

Release notes

0.8.6-X

Changes

- elementals:
 - `SQLCDatabase` supports the property `schema` (see section 7 on page 138)
 - `SQLCDatabase` supports foreign key and references (see section 7 on page 138)
 - new property `url.escape` for `FZZCWebAPIGetter` (see section 7 on page 125)
 - `MRKCCCCollector` supports `timeout` modes (see section 8 on page 148)
- primitives:
 - `sym.cat` behavior with *list terms* (see section 5.16 on page 93)
 - `sym.tokenize` will convert numbers into *numbers*.
 - `is.bound` accepts more than one *term*.

Additions

- elementals:
 - *class* `FZZCWebURLGetter` (see section 7 on page 128)
- terms:
 - `neq.nor.in` *variable constraint*
 - `frm.label` *lambda*
 - `now, lst.empty` as *primitives* for `FZZCFUNRunner`.
- primitives:
 - `lst.bulk` (see section 5.8 on page 68)
- volatiles ((see section 3.10 on page 18):
 - `seq`
 - `sym.16`
 - `uid`

Bug Fixes

- Incorrect handling of double quote and single quite characters in `SQLCDatabase`
- Incorrect error code returned when an error occurred in generating SQL statement in `SQLCDatabase`
- Padding wasn't zero-ed when flattening terms leading to possible non-match of terms stored as blob in `SQLCDatabase`
- *Boolean* properties for `Elemental` didn't accept symbols (such as `yes`, `false` ...)
- Crash when unifying a `functor` which label is a *variable*.
- primitive `fun.make` wasn't unifying correctly when its third *term* was not a final *term*.
- Issue with early fail reply when using cascade mode in `MRKCBFSolver`.
- Issue with `fzz.collect` not replying under some condition.
- `sym.tokenize` and `str.tokenize` did not accepts *split-list* as 3rd *term*.
- Issue with inference stopping seemingly randomly.

0.8.0-X

Changes

- elementals:
 - MRKCCSVStore can store *statements* (see section 6 on page 113)
 - FZZCFUNRunner support for primitive **frm.labels** (see section 6 on page 116)
 - FZZCFUNRunner support for primitive **cls** (see section 6 on page 116)
 - FZZCFUNRunner support for primitive **unify** (see section 6 on page 116)
 - FZZCFUNRunner support for primitive **inquire** (see section 6 on page 116)
 - FZZCFUNRunner support for primitive **lst.append** (see section 6 on page 116)
 - FZZCFUNRunner support for primitive **lst.prepend** (see section 6 on page 116)
 - FZZCWebPoster: support for **multipart/form-data** content (see section 7 on page 126)
 - MRKCLettered: support for property **loose** (see section 6 on page 122)
- primitives:
 - **console.gets** accepts two terms, the first one being a prompt to be printed
- samples:
- terms:
 - **if** variable constraint
 - **is.final** variable constraint
 - *frame* can be used as variable constraint
- console:
 - Solution can file can be made to load source files sequentially (see section 4.3 on page 26)
 - A custom **ttl** value can be specified for a *predicate*

Additions

- samples:
 - **cnet**
 - **sql**
 - **funx4.fizz**
 - **sentim.fizz**
 - **pyt**
- elementals:
 - PYTCElemental (see section 7 on page 145)
 - PYTCModule (see section 7 on page 144)
 - MRKCMingler (see section 6 on page 112)
 - MRKCCatcher (see section 6 on page 112)
 - SQLCDatabase (see section 7 on page 138)
 - FZZCWebAPIPoster (see section 7 on page 126)
 - MRKCAggregator (see section 6 on page 111)
 - MRKCProxy (see section 6 on page 114)
- terms:
 - **lambda** (see section 3.12 on page 19)
 - **split frame** (see section 3.4 on page 13)

