```
(RPAQQ CONDITION-HIERARCHY-ILCOMS ((STRUCTURES INTERLISP-ERROR DEFINER-MISMATCH NO-SUCH-DEFINITION
                                        STACK-POINTER-RELEASED UNDEFINED-FUNCTION-IN-APPLY INDEX-BOUNDS-ERROR
                                        UNDEFINED-CAR-OF-FORM ILLEGAL-STACK-ARG)
                                 ;; Should be in XCL

                                    (STRUCTURES CALL-ERROR TOO-MANY-ARGUMENTS TOO-FEW-ARGUMENTS
                                            INVALID-ARGUMENT-LIST)
                                    (PROP FILETYPE CONDITION-HIERARCHY-IL)))


(DEFINE-CONDITION INTERLISP-ERROR (CL:ERROR)
    (MESSAGE)
    (:REPORT (LAMBDA (CONDITION *STANDARD-OUTPUT*)
                (DESTRUCTURING-BIND (MESS1 . MESS2)
                        (INTERLISP-ERROR-MESSAGE CONDITION)
                    (ERRORMESS1 MESS1 MESS2 'ERROR)))))


(DEFINE-CONDITION DEFINER-MISMATCH (CONDITION)
    (NAME TYPE DEFINITION)
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                (CL:FORMAT T "Definition ~S isn't a ~A definition for ~S." (DEFINER-MISMATCH-DEFINITION
                                                                                CONDITION)
                        (DEFINER-MISMATCH-TYPE CONDITION)
                        (DEFINER-MISMATCH-NAME CONDITION)))))


(DEFINE-CONDITION NO-SUCH-DEFINITION (CL:ERROR)
    (NAME TYPE)
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                (CL:FORMAT T "No ~A definition for ~S." (NO-SUCH-DEFINITION-TYPE CONDITION)
                        (NO-SUCH-DEFINITION-NAME CONDITION)))))


(DEFINE-CONDITION STACK-POINTER-RELEASED (CELL-ERROR)
    NIL
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                (CL:FORMAT T "Stack ptr has been released~&~S" NAME))))


(DEFINE-CONDITION UNDEFINED-FUNCTION-IN-APPLY (UNDEFINED-FUNCTION)
    (ARGUMENTS))


(DEFINE-CONDITION INDEX-BOUNDS-ERROR (CELL-ERROR)
    (INDEX)
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                (CL:FORMAT T "Index out of bounds: ~D." (INDEX-BOUNDS-ERROR-INDEX CONDITION)))))


(DEFINE-CONDITION UNDEFINED-CAR-OF-FORM (CONTROL-ERROR)
    (FUNCTION))


(DEFINE-CONDITION ILLEGAL-STACK-ARG (CONTROL-ERROR)
    (ARG)
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                (CL:FORMAT T "Illegal stack arg: ~S" (ILLEGAL-STACK-ARG-ARG CONDITION)))))

;; Should be in XCL


(DEFINE-CONDITION CALL-ERROR (CONTROL-ERROR)
    (CALLEE))


(DEFINE-CONDITION TOO-MANY-ARGUMENTS (CALL-ERROR)
    (MAXIMUM ACTUAL)
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                (COND
                   ((AND (TOO-MANY-ARGUMENTS-MAXIMUM CONDITION)
                         (TOO-MANY-ARGUMENTS-ACTUAL CONDITION))
                    (CL:FORMAT T "Too many arguments to ~A:~%   ~D ~:*~[were~;was~:;were~] given but at most
```

```
                                        ~D ~:*~[are~;is~:;are~] accepted" (TOO-MANY-ARGUMENTS-CALLEE CONDITION)
                                       (TOO-MANY-ARGUMENTS-ACTUAL CONDITION)
                                       (TOO-MANY-ARGUMENTS-MAXIMUM CONDITION)))
                        ((TOO-MANY-ARGUMENTS-CALLEE CONDITION)
                         (CL:FORMAT T "Too many arguments to ~A" (TOO-MANY-ARGUMENTS-CALLEE CONDITION)))
                        (T (CL:PRINC "Too many arguments"))))))))


(DEFINE-CONDITION TOO-FEW-ARGUMENTS (CALL-ERROR)
    (MINIMUM ACTUAL)
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                    (CL:IF (AND (TOO-FEW-ARGUMENTS-MINIMUM CONDITION)
                                (TOO-FEW-ARGUMENTS-ACTUAL CONDITION))
                        (CL:FORMAT T "Too few arguments to ~A:~%    ~D ~:*~[were~;was~:;were~] given but at least
                                ~D ~:*~[are~;is~:;are~] necessary" (TOO-FEW-ARGUMENTS-CALLEE CONDITION)
                                    (TOO-FEW-ARGUMENTS-ACTUAL CONDITION)
                                    (TOO-FEW-ARGUMENTS-MINIMUM CONDITION))
                        (CL:FORMAT T "Too few arguments to ~A" (TOO-FEW-ARGUMENTS-CALLEE CONDITION)))))))


(DEFINE-CONDITION INVALID-ARGUMENT-LIST (CALL-ERROR)
    (ARGUMENT)
    (:REPORT (CL:LAMBDA (CONDITION *STANDARD-OUTPUT*)
                    (COND
                        ((NULL (INVALID-ARGUMENT-LIST-CALLEE CONDITION))
                         (CL:FORMAT T "Invalid argument: ~S" (INVALID-ARGUMENT-LIST-ARGUMENT CONDITION)))
                        (T (CL:FORMAT T "~S was given an invalid argument: ~S" (INVALID-ARGUMENT-LIST-CALLEE
                                                                                    CONDITION)
                            (INVALID-ARGUMENT-LIST-ARGUMENT CONDITION)))))))))

(PUTPROPS CONDITION-HIERARCHY-IL FILETYPE CL:COMPILE-FILE)

(PUTPROPS CONDITION-HIERARCHY-IL COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1988 1990))
```

STRUCTURE INDEX

PROPERTY INDEX