format

Synopsis

Displays a specified column in the complex search summary table in a specified format. Can be used only with **Display**, **Display only**, and **Grou p by** commands.

Syntax

```
in [format_type] format)(["Input_Unit"],)(["Output_Unit"])
```

Required Arguments

`format_type`

Syntax: number, **simple**, **time**, **date**, **volume**, **regexp**, **expression** or **query**

Description: The format in which to display the values of a specific column in the complex search summary table. For a time format_type, if no unit appears after time format, XpoLog assumes that the column value is in milliseconds and displays it in the maximal possible unit (for example, if the value is 2000, the output is 2 seconds; if the value is 120000, the output is 2 minutes, etc.).

Optional Arguments

`"Input_Unit"`

Syntax: Volume Units - **B**, **KB**, **MB**, **GB**; Time Units: **microsec**, **ms**, **sec**, **min**, **hour**, **day**

Description: The input unit of the format type.

`"Output_Unit"`

Syntax: Volume Units - **B**, **KB**, **MB**, **GB**; Time Units: **microsec**, **ms**, **sec**, **min**, **hour**, **day**

Description: The unit in which to convert the format type.

Note: If only one unit appears in the syntax, XpoLog assumes that it is the output unit, and that the input value is in milliseconds (for time) or bytes (for volume). If no unit appears in the syntax, XpoLog outputs the log value in milliseconds (for time) or bytes (for volume).

Description

Displays the column values in the specified format, assuming the default input and output units, if they are not specified, and converting to a specific output unit from a specific input unit, if specified.

Text can be formatted into the following format types:

- **number** – formats the text in the column to number format; (“#.##”) – the decimal format of the number
- **simple** – displays columns in difference format: (“column.name1 – column.name2”) – replace the columns with the values from the result
- **time** – displays the value in a time format of the default unit or of the indicated (“[OUTPUT_UNIT]”) , (“[INPUT_UNIT] “,“[OUTPUT_UNIT]”) – displays the column in output format and uses the input unit In case it is different from milliseconds.
  Time units: [microsec,ms,sec,min,hour,day]
- **date** – displays the value in day format: (“[SIMPLE DATE FORMAT]”) – change the date format
- **volume** – displays the value in volume format way: (“[OUTPUT_UNIT]”) , (“[INPUT_UNIT] “,“[OUTPUT_UNIT]”) – display the column in output format and use input unit in case it is different from bytes.
  Volume units: [B,KB,MB,GB]
- **regexp** – use regexp to extract values from the data: (“[REGEXP]”) – display the first group that is found from the regular expression
- **expression** – displays the column result after performing an expression on the original contents.
  Display Column_Name in regexp format("REGEXP"), where REGEXP is the regular expression to be executed on the value in Column_Name (“[EXPRESSION]”) – use an expression to calculate different result value
- **query** - displays an aggregated result broke into groups based on search queries constraints.
  ...group by FIELD_NAME in query format ("SEARCH_QUERY_1","RESULT_NAME_2","SEARCH_QUERY_2","RESULT_NAME_N",..., "SEARCH_QUERY_N","RESULT_NAME_N")
  It is possible to use '*' at the end as a query to group the undefined results of the other queries:
  status != NULL in log.access | count | group by status in query format ("status=200","VALID","*","ALL_THE_REST")
- **exception** - displays an aggregated result broke into groups based on number of lines in the stack trace.

...group by FIELD_NAME in exception format ("NUMBER_OF_LINES","SHOW_MESSAGE")

error in log.log4j log **|** *count* **|** *group by* message *in exception format* ("1","true")

- **replace** – use replace to replace a value from the data with a custom value.

  ....group by FIELD_NAME in replace format
  ("REGEXP_1","REPLACE_TEXT_1","REGEXP_2","REPLACE_TEXT_2",….,"REGEXP_N","REPLACE_TEXT_N")

  status != NULL in log.access | count | group by status in replace format ("200","OK","302","Resource temporarily moved to a new locatio n","304","Not Modified")
- **UserAgentDetect** – displays an aggregated result broke into groups based on types to view (browser,version,platform,os).