- `eq.or.in` *variable constraint*
- constants:
 - `self.path` (see section 3.9 on page 18)
- primitives:
 - `fzz.stats` (see section 5.11 on page 84)
 - `fzz.parse` (see section 5.11 on page 84)
 - `fzz.exists` (see section 5.11 on page 83)
 - `sleep` (see section 5.2 on page 54)
 - `console.quit` (see section 5.2 on page 48)
 - `rnd.list` (see section 5.13 on page 87)
 - `lst.permu` (see section 5.8 on page 74)
 - `frm.there` (see section 5.6 on page 66)
 - `frm.same` (see section 5.6 on page 66)
 - `var.make` (see section 5.11 on page 86)
 - `sym.end` (see section 5.16 on page 94)
 - `sym.cut` (see section 5.16 on page 94)
 - `bin.length` (see section 5.4 on page 59)
 - `bin.load` (see section 5.4 on page 59)
 - `bin.save` (see section 5.4 on page 59)
 - `sym.tolower` (see section 5.16 on page 95)
 - `sym.toupper` (see section 5.16 on page 96)
 - `var.tofu` (see section 5.11 on page 86)
 - `var.defu` (see section 5.11 on page 85)
 - `hash` (see section 5.2 on page 50)
 - `is.bound` (see section 5.18 on page 102)
 - `sym.tokenize` (see section 5.16 on page 95)
 - `lst.flat` (see section 5.8 on page 70)
 - `str.dist` (see section 5.17 on page 96)
 - `sym.stem` (see section 5.16 on page 95)
 - `lst.snap` (see section 5.8 on page 75)
 - `str.end` (see section 5.17 on page 97)
 - `lst.unique` (see section 5.8 on page 77)
 - `of.type` (see section 5.18 on page 106)

Bug Fixes

- issue with sharing of properties across multiple instances of the same *elemental*
- issue with trigger statement getting missed when frequently repeated
- issue with *primitive* `frm.labels` not unifying correctly with a *list* as second *term*
- issue cloned *elemental* not subscribing to the right label
- crash when re-loading a module
- predicate's limit not considered when collecting variables
- inquiry predicate prefix was not always working
- offloading mode for `MRKCSBFStore` was not working
- `div(42,-5)` was returning `NaN`

- variable's constraint was lost when unpinning (e.g. with primitive `cpy`)
- non-working negation of some primitives based predicate
- inconsistent hash code generation for Frame term (when the order of the slot is different)
- `fzz.collect` reusing query's context was causing issue within the *elemental* originator (early failure)
- issue with primitive `frm.labels` and `frm.pairs` not accepting its 2nd term to be a list
- issue with primitive `frm.pairs` not accepting a non-final *list* as second term
- issue with `MRKCCSVStore` always missing the last line

0.7.0-X

Breaking Changes

- predicates:
 - range check or unification to a *variable* of a *predicate*'s *truth value* requires an = character
- terms:
 - *regexp* is no longer an *atom*
 - *escaper* behavior have changed

Changes

- elementals:
 - FZZCFFBNetwork:
 - * new **datafile** property to save network to a binary file
 - * support for *list* and *data* terms
 - MRKCBFSolver:
 - * new property **reply.on** and **cascade**, **cascade.tmo**
 - FZZCCollector
 - * speed-up
 - * modified behavior of property **tmo** to be the time-out from the last received replies
 - * added property **ttl** to specify the time-to-live value for the query
 - MRKCCSVStore
 - * property **arity** to specify the arity of the statements (if the number of columns is greater than the arity, the extra will be grouped into a list as the last term)
 - EV3CSENLEGOgyros
 - * property **inverted**
 - EV3CBEVSonar
 - * support to *peek* at the current reading
- primitives:
 - **lst.sort** now accept as 3rd term a list of indexes to be used for sorting lists (+1 index will be used when the lists' terms are equal)
 - revisited the way the **sim** primitive compute the similarity between two numbers
- samples:
 - updated **irl2asm.fizz**
- terms:
 - a *frame*'s label can be any atom (and not just a *symbol*)
- console:
 - **/spy** output contains the timestamp
 - use **verbose** property to silence output
 - ignore *variables* with name starting with an upper case

