# Quick Reference

## by *M Gaffiero*

**gaffie@users.sourceforge.net**

Pequel ETL

**2.4**

## Section Types

### Expressions

A ***Pequel*** expression can contain a mix of Perl code, including regular expressions, *field-name*s, Pequel-macros, and Pequel-table lookups.

### Comments

Any text following and including the # symbol or `//` is considered as comment text. If the *cpp* pre-processor is available then comments are limited to C/C++ style comments with (`//` and `/* ... */`) — the # will be interpreted as a macro directive.

### Item Line Continuation

Each ***item*** within a section must appear on a single line. In order to break up an item statement (for better readability) us the line continuation character \.

### Pre-Processor

If your system provides the ***cpp*** pre-processor, your Pequel script may include any C/C++ style macros.

***options***

< *option-name*> [ **(**< *arg*> **)** ]

***description section***

< *free format text*>

***input section***

< *input-field-name*> [ **=>** < *calc-expression*> ]

calc-expression

A valid *Perl* statement which may contain *input-field-name*, Pequel macros, and Pequel-table lookup expressions.

***field preprocess***

< *input-field-name*> [ **=>** | **=~** < *calc-expression*> ]

***filter***

< *condition-expression*>

***reject***

< *condition-expression*>

***divert input record*** **(<** *filename* | *pequel-script name* | *db-connect-str(**)* | *socket(**)* **>)**

< *condition-expression*>

***copy input record*** **(<** *filename* | *pequel-script name* | *db-connect-str(**)* | *socket(**)* **>)**

< *condition-expression*>

***display message on input*(<** *message-expression*> **)**
< *condition-expression*>

***display message on input abort*(<** *message-expression*> **)**
< *condition-expression*>

***sort by***
< *input-field-name*> [ **numeric** | **string** ] [ **asc** | **des** ]

***group by***
< *input-field-name*> [ **numeric** | **decimal** | **string** ]

***dedup on***
< *input-field-name*> [ **numeric** | **decimal** | **string** ]

***output section***
< *pequel-type*> < *output-field-name*> < *input-field-name*>
| < *pequel-type*> < *output-field-name*> < *aggregation-expression*>
| < *pequel-type*> < *output-field-name*> **=** < *calc-expression*>

pequel-type
**numeric** | **decimal** | **string** | **date** [ **(**< *date-type*> **)** *]*

date-type
**YYYYMMDD** | **YYMMDD** | **DDMMYY** | **DDMMMYY** | **DDMMYYYY** | **DD/MM/YY** | **DD/MM/YYYY** | **MMDDYY** | **MMDDYYYY** | **MM/DD/YY** | **MM/DD/YYYY**

aggregation-expression
< *aggregate-type*> < *input-field-name*> [ **where** < *condition-expression*> ]
| **serial** < *start-num*> [ **where** < *condition-expression*> ]
| **count *** [ **where** < *condition-expression*> ]
| **flag *** [ **where** < *condition-expression*> ]

aggregate-type
**sum** | **maximum** | **max** | **minimum** | **min** | **avg** | **mean** | **first** | **last**
| **distinct** | **sum_distinct** | **avg_distinct** | **count_distinct**
| **median** | **variance** | **stddev** | **range** | **mode**
| **values_all** [ **(**< *delim*> **)** ] | **values_uniq** [ **(**< *delim*> **)** ]

condition-expression
*An expression that evaluates to true or false (non-zero, zero respectively).*

input-field-name
< *field-name*>

output-field-name
< *field-name*> | < *transparent-field-name*>

transparent-field-name
_< *field-name*>

field-name

**[_A-Za-z]+[0-9_A-Za-z]***

Case-sensitive and must begin with an alpha or **'_'** character, followed by zero or more alpha, numeric and **'_"**s characters.

