C:\Apps\TQLEditor\tqleditor.gif **TQL Quick Reference Guide V4.7.1**

© Copyright 2019, Apelon, Inc.

**Statements** – optional elements are shown in red

*query* := *statement query*

*statement\_block := statement* |{ *statement* ***…***}

A – unrestricted (anywhere)

N – non-contextual statement

R - restricted contextual statement

U - unrestricted contextual statement

**Collection**– a group of Concepts or Terms

FROM mods *collection* *statement\_block* (N)

*collection* := ALL WITH *selectors*

*context* WITH *selectors*

*context :=* [ *namespace* : version-name ] |

[*namespace* # version-date ] |

{ *subset* : version-name } |

{ *subset* # version-date } |

*conset*  | *termset*

**Conditional** – conditionally execute a statement block

IF *predicates* *statement\_block* (U)

ELSEIF *predicates* *statement\_block* (U)

ELSE *statement\_block* (U)

**Context** – create/delete/rename a DTS Context

CREATE < *authority* > ; (N)

CREATE [ *namespace* : *authority* : *type* : *link* ] ; (N)

*type* := THESAURUS | ONTYLOG | ONTYLOG\_EXTENSION

CREATE *create-ctx* FROM *collection* ; (N)

CREATE *create-ctx* FROM *build-ctx* *log-op* *build-ctx* ; (N)

*create-ctx* := { *subset : authority* } | *conset* | *termset*

*build-ctx* := { *subset* } | *conset* | *termset*

DELETE *context* ; (N)

RENAME *context1* TO *context2* ; (N)

**Edit** – edit data

CREATE\_CONCEPTS mods *create-concept-list* ; (R)

*create-concept* := concept :code-literal : id-literal

*concept* := name | *string-literal*  | ^string^

CREATE\_TERMS mods *create-term-list* ; (R)

*create-term* := term :code-literal : id-literal

*term* := name | *string-literal*  | ^string^

DELETE\_CONCEPTS *mods* ; (R)

DELETE\_CONCEPTS *mods* *concept-list* ; (R)

DELETE\_TREES *mods*; (R)

DELETE\_TREES *mods* *concept-list* ; (R)

DELETE\_TERMS *mods* ; (R)

DELETE\_TERMS mods *term-list* ; (R)

DELETE *delete-attr-list;* (U)

*delete-attr := attr-name . qual name = value . qual-value*

SET *set-attr&var-list*; (U)

*set-attr* := *attr-name . qual name = expr . qual-expr*

*attr-name*:= name(type ) |string-literal (type) | ^string^ (type)

UPDATE *set-attr-list* ; (U)

**Export** – export data

EXPORT\_CONCEPTS *mods*  ; (R)

EXPORT\_SUBSET *mods* ; (R)

EXPORT\_NAMESPACE *mod*s ; (R)

EXPORT *mods*  *expr*-*list* SORTED\_BY *sort-list* ;(R)

**For** – iterate over attribute instances (U)

FOR *attr-name statement-block*

**Output** – output a variable or value

LOG *expr-list;*  (A/U)

PRINT *expr-list;* (A/U)

**Read** – read values from a file

READ mods *file-name-expr* *statement\_block* (A)

**Set** **Variables**– set TQL variables

SET *set-var-list;* (A)

*set-var := TQLVariable-name = expr* |

*% UserVariable-name = expr*

***Parameters***

PARAMETER *parm-list*; (N)

*parm* := @name@ :type : help

CONSTRAIN *constrain-list*; (N)

*constrain* := TO op expr

***Predicates***

*predicate* | *predicates* *log-op* *predicates*  | ( *predicates* )

***Selectors***

*selector* | *selectors* *log-op* *selectors*  | ( *selectors* )

***Selector* –** see **Selector & Predicate Operators** table

*select-attribute op expr*

*function op expr*

***Logical Operators*** *– log-op*

AND | OR | AND\_NOT

**Expression** – string-valued expression

*expr* := *expr-element* *expr-op expr-element …*

*expr-element* := @name@

*literal*

*attr*

*variable*

*function*

(*attr*) “group values”

*concept-attr* -> *attr* “referencing”