Additions

- samples:
 - iris2, iris3
 - nlu
 - movies
 - ml
 - db
 - fuzzy
 - fun, eval, exec, sexp, lstrnd
 - funx, funx2, funx3
 - tasc (based on Hector Levesque’s book ”Thinking as Computation” (ISBN: 978-0-262-01699-5))
- console:
 - /trace (see section 4.4 on page 38)
- elementals:
 - MRKCStopper (see section 6 on page 115)
 - FZZCFUNRunner (see section 6 on page 116)
 - EV3CSRVMMapping (see section 7 on page 138)
- predicates:
 - ? prefix for *predicate* (see section 2.3 on page 4)
- terms:
 - data (see section 3.3 on page 12)
 - quirk (see section 3.11 on page 19)
- volatiles ((see section 3.10 on page 18):
 - sym.8
 - sym.6
 - now.ms
- constraints:
 - fun.label
- constants:
 - pi
- prototypes
 - support for alternate fuzzy and-or evaluation (see section 2.4 on page 7)
- primitives:
 - rnd.sint (see section 5.13 on page 89)
 - qrk.head (see section 5.12 on page 86)
 - qrk.tail (see section 5.12 on page 87)
 - qrk.make (see section 5.12 on page 87)
 - is.quirk (see section 5.18 on page 105)
 - lst.any (see section 5.8 on page 68)
 - lst.all (see section 5.8 on page 67)
 - qat.euler (see section 5.19 on page 107)

- `qat.apply` (see section 5.19 on page 107)
- `vec.lenght` (see section 5.19 on page 109)
- `vec.dist` (see section 5.19 on page 109)
- `vec.angle` (see section 5.19 on page 108)
- `vec.angle.signed` (see section 5.19 on page 109)
- `vec.norm` (see section 5.19 on page 110)
- `mat.make` (see section 5.19 on page 107)
- `mat.apply` (see section 5.19 on page 106)
- `min` (see section 5.1 on page 41)
- `max` (see section 5.1 on page 41)
- `cache` (see section 5.2 on page 45)
- `rng.not` (see section 5.14 on page 90)
- `rng.real` (see section 5.14 on page 92)
- `frm.swap` (see section 5.6 on page 65)
- `pull` (see section 5.2 on page 52)
- `push` (see section 5.2 on page 53)
- `drop` (see section 5.2 on page 49)
- `lst.split` (see section 5.8 on page 76)
- `lst.knit` (see section 5.8 on page 72)
- `mao.sin` (see section 5.10 on page 83)
- `mao.cos` (see section 5.10 on page 80)
- `mao.atan2` (see section 5.10 on page 79)
- `mao.d2r` (see section 5.10 on page 80)
- `spawn` (see section 5.2 on page 55)
- `cease` (see section 5.2 on page 46)
- `shoot` (see section 5.2 on page 55)
- `is.data` (see section 5.18 on page 103)
- `prune` (see section 5.2 on page 52)
- `daa.make` (see section 5.5 on page 61)
- `daa.length` (see section 5.5 on page 61)
- `daa.member` (see section 5.5 on page 61)
- `daa.format` (see section 5.5 on page 60)
- `daa.item` (see section 5.5 on page 60)
- `fzz.labels` (see section 5.11 on page 83)
- `is.primitive` (see section 5.18 on page 104)
- `exec` (see section 5.2 on page 50)
- `var.capture` (see section 5.11 on page 85)
- `var.release` (see section 5.11 on page 86)
- `var.collect` (see section 5.11 on page 85)
- `cpy` (see section 5.2 on page 48)
- `uny` (see section 5.2 on page 56)
- `nab` (see section 5.2 on page 51)
- `lst.combi` (see section 5.8 on page 68)
- `hush.if` (see section 5.2 on page 51)

