# Understanding SPL syntax

http://docs.splunk.com/Documentation/Splunk/latest/SearchReference/UnderstandingSPLsyntax

# Required and optional arguments

**Splunk SPL** commands consist of required and optional arguments. Optional arguments are enclosed in square brackets [ ].

Consider this command syntax:

```
bin [<bins-options>...] <field> [AS <newfield>]
```

The required argument is `<field>`. To use this command, at a minimum you must specify `bin <field>`.

The optional arguments are `[<bins-options>...]` and `[AS <newfield>]`.

## User input arguments

Required arguments are shown in angle brackets < >.

Consider this command syntax:

```
replace (<wc-string> WITH <wc-string>)... [IN <field-list>]
```

The user input arguments are: `<wc-string>` and `<field-list>`.

## Repeating arguments

Some arguments can be specified multiple times. The syntax displays ellipsis ... to specify which part of an argument can be repeated. The ellipsis always appear **immediately after** the part of the syntax that you can repeat.

Consider this command:

```
convert [timeformat=string] (<convert-function> [AS <field>] )...
```

The required argument is `<convert-function>`, with an option to specify a field with the `[AS <field>]` clause.

Notice the ellipsis at the end of the syntax, just after the close parenthesis. In this example, the syntax that is inside the parenthesis can be repeated `<convert-function>` `[AS <field>]`.

In the following syntax, you can repeat the `<bins-options>...`.

```
bin [<bins-options>...] <field> [AS <newfield>]
```

# Grouped arguments

Sometimes the syntax must display arguments as a group to show that the set of arguments are used together. Parenthesis ( ) are used to group arguments.

For example in this syntax:

```
replace (<wc-string> WITH <wc-string>)... [IN <field-list>]
```

The grouped argument is `(<wc-string> WITH <wc-string>)...`. This is a required set of arguments that you can repeat multiple times.

# Keywords

Many commands use keywords with some of the arguments or options. Examples of keywords include:

- AS
- BY
- OVER
- WHERE

You can specify these keywords in uppercase or lowercase in your search. However, for readability, the syntax in the Splunk documentation uses uppercase on all keywords.

# Argument order

In the command syntax, the command arguments are presented in the order in which the arguments are meant to be used.

In the descriptions of the arguments, the **Required arguments** and **Optional argument** sections, the arguments are listed alphabetically. For each argument, there is a **Syntax** and **Description**. Additionally, for Optional arguments, there might be a **Default**.

# Boolean values

In the values for a argument, where Boolean **<bool>** values are required, the documentation specifies 'true' or 'false'. Other variations are accepted in commands. For example, for 'true' you can also use 't', 'T', 'TRUE', and '1'.

# Boolean operators

When a boolean operator is included in the syntax of a command, you must always specify the operator in uppercase. Boolean operators include:

- AND
- OR
- NOT

To learn more about the order in which boolean expressions are evaluated, along with some examples, see Boolean expressions in the *Search Manual*.

# BY clauses

A <by-clause> and a <split-by-clause> are not the same argument.

A <by-clause> displays each unique item in a separate **row**. Think of the <by-clause> as a grouping.

The <split-by-clause> displays each unique item in a separate **column**. Think of the <split-by-clause> as a splitting or dividing.

Wildcard characters ( * ) are not accepted in BY clauses.

# Fields and wildcard fields

When the syntax contains `<field>` you specify a field name from your events.

Consider this syntax:

```
bin [<bins-options>...] <field> [AS <newfield>]
```

The `<field>` argument is required. You can specify that the field displays a different name in the search results by using the `[AS <newfield>]` argument. This argument is optional.

For example, if the field is `categoryId` and you want the field to be named `CategoryID` in the output, you would specify:

`categoryId AS CategoryID`

The `<wc-field>` argument indicates that you can use wild card characters when specifying field names. For example, if you have a set of fields that end with "log" you can specify `*log` to return all of those fields.