

File created: 5-May-2026 12:21:37 {MEDLEY}<library>IMPORTFONTS.;116

edit by: rmk

changes to: (FNS PEF)

previous date: 4-May-2026 15:26:51 {MEDLEY}<library>IMPORTFONTS.;115

Read Table: INTERLISP

Package: INTERLISP

Format: UTF-8

```
(RPAQQ IMPORTFONTS COMS
((FNS IMPORTFONTS FONT.TO.MCCS IMPORTFONTS.FONTSPEC IMPORTFONTS.CONTEXT IMPORTFONTS.NOCACHE
IMPORTFONTS.DIRECTORY IMPORTFONTS.CLEAR IMPORTFONTS.SUBDIR IMPORTFONTS.DIRSIZE)
(FNS IMPORTFONTS.AVAILABLE IMPORTFONTS.EXISTS?)
(FNS FAKEFACE FAKEFACE.FROMFILE FAKEFACE.FROMFONT)
(COMS
(FNS IMPORT.DISPLAY LEGACYDISPLAYFONT) ; For legacy display imports
(FILE ACFONT))
(COMS ; For testing
(FNS IPF IPFSIZES)
(FNS PEF AEF IEF MEF CEF FEF DEF EFCLOSE)
(FNS SHOWCHARS CSSOURCE FONTDEFFONTS)
(FILE EDITFONT))))
```

(DEFINEQ

IMPORTFONTS

[LAMBDA (PHASE FONTSPECS DEVICE FROMDIR TODIR IMPORTFN NODRIBBLE)

; Edited 4-May-2026 13:49 by rmk
; Edited 11-Apr-2026 10:55 by rmk
; Edited 5-Apr-2026 14:22 by rmk
; Edited 3-Apr-2026 08:15 by rmk
; Edited 1-Apr-2026 08:25 by rmk
; Edited 30-Mar-2026 16:41 by rmk
; Edited 29-Mar-2026 11:33 by rmk
; Edited 26-Mar-2026 12:47 by rmk
; Edited 24-Mar-2026 15:00 by rmk
; Edited 21-Mar-2026 11:16 by rmk
; Edited 4-Mar-2026 09:44 by rmk
; Edited 9-Oct-2025 15:56 by rmk

:: Device-dependent IMPORTFN must be provided if PHASE is IMPORT, the other phases (MCCS, COMPLETE, FAKE) operate on Medleyfont
:: files.

:: If PHASE is MCCS, recodes source FONTSPECS to MCCS without completion, otherwise coerces/completes the specified MCCS fonts.

:: Unless TODIRECTORY=DONT, writes the resulting fonts Medley fontfiles, otherwise collects and returns them in a list (which will eat up
:: storage).

```
(CL:UNLESS (MEMB (SETQ PHASE (U-CASE PHASE))
'(IMPORT MCCS COMPLETE FAKE DEPLOY))
(ILLEGAL.ARG PHASE))
```

```
(CL:UNLESS DEVICE
(SETQ DEVICE 'DISPLAY))
```

```
(CL:WHEN (AND (EQ PHASE 'IMPORT)
(NOT (GETD IMPORTFN)))
```

```
(ERROR "Importing from source requires an IMPORTFN"))
```

```
(PROG1 (RESETLST ; Close dribble outside of this context
```

```
(CL:MULTIPLE-VALUE-SETQ (FROMDIR TODIR)
(IMPORTFONTS.CONTEXT PHASE FROMDIR TODIR DEVICE))
```

```
(IMPORTFONTS.CLEAR PHASE FONTSPECS TODIR DEVICE)
```

```
(SETQ FONTSPECS (IMPORTFONTS.FONTSPEC PHASE FONTSPECS FROMDIR DEVICE))
```

```
(CL:WHEN (AND (IGEQ (LENGTH FONTSPECS)
```

```
5)
```

```
(NOT NODRIBBLE))
```

