level, in your `[tcpout]` stanza.

The defaultGroup specifies one or more target groups that you define later in `tcpout:<target_group>` stanzas. The forwarder sends all events to the specified groups.

```
[tcpout]
defaultGroup= <target_group1>, <target_group2>, ...
```

If you do not want to forward data automatically, do not set the `defaultGroup` attribute.

Outputs.conf target group stanza

The target group identifies a set of receivers. It also specifies how the forwarder sends data to those receivers. You can define multiple target groups.

Here is the basic pattern for the target group stanza.

```
[tcpout:<target_group>]
server=<receiving_server1>, <receiving_server2>, ...
<attribute1> = <val1>
<attribute2> = <val2>
...
```

You can specify a receiving server in a target group by using the format `<ipaddress_or_hostname>:<port>`, where `<port>` is the receiving host **receiving port**. For example, `myhost.splunk.com:9997`. When you specify multiple receivers, the forwarder load balances among them.

See Define typical deployment topologies later in this topic for information on how to use the target group stanza to define several deployment topologies.

Outputs.conf single-host stanza

You can define a specific configuration for an individual receiving indexer. However, the receiver must also be a member of a target group.

When you define an attribute at the single-host level, it takes precedence over any definition at the target group or global level.

Here is the syntax for defining a single-host stanza:

```
[tcpout-server://<ipaddress_or_hostname>:<port>]
<attribute1> = <val1>
<attribute2> = <val2>
...
```

Examples of outputs.conf

The following `outputs.conf` example contains three stanzas for sending data to Splunk receivers.

- Global settings. In this example, there is one setting, to specify a `defaultGroup`.
- Settings for a single target group consisting of two receivers. Here, we specify a load-balanced target group consisting of two receivers.
- Settings for one receiver within the target group. In this stanza, you can specify any settings specific to the `mysplunk_indexer1` receiver.

```
[tcpout]
defaultGroup=my_indexers

[tcpout:my_indexers]
server=mysplunk_indexer1:9997, mysplunk_indexer2:9996

[tcpout-server://mysplunk_indexer1:9997]
```

Define typical forwarder deployment topologies

You can configure a forwarder to support several typical deployment topologies. See the other topics in the "Forward data" chapter for information on how to configure forwarders for other topologies.

Configure load balancing on a universal forwarder with outputs.conf

When you specify a target group with multiple receivers in `outputs.conf` on a forwarder, the forwarder performs **load balancing** between the receivers.

In the example that follows, the target group consists of three receivers. The forwarder balances load between the three receivers you specify. If one receiver goes down, the forwarder automatically switches to the next available receiver.

```
[tcpout:my_LB_indexers]
server=10.10.10.1:9997,10.10.10.2:9996,10.10.10.3:9995
```

Note: While 9997 is the standard network port for receiving data from forwarders, you can specify any network port above 1024 to receive data.

Configure data cloning on a universal forwarder with outputs.conf

When you specify multiple target groups with a separate stanza for each group in `outputs.conf`, the forwarder performs **data cloning** between the groups. In data cloning, the forwarder sends copies of all its events to the receivers in two or more target groups. Data cloning usually results in similar, but not necessarily exact, copies of data on the receiving indexers. An example of how to configure data cloning follows.

```
[tcpout]
defaultGroup=indexer1,indexer2

[tcpout:indexer1]
server=10.1.1.197:9997

[tcpout:indexer2]
server=10.1.1.200:9997
```

The forwarder sends duplicate data streams to the servers specified in both the `indexer1` and `indexer2` target groups.

Configure data cloning with load balancing on a universal forwarder

You can combine load balancing with data cloning. For example:

```
[tcpout]
defaultGroup=cloned_group1,cloned_group2

[tcpout:cloned_group1]
```

```
server=10.10.10.1:9997, 10.10.10.2:9997, 10.10.10.3:9997

[tcpout:cloned_group2]
server=10.1.1.197:9997, 10.1.1.198:9997, 10.1.1.199:9997,
10.1.1.200:9997
```

The forwarder sends full data streams to both the `cloned_group1` and `cloned_group2` groups. The forwarders load-balance the data within each group, rotating among receivers every 30 seconds (the default frequency).

Common attributes for outputs.conf

The `outputs.conf` file provides a large number of configuration options that offer considerable control and flexibility in forwarding. Of the attributes available, several are of particular interest:

Attribute	Default	Where configured	Value
`defaultGroup`	n/a	global stanza	A comma-separated list of one or more target groups. Forwarder sends all events to all specified target groups.
`server`	n/a	target group stanza	Required. Specifies the hosts that function as receivers for the forwarder. This must be set to a value using the format `<ipaddress_or_servername>:<port>`, where `<port>` is the receiving server's receiving port.
`disabled`	false	any stanza level	Specifies whether the stanza is disabled. If set to "true", it is equivalent to the stanza not being there.
`sendCookedData`	true	global or target group stanza	Specifies whether data is cooked before forwarding.
`compressed`	false	global or target group stanza	Specifies whether the forwarder sends compressed data.
`ssl....`	n/a	any stanza level	Set of attributes for configuring SSL. See "About securing data from forwarders" in the *Securing*

			Splunk Enterprise manual for information on how to use these attributes.
`useACK`	false	global or target group stanza	Specifies whether the forwarder waits for indexer acknowledgment confirming that the data has been written to the file system.
`dnsResolutionInterval`	300	global or target group stanza	Specifies base time interval in seconds at which indexer DNS names will be resolved to IP address.

The `outputs.conf.spec` file, which you can find here, along with several examples, provides details for these and all other configuration options. In addition, most of these settings are discussed in topics that deal with specific forwarding scenarios.

DNS resolution interval

The `dnsResolutionInterval` attribute specifies the base time interval (in seconds) at which receiver DNS names will be resolved to IP addresses. The forwarder uses this value to compute the run-time interval as follows:

```
run-time interval = dnsResolutionInterval + (number of receivers in
server attribute - 1) * 30
```

The run-time interval increases by 30 seconds for each additional receiver that you specify in the `server` attribute (each additional receiver across which the forwarder load-balances.) The `dnsResolutionInterval` attribute defaults to 300 seconds.

For example, if you leave the attribute at the default setting of 300 seconds and the forwarder is load-balancing across 20 indexers, DNS resolution will occur every 14 1/2 minutes:

```
(300 + ((20 - 1) * 30)) = 870 seconds = 14.5 minutes
```

If you change `dnsResolutionInterval` to 600 seconds, and keep the number of load-balanced indexers at 20, DNS resolution will occur every 19 1/2 minutes:

```
(600 + ((20 - 1) * 30)) = 1170 seconds = 19.5 minutes
```