

IETF-126

Vienna

CDNI Discussion on Metadata Expression Language (MEL)

Agenda

Current Version: draft-ietf-cdni-metadata-expression-language-02

- Summary of changes since version 01:
 - Changed syntax definition from BNF to Augmented BNF (aBNF) as defined in RFC-5234.
 - Fixed minor typos.
 - `FCI.SupportedMELFeatures` list of supported variables updated to drop the `<name>` after `req.h` and `resp.h`
- Identify work to be completed over the next couple of months to:
 - Finalize the aBNF.
 - Clarify operator precedence and associativity.
 - Clarify Core Variable definitions.
 - Overall document review by more WG members.
- Identify any other open issues that we will need to cover to get to WG last call.
- Set date/time for working sessions to address all of the above and generate version 03 draft.

aBNF (draft-ietf-cdni-metadata-expression-language-02)

```
; Top-level

expression = identifier "=" expression
           / expression "?" expression ":" expression
           / function-call
           / expression operator expression
           / prefix-op expression
           / "(" expression ")"
           / literal
           / list

; List

list       = "[" "]"
           / "[" expr-list "]"

expr-list  = expression *("," expression)

; Function calls

function-call = function-name "(" ")"
              / function-name "(" expr-list ")"

function-name = "integer" / "real" / "string" / "boolean"
              / "upper" / "lower"
              / "match" / "match_replace"
              / "add_query" / "add_query_multi"
              / "remove_query" / "remove_query_multi"
              / "path_element" / "path_elements"

; Operators

operator = logic-operator
         / math-operator
         / bit-operator
         / string-operator
         / comparison-operator
         / regex-operator
         / glob-operator
         / ip-operator
```

```
prefix-op   = "+" / "-" / "!" / "~"

logic-operator = "and" / "or"

math-operator = "+" / "-" / "*" / "/" / "%"

bit-operator  = "|" / "&" / "~" / "<<" / ">>"

string-operator = "."

comparison-operator = "==" / "!=" / ">=" / "<=" / ">" / "<"

regex-operator = "regexmatch" / "regexmatchi"
               / "!regexmatch" / "!regexmatchi"
               / "~="

glob-operator  = "globmatch" / "globmatchi"
               / "!globmatch" / "!globmatchi"
               / "*=" / "!*="
               / "%*=" / "!%*="

ip-operator    = "ipmatch" / "!ipmatch"

; Literals

literal        = number / string / "true" / "false" / "nil"

number         = ["-"] 1*DIGIT ["." 1*DIGIT]

string         = DQUOTE *string-char DQUOTE

string-char    = %x20-21 / %x23-7E / escaped-char
               ; printable ASCII except DQUOTE, plus escapes

escaped-char   = "\\" (DQUOTE / "\" / "n" / "r" / "t")

; Identifiers

identifier     = id-start *id-char

id-start       = ALPHA / "_"

id-char        = ALPHA / DIGIT / "." / "-" / "_" / "#"
```

Clarifying Operator Precedence and Associativity

- In the current draft, Section 6 (Operators and Keywords) has 2 tables that require updating:
 - The Operator table will have 2 new columns (Associativity and Precedence), where Associativity will be either “left-to-right” or “right-to-left” and Precedence will be an ordinal starting at 1 and increasing to rank the precedence of all operators relative to each other.
 - A “member access” operator (“.”) will be defined and added to aBNF.
 - “and”, “or”, and “not” will be moved from the Keywords table to the Operators table.

Clarifying Core Variable Definitions

- In the current draft, Section 5 (Core Variables) requires more details to clarify a variety of issues such as normalization of request header names and variables referencing headers that may not be present in a request or response.

Will Hawkins to give a brief overview of his suggestions on the above topics. Will's notes can be found here: <https://gist.github.com/hawkinsw/4635842fd54029840050dcd14a761366>