<https://code.kx.com/q/releases/ChangesIn3.6/>

Changes in 3.6

Below is a summary of changes from V3.5. Commercially licensed users may obtain the detailed change list / release notes from [http://downloads.kx.com](http://downloads.kx.com/)

Production release date

2018.05.16

Update 2019.11.13

[.z.ph](https://code.kx.com/q/ref/dotz/#zph-http-get) now calls [.h.val](https://code.kx.com/q/ref/doth/#hval-value) instead of [value](https://code.kx.com/q/ref/value/), to allow users to interpose their own evaluation code. .h.val defaults to value.

Deferred response

More efficient gateways: a server process can now use [-30!x](https://code.kx.com/q/basics/internal/#-30x-deferred-response) to defer responding to a sync query until, for example, worker process have completed their tasks.

64-bit enumerations

Enums, and linked columns, now use 64-bit indexes:

q)(type a:`sym?`a`b;type `zym!4000000000)

20 -20h

* 64-bit enums are type 20 regardless of their domain.
* There is no practical limit to the number of 64-bit enum domains.
* 64-bit enums save to a new file format which is not readable by previous versions.
* 32-bit enums files from previous versions are still readable, use type space 21 thru 76 and all ops cast them to 64-bit.

Anymap

A new mappable type, 'mapped list' (77h) improves on existing mapped nested types (77h+t) by removing the uniformity restriction. e.g.

q)a:get`:a set (1 2 3;"cde"); b:get`:b set ("abc";"def")

q)77 77h~(type a;type b)

Mapped lists' elements can be of any type, including lists, dictionaries, tables. e.g.

q)a:get`:a set ((1 2;3 4);`time`price`vol!(.z.p;1.;100i);([]a:1 2;b:("ab";"cd")))

q)77 0h~(type a;type first a)

A new write primitive alternative to [set](https://code.kx.com/q/ref/get/#set), `:a 1: x, allows mapped lists to nest within other mapped lists. For files written with 1:, vectors within all structures remain mapped, no matter the depth, and can be used without being copied to the heap. e.g.

q)a:get`:a 1: ((1 2;3 4);([]time:1000?.z.p;price:1000?100.);([]time:1000?.z.p;price:1000?200))

q)77 77h~(type a;type first a)

q).Q.w[]`used`mmap / 336736 40432

q)p:exec price from a[1]

q).Q.w[]`used`mmap /336736 40432

* Symbol vectors/atoms are automatically enumerated against file## and deenumerated (and therefore always copied) on access. e.g.

q)`:file set((`a`b;`b`c);0 1) / symbols cause a 3rd file to be created, file##, which contains the enumeration domain

* The underlying storage (file#) stays mapped as long as there exists a reference to any mapped object within. Hence, care should be taken when working with compressed data, as anything ever decompressed in a file would stay in memory until the last reference is gone.

GUID hashing

Hash now considers all bits of the guid. Guids with u, p or g attribute use a new file format, unreadable by previous versions.

File compression

Added lz4hc as file-compression algorithm #4. e.g.

q).z.zd:17 4 16;`:z set z:100000?200;z~get`:z

**lz4 compression**

Certain [releases](https://github.com/lz4/lz4/releases) of lz4 do not function correctly within kdb+.

Notably, lz4-1.7.5 does not compress, and lz4-1.8.0 appears to hang the process.

Kdb+ requires at least lz4-r129. lz4-1.8.3 works. We recommend using the latest lz4 release available.

NUCs – not upwardly compatible

We have tried to make the process of upgrading seamless, however please pay attention to the following NUCs to consider whether they impact your particular installation.

The following files use a new file format. They are therefore unreadable by the previous versions. However, 3.6 can read all formats 3.5 can:

* 64-bit enumerations use a new file format. 3.5 enum files are read-only.
* Mapped list type (77h) deprecates old mapped nested types (77h+t). 77h+t files are read-only.
* Guids with u, p or g attribute use a new file format.

Added ajf and ajf0, to behave as V2.8 aj and aj0, i.e. they fill from LHS if RHS is null. e.g.

q)a:([]time:2#00:00:01;sym:`a`b;p:1 1;n:`r`s)

q)b:([]time:2#00:00:01;sym:`a`b;p:0 1)

q)c:([]time:2#00:00:00;sym:`a`b;p:1 0N;n:`r`s)

q)a~ajf[`sym`time;b;c]

1b

Suggested upgrade process

Even though we have run a wide range of tests on V3.6, and various customers have been kind enough to repeatedly run their own tests during the last few months of development, customers who wish to upgrade to V3.6 should run their own tests on their own data and code/queries before promoting to production usage. **Most importantly, be aware that rolling back to a previous version will be complicated by the fact that files written by v3.6 are not readable by prior versions**, hence users should test thoroughly prior to committing to an upgrade. In the event that you do discover a suspected bug, please email [tech@kx.com](mailto:tech@kx.com)