XpoLog Patterns Language

XpoLog enables users to use the *Patterns* language to normalize log records into tabular format.

Records in the log can usually be presented by a combination of types. It is recommended to present the logged information in as detailed way as possible. Doing so gives greater possibilities in manipulating and analyzing the data, such as filtering by a specific ip, priority, date, or specific text column. You should try to be as descriptive as possible.
Note that you can configure several patterns for one log using XpoLog multi pattern.
Each command will be treated as a column of data in the log view table.

Note: If a pattern is formulated incorrectly, the log records may display wrong data or no data at all.

IMPORTANT!!!
After defining a pattern, it is highly recommended to click the **verify pattern** link to see the results of your definition in the table at the bottom of the page.
If you do not see data at all or you see wrong data, check your pattern definition.

For any definition problem, send XpoLog support the log example by email to support@xplg.com, and we will help you define a pattern.

The following table describes the *Patterns* language:

Pattern Type		Syntax
String	any string of characters, including multi line strings	{string}
Text	any single-line string of characters	{text}
Date	a date string	{date, MM-dd-yyyy}
Timestamp	a timestamp representing a date string	{timestamp}
Number	a numeric string	{number}
Choice	a set of strings that can appear in a record	{choice,value1;value2...,valueN} **Note:** All optional choices should appear as a semicolon separated list inside the tag.
IP Address	An IP address	{ip}
Geo IP Address	A Geo IP address	{geoip,type=country:region:city} **Note:** All optional combinations of country, region and city are valid, for example: country:region, region:city, country etc.
Priority	A set of priorities that can appear in a record	{priority,priority1;priority2...,priorityN} **Note:** All optional priorities should appear as a semi-colon separated list inside the tag.

Expression	The expression that will be used according to the source columns given	{expression}
Regular Expression	A regular expression, used to extract part of the data from another column read more about regular expressions in the regular expressions help page	{regexp,refIndex=index | refName=column_name;columnType=date/timestamp/number;dateUIFormat=DISPLAY_DATE_FORMATmultiLine=true/false,(regular_expression_to_ext
Properties	A set of key-value properties that can appear in a record	{properties,keysSep=[Keys_Separator];propSep=[Properties_Separator];,key1;key2;....;keyN}
Json	A set of key-value pairs	{json,key1;key2;key3;....;keyN}
Term	a constant string that appears in a record and needs to be displayed in the log view	{term,TERM} TERM is the constant string
Free Text	any text you wish to display in the log view, usually used in multi-pattern logs to distinguish records	{freetext,FREETEXT} FREETEXT is the text that you wish to display in the log view.

Value Mapping Option I (manual mapping)	maps an original value from log to a converted value	{map,val1=convertedVal1;val2=convertedVal2;val3=convertedVal3;...;valN=convertedValN}
Value Mapping Option II (mapping based on an external file)	maps an original value from log to a converted value	{map,refIndex=ORIG_COL_INDEX,file:FULL_PATH_TO_FILE}
Value Mapping Option III (r egular expression manual mapping)	maps an original value from log to a converted value	{regexp,refIndex=index | refName=column_name;columnType=map;mapping=val1:convertedVal1^val2:convertedVal2^...^valN:convertedValN,(regular_expression_to_extract)}
Block	an optional string that does not appear in all records	{block,start,emptiness=true}XXX{block,end,emptiness=true}
Horizontal Tab	a tab delimiter	{tab}
End of Line	end of line, used in records that spread over more than one line	{eol}
End of Entity	end of entity, used to mark the end of a record, improves the efficiency of the parsing process	{eoe}

Optional Identifiers for Date Pattern

The following table provides examples of optional identifiers that can be used in a Date pattern.

Identifier	Text in Log	Pattern
MM - numeric month	01-25-1986	{date,MM-dd-yyyy}
MMMMM - full textual month	25/July/1986	{date,dd/MMMMM/yyyy}
MMM - textual month	25/Jul/1986	{date,dd/MMM/yyyy}
dd - numeric day	01:25:1986	{date,MM:dd:yyyy}
EEEEE - full textual day	Friday 01-25-00	{date,EEEEE MM-dd-yy}
EEE - textual day	Fri 01-25-00	{date,EEE MM-dd-yy}
yy - 2 digit year	25/Jul/86	{date,dd/MMM/yy}

yyyy - 4 digit year	25/Jul/1986	{date,dd/MMM/yyyy}
HH - 24 hour	18:05:23	{date,HH:mm:ss}
hh - 12 hour	6:05:23 PM	{date,hh:mm:ss}
a - AM/PM marker	6:05:23 PM	{date,hh:mm:ss a}
mm - minute	18-05-23	{date,HH-mm-ss}
ss - second	18:05:23	{date,HH:mm:ss}
SSS - millisecond	18:05:23 253	{date,HH:mm:ss SSS}
z - general time zone	18:05:23 EST	{date,HH:mm:ss z}
Z - RFC 822 time zone	18:05:23 -0400	{date,HH:mm:ss Z}
X - ISO 8601 time zone	18:05:23 -04:00	{date,HH:mm:ss XXX}
'TEXT' - a constant text that appears in the date string	07-1986D25	{date,MM-yyyy'D'dd}

Attributes Supported by All Types

All pattern types support the attributes described in the following table.