; Put all the dribbles together one up

```
[DRIBBLE (PSEUDOFILNAME (PACKFILENAME 'BODY (OR (SUBSTRING TODIR 1
(STRPOS ">" TODIR -2 NIL NIL
NIL NIL T))
```

```
LOGINHOST/DIR)
```

```
'NAME PHASE 'EXTENSION 'DRIBBLE]
```

```
(PRINTOUT T "Dribbling to " (FULLNAME (DRIBBLEFILE))
```

```
T))
```

```
(PRINTOUT T (SELECTQ PHASE
```

```
(IMPORT "Importing ")
```

```

(MCCS "MCCS recoding ")
(COMPLETE "Completing ")
(FAKE "Faking ")
(DEPLOY "Deploying ")
NIL)
(LENGTH FONTSPECS)
" " DEVICE " font" (CL:IF (IGEQ (LENGTH FONTSPECS)
2)
"s"
""))
(if TODIR
then (PRINTOUT T 3 "from " FROMDIR 3 "to " TODIR T)
else (PRINTOUT T " from " FROMDIR T 3 "(but not writing)" T)
(BKSYSBUF " "))
(IMPORTFONTS.CLEAR PHASE FONTSPECS TODIR DEVICE)
(for FS FONT FONT FONTSTART TOFILE FROMFILE CAPTION THISTIME FROMSIZE TOSIZE
CHANGED NOTINSTANTIATED TOTALTIME TODIRSIZE (FROMDIRSIZE ←
(CL:IF (EQ PHASE
'IMPORT)
0
(IMPORTFONTS.DIRSIZE
PHASE DEVICE
FROMDIR)))
(NNOCHARSETS ← 0) in FONTSPECS as I from 1 first (SETQ TOTALTIME (CLOCK 0))
eachtime (PRINTOUT T .I3 I " " (fetch (FONTSPEC FSFAMILY) of FS)
" " .I2 (fetch (FONTSPEC FSSIZE) of FS)
" ")
(FONTFACETOATOM (fetch (FONTSPEC FSFACE) of FS)))
(SETQ THISTIME (CLOCK 0))
collect (SELECTQ PHASE
(IMPORT (SETQ FONT (APPLY* IMPORTFN FS FROMDIR))
(SETQ CHANGED T))
((MCCS COMPLETE)
(SETQ FROMFILE (INFILEP (MEDLEYFONT.FILENAME FS FROMDIR)))
(if FROMFILE
then (SETQ FONT (MEDLEYFONT.READ.FONT FROMFILE 'ALL))
(SETQ FROMSIZE (GETFILEINFO FROMFILE 'LENGTH))
else (SETQ FONT (FONTCREATE.SLUGFD FS))
(SETQ FROMSIZE 0))
(SETQ CHANGED (SELECTQ PHASE
(MCCS (FONT.TO.MCCS FONT))
(COMPLETE (COMPLETE.FONT FONT T))
NIL)))
(FAKE (CL:MULTIPLE-VALUE-SETQ (FONT FROMFILE)
(FAKEFACE.FROMFONT FS FROMDIR))
(SETQ FROMSIZE (GETFILEINFO FROMFILE 'LENGTH))
(SETQ CHANGED (FAKEFACE FONT)))
(DEPLOY (SETQ FROMFILE (MEDLEYFONT.FILENAME FS FROMDIR))
(SETQ FROMSIZE (GETFILEINFO FROMFILE 'LENGTH)))
NIL)
(CL:WHEN [SETQ NOTINSTANTIATED (EQ 0 (FONTPROP FONT
'NINSTANTIATEDCHARSETS])
(add NNOCHARSETS 1))
(CL:WHEN TODIR
(SETQ TOFILE (MEDLEYFONT.FILENAME FS TODIR))
(SETQ TOFILE (CL:IF CHANGED
(MEDLEYFONT.WRITE.FONT FONT TOFILE)
(COPYFILE FROMFILE TOFILE))))
(SETQ THISTIME (FQUOTIENT (IDIFFERENCE (CLOCK 0)
THISTIME)
1000))
(if NOTINSTANTIATED
then (PRINTOUT T 26 "No instantiated charsets" 41 .F7.2 THISTIME)
elseif TOFILE
then (SETQ TOSIZE (GETFILEINFO TOFILE 'LENGTH))
(SELECTQ PHASE
(IMPORT (PRINTOUT T 25 .I6 TOSIZE " " .F7.2 THISTIME))
((MCCS COMPLETE FAKE DEPLOY)
(if CHANGED

