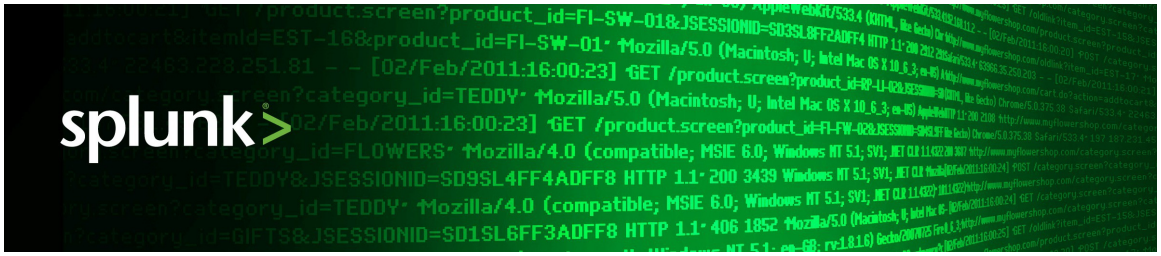

Splunk® Enterprise Knowledge Manager Manual 7.2.5

Configure external lookups

Generated: 4/05/2019 10:14 am

Configure external lookups

External lookups invoke a script that matches fields in your events with fields in an external source and outputs corresponding fields from that external source and adds them to your events.

External lookups are often referred to as scripted lookups, because they are facilitated through the use of a script. See About the external lookup script for information about how these scripts work.

Create an external lookup

The following is the steps required to create an external lookup for a Splunk Enterprise deployment. If you have Splunk Cloud and want to define external lookups, file a Support ticket.

Prerequisites

- About lookups
- Define an external lookup in Splunk Web
- Add field matching rules to your lookup configuration
- Configure a time-bounded lookup
- Make your lookup automatic

Do not edit configuration files in `$SPLUNK_HOME/etc/system/default`.

Steps

1. Add the script for the lookup to your Splunk deployment.
   The script must be located in one of two places:
   - `$SPLUNK_HOME/etc/searchscripts`
   - `$SPLUNK_HOME/etc/apps/<app_name>/bin`
2. Add an external lookup stanza to `transforms.conf`.
   If you want the lookup to be available globally, add its lookup stanza to the version of `transforms.conf` in `$SPLUNK_HOME/etc/system/local/`. If you want the lookup to be specific to a particular app, add its stanza to the version of `transforms.conf` in `$SPLUNK_HOME/etc/apps/<app_name>/local/`. The external lookup stanza names the lookup table, provides the lookup script and argument, identifies the script type, and supplies a list of fields that are supported by the script. It uses the following required attributes.
   - `[<lookup_name>]`: The name of the lookup.

- `external_cmd = <string>`: The command and arguments issued to perform the lookup. The command must be the name of the script, such as `external_lookup.py`. The arguments are the names of the fields that you want to pass to the script, separated by spaces, like this: `clienthost clientip`.
- `external_type = [python|executable|kvstore|geo]`: The type of script being used for the lookup. The values include `python`, for a Python script, or `executable`, for a binary executable, such as a C++ executable. The `kvstore` and `geo` values are reserved for KV store lookups and geospatial lookups, respectively.
- `fields_list = <string>`: This is a list of all fields that are supported by the external lookup. The fields must be delimited by a comma followed by a space.

3. (Optional) Set up field/value matching rules for the external lookup.
4. (Optional) If the data source for the external lookup contains time fields, make the external lookup time-bounded.
5. (Optional) Make the external lookup automatic by adding a configuration to `props.conf`.

   If you want the automatic lookup to be available globally, add its lookup stanza to the version of `props.conf` in `$SPLUNK_HOME/etc/system/local/`. If you want the lookup to be specific to a particular app, add its stanza to the version of `props.conf` in `$SPLUNK_HOME/etc/apps/<app_name>/local/`.
6. Restart Splunk Enterprise to implement your changes.

   If you have set up an automatic lookup, after restart you should see the `output` fields from your lookup table listed in the fields sidebar. From there, you can select the fields to display in each of the matching search results.

External lookup example

Here's an example of an external lookup that is delivered with Splunk software. It matches with information from a DNS server. It does not have a `props.conf` component, so it is not an automatic lookup. You access it by running a search with the `lookup` command.

Splunk Enterprise ships with a script located in `$SPLUNK_HOME/etc/system/bin/` called `external_lookup.py`, which is a DNS lookup script that:

- if given a host, returns the IP address.
- if given an IP address, returns the host name.

The configuration for this script resides in `$SPLUNK_HOME/etc/system/default/transforms.conf`.

```
[dnslookup]
external_cmd = external_lookup.py clienthost clientip
fields_list = clienthost,clientip
```

You can run a search with the `lookup` command that uses the `[dnslookup]` stanza from the default `transforms.conf`.

```
sourcetype=access_combined | lookup dnslookup clienthost AS host |
stats count by clientip
```

This search:

- Matches the `clienthost` field in the external lookup table with the `host` field in your events</code>
- Returns a table that provides a count for each of the `clientip` values that corresponds with the `clienthost` matches.

This search does not add fields to your events.

You can also design a search that performs a reverse lookup, which in this case returns a host value for each IP address it receives.

```
sourcetype=access_combined | lookup dnslookup clientip | stats count by
clienthost
```

Note that this reverse lookup search does not include an AS clause. This is because Splunk automatically extracts IP addresses as `clientip`.

About the external lookup script

Your external lookup script can be a Python script or a binary executable (such as a C++ executable). The script must take in data formatted as a partially empty CSV table and output data formatted as a filled-in CSV table. The arguments that you pass to the script are the headers for these input and output CSV tables.

In the DNS lookup example above, the CSV table contains two fields: `clienthost` and `clientip`. The fields that you pass to this script are the ones you specify in `transforms.conf` using the `external_cmd` attribute. If you do not pass these arguments, the script returns an error.

```
external_cmd = external_lookup.py clienthost clientip
```
When you run this search string:

```
... | lookup dnsLookup clienthost
```

You are telling Splunk software to:

1. Use the lookup table that you defined in `transforms.conf` as `[dnsLookup]`
2. Pass the values for the `clienthost` field into the external command script in CSV table format. The CSV table looks like this:

```
clienthost,clientip
work.com
home.net
```
This is a CSV table with `clienthost` and `clientip` as column headers, but without values for `clientip`. The script includes the two headers because they are the fields you specified in the `fields_list` attribute of the `[dnslookup]` stanza in the default `transforms.conf`.

The script then outputs the following CSV table, which is used to populate the `clientip` field in your results:

```
host,ip
work.com,127.0.0.1
home.net,127.0.0.2
```
Your script does not have to refer to actual external CSV files. But if it does refer to external CSV files, the filepath references must be relative to the directory where the scripts are located.

See also

In addition to using external lookups to add fields from external sources to events, you might use a scripted input to send data from non-standard sources for indexing or to prepare this data for parsing. For more information, see the Scripted inputs overview in *Developing Views and Apps for Splunk Web*.