

File created: 22-Sep-87 13:17:50 {DSK}<XAVIER>FORMACRO.;10

changes to: (IL:FUNCTIONS SDC-FOR XCL-USER::SELECTC ISOPRP FROMTOTEST)
(IL:VARS IL:FORMACROCOMS)
(IL:PROPS (AS FORWARD)
(THEREIS FORWARD)
(BY FORWARD)
(FROM FORWARD)
(IN FORWARD)
(ON FORWARD)
(TO FORWARD)
(JOIN FORWARD))

previous date: 21-Sep-87 17:07:45 {DSK}<XAVIER>FORMACRO.;9

Read Table: XCL

Package: USER

Format: XCCS

; Copyright (c) 1987 by System Development Corp.. All rights reserved.

```
(IL:RPAQQ IL:FORMACROCOMS
  ((IL:P (EXPORT 'FOR))
  (IL:FUNCTIONS EVLIST FOR FROMTOTEST ISOPRP SDC-FOR XCL-USER::SELECTC TESTFOR)
  (IL:PROP FORWARD ALWAYS AS BIND BY COLLECT COUNT DO EACHTIME FINALLY FIRST FROM IN INSIDE JOIN
    LARGEST MAX MIN NEVER ON REPEATWHILE REPEATUNTIL SMALLEST SUM TO THEREIS UNION UNLESS UNTIL
    WHEN WHILE)
  (IL:PROP IL:MAKEFILE-ENVIRONMENT IL:FORMACRO)
  (IL:EDITHIST IL:FORMACRO)))

(EXPORT 'FOR)

(DEFUN EVLIST (L)
  (NREVERSE (MAPCAR #'EVAL L)))

(DEFMACRO FOR (&BODY X)
  (SDC-FOR X))

(DEFUN FROMTOTEST (IV FROM TO BY BYVAR TOVAR)
  (LET ((BYVAL (SYMBOL-VALUE BY)))
    (IF (CONSTANTP BYVAL)
      (IF (AND (NUMBERP FROM)
              (NUMBERP TO)
              (< TO FROM)
              (= BYVAL 1))
        (PROGN
          (SET BY -1)
          `(IF (< ,IV ,TO)
              (GO $$OUT)))
          `(IF (, (IF (MINUSP BYVAL)
                    '<'
                    '>')
                ,IV
                , (IF (CONSTANTP TO)
                    TO
                    TOVAR))
              (GO $$OUT)))
          `(IF (AND ,BYVAR (OR (= 0 ,BYVAR)
                              (IF (MINUSP ,BYVAR)
                                  (< ,IV , (IF (CONSTANTP TO)
                                                TO
                                                TOVAR))
                                  (> ,IV , (IF (CONSTANTP TO)
                                                TO
                                                TOVAR))))))
              (GO $$OUT))))))

; BY was probably a default +1 so silently force to -1 before the
; incremter gets expanded

(DEFUN ISOPRP (X)
  (IF (SYMBOLP X)
    (GET (INTERN (STRING-UPCASE X)
                "USER")
      'FORWARD)))

(DEFUN SDC-FOR (X)
  (DECLARE (SPECIAL X))
  (PROG (VARLIST IV IV1 IOV INITS EACHS PRETESTS BODYS POSTTESTS UPDATES FINALS OP (INCCNT 0)
        $$INC0 $$INC1 $$INC2 $$INC3 $$INC4 $$INC5 $$INC6 $$INCVAR0 $$INCVAR1 $$INCVAR2 $$INCVAR3
        $$INCVAR4 $$INCVAR5 $$INCVAR6 $$INIT0 $$INIT1 $$INIT2 $$INIT3 $$INIT4 $$INIT5 $$INIT6 $$END0
        $$END1 $$END2 $$END3 $$END4 $$END5 $$END6)
    (DECLARE (SPECIAL VARLIST IV IV1 IOV INITS EACHS PRETESTS BODYS POSTTESTS UPDATES FINALS INCCNT $$INC0
```