```

```

then (PRINTOUT T 25 .I6 FROMSIZE " -> " .I6 TOSIZE " " .F6.2
      THISTIME 50 .I5 (IDIFFERENCE TOSIZE FROMSIZE))
else (PRINTOUT T 28 "Copied " .I6 TOSIZE " " .F6.2 THISTIME))
NIL))
(TERPRI T)
(CL:IF TODIR
  TOFILE
  FONT)
finally (SETQ TOTALTIME (FIXR (FQUOTIENT (IDIFFERENCE (CLOCK 0)
                                                    TOTALTIME)
                                      1000)))
(PRINTOUT T 4 (IDIFFERENCE (LENGTH $$VAL)
                          NNOCHARSETS)
  " font"
  (CL:IF (IGEQ (LENGTH $$VAL)
              2)
    "s "
    ""))
(SELECTQ PHASE
  (IMPORT "imported")
  (MCCS "recoded to MCCS")
  (COMPLETE "completed")
  (FAKE "faked")
  (DEPLOY "deployed")
  NIL))
(if TODIR
  then (SETQ TODIRSIZE (IMPORTFONTS.DIRSIZE PHASE DEVICE TODIR))
  (PRINTOUT T " and written in " TOTALTIME " seconds" 4 "Total size is "
    (FIXR (FQUOTIENT TODIRSIZE 1024))
    " KB, grew by "
    (FIXR (FQUOTIENT (IDIFFERENCE TODIRSIZE FROMDIRSIZE)
                    1024))
    " KB" T)
  (CL:UNLESS (EQ 0 NNOCHARSETS)
    (PRINTOUT T 4 NNOCHARSETS " fonts had no character sets" T))
  (CL:WHEN (AND TODIR (IGEQ (LENGTH $$VAL)
                            5))
    (SETQ $$VAL TODIR))
  else (PRINTOUT T " (but not written) in " TOTALTIME " seconds" T)))
(CL:WHEN (DRIBBLEFILE)
  (PRINTOUT T "Dribbled to " (FULLNAME (DRIBBLEFILE))
    T)
  [TEDIT (DRIBBLE)
    'Dribble NIL '(TITLE ,(CONCAT PHASE " font dribble" " " (DATE))
      LEAVETTY T READONLY QUIET PARABREAKCHARS NIL FONT
      DEFAULTFONT OPENWIDTH ,(fetch (REGION WIDTH) of (WINDOWPROP
                                                    (WFROMMS T)
                                                    'REGION))
      OPENHEIGHT
      ,(fetch (REGION HEIGHT) of (WINDOWPROP (WFROMMS T)
                                                    'REGION)))]])

```

(FONT.TO.MCCS
[LAMBDA (FONT)

- ; Edited 11-Apr-2026 15:43 by rmk
- ; Edited 10-Mar-2026 00:23 by rmk
- ; Edited 7-Mar-2026 12:55 by rmk
- ; Edited 1-Mar-2026 13:43 by rmk
- ; Edited 7-Oct-2025 17:13 by rmk
- ; Edited 6-Sep-2025 16:43 by rmk
- ; Edited 2-Sep-2025 15:20 by rmk
- ; Edited 29-Aug-2025 11:25 by rmk
- ; Edited 27-Aug-2025 17:36 by rmk

:: Move character information in font to their MCCS positions, with coercions otherwise suppressed.

:: If there are no mappings, prints a message and returns INFONT.

(LET [(PAIRS (MCCSMAPPINGS (FONTPROP FONT 'CHARENCODING)
(CL:WHEN PAIRS

```

(MOVEFONTCHARS PAIRS FONT FONT)
(replace (FONTDESCRIPTOR FONTCOMPLETEP) of FONT with NIL)
;; Keep the map function even for coerced MCCS fonts--can still be used for code conversion (e.g. Tedit file updating)
[replace (FONTDESCRIPTOR FONTTOMCCSFN) of FONT with (MCCSMAPFN (FONTPROP FONT
'CHARENCODING]
(CL:WHEN (MEMB (FONTPROP FONT 'CHARENCODING)
'(GACHA XCCS$ ALTOTEXT PALATINO UNICODE HIPPO CYRILLIC))
(replace (FONTDESCRIPTOR FONTCHARENCODING) of FONT with 'MCCS)
; These fonts made it all the way
(Charsetprop (\GETCHARSETINFO FONT 0)
'CSCHARENCODING
'MCCS))
FONT))

```

**(IMPORTFONTS.FONTSPecs
[LAMBDA (PHASE FONTSPecs FROMDIR DEVICE)**

```

; Edited 4-May-2026 08:59 by rmk
; Edited 4-Apr-2026 11:41 by rmk
; Edited 3-Apr-2026 00:51 by rmk
; Edited 1-Apr-2026 12:50 by rmk
; Edited 30-Mar-2026 23:27 by rmk
; Edited 23-Mar-2026 13:17 by rmk
; Edited 21-Mar-2026 08:58 by rmk
; Edited 14-Mar-2026 23:50 by rmk
; Edited 13-Mar-2026 10:53 by rmk
; Edited 4-Mar-2026 10:44 by rmk

```

;; Crucially, the fontspecs for COMPLETE and FAKEFACE must be ordered so that coercion or faces come before the fontspecs that depend on them. E.g. the order of faces has to be MRR MIR/BRR BIR. That means that the TODIR can be the source for the completion/faking of later fonts.