- `hush.if.not` (see section 5.2 on page 51)
- `cut.if` (see section 5.2 on page 48)
- `cut.if.not` (see section 5.2 on page 48)
- `lst.min` (see section 5.8 on page 73)
- `lst.max` (see section 5.8 on page 73)
- `lst.avg` (see section 5.8 on page 68)
- `vec.add` (see section 5.19 on page 108)
- `vec.sub` (see section 5.19 on page 110)
- `vec.mul` (see section 5.19 on page 110)
- `vec.div` (see section 5.19 on page 109)
- `rng.norm` (see section 5.14 on page 90)
- `daa.find` (see section 5.5 on page 60)
- `daa.min` (see section 5.5 on page 62)
- `daa.max` (see section 5.5 on page 61)
- `daa.avg` (see section 5.5 on page 60)
- `qat.add` (see section 5.19 on page 107)
- `qat.sub` (see section 5.19 on page 108)
- `qat.length` (see section 5.19 on page 108)
- `lst.it` (see section 5.8 on page 71)
- `any` (see section 5.2 on page 43)

Bug Fixes

- issue with *variables* in the `define primitive`
- issue with `fun.terms` not unifying to a `split-list` (as 2nd term)
- issue with property `clone` not finding the *elemental* to clone from
- issue with `rnd.sint` and `rnd.uint` crashing with "Floating point exception (core dumped)" when the range was given using the same value
- issue with `then primitive` confusing minutes and seconds
- issue with `mao.abs primitive` returning 0 when the first term was a negative floating point value
- issue with `frm.make primitive` failing when an empty *list* was used as one of the *term*
- issue in `FZZCCLUGateway` leading to long delay in further transmission after a large one

0.6.0-X

Breaking Changes

- `MRKCSBFStore` *elemental class* is impacted by a bug in storing `GUID term`.
- Many of the non-core *elementals* have been moved to individual modules (see 7 on page 123).

Changes

- primitives:
 - `str.tokenize` support optional fourth *term* which is a *list* of flags.
 - `peek` accepts a third *term* which is a value to be unified to the 2nd *term* if the label doesn't exist in the properties.
- config:
 - `spinning` meaning changed (see 4.2 on page 23)
- elementals:
 - new `ttl` property to set the time-to-live of any query sent by the *elemental* (instead of using the system default)
 - `MRKEvaluator`: when no "tmo" is specified, the *substrate* or *elemental* TTL value is used
- predicates:
 - `~` can be used with any label other than `self` (see section 2.3 on page 4).

Additions

- samples:
 - bigrams
 - clu
 - ecalculus
 - robin
- modules:
 - CLU (see section 7 on page 128)
 - EV3 (see section 7 on page 130)
- elementals:
 - constants `$self` and `$guid`
- primitives:
 - `rnd.sint` (see section 5.13 on page 89)
- predicates:
 - `*` prefix for *predicate* (see section 2.3 on page 4)

Bug Fixes

- `guid term` wasn't flattened and thus wouldn't get saved in *SBFStore*.
- trigger based *prototypes* where not respecting the 'cut' directives.
- Unfrequent crashes when pasting into the console (outside of the input mode)

0.5.0-X

Breaking Changes

- Pre 0.5 *kindled runtime* (`.bizz`) files can't be loaded
- `MRKCSBFStore` *elemental class* is impacted by hashing changes to *numbers*

Changes

- support for modules (shared library) that can be loaded at runtime (SDK to come in a future release)
- console:
 - previous query is no longer cancelled when a new one is issued
 - query specified via the *command line* gets executed once all the files specified in the *command line* have been loaded
- new *elemental* properties:
 - `chatty` (see section 2.5 on page 8)
 - `noisy` (see section 2.5 on page 8)
 - `clone` (see section 2.5 on page 8)
- any *elemental* property can be read using the *constant* syntax
- new property for *elemental* of class `MRKCBSolver`:
 - `memoize` (see. fibonacci sample)
- new property for *elemental* of class `MRKCLettered`:
 - `recall.frq`, `recall.ttl`, `recall.add`, `recall.mul`, `recall.thd` (see section 6 on page 122)
- *primitives* `gt`, `gte`, `lt` and `lte` now also works with *strings* and *symbols*

