```
(RPAQQ ERROR-RUNTIME-AFTER-FASLCOMS ((PROP FILETYPE ERROR-RUNTIME-AFTER-FASL)
```

;;; These functions must be PavCompiled.

```
                                    (FUNCTIONS SI::CREATE-CONDITION-CASE-LITERALS)
```

;;; These don't need to be, but depend on the previous stuff.

```
                                    (FUNCTIONS ABORT CONDITIONS:CONTINUE CONDITIONS:MUFFLE-WARNING USE-VALUE
                                          STORE-VALUE)))
(PUTPROPS ERROR-RUNTIME-AFTER-FASL FILETYPE :COMPILE-FILE)
```

;;; These functions must be PavCompiled.

```
(CL:DEFUN SI::CREATE-CONDITION-CASE-LITERALS (SI::TYPE-LIST)
   (LET ((SI::CATCH-TAG (LIST NIL)))                     ; use a separate CONS for the catch tag so that you don't get a
                                                         ; circular structure (you could save one by making the tag be the
                                                         ; handler binding itself, then smashing its CDR with the closure).
        (LIST* SI::CATCH-TAG SI::TYPE-LIST #'(CL:LAMBDA (SI::CONDITION)
                                               (CL:THROW SI::CATCH-TAG SI::CONDITION)))))
```

;;; These don't need to be, but depend on the previous stuff.

```
(DEFINE-PROCEED-FUNCTION ABORT :REPORT "Abort"
   (CONDITION XCL:*CURRENT-CONDITION*))

(CL:DEFUN CONDITIONS:CONTINUE ()
   (LET ((CONDITIONS::R (CONDITIONS:FIND-RESTART 'CONDITIONS:CONTINUE)))
        (CL:WHEN CONDITIONS::R (CONDITIONS:INVOKE-RESTART CONDITIONS::R))))

(CL:DEFUN CONDITIONS:MUFFLE-WARNING ()
   (LET ((CONDITIONS::R (CONDITIONS:FIND-RESTART 'CONDITIONS:MUFFLE-WARNING)))
        (CL:WHEN CONDITIONS::R (CONDITIONS:INVOKE-RESTART CONDITIONS::R))))

(DEFINE-PROCEED-FUNCTION USE-VALUE :REPORT "Use a different value"
   XCL::VALUE)

(DEFINE-PROCEED-FUNCTION STORE-VALUE :REPORT "Store a new value and use it"
   XCL::VALUE)

(PUTPROPS ERROR-RUNTIME-AFTER-FASL COPYRIGHT ("Venue & Xerox Corporation" 1987 1988 1990))
```

FUNCTION INDEX

PROPERTY INDEX