\$\$INC1 \$\$INC2 \$\$INC3 \$\$INC4 \$\$INC5 \$\$INC6 \$\$INCVAR0 \$\$INCVAR1 \$\$INCVAR2 \$\$INCVAR3
 \$\$INCVAR4 \$\$INCVAR5 \$\$INCVAR6 \$\$INIT0 \$\$INIT1 \$\$INIT2 \$\$INIT3 \$\$INIT4 \$\$INIT5 \$\$INIT6
 \$\$END0 \$\$END1 \$\$END2 \$\$END3 \$\$END4 \$\$END5 \$\$END6))

```
(PUSH '$$VAL VARLIST)
(COND
  ((CONSP (CAR X))
   (COND
    ((CONSP (CAAR X))
     (SETQ IOV (SETQ IV1 (SETQ IV (CAAR X)))))) ;(FOR ((VAR "...") "...") "...")
    (T
     (SETQ IV1 (SETQ IV (CAAR X)))) ;(FOR (VAR "...") "...")
     (SETQ VARLIST (NCONC (REVERSE (CAR X))
                          VARLIST))
     (SETQ X (CDR X))
     ((AND (SYMBOLP (CAR X))
          (NOT (ISOPRP (CAR X)))) ;(FOR VAR "...")
      (SETQ IOV (SETQ IV1 (SETQ IV (CAR X))))
      (PUSH (LIST (CAR X)
                  1)
            VARLIST)
      (SETQ X (CDR X)))
    (T ;E.G. (FOR DO "...")
     (SETQ IOV (SETQ IV1 (SETQ IV '$$ITER)))
     (PUSH '($$ITER 1)
           VARLIST)))
```

```
TOP (IF (NULL X)
        (RETURN `(PROG , (NREVERSE VARLIST)
                    ,@(EVLIST INITS)
                    $$LP
                    ,@(EVLIST EACHS)
                    ,@(EVLIST PRETESTS)
                    ,@(EVLIST BODYYS)
                    $$ITERATE
                    ,@(EVLIST POSTTESTS)
                    ,@(EVLIST UPDATES)
                    (GO $$LP)
                    $$OUT
                    ,@(EVLIST FINALS)
                    (RETURN $$VAL))))
      (COND
       ((SETQ OP (ISOPRP (CAR X)))
        (EVAL OP)
        (COND
         ((AND (CDDR X)
              (NOT (ISOPRP (CADDR X)))
              (NOT (EQL 'DO (INTERN (STRING-UPCASE (CAR X))
                                   "USER")))))
          (PRINT (LIST "Warning, no implicit PROG in Unisys FOR macro:" X)
                 T)))
         (SETQ X (CDDR X)))
        (T (PUSH (LIST 'QUOTE (CAR X))
                BODYYS)
          (SETQ X (CDR X))))
      (GO TOP)))
```

```
(DEFMACRO XCL-USER::SELECTC (XCL-USER::SELECTOR &REST XCL-USER::CASES)
  `(CASE ,XCL-USER::SELECTOR
    (IL:\,\,@ (FOR CASE XCL-USER::ON XCL-USER::CASES XCL-USER::WHILE (CDR CASE)
                  XCL:COLLECT
                  (CONS (EVAL (CAAR CASE))
                        (CDAR CASE))))
    (OTHERWISE ,(CAR (LAST XCL-USER::CASES))))))
```

```
(DEFMACRO TESTFOR (L . BODY)
  (PROG (X)
    (SETQ X (SDC-FOR (CONS L BODY)))
    (TERPRI)
    (WRITE X :PRETTY T)
    (TERPRI)
    (RETURN X)))
```

```
(IL:PUTPROPS ALWAYS FORWARD (PROGN (PUSH '' (SETQ $$VAL T)
                                       INITS)
                                     (PUSH (LIST 'QUOTE `(COND ((NULL ,(CADR X))
                                                                (SETQ $$VAL NIL)
                                                                (GO $$OUT))))
                                       BODYYS)))
```

```
(IL:PUTPROPS AS FORWARD (PROGN (INCF INCCNT)
                                 (SETQ IOV (SETQ IV (CADR X)))
                                 (OR IV1 (SETQ IV1 IV))
                                 (PUSH (LIST IV 1)
                                       VARLIST)))
```

```
(IL:PUTPROPS BIND FORWARD (IF (CONSP (CADR X))
```

```
(SETQ VARLIST (APPEND (REVERSE (CADR X))
                      VARLIST))
(PUSH (CADR X)
      VARLIST)))