*prop-attr* | *encode-attr* “encoded by”

expr-op := + | - | &

***Comments***

-- comment text to end of line

/\* comment text \*/

***Modifiers*** – see also **Variables** Table

/APPEND = *boolean-literal*

/AXIS = *string-literal*

/CONCEPTS

/DELIMITER = *string*-*literal*

/EXPORTFILE = *string*-*literal*

/HEADER = *boolean-literal*

/IGNORE\_EXISTENCE

/LIMIT = *integer-literal*

/PERMANENT

/PRUNE\_TERMS = *boolean-literal*

/RETAIN\_HEAD = *boolean-literal*

/STATUS = *status*

/TERMS

/TYPEDEFS = *boolean-literal*

/UNIQUE = *boolean-literal*

status := ALL | ACTIVE | INACTIVE | DELETED

**Attribute Keywords**

|  |  |  |
| --- | --- | --- |
| ***Attribute Name*** | ***Usee*** | ***Description*** |
| CONCEPT | EFS | The Concept itself. |
| CONCEPT\_NAME | EFS | The Concept’s Name. |
| CONCEPT\_CODE | EFS | The Concept’s Code. |
| CONCEPT\_ID | EFS | The Concept’s Id. |
| CONCEPT\_STATUS | EF | The Concept’s Status. |
| TERM | EFS | The Term itself |
| TERM\_NAME | EFS | The Term’s name. |
| TERM\_CODE | EFS | The Term’s Code. |
| TERM\_ID | EFS | The Term’s Id. |
| TERM\_STATUS | EFS | The Term’s Status. |
| NAME | EFS | CONCEPT\_NAME or TERM\_NAME |
| CODE | EFS | CONCEPT\_CODE or TERM\_CODE |
| ID | EFS | CONCEPT\_ID or TERM\_ID |
| STATUS | EFS | CONCEPT\_STATUS or TERM\_STATUS |
| NAMESPACE | EFS | The Concept’s or Term’s Namespace. |
| VERSION\_NAME | EFS | The name of the Concept or Term snapshot Version. |
| VERSION\_DATE | EFS | The date of the Concept or Term snapshot Version. |
| PREFERRED\_NAME | EFS | The Concept’s Preferred Term (Synonym) or empty string. |
| RESOLVED\_NAME | EFS | The Concept’s Preferred Name or the Concept Name. |
| QUALIFIED\_NAME | EFS | The Concept Name followed by its Namespace name. |
| PRIMITIVE | EFS | “Primitive” or “Defined” if Concept is Ontylog.. |
| SYNONYM | EF | The Concept’s Synonyms |
| PROPERTY | EF | The Concept’s Properties |
| ASSOCIATION | EF | The Concept’s Associations. |
| ROLE | EF | The Concept’s Roles. |
| KIND | EFS | The Concept’s Kind. |
| SUBSET | EFS | The name(s) of the Concept’s Subsets. |
| DEFINING\_CONCEPT | EFS | The Ontylog Concept’s Defining Concepts.  . |
| CHILD | EFS | The Concept’s Children. (Subconcepts).Uses inv AXIS. |
| DESCENDANT | EFS | The Concept’s Descendants. Uses inv Axis. |
| PARENT | EFS | The Concept’s Parents (Superconcepts).Uses AXIS. |
| PARENT\_PLUS | S | Same as above but includes the Concept. |
| ANCESTOR | S | The Concept’s Ancestors. Uses AXIS. |
| ANCESTOR\_PLUS | S | Same as above but includes Concept. |