Attribute	Remark	Examples
Name	This attribute should always follow the tag name with a leading colon.	{string:Title} {date:Start Date,dd/mm/yyyy} {priority:Severity,DEBUG;INFO;ERROR} {number:Status Code}
uiMessageLength	This attribute allows you to limit the length of data displayed in a specific column. If the data is longer than specified, it will be divided to several lines.	{string:Title,uiMessageLength=20}
charsLength	This attribute allows you to force the existence of a fixed number of characters in a string, even if there are less characters in the record.	{string:Title,charsLength=10}
stopPattern	This attribute allows you to set a regular expression that will serve as the column's delimiter. This is useful in case there is no natural delimiter (such as space or a non-word character) between two columns.	{text,stopPattern=\d+.\d+.\d+}
masker	This attribute allows you to set a regular expression for masking the column's data. If a match is found for the column's value, then the matched part will be displayed as a string of asterisks ('*').	{string:UserID,masker=(.+)}

Special Cases

- The right/left curly brackets characters ({ }) are reserved in XpoLog syntax, therefore they can be used as literal only if as \u007B (left curly bracket) or \u007D (right curly bracket).
- The quote character (') can be used as literal only if preceded by another quote **''**.

Examples of Patterns Used on Logs

The following are examples of patterns that can be used to tune the parsing results of logs.

Log	Parsed Records	Data Pattern

Log 1	2003-02-12 12:37:26 ContextConfig[/examples]: Missing application web.xml, using defaults only 2003-02-12 12:37:26 StandardManager[/examples]: Seeding random number generator class java.security.SecureRandom 2003-02-12 12:37:30 StandardManager[/examples]: Seeding of random number generator has been completed 2003-02-12 12:37:30 StandardWrapper[/examples:default]: Loading container servlet default 2003-02-12 12:37:30 StandardWrapper[/examples:invoker]: Loading container servlet invoker	{date,yyyy-MM-dd HH:mm:ss} {string}[/{string}]: {string}
Log2	127.0.0.1 - - [26/Dec/2001:19:49:23 +0200] "GET / HTTP/1.1" 200 1494 127.0.0.1 - - [26/Dec/2001:19:49:23 +0200] "GET /apache_pb.gif HTTP/1.1" 200 2326 127.0.0.1 - - [26/Dec/2001:19:52:48 +0200] "GET /examples/ HTTP/1.1" 404 277 127.0.0.1 - - [26/Dec/2001:19:54:37 +0200] "GET /examples/jsp/snp/snoop.jsp HTTP/1.1" 404 294 127.0.0.1 - - [28/Dec/2001:09:54:37 +0200] "GET /puga/main.html HTTP/1.1" 404 282	{string} - - [{date,dd/MMM/yyyy:HH:mm:ss Z} +0200] "{string}" {number} {number}
Log3	[Wed Dec 26 19:52:48 2001] [error] [client 127.0.0.1] File does not exist: c:/devapp/apache/apache/htdocs/examples/ [Wed Dec 26 19:55:01 2001] [error] [client 127.0.0.1] File does not exist: c:/devapp/apache/apache/htdocs/_vti_bin/owssvr.dll [Wed Dec 26 19:55:01 2001] [error] [client 127.0.0.1] File does not exist: c:/devapp/apache/apache/htdocs/msoffice/cltreq.asp	[{string} {date,EEE MMM dd HH:mm:ss yyyy] [{priority,debug;info;warn;error;fatal}
Log4	28/02/03 20:23:16 ERR Critical error on section 34 on module 5 [Channel 9] 4.4.4.4 28/02/03 20:25:35 DBG information arrived to fusion zone, restoring states [Channel 39] 4.8.4.9 28/02/03 20:33:22 WRN port collision seeking another [Channel 19] 4.4.4.4 28/02/03 20:33:22 FLW DB connection open structure initiated [Channel 9] 4.23.12.5	{date,dd/MM/yy HH:mm:ss} {priority,DBG;FLW;WRN;ERR} {string} [{string}] {string
Log5	5 d MBGN Talk to port 9 f MLPT1 Port open 0 x MCOM Com port open	{number}{tab}{string}{tab}{string}{tab}{string}
Log6	5 d MBGN procId=123 Talk to port 9 f MLPT1 Port open 0 x MCOM procId=456 Com port open	{number}{tab}{string}{tab}{block,start,emptiness=true}procId={string}{block,end,e