```

```
(IL:PUTPROPS BY FORWARD (LET ((INCAMT (INTERN (CONCATENATE 'STRING "$$INC" (PRINC-TO-STRING INCCNT))
                                                "USER")))
  (COND ((NUMBERP (CADR X))
        (SET INCAMT (CADR X)))
        (T (SET INCAMT (SUBST IOV IV (CADR X)))
            (LET ((BYVAR (GENSYM)))
              (PUSH BYVAR VARLIST)
              (SET (INTERN (CONCATENATE 'STRING "$$INCVAR" (PRINC-TO-STRING
                                                INCCNT))
                          "USER")
                    BYVAR))))))
```

```
(IL:PUTPROPS COLLECT FORWARD (PROGN (PUSH (LIST 'QUOTE `(SETQ $$VAL (CONS ,(CADR X)
                                                                              $$VAL)))
      BODY)
  (PUSH `(SETQ $$VAL (NREVERSE $$VAL))
        FINALS)))
```

```
(IL:PUTPROPS COUNT FORWARD (PROGN (PUSH `(SETQ $$VAL 0)
      INITS)
  (PUSH (LIST 'QUOTE `(IF ,(CADR X)
                          (SETQ $$VAL (1+ $$VAL))))
        BODY)))
```

```
(IL:PUTPROPS DO FORWARD (IF (NULL (ISOPRP (CADR X)))
  (PUSH (LIST 'QUOTE (CADR X))
        BODY)
  (SETQ X (CONS 'X X))))
```

```
(IL:PUTPROPS EACHTIME FORWARD (PUSH (LIST 'QUOTE (CADR X))
      EACH))
```

```
(IL:PUTPROPS FINALLY FORWARD (PUSH (LIST 'QUOTE (CADR X))
      FINALS))
```

```
(IL:PUTPROPS FIRST FORWARD (PUSH (LIST 'QUOTE (CADR X))
      INITS))
```

```
(IL:PUTPROPS FROM FORWARD (LET ((INCREM (INTERN (CONCATENATE 'STRING "$$INC" (PRINC-TO-STRING INCCNT))
                                                "USER")))
  (IF (EQL IV (CADR (CADAR INITS)))
      (RPLACA (CDDR (CADAR INITS))
              (CADR X))
      (PUSH `(SETQ ,IV ,(CADR X))
            INITS))
  (SET (INTERN (CONCATENATE 'STRING "$$INIT" (PRINC-TO-STRING INCCNT))
          "USER")
        (CADR X))
  (OR (SYMBOL-VALUE INCREM)
      (SET INCREM 1))
  (OR (EQL IV (CADR (CADDAR UPDATES)))
      (PUSH `(LIST 'INCF ',IV (IF (CONSTANTP ,INCREM)
                                  ,INCREM
                                  (LIST 'SETQ ,(INTERN (CONCATENATE 'STRING
                                                                    "$$INCVAR"
                                                                    (PRINC-TO-STRING
                                                                    INCCNT))
                                                                    "USER")
                                      ,INCREM))))
        UPDATES))))
```

```
(IL:PUTPROPS IN FORWARD (PROG (INCEXP)
  (SETQ IOV (GENSYM))
  (SETQ INCEXP (INTERN (CONCATENATE 'STRING "$$INC" (PRINC-TO-STRING INCCNT))
                        "USER")))
  (SET INCEXP (LIST 'CDR IOV))
  (PUSH (LIST IOV (CADR X))
        VARLIST)
  (PUSH (LIST 'QUOTE `(IF (NOT ,IOV)
                          (GO $$OUT)
                          (SETQ ,IV (CAR ,IOV))))
        EACH)
  (PUSH `(LIST 'SETQ ,(LIST 'QUOTE IOV)
                ,INCEXP)
        UPDATES)))
```

```
(IL:PUTPROPS INSIDE FORWARD (PROGN (SETQ IOV (GENSYM))
  (PUSH (LIST IOV (CADR X))
        VARLIST)
  (PUSH (LIST 'QUOTE `(COND ((NULL ,IOV)
                              (GO $$OUT))
                              ((NOT (CONSP ,IOV))
```

```

      (SETQ ,IV ,IOV)
      (SETQ ,IOV NIL))
      (T (SETQ ,IV (CAR ,IOV))
         (SETQ ,IOV (CDR ,IOV))))))

