Configuration parameters and the data pipeline

Data goes through several phases as it transitions from raw input to searchable events. This process is called the data pipeline and consists of four phases:

- Input
- Parsing
- Indexing
- Search

Each phase of the data pipeline relies on different configuration file parameters. Knowing which phase uses a particular parameter allows you to identify where in your Splunk deployment topology you need to set the parameter.

What the data pipeline looks like

This diagram outlines the data pipeline:
Monitor Input  FIFO Input  UDP Input  TCP Input  Scripted Input

parsingQueue

Parsing Pipeline
- source, event typing
- character set normalization
- line termination identification
- timestamp identification / normalization, event boundary identification
- regex transforms, parse-time field creation

indexQueue

Indexing Pipeline
- segmentation
- index building

rawdata → index files → Search
- Managing user interaction
- Displaying results
- Storing knowledge objects
The Distributed Deployment manual describes the data pipeline in detail, in "How data moves through Splunk: the data pipeline".

**How Splunk Enterprise components correlate to phases of the pipeline**

One or more Splunk Enterprise components can perform each of the pipeline phases. For example, a universal forwarder, a heavy forwarder, or an indexer can perform the input phase.

Data only goes through each phase once, so each configuration belongs on only one component, specifically, the first component in the deployment that handles that phase. For example, say you have data entering the system through a set of universal forwarders, which forward the data to an intermediate heavy forwarder, which then forwards the data onwards to an indexer. In that case, the input phase for that data occurs on the universal forwarders, and the parsing phase occurs on the heavy forwarder.

<table>
<thead>
<tr>
<th>Data pipeline phase</th>
<th>Components that can perform this role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>indexer</td>
</tr>
<tr>
<td></td>
<td>universal forwarder</td>
</tr>
<tr>
<td></td>
<td>heavy forwarder</td>
</tr>
<tr>
<td>Parsing</td>
<td>indexer</td>
</tr>
<tr>
<td></td>
<td>heavy forwarder</td>
</tr>
<tr>
<td></td>
<td>light/universal forwarder (in conjunction with the INDEXED_EXTRACTIONS attribute only)</td>
</tr>
<tr>
<td>Indexing</td>
<td>indexer</td>
</tr>
<tr>
<td>Search</td>
<td>indexer</td>
</tr>
<tr>
<td></td>
<td>search head</td>
</tr>
</tbody>
</table>

Where to set a configuration parameter depends on the components in your specific deployment. For example, you set parsing parameters on the indexers in most cases. But if you have heavy forwarders feeding data to the indexers, you instead set parsing parameters on the heavy forwarders. Similarly, you set search parameters on the search heads, if any. But if you aren't deploying dedicated search heads, you set the search parameters on the indexers.

For more information, see "Components and the data pipeline" in the *Distributed Deployment Manual*. 
How configuration parameters correlate to phases of the pipeline

This is a non-exhaustive list of configuration parameters and the pipeline phases that use them. By combining this information with an understanding of which Splunk component in your particular deployment performs each phase, you can determine where to configure each setting.

For example, if you are using universal forwarders to consume inputs, you need to configure `inputs.conf` parameters on the forwarders. If, however, your indexer is directly consuming network inputs, you need to configure those network-related `inputs.conf` parameters on the indexer.

The following items in the phases below are listed in the order Splunk applies them (ie `LINE_BREAKER` occurs before `TRUNCATE`).

**Input phase**

- `inputs.conf`
- `props.conf`
  - CHARSET
  - NO_BINARY_CHECK
  - CHECK_METHOD
  - CHECK_FOR_HEADER (deprecated)
  - PREFIX_SOURCETYPE
  - sourcetype
- `wmi.conf`
- `regmon-filters.conf`

**Structured parsing phase**

- `props.conf`
  - INDEXED_EXTRACTIONS, and all other structured data header extractions

**Parsing phase**

- `props.conf`
  - LINE_BREAKER, TRUNCATE, SHOULD_LINEMERGE, BREAK_ONLY_BEFORE_DATE, and all other line merging settings
  - TIME_PREFIX, TIME_FORMAT, DATETIME_CONFIG (datetime.xml), TZ, and all other time extraction settings and rules
TRANSFORMS which includes per-event queue filtering, per-event index assignment, per-event routing

• SEDCMD
• MORE_THAN, LESS_THAN

• transforms.conf
  • stanzas referenced by a TRANSFORMS clause in props.conf
  • LOOKAHEAD, DEST_KEY, WRITE_META, DEFAULT_VALUE, REPEAT_MATCH

Indexing phase

• props.conf
  • SEGMENTATION
• indexes.conf
• segmenters.conf

Search phase

• props.conf
  • EXTRACT
  • REPORT
  • LOOKUP
  • KV_MODE
  • FIELDALIAS
  • EVAL
  • rename
• transforms.conf
  • stanzas referenced by a REPORT clause in props.conf
  • filename, external_cmd, and all other lookup-related settings
  • FIELDS, DELIMS
  • MV_ADD
• lookup files in the lookups folders
• search and lookup scripts in the bin folders
• search commands and lookup scripts
• savedsearches.conf
• eventtypes.conf
• tags.conf
• commands.conf
• alert_actions.conf
• macros.conf
• fields.conf
• transactiontypes.conf
• multikv.conf
Other configuration settings

There are some settings that don't work well in a distributed Splunk environment. These tend to be exceptional and include:

- props.conf
  - CHECK_FOR_HEADER (deprecated), LEARN_MODEL, maxDist. These are created in the parsing phase, but they require generated configurations to be moved to the search phase configuration location.