

File created: 29-Jan-2025 19:34:13 {WMEDLEY}<lispusers>MULTI-ALIST.;15

edit by: rmk

changes to: (FNS MAPMULTI)

previous date: 25-Jan-2025 15:04:13 {WMEDLEY}<lispusers>MULTI-ALIST.;14

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

## (RPAQQ MULTI-ALISTCOMS

```
((MACROS GETMULTI PUTMULTI PUTMULTI-D PUTMULTI-NEW PUTMULTI-COUNT PUTMULTI-SUM REMOVEMULTI REMOVEMULTIALI
```

```
)  
(MACROS FGETMULTI FPUTMULTI FPUTMULTI-D FPUTMULTI-NEW)  
(FNS MAPMULTI MAPMULTI1 COLLECTMULTI)  
(FNS GETMULTI.EXPAND PUTMULTI.EXPAND REMOVEMULTI.EXPAND)  
(MACROS ADDTOMULTI)  
(FNS ADDTOMULTI1)  
(LOCALVARS . T)))
```

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS GETMULTI MACRO (ARGS (GETMULTI.EXPAND 'SASSOC ARGS)))
```

```
(PUTPROPS PUTMULTI MACRO (ARGS (PUTMULTI.EXPAND 'SASSOC ARGS)))
```

```
(PUTPROPS PUTMULTI-D MACRO (ARGS (PUTMULTI.EXPAND 'SASSOC ARGS NIL T)))
```

```
(PUTPROPS PUTMULTI-NEW MACRO (ARGS (PUTMULTI.EXPAND 'SASSOC ARGS)))
```

```
(PUTPROPS PUTMULTI-COUNT MACRO (ARGS (PUTMULTI.EXPAND 'SASSOC (APPEND ARGS '(1))  
NIL NIL T)))
```

```
(PUTPROPS PUTMULTI-SUM MACRO (ARGS (PUTMULTI.EXPAND 'SASSOC ARGS NIL NIL T)))
```

```
(PUTPROPS REMOVEMULTI MACRO (ARGS (REMOVEMULTI.EXPAND ARGS)))
```

```
(PUTPROPS REMOVEMULTIALI MACRO (ARGS (REMOVEMULTI.EXPAND ARGS T)))
```

```
)
```

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS FGETMULTI MACRO (ARGS (GETMULTI.EXPAND 'FASSOC ARGS)))
```

```
(PUTPROPS FPUTMULTI MACRO (ARGS (PUTMULTI.EXPAND 'FASSOC ARGS)))
```

```
(PUTPROPS FPUTMULTI-D MACRO (ARGS (PUTMULTI.EXPAND 'FASSOC ARGS NIL T)))
```

```
(PUTPROPS FPUTMULTI-NEW MACRO (ARGS (PUTMULTI.EXPAND 'FASSOC ARGS)))
```

```
)
```

```
(DEFINEQ
```

## (MAPMULTI

```
[LAMBDA (MULTIALIST MAPFN)
```

```
; Edited 29-Jan-2025 19:33 by rmk  
; Edited 25-Jan-2025 14:51 by rmk  
; Edited 16-Jan-2025 10:32 by rmk  
; Edited 6-Jan-2020 10:15 by rmk:
```

```
;; MAPMULTI applies a mapping function of N args to each item in an N-way item in the multi-alist at MULTIALIST. If an item C is inserted by  
;; (PUTMULTI FOO A B C), then MAPFN should be a 3 argument function and it will be applied to A B C. The caller is responsible for making sure  
;; the arities of the index and the mapfn correspond.
```

```
(DECLARE (SPECVARS MAPFN))
```

```
(LET ($$LISTFORARGS$$)
```

```
(DECLARE (SPECVARS $$LISTFORARGS$$))
```

```
(SETQ $$LISTFORARGS$$ (FOR I FROM 1 TO (NARGS MAPFN) COLLECT NIL))
```

```
(MAPMULTI1 MULTIALIST $$LISTFORARGS$$ (NARGS MAPFN))
```

## (MAPMULTI1

```
[LAMBDA (SUBALIST ARGLIST NREMAINING)
```

```
; Edited 25-Jan-2025 15:03 by rmk  
; Edited 22-Jan-2025 23:42 by rmk  
; Edited 16-Jan-2025 10:29 by rmk  
; Edited 6-Jan-2020 10:21 by rmk:
```

```
(DECLARE (USEDFREE $$LISTFORARGS$$ MAPFN))
```

```
(if [AND (IGREATERP NREMAINING 1)
```

```
(LISTP (CAR (LISTP SUBALIST)
```

```
then
```

```
;; Still a list of alists.
```

```
(for SI in SUBALIST do (RPLACA ARGLIST (CAR SI))
```

```
(MAPMULTI1 (CDR SI)
```

```
(CDR ARGLIST)
```

```
(SUB1 NREMAINING)))
```

```
else (for ITEM inside SUBALIST do (RPLACA ARGLIST ITEM)
```