  ...group by FIELD_NAME in UserAgentDetect format ("TYPE_1+….+TYPE_N")

  status != NULL in log.Access Log | *count* | *group by* user agent *in useragentdetect format* ("browser+version")

Examples – Volume Format: bytes sent column contains numeric values representing volume.

Example 1:

```
* in log.access | avg bytes sent | display avg in volume format
```

XpoLog formats **avg** of **bytes sent** in volume format, automatically assuming that the log value is in bytes.

Example 2:

```
* in log.access | avg bytes sent | display avg in volume format(“MB”)
```

XpoLog formats **avg** of **bytes sent** in volume format, automatically assuming that the log value is in bytes, and converts and outputs the value in MB.

Example 3:

```
* in log.access | avg bytes sent | display avg in volume format(“KB”,”MB”)
```

XpoLog formats **avg** of **bytes sent** in volume format, assuming that the log value is in KB, and converts and outputs the value in MB.

Examples – Time Format: time taken column contains numeric value representing time.

Example 1:

```
* in log.access | avg time taken | display avg in time format
```

XpoLog formats **avg** of **time taken** in time format, automatically assuming that the log value is in milliseconds.

Example 2:

```
* in log.access | avg time taken | display avg in time format(“SEC”) à format to seconds
```

XpoLog formats **avg** of **time taken** in time format, automatically assuming that the log value is in milliseconds, and converts and outputs the value in seconds.

Example 3:

```
* in log.access | avg time taken | display avg in time format(“SEC”,”MIN”) à format from seconds to minutes
```

XpoLog formats **avg** of **time taken** in time format, assuming that the log value is in seconds, and converts and outputs the value to minutes.

Regular Expressions:

1. XpoLog groups by URL field which has multiple parts divided by slashes / and then uses a regular expression to format the result to present only part of the URL based on the regular expression criteria, I.E. present only the last part after the last slash / in the URL:

   URL Example:
   [URL] /home/web-main/css/texts.css

   ### XpoLog Query:

   * in log.access log | count | group by url as formatted-url | order by count desc | display formatted-url in regexp format (".*/([^/]+)")

   ### Result:

formatted-url	Count
iframebackground.html	1,122
xplg.css	309
links.css	296
texts.css	295
logo.gif	277
brochure_inter.gif	234

2. XpoLog uses a regular expression to format the Description field which contains multiple lines with different values based on the regular expression criteria, I.E. extract from the entire Description field only the value which comes after 'Account Name:' and group by it only (as if it was a pre-configured field in the log):

   Description Example:
   ...[Description] An account was logged off.
   Subject:
   Security ID: S-1-5-21-3480273402-748593870-3636473903-1144
   Account Name: xplg
   Account Domain: XPOLOG
   Logon ID: 0xa078ea24
   Logon Type: 3
   This event is generated when a logon session is destroyed. It may be positively correlated with a logon event using the Logon ID value. Logon IDs are only unique between reboots on the same computer.

   XpoLog Query:
   (*) in log.application | count | group by Description as UserName in regexp format ("Account Name:\s+(\w+)")

   Result:

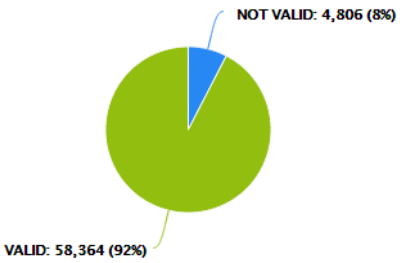
UserName	Count
	116
SBSMonAcct	438
SYSTEM	9
xplg	165
XPOLOGDOMAIN	322

Query Format:

1. XpoLog groups by STATUS which has multiple values, and then based on query criteria it breaks the result to different pieces:
   Status values may vary from 200, 302, 404, 500, etc. but in order to break it into two groups 200 - defined as valid and not 200 as not valid the query format handles it:

   XpoLog Query:
   * in log.access | count | group by status in query format ("status=200","VALID","status != 200","NOT VALID")

   Result:



2 results (based on 63,171 events) in 3 logs | All Time

Viewing 1 - 2

status	Count
NOT VALID	4,806
VALID	58,364