(CL:UNLESS DEVICE

(SETQ DEVICE 'DISPLAY))

(SETQ FROMDIR (IMPORTFONTS.DIRECTORY DEVICE FROMDIR PHASE))

(RESETLST

(IMPORTFONTS.NOCACHE)

[RESETSAVE (FONTDEVICEPROP DEVICE 'FONTDIRECTORIES FROMDIR)

'(PROGN (FONTDEVICEPROP 'DEVICE 'FONTDIRECTORIES OLDVALUE]

(LET [(EXPANDED (CL:REMOVE-DUPLICATES (for F inside (CL:IF (type? FONTSPEC FONTSPECS)

(CONS FONTSPECS)

(OR FONTSPECS '))**

join (if (type? FONTSPEC F)

then (for FACE in (FONTFACE.STARS (fetch (FONTSPEC F

FSFACE) of (\FONT.CHECKARGS F)))

collect (create FONTSPEC using F FSFACE ← FACE FSDEVICE ← DEVICE FSROTATION ← 0))

elseif (LITATOM F)

then ;; Looks in FROMDIR

(IMPORTFONTS.AVAILABLE PHASE (MAKEFONTSPEC (OR F ')**

**** 0 DEVICE)**

FROMDIR)

else (ILLEGAL.ARG F)))

:TEST (FUNCTION EQUAL]

(SELECTQ PHASE COMPLETE

;; Fonts have to be ordered so that a coercion target is done before the font that needs it (e.g. CLASSIC ;; before MODERN before TERMINAL before GACHA). This is essentially a topsort

(for FONTSPEC NEWFONTS in EXPANDED do (for F in (DREVERSE (CONS FONTSPEC (COERCEFONTSPEC

FONTSPEC

CHARCOERCIONS
T)))

unless (MEMBER F NEWFONTS)
when (IMPORTFONTS.EXISTS?
'COMPLETE F FROMDIR)
do (push NEWFONTS F))

finally ;; We may have requested a font that doesn't exist at all on its own: HELVETICA 10 MIR

(for F in EXPANDED unless (MEMBER F NEWFONTS) do (push NEWFONTS F)
)
(RETURN (DREVERSE NEWFONTS))))

(FAKE ;; If e.g. BIR is in the list, then so should be BMR, MIR, and if those are there, then so should MRR

(for ETAIL FS FACE SOURCE NEWFONTS on EXPANDED

do (SETQ FS (CAR ETAIL))

(SETQ FACE (fetch (FONTSPEC FSPACE) of FS))

(CL:WHEN (EQ 'BOLD (fetch (FONTFACE WEIGHT) of FACE))

; BRR -> MRR, BIR ->MIR which goes to MRR

(for FC SOURCE in (FONTFACE.STARS (create FONTFACE using FACE WEIGHT ←
'MEDIUM))

eachtime (SETQ SOURCE (create FONTSPEC using FS FSPACE ← FC))
when (IMPORTFONTS.EXISTS? 'FAKE SOURCE FROMDIR) do (push

NEWFONTS
SOURCE)))

(CL:WHEN (EQ 'ITALIC (fetch (FONTFACE SLOPE) of FACE))

; MIR -> MRR, BIR ->BMR

(for F SOURCE in (FONTFACE.STARS (create FONTFACE using FACE SLOPE ←
'REGULAR))

eachtime (SETQ SOURCE (create FONTSPEC using FS FSPACE ← F))
when (IMPORTFONTS.EXISTS? 'FAKE SOURCE FROMDIR) do (push

NEWFONTS
SOURCE)))

(CL:WHEN (EQ 'COMPRESSED (fetch (FONTFACE EXPANSION) of FACE))

; MRC -> MRR

(for F SOURCE in (FONTFACE.STARS (create FONTFACE using FACE EXPANSION
← 'REGULAR))

eachtime (SETQ SOURCE (create FONTSPEC using FS FSPACE ← F))
when (IMPORTFONTS.EXISTS? 'FAKE SOURCE FROMDIR) do (push

NEWFONTS
SOURCE)))

finally (for F in EXPANDED unless (MEMBER F NEWFONTS) do (push NEWFONTS F))
(RETURN (DREVERSE NEWFONTS))))