(APPLY MAPFN \$\$LISTFORARGSS\$)

(COLLECTMULTI

[LAMBDA (MULTIALIST MAPFN)

; Edited 25-Jan-2025 15:00 by rmk
; Edited 22-Jan-2025 23:44 by rmk
; Edited 6-Jan-2020 10:15 by rmk:

(LET (\$\$COLLECT)
(DECLARE (SPECVARS \$\$COLLECT))
(MAPMULTI MULTIALIST MAPFN
\$\$COLLECT])

)

(DEFINEQ

(GETMULTI.EXPAND

[LAMBDA (ASSOCFN ARGS)

; Edited 16-Jan-2025 10:27 by rmk
; Edited 19-Jul-2020 00:38 by rmk:
; Edited 22-Mar-2020 13:21 by rmk:
; Edited 27-Feb-2020 13:44 by rmk:
; Edited 30-Dec-2019 20:50 by rmk:

:: If SUM, returns the value after the last argument, paired with PUTMULTISUM

(IF (CDR ARGS)
THEN `(LET (\$\$CELL\$\$)
(DECLARE (LOCALVARS \$\$CELL\$\$))
,@[FOR ATAIL (HEAD \_ (CAR ARGS)) ON (CDR ARGS)
COLLECT (PROG1 `[SETQ \$\$CELL\$\$ (CDR (,ASSOCFN , (CAR ATAIL)
,HEAD]
(SETQ HEAD '\$\$CELL\$\$) )]
\$\$CELL\$\$)
ELSE (CAR ARGS])

(PUTMULTI.EXPAND

[LAMBDA (ASSOCFN ARGS ALLOWREPEATS SINGLEVALUE SUM)

; Edited 23-Jan-2025 09:40 by rmk
; Edited 16-Jan-2025 10:18 by rmk
; Edited 17-Aug-2020 14:09 by rmk:

:: If ALLOWREPEATS, doesn't test (MEMBER) for preexisting values, just accumulates

:: If SINGLEVALUE, new value smashes out old

:: For SUM, the last argument is the increment to be added to the current value, and the incremented value is returned for PUTMULTISUM and for

:: GETMULT

::

::

:: We get the self method so that any expressions in the form will be evaluated only once.

(CL:MULTIPLE-VALUE-BIND (TEMPVARS VALFORMS STOREVARS STOREFORM ACCESSFORM)
(CL:GET-SETF-METHOD (CAR ARGS))
(CL:IF (CDR ARGS)
`(LET\*
, (FOR VF IN VALFORMS AS TV IN TEMPVARS COLLECT (LIST TV VF))
(DECLARE (LOCALVARS ,@TEMPVARS))
(LET
(\$\$ARG1\$\$ \$\$ARG2\$\$)
(DECLARE (LOCALVARS \$\$ARG1\$\$ \$\$ARG2\$\$))
,@[FOR ATAIL (HEAD \_ ACCESSFORM) ON ARGS WHILE (CDR ATAIL)
JOIN
(IF (AND SUM (NULL (CDDR ATAIL)))
THEN (POP ATAIL)
`[(CL:UNLESS ,HEAD (RPLACD \$\$ARG1\$\$ 0))
(SETQ \$\$ARG2\$\$ (ADD ,HEAD , (CAR ATAIL))
ELSE (PROG1 `[(SETQ \$\$ARG2\$\$ , (CADR ATAIL))
, (IF (CDDR ATAIL)
THEN `[SETQ \$\$ARG1\$\$ (OR (,ASSOCFN \$\$ARG2\$\$ ,HEAD)
(CAR (CL:PUSH (CONS \$\$ARG2\$\$)
,HEAD]
ELSEIF ALLOWREPEATS
THEN `(push ,HEAD \$\$ARG2\$\$)
ELSEIF SINGLEVALUE
THEN `(RPLACD \$\$ARG2\$\$)
ELSE `(OR (MEMBER \$\$ARG2\$\$ ,HEAD)
(push ,HEAD \$\$ARG2\$\$)
(SETQ HEAD '(CDR \$\$ARG1\$\$)))]
\$\$ARG2\$\$)
(CAR ARGS)))]

(REMOVEMULTI.EXPAND

[LAMBDA (ARGS ALLFLAG)

; Edited 16-Jan-2025 10:34 by rmk
; Edited 17-Aug-2020 15:12 by rmk:
; Edited 17-May-2020 17:25 by rmk:
; Edited 14-Feb-2020 11:24 by rmk:
; Edited 25-Dec-2019 09:57 by rmk:

:: If ALLFLAG, then all data after the last of ARGS, if any, is removed. That is, if there are 3 keys to the index, and REMOVEMULTIALL is invoked
:: with 2 keys, then it's as if no entries were made for any of the third keys after those first two. In the case of REMOVEMULTIALL, it returns the

;; previous tail.