*Notes:* E = export F = function S = selector

**Selector & Predicate Operators** - *obj op operand*

| ***Operator String*** | ***Right Operand Type*** | ***Description*** |
| --- | --- | --- |
| MEMBER\_OF | *context* | *obj* is a member of the context. |
| NOT\_MEMBER\_OF | *context* | *obj* is not a member of the context. |
| CHILD\_OF | Concept name literal | *obj* is a Child (Subconcept) of the referenced Concept. \* |
| NOT\_CHILD\_OF | Concept name literal | *attr* is not a Child (Subconcept) of the referenced Concept. \* |
| CHILD\_OF\_PLUS | Concept name literal | Same as CHILD\_OF but includes referenced Concept. \* |
| NOT\_CHILD\_OF\_ PLUS | Concept name literal | Same as NOT\_CHILD\_OF but includes referenced Concept. \* |
| DESCENDANT\_OF | Concept name literal | *obj* is a descendant of the referenced Concept. \* |
| NOT\_DESCENDANT\_ OF | Concept name literal | *obj* is not a descendant of the referenced Concept. \* |
| DESCENDANT\_OF\_ PLUS | Concept name literal | Same as DESCENDANT\_OF but includes referenced Concept. \* |
| NOT\_DESCENDANT\_ OF\_ PLUS | Concept name literal | Same as NOT\_DESCENDANT\_OF but includes referenced Concept. \* |
| FOLLOWS | String literal | *obj value* follows operand lexically. |
| PRECEDES | String literal | *obj value* precedes operand lexically. |
| EQUALS | String literal (wildcards) | *obj* value equals the literal. |
| NOT\_EQUALS | String literal (wildcards) | *obj* value does not equal the literal. |
| EXISTS | None | At least one instance of the *obj* exists on the collection object. Same as EQUALS “\*”. |
| NOT\_EXISTS | None | *obj* does not exist on the collection object. Searches full *context*. |
| MATCHES | RegEx string literal | *obj* value matches the RegEx. |
| NOT\_MATCHES | RegEx string literal | *obj* value does not match the RegEx. |
| IN\_RANGE | Numeric literal “:” numeric literal | Numeric interpretation of the *obj* value lies between the lower and upper limit (incl) |
| NOT\_IN\_RANGE | Numeric literal “:” numeric literal | Numeric interpretation of the *obj* value does not lie between the lower and upper limit (incl) |
| = | Numeric literal | Numeric interpretation of the *obj* value equals the literal. |
| <> | Numeric literal | Numeric interpretation of the *obj* value does not equal the literal. |
| < | Numeric literal | Numeric interpretation of the *obj* value is less than the literal. |
| <= | Numeric literal | Numeric interpretation of the *obj* value is less than or equal to literal. |
| > | Numeric literal | Numeric interpretation of the *obj* value is greater than the literal. |
| >= | Numeric literal | Numeric interpretation of the *obj* value is greater than or equal to the literal. |

*Notes*: \* Not permitted when *context* is a ConSet or TermSet.

**TQL Functions** – *function*(*attr*)

|  |  |
| --- | --- |
| ***Function Name*** | ***Description*** |
| LENGTH | Returns the string length of the *attr*. |
| COUNT | Returns the number of instances of the *attr* on the collection object. |

**Variables**

| ***Variable*** | ***Applies To*** | ***Description*** |
| --- | --- | --- |
| APPEND | EXPORT | If “Yes” or “True”, export is appended to the export file. TQL default is “False”. |
| AXIS | Hierarchy operations and DELETE\_TREES | The name of the role or association used as the parent-to-child “axis”. Default is “Parent Of. Can have the “INV” prefix . |
| DATE\_FORMAT | EXPORT | Format of date values. |
| DELIMITER | EXPORT | Two character delimiter. First character is field delimiter; second (optional) character is group delimiter. Field default is “|” and group default is the colon “:”. |
| EXPORTFILE | EXPORT | The name of the export file. |
| HEADER | EXPORT | “Yes” or “True” to include a header line in the output. Default is “Yes”. |
| LIMIT | EXPORT | Limits the number of lines exported. Default is “0” meaning no limit. |
| PRUNE\_TERMS | DELETE\_ CONCEPTS  DELETE\_TREES | “Yes” or “True” to delete orphan Terms when Concepts or Synonyms are deleted. Default is “No”. |
| RETAIN\_HEAD | DELETE\_TREES | “Yes” or “True” to retain the named, “head”, concepts when trees are deleted. Default is “No”. |
| SIZE | *collections* | Read-only size of last *collection*. |
| TIME\_FORMAT | EXPORT | Format of time values. |
| TYPEDEFS | EXPORT\_ CONCEPTS  EXPORT\_ SUBSET | “Yes” or “True” to include Type definitions in the export. Default is “False”. |
| UNIQUE | EXPORT | If “Yes” or “True”, only unique lines are exported. Default is “False”. |
|  |  |  |

**Special Codes**

|  |  |
| --- | --- |
| **Code** | **Replaced With** |
| @D or @d | The current date. Uses DATE\_FORMAT. |
| @T or @t | The current time. Uses TIME\_FORMAT. |