(DEPLOY ; Make sure we copy consistent versions of the indirects

(SORTFONTSPECS (CL:REMOVE-DUPLICATES (APPEND (for FS in EXPANDED

join (MEDLEYFONT.GETFILEPROP
(MEDLEYFONT.FILENAME FS
FROMDIR)

'INDIRECTS))

EXPANDED)

:TEST
(FUNCTION EQUAL))))

(SORTFONTSPECS EXPANDED))))))

(IMPORTFONTS.CONTEXT
[LAMBDA (PHASE FROMDIR TODIR DEVICE)

- ; Edited 4-May-2026 00:16 by rmk
- ; Edited 4-Apr-2026 09:33 by rmk
- ; Edited 24-Mar-2026 23:11 by rmk
- ; Edited 23-Mar-2026 13:17 by rmk
- ; Edited 21-Mar-2026 09:53 by rmk
- ; Edited 18-Mar-2026 22:36 by rmk
- ; Edited 16-Mar-2026 09:01 by rmk
- ; Edited 14-Mar-2026 23:56 by rmk
- ; Edited 13-Mar-2026 10:52 by rmk
- ; Edited 3-Mar-2026 21:58 by rmk

:: This sets up the FONTDEVICE props according to PHASE, using default values for the directories if they aren't specified. Returns the TODIR, or NIL if the TODIR was DONT=don't file.

```
(SETQ FROMDIR (IMPORTFONTS.DIRECTORY DEVICE FROMDIR (SELECTQ PHASE
  (IMPORT 'SOURCE)
  (MCCS 'IMPORT)
  (COMPLETE 'MCCS)
  (FAKE 'COMPLETE)
  (DEPLOY (for P in '(FAKE COMPLETE MCCS
    IMPORT)
    suchthat (DIRECTORYNAMEP
      (IMPORTFONTS.DIRECTORY
        DEVICE FROMDIR P))))
  )
  NIL)))
```

```
(SETQ TODIR (CL:IF (EQ PHASE 'DEPLOY)
  (CONCAT [CAR (MKLIST (FONTDEVICEPROP DEVICE 'FONTDIRECTORIES)
    ">")
    (IMPORTFONTS.DIRECTORY DEVICE TODIR PHASE)))
  (PROGN (FONTDEVICEPROP 'DEVICE 'FONTCOERCIONS OLDVALUE]
  (SELECTQ PHASE
    ((IMPORT MCCS)
      [RESETSAVE (FONTDEVICEPROP DEVICE 'FONTDIRECTORIES FROMDIR)
        '(PROGN (FONTDEVICEPROP ',DEVICE 'FONTDIRECTORIES OLDVALUE])
    (COMPLETE
      ;; Make FONTCREATE1 in the COMPLETE phase draw from previously import-completed fonts, not from the currently
      ;; active fonts
      [RESETSAVE (FONTDEVICEPROP DEVICE 'FONTDIRECTORIES (LIST TODIR FROMDIR))
        '(PROGN (FONTDEVICEPROP ',DEVICE 'FONTDIRECTORIES OLDVALUE]
        ; Suppress face faking
      [RESETSAVE (FONTDEVICEPROP DEVICE 'FACECOERCIONS NIL)
        '(PROGN (FONTDEVICEPROP ',DEVICE 'FACECOERCIONS OLDVALUE])
    (FAKE
      ;; Make FONTCREATE1 in the FAKE phase draw from previously import-completed fonts, not from the currently active fonts
      [RESETSAVE (FONTDEVICEPROP DEVICE 'FONTDIRECTORIES (LIST TODIR FROMDIR))
        '(PROGN (FONTDEVICEPROP ',DEVICE 'FONTDIRECTORIES OLDVALUE]
        ; Completion has already been done
      [RESETSAVE (FONTDEVICEPROP DEVICE 'FONTCOERCIONS NIL)
        '(PROGN (FONTDEVICEPROP ',DEVICE 'FONTCOERCIONS OLDVALUE])
    NIL)
  (IMPORTFONTS.NOCACHE)
  (CL:VALUES FROMDIR TODIR))
```

```
(IMPORTFONTS.NOCACHE
[LAMBDA NIL ; Edited 3-Mar-2026 11:54 by rmk
  (RESETSAVE \FONTSINCORE NIL)
  (RESETSAVE \FONTEXISTS?-CACHE NIL)
  (RESETSAVE \FONSAVAILABLEFILECACHE NIL])
```

```
(IMPORTFONTS.DIRECTORY
[LAMBDA (DEVICE DIRECTORY PHASE) ; Edited 4-May-2026 00:52 by rmk
  ; Edited 5-Apr-2026 14:17 by rmk
  ; Edited 22-Mar-2026 22:59 by rmk
  ; Edited 21-Mar-2026 22:40 by rmk
  ; Edited 12-Mar-2026 09:35 by rmk
  ; Edited 9-Mar-2026 09:14 by rmk
  ; Edited 4-Mar-2026 00:46 by rmk
```

```
(CL:UNLESS (MEMB DIRECTORY '(DONT DON'T))
  [PSEUDOFILENAME (OR DIRECTORY (SELECTQ PHASE
    (DEPLOY (CONCAT [CAR (MKLIST (FONTDEVICEPROP DEVICE
      'FONTDIRECTORIES)
        ">")
        (CONCAT "{MEDLEY}<internal>fonts>"
          (L-CASE (CONCAT (OR (if (type? FONTDESCRIPTOR DEVICE)
            then (FONTPROP DEVICE 'DEVICE)
            elseif (type? FONTSPEC DEVICE)
```

```

then (fetch (FONTSPEC FSDEVICE
)
of (\FONT.CHECKARGS
DEVICE))
elseif (NOT (LITATOM DEVICE))
then (\ILLEGAL.ARG DEVICE)
elseif DEVICE
else 'DISPLAY))
">"
(CL:IF PHASE
(CONCAT (IMPORTFONTS.SUBDIR PHASE)
">")
"")))]))

