

# Retro 11.0 Porting Notes

## Overview

Retro 11 has numerous significant changes from Retro 10. This document addresses many of the changes users are likely to run into when porting code to Retro 11.

For those familiar with traditional Forth implementations, this may also prove helpful in getting started with Retro 11.

Many of these changes are the result of the introduction of *quotes* (anonymous code blocks) and *combinators* (functions operating on quotes).

## Conditionals

A significant change is the handling of conditionals. In Retro 10, a large number of functions were provided.

```
=if !if <if >if if
then ;then else if; 0;
= < > <>
```

The basic forms conditionals appeared in took one of two forms:

```
<if> ... true ... <then>
<if> ... true ... <else> ... false ... <then>
```

Retro 11 changes this significantly. The following functions have been removed:

```
=if !if <if >if if
then ;then else
```

New functions have been added.

```
if ifTrue ifFalse
when >= <=
```

In Retro 10, `<` and `>` actually checked for *less than or equal to* and *greater than or equal to*, rather than *less than* and *greater than*. This has been corrected in Retro 11.

Conditionals are now handled by the use of quotes and combinators. The basic forms are:

```
... flag ... [ ... true ... ] [ ... false ... ] if
... flag ... [ ... true ... ] ifTrue
... flag ... [ ... false ... ] ifFalse
```

Additionally, Retro 11 provides a case style combinator named **when**. It is used like:

```
: foo ( n-m )
  [ 1 = ] [ drop 14 ] when
  [ 2 = ] [ drop 15 ] when
  [ 3 = ] [ drop 16 ] when
  ( default action )
  drop 999 ;
```

# Loops

Counted loops are now handled using quotes and combinators. Retro 10 provided the following for creating simple counted loops:

```
... count ... for ... code ... next
... count ... fori ... code using loop index ... nexti
```

In place of this, Retro now has three combinators.

```
... count ... [ ... code ... ] times
... count ... [ ... code using loop index ... ] iter
... count ... [ ... code using loop index ... ] iterd
```

Note here that we have two combinators providing loop indexes. The first, **iter**, counts up, while the second, **iterd**, counts down.

It is no longer possible to directly access the loop counter using **r**.

In addition, a new combinator has been added:

```
[ ... code returning a flag ... ] while
```

This will execute the quote repeatedly, until the flag returned is set to zero.

The old **repeat** / **again** loops remain intact.

# Address Stack

Under Retro 10, it was common to see functions using the address stack quite freely. This is now discouraged.

We can replace some forms using quotes and combinators:

```
1 2 push 3 + pop
1 2 push 3 + push 4 pop + pop
```

Becomes:

```
1 2 [ 3 + ] dip
1 2 [ 3 + [ 4 ] dip + ] dip
```

The use of combinators ensures that the stack remains balanced, and that Code retains a greater consistency across the board. When coupled with an editor that is aware of brackets, it can also help to quickly identify the overall stack flow.

As a side effect, early exits via **pop drop** sequences are no longer guaranteed to work. Rewrite code using these to exit in a more structured manner, or create new combinators to deal with situations that need them.

# The 2 Prefix

Retro 11 introduces the use of a new prefix, 2 for executing functions twice. This has one significant side effect that you need to watch for: **2dup** is now the same as **dup dup**, not **over over**. Use **2over** instead:

```
2dup ( x-xxx )
2over ( xy-xy )
```

**2drop** will still work as expected. This change also allows for some other interesting things such as:

```
1 2 3 2+
1 2 3 4 2nip .s
```

Also, **-rot** is now done by doing **2rot**:

```
1 2 3 rot 2rot
```

## Removals

In addition to the loop and conditional functions, a number of other functions have been removed.

- The **quotes'** vocabulary is gone; it is now part of the core language.
- The **canvas'** vocabulary is now part of the library as most targets do not support the canvas devices.
- The **net** vocabulary is gone; Retro 11 currently lacks a standard socket or networking interface.
- **-rot**, **2drop** and **2dup** are removed as they are now redundant.

Additionally, many functions have been renamed for clarity and/or to follow the naming guidelines in the code style guide.

Old	New	Notes
>number	toNumber	Renamed to follow code style guide
with-class	withClass	Renamed to follow code style guide
remap-keys	remapKeys	Renamed to follow code style guide
console	console'	All vocabularies now end in '
\$	strings'	All vocabularies now end in '
file	files'	All vocabularies now end in '
whitespace	remapping	Renamed to reflect intent better; toggles character remapping, not just whitespace
[	[[	The single bracket [ is now used to start a quote
]	]]	The single bracket ] is now used to end a quote
key	getc	Renamed to be more consistent with other I/O functions
emit	putc	Renamed to be more consistent with other I/O functions
type	puts	Renamed to be more consistent with other I/O functions
(.)	putn	Renamed to be more consistent with other I/O functions
<	<=	Renamed to reflect actual usage. < now acts as expected
<=	>=	Renamed to reflect actual usage. > now acts as expected
neg	negate	Renamed for clarity purposes