Additions

- *solution* files (see section 4.3 on page 26)
- new console *command*: `/use` (see section 4.4 on page 39)
- new *syntax*:
 - `~` prefix for *predicate* (see section 2.3 on page 4)
 - `self` *predicate* (see section 2.3 on page 4)
- new *terms*:
 - `regexp` (see section ?? on page ??)
- new *primitives*:
 - `frm.erase` (see section 5.6 on page 62)
 - `lst.mix` (see section 5.8 on page 74)
 - `lst.sort` (see section 5.8 on page 76)
 - `lst.sub` (see section 5.8 on page 77)
 - `rex.make` (see section 5.15 on page 93)
 - `rex.match` (see section 5.15 on page 93)
 - `rng.rand` (see section 5.14 on page 92)
- new *constraints*:
 - `eq`
 - `is.regexp`
 - `is.bound`
- new *volatiles*: `sym.3`, `sym.4` and `sym.10` (see section 3.10 on page 18)

Bug Fixes

- `lst.item`, `lst.head`, `lst.tail` would not unify their last *term* with a *list*.
- `MRKCTicker` wouldn't accept a property as a *constant*.
- `peek(guid, :x)` was unifying `:x` with a *string* instead of a *guid*.
- `frm.fetch(a = [1,2], a, [-, :v])` wasn't returning 2.
- re-saving an *elemental* into a *fizz* file was failing.
- *terms* in a *range* couldn't be a *constant*.
- the hashcode of real *number* was the same regardless of the sign.
- `lst.tail` was not unifying its second *term* with `[]` when the first *term* was an empty *list*.

0.4.0-X

Additions

- new *elementals*:
 - `MRKCSBFStore` (see section 6 on page 115)
 - `MRKCCSVStore` (see section 6 on page 113)
 - `FZZCLGRProcessor` (see section 7 on page 123)
- new *terms*:
 - `guid` (see section 3.1.5 on page 11)
- new *primitives*:
 - `str.trim.head` (see section 5.17 on page 101)
 - `str.trim.tail` (see section 5.17 on page 101)
 - `str.tail` (see section 5.17 on page 99)
 - `str.head` (see section 5.17 on page 98)
 - `lst.incl` (see section 5.8 on page 71)
 - `lst.excl` (see section 5.8 on page 70)
 - `lst.join` (see section 5.8 on page 72)
 - `lst.init` (see section 5.8 on page 71)
 - `sym.cmp` (see section 5.16 on page 94)
 - `sim` (see section 5.1 on page 42)
 - `is.even` (see section 5.18 on page 103)
 - `is.odd` (see section 5.18 on page 104)
 - `gid.make` (see section 5.11 on page 84)
- new *constraints*:
 - `lst.incl`
 - `lst.excl`
 - `is.guid`
 - `is.even`
 - `is.odd`

Changes

- modified *primitives*:
 - `lst.remove` was changed to succeed when the item to remove isn't found in the *list*.
 - `str.trim` was changed to accept an optional third *term*: the *string* to be trimmed from the 1st *term*.
 - `lst.length` was changed to accept a third *term* which is the *term* to be assigned to each of the *list's* *terms* when the first *term* of the primitive is an *unbound variable*.
 - `fzz.lst` was changed to returns a *list* of *guid terms* instead of a *list* of *strings*.
 - `guid.str` and `guid.sym` were renamed `gid.str` and `gid.sym`.
- modified *console commands*:
 - `/peek` now accepts a *guid*.
 - `/poke` now accepts a *guid*.
 - `/tells` now accepts a *guid* as well as a *symbol*.
 - `/knows` now accepts a *guid*.
- modified *terms*:
 - *binary* syntax has changed to single quote *functor*.
 - *symbol* can now include `+` or `*` as long as they are not on the first character.