```

**(IMPORTFONTS.CLEAR
[LAMBDA (PHASE FONTSPESCS TODIR DEVICE)**

; Edited 3-May-2026 22:40 by rmk
; Edited 4-Apr-2026 10:03 by rmk
; Edited 3-Apr-2026 01:06 by rmk
; Edited 30-Mar-2026 12:37 by rmk
; Edited 23-Mar-2026 13:18 by rmk
; Edited 14-Mar-2026 22:51 by rmk
; Edited 12-Mar-2026 09:31 by rmk

(CL:WHEN [AND TODIR (MEMB PHASE '(COMPLETE FAKE)

; Previous completions could serve as inputs for later ones, start
; fresh.

```

(LET (NDELETED)
[SETQ NDELETED (if FONTSPESCS
then (for FS in (CL:IF (type? FONTSPEC FONTSPESCS)
(CONS FONTSPESCS)
FONTSPESCS)
sum (for FILE in (FILDIR (PACKFILENAME 'VERSION '* 'BODY (
MEDLEYFONT.FILENAME
FS TODIR)))
count (DELFILE FILE)))
else (for FILE in [FILDIR (PACKFILENAME 'DIRECTORY TODIR 'NAME '* 'VERSION
'*
'EXTENSION
(CAR (MKLIST (FONTDEVICEPROP DEVICE
'FONTEXTENSIONS])
count (DELFILE FILE])
(CL:UNLESS (EQ 0 NDELETED)
(PRINTOUT T (SELECTQ NDELETED
(1 "1 font")
(CONCAT NDELETED " fonts"))
" deleted from "
(PSEUDOFILNAME TODIR
T)))))))]

```

**(IMPORTFONTS.SUBDIR
[LAMBDA (PHASE)**

; Edited 4-May-2026 00:27 by rmk
; Edited 30-Mar-2026 16:27 by rmk
; Edited 23-Mar-2026 13:18 by rmk
; Edited 21-Mar-2026 09:53 by rmk

```

(SELECTQ PHASE
(SOURCE "source")
(IMPORT "imported")
(MCCS "mccs")
(COMPLETE "completed")
(FAKE "faked")
(DEPLOY)
(ILLEGAL.ARG])

```

**(IMPORTFONTS.DIRSIZE
[LAMBDA (PHASE DEVICE DIRECTORY)**

; Edited 4-May-2026 00:26 by rmk
; Edited 30-Mar-2026 16:30 by rmk

```

(DIRECTORY (PACKFILENAME 'DIRECTORY (IMPORTFONTS.DIRECTORY DEVICE DIRECTORY PHASE)
'NAME

```

```

      '*
      'EXTENSION
      [CAR (MKLIST (FONTDEVICEPROP (OR DEVICE 'DISPLAY)
      'FONTEXTENSIONS]
      'VERSION "")
      'COUNTLENGTH])

```

)

