MatchBuss

maintains the buss containing symbols in the order that they were added.

MatchDefs update from the MatchBuss when the patterns match in sequence.

for example

given a Matchbuss containing

```
[\mainmix, \scene_0, \b_59, \k_a, \midiNote_30]
```

a MatchDef with pattern

```
[\scene_0, \k_a]
```

will match because the symbols \scene_0 and \k_a appear in the right order

HOWEVER

given a Matchbuss containing

```
[\mainmix, \b_59, \k_a, \midiNote_30, \scene_0]
```

a MatchDef with pattern

```
[\scene_0, \k_a]
```

will not match because they are in the wrong order.

again ans more simply

a MatchDef with pattern [\scene_0, \k_a] will match a MatchBuss with [\mainmix, \scene_0, \k_a]

a MatchDef with pattern [\scene_0, \k_a] will not match a MatchBuss with [\mainmix, \k_a, \scene_0]

instance methods

.addDef(name, pattern, exclude, unless)
name - a symbol
pattern - an array of symbols
exclude - an array of symbols
unless - an array of symbols

create and attach a Matchdef to the MatchBuss. The MatchDef of name will match the the MatchBuss contains pattern.
When the pattern matches the MatchDef will stop any MatchDefs that match exclude. Then the MatchDef pattern matches the will not start its EventDefs if the pattern unless already matches.

.add(key, chan)
key - a symbol that wil be added to the MatchBuss
chan - a symbol, optional. When in a CtrlBuss network scenario, add this symbol to all network Matchbuss listening on channel.

.remove(key, chan)
key - a symbol that wil be removed from the MatchBuss
chan - a symbol, optional. When in a CtrlBuss network scenario, remove this symbol from all network Matchbuss listening on channel.

.at(key)
retreive a value from the local environment

.put(key, value)
put a value into the local environment

.setLast(matchName)
matchName - a symbol
Commonly used internally and may only be of limited user application.

.signal(...args)
args are passed to the MatchDef that has focus

.disconnect(matchDefNames)
matchDefNames - a symbol, name of MatchDef(s)
MatchDef(s) with the names argued will set .listen(false)

.connect(matchDefNames)
matchDefNames - a symbol, name of MatchDef(s)
MatchDef(s) with the names argued will set .listen(true)

.toggle(key)
key - a symbol
add or remove the the key depending whether the exists in the MatchBuss

.update
Commonly used internally and may only be of limited user application.

communicate with all MatchDef the state of things. prompt MatchDefs

//
//handlers
//
//handlers are an identity dictionary of named functions
//when a matchbuss .add or .remove, the \key and \chan are argued to the \add and \remove handlers (if they exist)

.addHandler()
Commonly used internally and may only be of limited user application.

.removeHandler()
Commonly used internally and may only be of limited user application.