Bug Fixes

- *constraint* `is.string` was testing for a variable to be bound to a *symbol*
- *primitive* `str.swap` in some condition was repeating part of the tail of the *string* where the replacement was occurring
- *primitive* `add` was returning 0 when used with an unsigned number as the first term and a negative number as the second term (e.g. `add(23u,-18,:v)`)
- *string terms* with control characters were not rendered properly when they are embedded in other terms

0.3.0-X

Additions

- *live code reload* functionality
- new *constant* `$cores`
- new *primitives*:
 - `aeq` (see section 5.3 on page 57)
 - `bundle` (see section 5.2 on page 44)
 - `div.int` (see section 5.1 on page 40)
 - `fzz.lst` (see section 5.11 on page 84)
 - `lst.remove` (see section 5.8 on page 74)
 - `mao.sign` (see section 5.10 on page 82)
 - `str.find` (see section 5.17 on page 97)
 - `str.flip` (see section 5.17 on page 97)
 - `str.trim` (see section 5.17 on page 101)
 - `str.rest` (see section 5.17 on page 98)
 - `str.swap` (see section 5.17 on page 99)
 - `sym.cat` (see section 5.16 on page 93)
- new *console commands*:
 - `/reload` (see section 4.4 on page 36)
 - `/import.txt` (see section 4.4 on page 33)
- new *class* `FZZCWebAPIGetter` (see section 7 on page 125)

Changes

- increased the maximum number of threads that can be used by the console
- added support for `str.find` as a *variable's constraint*
- *primitive* `frm.fetch` allows for a fourth *term* to specify a default value to use if the label isn't found
- when the first *term* of the `/peek` and `/poke` *console commands* is a *symbol*, all *elemental* of that label will be targetted
- the `fzz.eval` service now accept a *list* as second *term* to describe the *functor* to be evaluated
- changed *class* `FZZCTicker` to support the property `tick.on.attach`
- changed *class* `MRKCBFSolver` to support the property `replies.are.triggers`
- changed *class* `MRKCLettered` to support the property `nearest.only`

Bug Fixes

- minor performance tweaks when parsing *list* in *fizz* source files
- *primitive* `str.sub` was not properly handling negative offset
- on occasion queries/replies where not being sent/received
- JSON support wasn't handling 'null' value (causing crash)
- chunked transfer encoding wasn't supported by the builtin web client

0.2.0-X

Additions

- added console commands `/import.json` and `/export.json` to import and export JSON files (see section 4.4 on page 32 and 4.4 on page 29)
- added *primitive* `change` (see section 5.2 on page 47)
- added *primitive* `console.exec` (see section 5.2 on page 47)
- added *primitive* `then` (see section 5.2 on page 56)
- added *primitive* `tme.str` (see section 5.2 on page 56)
- added *primitive* `str.cmp` (see section 5.17 on page 96)
- added *elemental* class `FZZCWebAPIPuller` for fetching JSON data from web services (see section 7 on page 127)

Changes

- console commands `/import` and `/export` were renamed `/import.csv` and `/export.csv`
- the *elemental* class `FZZCTicker` now also supports time interval expressed in seconds (see section 6 on page 121)

Bug Fixes

- published statements could stop from being received by *elementals* referencing them as trigger
- *primitive* `str.tosym` was failing when the first *term* was already a *symbol*

0.1.4-X

Changes

Initial Release

Bug Fixes

Initial Release

Known issues

- Poor performance with *inferences* that involves *combinatorial exploration*
- Parser's error handling is too terse
- An empty comment line will cause a parsing error in a `fizz` file