(DEFINEQ

```

(IMPORTFONTS.AVAILABLE
[LAMBDA (PHASE FONTSPEC FROMDIR)
; Edited 4-May-2026 00:00 by rmk
; Edited 22-Mar-2026 13:36 by rmk

```

```

  (RESETLST
    [RESETSAVE (FONTDEVICEPROP FONTSPEC 'FONTDIRECTORIES (IMPORTFONTS.DIRECTORY FONTSPEC
      FROMDIR PHASE))
      '(PROGN (FONTDEVICEPROP ',FONTSPEC 'FONTDIRECTORIES OLDVALUE]
    (IMPORTFONTS.NOCACHE)
    (FONTSAVAILABLE FONTSPEC NIL NIL NIL NIL 'ONLY)))])

```

```

(IMPORTFONTS.EXISTS?
[LAMBDA (PHASE FONTSPEC FROMDIR)
; Edited 4-May-2026 00:00 by rmk
; Edited 22-Mar-2026 14:55 by rmk

```

```

  (RESETLST
    [RESETSAVE (FONTDEVICEPROP FONTSPEC 'FONTDIRECTORIES (IMPORTFONTS.DIRECTORY FONTSPEC
      FROMDIR PHASE))
      '(PROGN (FONTDEVICEPROP ',FONTSPEC 'FONTDIRECTORIES OLDVALUE]
    (IMPORTFONTS.NOCACHE)
    (FONTEXISTS? FONTSPEC NIL NIL NIL NIL T)))])

```

)

(DEFINEQ

```

(FAKEFACE
[LAMBDA (FONT)
; Edited 5-Apr-2026 14:10 by rmk
; Edited 4-Apr-2026 09:38 by rmk
; Edited 1-Apr-2026 09:00 by rmk
; Edited 24-Mar-2026 22:14 by rmk
; Edited 21-Mar-2026 14:37 by rmk
; Edited 14-Mar-2026 23:53 by rmk

```

```

;; FONTSPEC describes a font that has already been completed in terms of its CHARCOERCIONS.
;; Suppose FONTSPEC describes a (display) BRR font. If it has not been completed (i.e., it is still just MCCS), then all of the glyphs in all of its
;; character sets are natively bold.
;; If it has been completed without facefaking, then some of its characters may have been retrieved from other fonts, and those are mixed in with
;; its native glyphs. But presumably, those characters are also drawn from a BRR font, on the assumption that CHARCOERCIONS tends to
;; preserve faces.
;; So: it is correct to run through all of the slugs, grab the corresponding glyphs from the already completed MRR, if it exists, and insert the
;; bolded versions of those characters.
;;

```

```

(LET* ((FONTSPEC (FONTPROP FONT 'DEVICESPEC))
      (FACE (fetch (FONTSPEC FSDFACE) of FONTSPEC))
      SOURCEFONT CHANGED)
  (CL:UNLESS (EQ 'DISPLAY (FONTPROP FONT 'DEVICE))
    (ERROR "Only display fonts can be face-faked" FONT))
  (CL:UNLESS (fetch (FONTDESCRIPTOR FONTCOMPLETEP) of FONT)
    (ERROR "Fonts must be completed before face-faking" FONTSPEC))
  (CL:WHEN [AND (EQ 'BOLD (fetch (FONTFACE WEIGHT) of FACE))
    (SETQ SOURCEFONT (FONTCREATE1 (create FONTSPEC using FONTSPEC FSWEIGHT ←
      'MEDIUM]
      (for CHARSET from 0 to (MAXCHARSET FONT) when (FAKEFACE.CHARSET FONT CHARSET
        (FUNCTION MAKEBOLD.CHAR)
        SOURCEFONT)
      do (SETQ CHANGED T)))])

```

```

(CL:WHEN [AND (EQ 'ITALIC (fetch (FONTFACE SLOPE) of FACE))
              (SETQ SOURCEFONT (FONTCREATE1 (create FONTSPEC using FONTSPEC FSSLOPE ←
              'REGULAR]
              (for CHARSET from 0 to (MAXCHARSET FONT) when (FAKEFACE.CHARSET FONT CHARSET
              (FUNCTION MAKEITALIC.CHAR)
              SOURCEFONT)
              do (SETQ CHANGED T)))
(CL:WHEN [AND (EQ 'COMPRESSED (fetch (FONTFACE EXPANSION) of FACE))
              (SETQ SOURCEFONT (FONTCREATE1 (create FONTSPEC using FONTSPEC FSEXPANSION
              ← 'REGULAR]
              (for CHARSET from 0 to (MAXCHARSET FONT) when (FAKEFACE.CHARSET FONT CHARSET
              (FUNCTION MOVEFONTCHARS)
              SOURCEFONT)
              do (SETQ CHANGED T)))
CHANGED])