***sort output***

< *output-field-name*> [ **numeric** | **string** ] [ **asc** | **des** ]

***field postprocess***

< *output-field-name*> [ **=>** | **=~** < *calc-expression*> ]

***having***

< *condition-expression*>

***divert output record*** **(<** *filename* | *pequel-script-name* | *db-connect-str(**)* | *socket(**)* **>)**

< *condition-expression*>

***copy output record*** **(<** *filename* | *pequel-script-name* | *db-connect-str(**)* | *socket(**)* **>)**

< *condition-expression*>

***display message on output*****(<** *message-expression*> **)**

< *condition-expression*>

***display message on output abort*****(<** *message-expression*> **)**

< *condition-expression*>

***use package***

< *Perl package name*>

***init table***

< *table-name*> < *key*> < *value*> [, < *value*> ... ]

***load table***

< *table-name*> [ < *filename*> [ < *key-col*> [ < *val-col*> ] ] ] [, ...]

***load table pequel***

< *table-name*> < *pequel-script-name*> [ < *keyfield-name*> [ < *keyfield-type*> ] ]

keyfield-name

*pequel-script-name.output-field-name*

keyfield-type

**STRING** | **NUMERIC**

***load table sqlite***
< *table-name*> < *filename*> < *key-col*> < *keyfield-type*> [ < *table-field-name*> **=** < *field-col*> [...] ]

keyfield-type
**INTEGER** | **VARCHAR**

***load table sqlite merge***
< *table-name*> < *filename*> < *key-col*> < *keyfield-type*> [ < *table-field-name*> **=** < *field-col*> [...] ]

***load table oracle***
< *table-name*> < *filename*> < *connect-str*> < *key-col*> < *key-type*> \
[ < *field-name*> **=** < *field-col*> [...] ]

key-type
***Oracle Data Type***
connect-str
< *user*> **/**< *password*> @< *db-name*>

***load table oracle merge***
< *table-name*> < *filename*> < *connect-str*> < *key-col*> < *key-type*> \
[ < *field-name*> **=** < *field-col*> [...] ]

## Aggregate Types

***count***
Output the count of records in the group.

***count_distinct***
Output the count of unique values for *field-name* in the group.

***distinct***
Output the count of unique values for *field-name* in the group. Same as ***count_distinct***.

***sum***
Accumulate the total for all values in the group.

***sum_distinct***
Accumulate the total for unique values for *field-name* in the group.

***maximum***
Output the maximum value in the group.

***max***
Output the maximum value in the group. Same as ***maximum***.

***minimum***
Output the minimum value in the group.

***min***

Output the minimum value in the group. Same as ***minimum***.

***avg***
Output the average value in the group.

***avg_distinct***
Output the average value for unique values for *field-name* in the group.

***mean***
Output the average value in the group. Same as ***avg***.

***first***
Output the first value in the group.

***last***
Output the last value in the group.

***median***
Output the middle vale in the group, or, in the case of an even number of values, output the mean of the two middle values in the group.

***variance***
Output ( *sum squares* / *count* ) - ( *mean* ** 2); *sum_squares* is each value in the distribution squared (** 2); *count* is the number of values in the distribution; *mean* is discussed above.

***stddev***
Output the square-root of *variance*.

***range***
Output the maximum value minus the minimum value in a distribution.

***mode***
Output the most frequently occuring score or scores (space delimited) in a distribution.

***values_all*** [ **(**< *delim*> **)** ]
Output the list of all values in the group.

***values_uniq*** [ **(**< *delim*> **)** ]
Output the list of unique values in the group.

## Macros

***String Macros***

**&length (** < *field-name*> **)**
**&substr (** < *field-name*> , < *offset*> , < *len*> **)**
**&index (** < *field-name*> , < *substr*> , < *offset*> **)**
**&rindex (** < *field-name*> , < *substr*> , < *offset*> **)**
**&lc (** < *field-name*> **)**
**&uc (** < *field-name*> **)**
**&lc_first (** < *field-name*> **)**
**&uc_first (** < *field-name*> **)**
**&commify (** < *field-name*> **)**
**&trim (** < *field-name*> [**,** < *character-list*> ] **)**
**&spaceout (** < *field-name*> **)**

