# iis-search-examples.md




---

id: iis-search-examples

title: IIS Search Examples Cheat Sheet

sidebar_label: IIS Search Examples

description: The IIS Search Examples cheat sheet provides examples of useful IIS search queries for different use cases.

---

The IIS Search Examples cheat sheet provides examples of useful IIS search queries for different use cases.

The examples use this sample Access log message where applicable:

`2015-06-03 00:02:48 GET /myurl dp=mysearch 8200 10.1.1.1 Windows-RSS-Platform/2.0+ (IE+11.0;+Windows+NT+6.2) - - abcd.com 200 0 0 2583 271 15`

## Keyword Expressions

| Use Case | Sumo Logic Query Example |

| :-- | :-- |

| Look for failures or errors with a specific message. | `"ID = 123456" AND (fail* OR error)` |

| Look for errors in sshd logs. AND is assumed. Case insensitive, unless double-quoted. | `sshd (fail* OR error OR allowed OR identity)` |

| Look for general authorization failures excluding router messages. | `(fail* OR error?) NOT _source=routers` |

:::sumo More Info

For more information, see [Keyword Search Expressions](../get-started-with-search/build-search/keyword-search-expressions.md).

:::

## Parse, Count, and Top Operators

<table>

<tr>

<td>Use Case</td>

<td>Sumo Logic Query Example</td>

</tr>

<tr>

<td>Extract "from" and "to" fields using a simple wild card. For example, if a raw event contains "From: Jane To: John", then from=Jane and to=John.</td>

<td><code>* | parse "From: * To: *" as from, to</code></td>

</tr>

<tr>

<td>Extract IP address using a regex pattern.</td>

<td><code>* | parse regex <br/> "(?<c_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
</code></td>

</tr>

<tr>

<td>Identify pages visited, extracted as the "cs_uri_stem" field.</td>

<td><code>_source=IIS </code>

<br/><code>| parse "GET * " as cs_uri_stem</code></td>

</tr>

<tr>

<td>Identify messages with status code “200” and extract the sc_substatus, sc_win32_status, and sc_bytes fields.</td>

<td><code>_source=IIS</code>

<br/><code>| parse " 200 * * * " as sc_substatus, sc_win32_status, sc_bytes</code></td>

</tr>

<tr>

<td> </td>

<td>Examples below assume the parsing used above</td>

</tr>

<tr>

<td>Calculate the total number of bytes transferred to each client IP address.</td>

<td><code>| count, sum(sc_bytes) by c_ip</code></td>

</tr>

<tr>

<td>Calculate the average size of successful HTTP responses.</td>
<td><code>| avg(sc_bytes)</code></td>

</tr>

<tr>

<td>If the "sc_substatus" field is missing don’t exclude those messages (nodrop)…otherwise non-matches would be filtered out.</td>

<td><code>| parse " 200 * " as sc_substatus nodrop</code></td>

</tr>

<tr>

<td>Calculate the number of times a page has been visited.</td>

<td><code>| count by cs_uri_stem</code></td>

</tr>

<tr>

<td>Calculate the total number of pages by client IP addresses.</td>

<td><code>| count by c_ip</code></td>

</tr>

<tr>

<td>Calculate the total number of pages by client IP address, sort them highest to lowest.</td>

<td><code>| count by c_ip </code><br/><code>| sort by _count desc</code></td>

</tr>

<tr>

<td>Identify the top 10 pages.</td>

<td><code>| count by cs_uri_stem </code><br/><code>| top 10 cs_uri_stem by _count</code>
</td>

</tr>

<tr>

<td>Identify the top 10 client IP addresses by bandwidth usage.</td>

<td><code>| sum(sc_bytes) as total_bytes by c_ip</code><br/><code>| top 10 c_ip by
total_bytes</code></td>

</tr>

<tr>

<td>Identify the top 100 client IP addresses by number of hits.</td>

<td><code>| count by c_ip</code><br/><code>| top 100 c_ip by _count</code></td>

</tr>

</table>

:::sumo More Info

For more information, see [Parsing](/docs/search/search-query-language/parse-operators), [Count](/docs/search/search-query-language/group-aggregate-operators/count-count-distinct-and-count-frequent), and [Top](/docs/search/search-query-language/search-operators/top).

:::

## Timeslice and Transpose

<table>

<tr>

<td>Use Case</td>

<td>Sumo Logic Query Example</td>

</tr>

<tr>

<td>For the host / domain "abcd.com", count by sc_status with a timeslice of 15m</td>

<td><code>source=IIS</code><br/>

<code>| parse "abcd.com * " as sc_status</code><br/>

<code>| timeslice 15m</code><br/>

<code>| count by _timeslice, sc_status</code></td>

</tr>

<tr>

<td>Pivot the results so that time is on the X axis and sc_status is on the Y axis (values can be displayed in legend)</td>

<td><code>| transpose row _timeslice column sc_status</code></td>

</tr>

</table>

:::info

For more information, see [Timeslice](/docs/search/search-query-language/search-operators/timeslice) and [Transpose](/docs/search/search-query-language/search-operators/transpose).

:::
## Conditional Operators

<table>

<tr>

<td>Use Case</td>

<td>Sumo Logic Query Example</td>

</tr>

<tr>

<td>For the source "IIS", find all messages with a client error status code (40*)</td>

<td><code>_source=IIS 40*</code><br/>

<code>| parse "abcd.com * " as sc_status</code><br/>

<code>| where sc_status matches "40*"</code></td>

</tr>

<tr>

<td>For the source "IIS/Access", count hits by browser</td>

<td><code>source=IIS/Access </code><br/>

<code>| parse "* * * * * * * * " as date, time, csmethod, cs_uri_stem, cs_uri_query, s_port, c_ip, cs_UserAgent </code><br/>

<code>| if (cs_UserAgent matches "*MSIE*",1,0) as ie </code><br/>

<code>| if (cs_UserAgent matches "*Firefox*",1,0) as firefox </code><br/>

<code>| if (cs_UserAgent matches "*Safari*",1,0) as safari</code><br/>

<code>| if (cs_UserAgent matches "*Chrome*",1,0) as chrome</code><br/>

<code>| sum(ie) as ie, sum(firefox) as firefox, sum(safari) as safari, sum(chrome) as chrome</code></td>

</tr>

<tr>

<td>Use the where operator to match only weekend days.</td>

<td><code>* | parse "day=*:" as day_of_week </code><br/>

<code>| where day_of_week in ("Saturday","Sunday")</code></td>

</tr>

<tr>

<td>Identify all URLs that contain the subdirectory "Courses" in the path.</td>

<td><code>* | parse "GET * " as cs_uri_stem </code><br/>

<code>| where cs_uri_stem matches "*Courses*"</code></td>

</tr>

<tr>

<td>Find version numbers that match numeric values 2, 3 or 6. Use the num operator to change the string into a number.</td>

<td><code>* | parse "Version=*." as number </code><br/>

<code>| num(number) | where number in (2,3,6)</code></td>

</tr>

</table>

:::sumo More Info

For more information, see [Where](/docs/search/search-query-language/search-operators/where) and [If](/docs/search/search-query-language/search-operators/if).

:::

For any query, you can increase specificity by adding metadata fields to the keyword expression. Metadata fields include `_sourceCategory`, `_sourceHost `, and `_sourceName`. Edit Source metadata in the **Collection** tab. For details, see [Search Metadata](/docs/search/get-started-with-search/search-basics/built-in-metadata).