



TQL Quick Reference Guide V4.6

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

query := *statement* **query**

statement_block := *statement* | { *statement* ... }

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)

Create Context – create 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 – delete a context

DELETE *context*; (N)

Edit – edit data

CREATE_CONCEPTS *create-concept-list*; (R)

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

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

CREATE_TERMS *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 **mods** *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 **mods**; (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*; (U)

PRINT *expr-list*; (U)

Set Variables – set TQL variables

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

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*

/HEADER = **boolean-literal**

/LIMIT = *integer-literal*

/EXPORTFILE = *string-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	Use	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
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 Ontolog..
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
DEFINING_CONCEPT	EFS	The Ontolog 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	Description
MEMBER_OF	<i>context</i>	<i>obj</i> is a member of the context.
NOT_MEMBER_OF	<i>context</i>	<i>obj</i> is not a member of the context.
CHILD_OF	Concept name literal	<i>obj</i> is a Child (Subconcept) of the referenced Concept. *
NOT_CHILD_OF	Concept name literal	<i>attr</i> 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	<i>obj</i> is a descendant of the referenced Concept. *
NOT_DESCENDANT_OF	Concept name literal	<i>obj</i> is not a descendant of the referenced Concept. *

Operator String	Right	Description
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	<i>obj value</i> follows operand lexically.
PRECEDES	String literal	<i>obj value</i> precedes operand lexically.
EQUALS	String literal (wildcards)	<i>obj value</i> equals the literal.
NOT_EQUALS	String literal (wildcards)	<i>obj value</i> does not equal the literal.
EXISTS	None	At least one instance of the <i>obj</i> exists on the collection object. Same as EQUALS <i>"*"</i> .
NOT_EXISTS	None	<i>obj</i> does not exist on the collection object. Searches full <i>context</i> .
MATCHES	RegEx string literal	<i>obj value</i> matches the RegEx.
NOT_MATCHES	RegEx string literal	<i>obj value</i> does not match the RegEx.
IN_RANGE	Numeric literal <i>"."</i> numeric literal	Numeric interpretation of the <i>obj value</i> lies between the lower and upper limit (incl)
NOT_IN_RANGE	Numeric literal <i>"."</i> numeric literal	Numeric interpretation of the <i>obj value</i> does not lie between the lower and upper limit (incl)
=	Numeric literal	Numeric interpretation of the <i>obj value</i> equals the literal.
<>	Numeric literal	Numeric interpretation of the <i>obj value</i> does not equal the literal.
<	Numeric literal	Numeric interpretation of the <i>obj value</i> is less than the literal.
<=	Numeric literal	Numeric interpretation of the <i>obj value</i> is less than or equal to literal.
>	Numeric literal	Numeric interpretation of the <i>obj value</i> is greater than the literal.
>=	Numeric literal	Numeric interpretation of the <i>obj value</i> is greater than or equal to the literal.

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

TQL Functions – *function(attr)*

Function	Description
LENGTH	Returns the string length of the <i>attr</i> .
COUNT	Returns the number of instances of the <i>attr</i> 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	<i>collections</i>	Read-only size of last <i>collection</i> .
TIME_FORMAT	EXPORT	Format of time values.
TYPEDFS	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.