**&trim_trailing (** < *field-name*> [**,** < *character-list*> ] **)**
**&trim_leading (** < *field-name*> [**,** < *character-list*> ] **)**
**&clip_str (** < *field-name*> **)**
**&left_clip_str (** < *field-name*> **)**
**&right_clip_str (** < *field-name*> **)**
**&left_pad_str (** < *field-name*> , <pad-char>, <len> **)**
**&right_pad_str (** < *field-name*> , <pad-char>, <len> **)**
**&remove_spaces (** < *field-name*> **)**
**&to_number (** < *field-name*> **)**
**&extract_numeric (** < *field-name*> **)**
**&remove_non_numeric (** < *field-name*> **)**
**&remove_numeric (** < *field-name*> **)**
**&remove_special (** < *field-name*> **)**
**&translate (** < *field-name*> , <from-list>, <to-list> [**,** <modifier> ] **)**
**&initcap (** < *field-name*> **)**
**&extract_init (** < *field-name*> **)**

***Arithmentic Macros***

**&ord (** < *field-name*> **)**
**&sqrt (** < *field-name*> **)**
**&rand (** < *field-name*> **)**
**&sin (** < *field-name*> **)**
**&exp (** < *field-name*> **)**
**&cos (** < *field-name*> **)**
**&log (** < *field-name*> **)**
**&chr (** < *field-name*> **)**
**&abs (** < *field-name*> **)**
**&int (** < *field-name*> **)**
**&atan2 (** < *field-name*> **)**
**&sign (** < *field-name*> **)**
**&trunc (** < *field-name*> , < *dec*> **)**
**&lshift (** < *field-name>, bits* **)** (**)
**&rshift (** < *field-name>, bits* **)** (**)

***Date Macros***

**&date (** < *field-name*> [**,** < *date-type*> ] **)**
**&months_since (** < *field-name*> [**,** < *date-type*> ] **)**
**&months_between (** < *field-name*> , < *field-name*> <n> **)**
**&date_last_day (** < *field-name*> **)**
**&last_day (** < *field-name*> **)**
**&date_next_day (** < *field-name*> **)**
**&day_number (** < *field-name*> **)**
**&y (** < *field-name*> [ **,** < *date-type*> ] **)**
**&d (** < *field-name*> [ **,** < *date-type*> ] **)**
**&m (** < *field-name*> [ **,** < *date-type*> ] **)**
**&today ( )**

***Array Macros***

**&to_array (** < *field-name*> **)**
**&arr_size (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_sort (** < *field-name*> **)**
**&arr_reverse (** < *field-name*> **)**
**&arr_values_uniq (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_sum (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_sum_distinct (** < *field-name*> [**,** < *field-name*> , ...] **)** (**)

**&arr_avg (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_avg_distinct (** < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_mean (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_first (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_last (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_min (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_max (** < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_median (** < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_variance (** < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_stddev (** < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_range (** < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_max (** < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_lookup (** < *value*> , < *field-name*> [**,** < *field-name*> , ...] **)**
**&arr_pack (** < *pack-format*> , < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_unpack (** < *pack-format*> , < *field-name*> [**,** < *field-name*> , ...] **)** (**)
**&arr_set_and (** < *field-name*>*, field-name* **)** (**)
**&arr_set_or (** < *field-name*>*, field-name* **)** (**)
**&arr_set_xor (** < *field-name*>*, field-name* **)** (**)

***Miscellaneous Macros***

**&banding (** < *field-name*> , <band-divisor> **)**
**&env (** < *env-var-name*> **)**
**&option (** < *pequel-option-name*> **)**
**&select (** < *field-name*> , < *value*> [ [, < *field-name*> , < *value*> ] [ ,...] ], < *default-value*> **)**
**&match_any (** < *field-name*> , <match list> **)**
**&match (** < *field-name*> , <match list> **)**
**&map (** < *table-name*> , < *field-name*> [, ...] **)**
**&input_record_count()**
**&soundex (** < *field-name*> **)**
**&pack (** < *pack-format*> , < *field-name*> [, ...] **)**
**&unpack (** < *pack-format*> , < *field-name*> [, ...] **)**
**&sprintf (** < *print-format*> , < *field-name*> [, ...] **)**

(**) Forthcoming.