```

EACHS)))

```

(IL:PUTPROPS JOIN FORWARD (PROGN (PUSH (LIST 'QUOTE `(SETQ $$VAL (NCONC (NREVERSE ,(CADR X))
                                                                           $$VAL)))

```

```

      BODY))
      (PUSH ''(SETQ $$VAL (NREVERSE $$VAL))
            FINALS)))

```

```

(IL:PUTPROPS LARGEST FORWARD (PROGN (PUSH '$$EXTREME VARLIST)
                                     (PUSH (LIST 'QUOTE `(COND ((OR (NULL $$EXTREME)
                                                                    (> ,(CADR X)
                                                                    $$EXTREME))
                                                                    (SETQ $$EXTREME ,(CADR X))
                                                                    (SETQ $$VAL ,IV))))

```

BODY)))

```

(IL:PUTPROPS MAX FORWARD (PUSH (LIST 'QUOTE `(IF (NULL $$VAL)
                                                    (SETQ $$VAL ,(CADR X))
                                                    (SETQ $$VAL (MAX $$VAL ,(CADR X))))))

```

BODY))

```

(IL:PUTPROPS MIN FORWARD (PUSH (LIST 'QUOTE `(IF (NULL $$VAL)
                                                    (SETQ $$VAL ,(CADR X))
                                                    (SETQ $$VAL (MIN $$VAL ,(CADR X))))))

```

BODY))

```

(IL:PUTPROPS NEVER FORWARD (PROGN (PUSH ''(SETQ $$VAL T)
                                      INITS)
                                    (PUSH (LIST 'QUOTE `(COND (,(CADR X)
                                                                (SETQ $$VAL NIL)
                                                                (GO $$OUT))))

```

BODY)))

```

(IL:PUTPROPS ON FORWARD (PROG (INCEXP)
                               (SETQ INCEXP (INTERN (CONCATENATE 'STRING "$$INC" (PRINC-TO-STRING (INCF INCCNT

```

```

                                     "USER"))
                               (SET INCEXP (LIST 'CDR IOV))
                               (COND ((EQL IV (CAAR VARLIST))
                                       (RPLACA (CDAR VARLIST)
                                              (CADR X)))
                                       (T (PUSH (LIST 'QUOTE `(SETQ ,IV ,(CADR X))
                                                    INITS)))
                                       (PUSH `(LIST 'SETQ ,(LIST 'QUOTE IV)
                                                ,INCEXP)
                                             UPDATES)
                                       (PUSH (LIST 'QUOTE `(IF (NOT ,IV)
                                                                (GO $$OUT))))
                                       PRETESTS)))

```

```

(IL:PUTPROPS REPEATWHILE FORWARD (PUSH (LIST 'QUOTE `(IF (NOT ,(CADR X))
                                                           (GO $$OUT)))

```

POSTTESTS))

```

(IL:PUTPROPS REPEATUNTIL FORWARD (IF (NUMBERP (CADR X))
                                       (PUSH (LIST 'QUOTE `(IF (> ,IV ,(CADR X))
                                                               (GO $$OUT)))

```

```

                                       POSTTESTS)
                                       (PUSH (LIST 'QUOTE `(IF ,(CADR X)
                                                               (GO $$OUT)))

```

POSTTESTS)))

```

(IL:PUTPROPS SMALLEST FORWARD (PROGN (PUSH '$$EXTREME VARLIST)
                                       (PUSH (LIST 'QUOTE `(COND ((OR (NULL $$EXTREME)
                                                                       (< ,(CADR X)
                                                                       $$EXTREME))
                                                                       (SETQ $$EXTREME ,(CADR X))
                                                                       (SETQ $$VAL ,IV))))

```

BODY)))

```

(IL:PUTPROPS SUM FORWARD (PROGN (PUSH ''(SETQ $$VAL 0)
                                   INITS)
                                  (PUSH (LIST 'QUOTE `(SETQ $$VAL (+ ,(CADR X)
                                                                       $$VAL)))