;; No point in distinguishing FASSOC from SASSOC here.

```
(CL:MULTIPLE-VALUE-BIND (TEMPVARS VALFORMS STOREVARS STOREFORM ACCESSFORM)
  (CL:GET-SETF-METHOD (CAR ARGS))
  (CL:IF (CDR ARGS)
    \ (LET*
      , (FOR VF IN VALFORMS AS TV IN TEMPVARS COLLECT (LIST TV VF))
      (DECLARE (LOCALVARS ,@TEMPVARS))
      (LET
        ($$ARG1$$ $$ARG2$$)
        (DECLARE (LOCALVARS $$ARG1$$ $$ARG2$$))
        ,@ (FOR ATAIL (HEAD _ ACCESSFORM) ON ARGS WHILE (CDR ATAIL)
          JOIN (PROG1 \ [(SETQ $$ARG2$$ , (CADR ATAIL))
            , (IF (CDDR ATAIL)
              THEN \ (SETQ $$ARG1$$ (SASSOC $$ARG2$$ , HEAD))
              ELSEIF ALLFLAG
              THEN \ (CL:WHEN (SETQ $$ARG1$$ (SASSOC $$ARG2$$ , HEAD))
                (SETQ $$ARG2$$ (CDR $$ARG1$$))
                (RPLACD $$ARG1$$))
              ELSE \ (AND (SETQ $$ARG2$$ (MEMBER $$ARG2$$ , HEAD))
                (RPLACD $$ARG1$$ (DREMOVE (SETQ $$ARG2$$ (CAR $$ARG2$$))
                  , HEAD]
                (SETQ HEAD ' (CDR $$ARG1$$)))]
          $$ARG2$$)
        (CAR ARGS)))]])
```

)

(DECLARE%: EVAL@COMPILE

```
(PUTPROPS ADDTOMULTI MACRO [ARGS (CL:MULTIPLE-VALUE-BIND (TEMPVARS VALFORMS STOREVARS STOREFORM ACCESSFORM)
  (CL:GET-SETF-METHOD (CAR ARGS))
  \ (LET* [, @ (FOR VF IN VALFORMS AS TV IN TEMPVARS
    COLLECT (LIST TV VF))
    ($$KEYS , (CADR ARGS))
    (DECLARE (LOCALVARS $$KEYS ,@TEMPVARS))
    (COND
      [(LISTP $$KEYS)
        (CL:UNLESS (SASSOC (CAR $$KEYS)
          , ACCESSFORM)
          (CL:PUSH (CONS (CAR $$KEYS)
            , ACCESSFORM))
          (ADDTOMULTI1 , ACCESSFORM $$KEYS , (CADDR ARGS)
            (T (CL:SETF , ACCESSFORM , (CADDR ARGS)))]])
```

)

(DEFINEQ

**(ADDTOMULTI1**

[LAMBDA (PLACE KEYS VAL)

; Edited 22-Jan-2025 23:47 by rmk  
; Edited 17-Aug-2020 15:05 by rmk:

;; This allows the keys to be provided in a single list rather than as separate arguments.

```
(FOR I (P _ PLACE) IN KEYS DO [SETQ P (OR (SASSOC I P)
  (CAR (PUSH (CDR P)
    (CONS I
      (FINALLY (PUSH (CDR P)
        VAL))
      VAL))
  VAL])
```

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)

)

---

**FUNCTION INDEX**

ADDTOMULTI1 .....3    GETMULTI.EXPAND .....2    MAPMULTI1 .....1    REMOVEMULTI.EXPAND .....2  
COLLECTMULTI .....2    MAPMULTI .....1    PUTMULTI.EXPAND .....2

---

**MACRO INDEX**

ADDTOMULTI .....3    FPUTMULTI-D .....1    PUTMULTI .....1    PUTMULTI-NEW .....1    REMOVEMULTIALL .....1  
FGETMULTI .....1    FPUTMULTI-NEW .....1    PUTMULTI-COUNT .....1    PUTMULTI-SUM .....1  
FPUTMULTI .....1    GETMULTI .....1    PUTMULTI-D .....1    REMOVEMULTI .....1

---