## Pequel Data Types

***string***
***numeric***
***decimal***
***date***
***array***

## Option Types

***Basic Options***

**verbose**
**noverbose**
**input_delimiter_extra**
**input_delimiter**
**output_delimiter**
**input_file**
**output_file**
**script_name**
**discard_header**

**header**
**noheader**
**addpipe**
**noaddpipe**
**optimize**
**nooptimize**
**nulls**
**nonulls**
**reject_file**
**default_datetype**
**default_list_delimiter**
**hash**
**transfer**
**suppress_output**
**num_threads**
**sort_tmp_dir**
**logfilename**
**logging**
**prefix**
**lock_output**
**output_file_append**
**sort_cmd**
**sort_args**
**cpp_cmd**
**cpp_args**
**gzcat_cmd**
**gzcat_args**
**cat_cmd**
**cat_args**
**pack_output**
**output_pack_fmt**
**unpack_input**
**input_pack_fmt**
**input_record_limit**
**rmctrlm**
**show_synonyms**
**exec_min_lines**

***General Table Options***

**display_table_stats**
**reload_tables**
**load_tables_only**
**table_drop_unused_fields**
**table_dir**

***Oracle Table Options***

**oracle_prefetch_count**
**oracle_home**
**oracle_sqlldr_rows**
**oracle_use_merge_fetch_macro**

***Sqlite Table Options***

**sqlite_dir**
**sqlite_merge_optimize**
**sqlite_merge_optimize_count**

***Inline Options***

**use_inline**
**inline_cc**
**inline_libs**
**inline_inc**
**inline_ccflags**
**inline_optimize**
**inline_lddlflags**
**inline_make**
**inline_clean_after_build**
**inline_clean_build_area**
**inline_build_noisy**
**inline_build_timers**
**inline_force_build**
**inline_print_info**
**inline_directory**
**inline_cache_recs**
**use_av_store_macro**
**inline_merge_optimize**
**inline_merge_optimize_count**

***Document Generation Options***

**doc_title**
**doc_version**
**doc_email**

***Developer Options***

**dumpcode**
**debug_show_caller**
**debug**
**debug_generate**
**debug_parser**
**diagnostics**
**tinfo**
**minfo**
**pequelsrclist**
**pequelprogref**

***Command Line Options***

**version**
**usage**
**viewcode**
**viewraw**
**syntax_check**
**list**
**option**
**pequeldoc**
**detail**

## Command Line Usage

***pequel scriptfile.pql*** **<** *file_in* **>** *file_out*
Execute ***pequel*** with *scriptfile.pql* script to process *file_in* data file, resulting in *file_out*.

***pequel -c scriptfile.pql***

Check the syntax of the pequel script *scriptfile.pql*.

***pequel -viewcode scriptfile.pql***
Generate and display the code for the pequel script *scriptfile.pql*.

***pequel -dumpcode scriptfile.pql***
Generate the pequel code for the script *scriptfile.pql* and save generated code in the file *scriptname.pql.2.code*.

***pequel -v***
Display version informatio for ***Pequel***.

***pequel -usage***
Display Pequel usage command summary.

***pequel -pequeldoc pdf -detail scriptfile.pql***
Generate the Script Reference document in pdf format for the Pequel script *scriptfile.pql*. The document will include a section showing the generated code (***-detail***).

***—prefix, —prefix_path***
Prefix for filenames directory path.

***—noverbose, —silent***
Do not display progress counter and messages.

## COPYRIGHT