```

BODY)))

```

(IL:PUTPROPS TO FORWARD (LET (LIMIT (INITAMT (INTERN (CONCATENATE 'STRING "$$INIT" (PRINC-TO-STRING INCCNT)
                                                                    "USER"))
                                                (INCREM (INTERN (CONCATENATE 'STRING "$$INC" (PRINC-TO-STRING INCCNT)
                                                                    "USER"))
                                                (BYVAR (INTERN (CONCATENATE 'STRING "$$INCVAR" (PRINC-TO-STRING INCCNT)
                                                                    "USER"))
                                                (END (INTERN (CONCATENATE 'STRING "$$END" (PRINC-TO-STRING INCCNT)

```

```

                                "USER"))
(OR (EQL IV (CADR (CADAR INITS)))
    (PUSH '(SETQ ,IV 1)
          INITS))
(OR (SYMBOL-VALUE INCREM)
    (SET INCREM 1))
(OR (CONSTANTP (CADR X))
    (PUSH (LIST (SETQ LIMIT (GENSYM)
                (CADR X))
              VARLIST))
(OR (SYMBOL-VALUE INITAMT)
    (SET INITAMT 1))
(OR (EQL IV (CADR (CADDAR UPDATES)))
    (PUSH '(LIST 'INCF ',IV (IF (CONSTANTP ,INCREM
                                ,INCREM
                                (LIST 'SETQ ,BYVAR ,INCREM)))
          UPDATES))
(SET END (CADR X))
(PUSH (LIST 'FROMTOTEST ',IV INITAMT END ',INCREM BYVAR
          ',LIMIT)
      PRETESTS))

(IL:PUTPROPS THEREIS FORWARD (PROGN (PUSH (LIST 'QUOTE `(COND (, (CADR X)
                                                    (SETQ $$VAL (OR ,IV1 T))
                                                    (GO $$OUT))))
                                      BODY))

(IL:PUTPROPS UNION FORWARD (PUSH (LIST 'QUOTE `(SETQ $$VAL (UNION , (CADR X)
                                                                    $$VAL)))
                                      BODY))

(IL:PUTPROPS UNLESS FORWARD (PUSH (LIST 'QUOTE `(IF , (CADR X)
                                              (GO $$ITERATE)))
                                      PRETESTS))

(IL:PUTPROPS UNTIL FORWARD (IF (NUMBERP (CADR X))
    (PUSH (LIST 'QUOTE `(IF (> ,IV , (CADR X))
                            (GO $$OUT)))
          PRETESTS)
    (PUSH (LIST 'QUOTE `(IF , (CADR X)
                            (GO $$OUT)))
          PRETESTS)))

(IL:PUTPROPS WHEN FORWARD (PUSH (LIST 'QUOTE `(IF (NOT , (CADR X))
                                                      (GO $$ITERATE)))
                                      PRETESTS))

(IL:PUTPROPS WHILE FORWARD (PUSH (LIST 'QUOTE `(IF (NOT , (CADR X))
                                                      (GO $$OUT)))
                                      PRETESTS))

(IL:PUTPROPS IL:FORMACRO IL:MAKEFILE-ENVIRONMENT (:PACKAGE "USER" :READTABLE "XCL"))
(IL:DECLARE\ : IL:DONTCOPY
(IL:ADDTOVAR IL:EDITHISTALIST )
)
(IL:PUTPROPS IL:FORMACRO IL:COPYRIGHT ("System Development Corp." 1987))

```

FUNCTION INDEX

EVLIST1 FROMTOTEST1 ISOPRP1 SDC-FOR1

PROPERTY INDEX

ALWAYS2 COUNT3 IL:FORMACRO .5 LARGEST4 REPEATUNTIL .4 TO4 WHILE5
AS2 DO3 FROM3 MAX4 REPEATWHILE .4 UNION5
BIND2 EACHTIME3 IN3 MIN4 SMALLEST4 UNLESS5
BY3 FINALLY3 INSIDE3 NEVER4 SUM4 UNTIL5
COLLECT3 FIRST3 JOIN4 ON4 THEREIS5 WHEN5

MACRO INDEX

FOR1 XCL-USER::SELECTC2 TESTFOR2

VARIABLE INDEX

IL:EDITHISTALIST5