```

```

(FAKEFACE.FROMFILE
[LAMBDA (FONTSPEC FROMDIR)

```

```

; Edited 3-May-2026 22:37 by rmk
; Edited 4-Apr-2026 09:42 by rmk
; Edited 30-Mar-2026 23:17 by rmk
; Edited 21-Mar-2026 09:03 by rmk
; Edited 19-Mar-2026 11:53 by rmk
; Start with MRR for a face that doesn't yet exist (HELVETICA
; MIR).

```

```

(OR (INFILEP (MEDLEYFONT.FILENAME FONTSPEC FROMDIR))
    (LET ((FACE (fetch (FONTSPEC FSPACE) of FONTSPEC)))
        (OR (AND (EQ 'BOLD (fetch (FONTFACE WEIGHT) of FACE))
                (INFILEP (MEDLEYFONT.FILENAME (create FONTSPEC using FONTSPEC FSPACE ← (
                MAKEFONTFACE
                'MEDIUM NIL
                NIL FACE))
                FROMDIR)))
            (AND (EQ 'ITALIC (fetch (FONTFACE SLOPE) of FACE))
                (FAKEFACE.FROMFILE (create FONTSPEC using FONTSPEC FSPACE ← (MAKEFONTFACE NIL
                'REGULAR NIL
                FACE))
                FROMDIR)))
            (AND (EQ 'COMPRESSED (fetch (FONTFACE EXPANSION) of FACE))
                (FAKEFACE.FROMFILE (create FONTSPEC using FONTSPEC FSPACE ← (MAKEFONTFACE NIL NIL
                'REGULAR
                FACE))
                FROMDIR)))
        (ERROR "No source for face-faking" FONTSPEC])

```

```

(FAKEFACE.FROMFONT
[LAMBDA (FONTSPEC FROMDIR)

```

```

; Edited 3-May-2026 22:37 by rmk
; Edited 19-Mar-2026 20:42 by rmk

```

;; If FONTSPEC doesn't identify an existing font that needs to be coerced, we create an empty starting font full of slug charsets.

```

(LET (FROMFILE FONT)
    (SETQ FROMFILE (FAKEFACE.FROMFILE FONTSPEC FROMDIR))
    (SETQ FONT (MEDLEYFONT.READ.FONT FROMFILE 'ALL))
    (CL:UNLESS (EQUAL FONTSPEC (FONTPROP FONT 'SPEC))
        ;; We catch all the other properties of the backing font--encoding etc.
        (SETQ FONT (FONTCREATE.SLUGFD FONTSPEC FONT)))
    (CL:VALUES FONT FROMFILE)
)

```

;; For legacy display imports

```

(DEFINEQ

```

```

(IMPORT.DISPLAY
[LAMBDA (FONTSPECS)

```

```

; Edited 4-May-2026 15:18 by rmk
; Edited 28-Mar-2026 23:08 by rmk

```

; Edited 18-Mar-2026 23:54 by rmk

(IMPORTFONTS 'IMPORT FONTSPECS 'DISPLAY '{MEDLEY}/fonts/displayfonts/ NIL
(FUNCTION LEGACYDISPLAYFONT])

(LEGACYDISPLAYFONT
[LAMBDA (FONTSPEC FROMDIR)

; Edited 3-May-2026 22:38 by rmk
; Edited 16-Apr-2026 22:37 by rmk
; Edited 12-Apr-2026 13:22 by rmk
; Edited 31-Mar-2026 15:01 by rmk
; Edited 28-Mar-2026 09:27 by rmk

;; Loads legacy display fonts (ac or strike format, gacha, terminal, helevetica...) from FROMDIR. If NIL, the current directory

(SETQ FONTSPEC (\FONT.CHECKARGS FONTSPEC NIL NIL NIL 'DISPLAY T))
(RESETLST

[RESETSAVE (FONTDEVICEPROP 'DISPLAY 'FONTDIRECTORIES FROMDIR)
'(PROGN (FONTDEVICEPROP 'DISPLAY 'FONTDIRECTORIES OLDVALUE]
[RESETSAVE (FONTDEVICEPROP 'DISPLAY 'FONTEXTENSIONS '(DISPLAYFONT)
FROMDIR)

'(PROGN (FONTDEVICEPROP 'DISPLAY 'FONTEXTENSIONS OLDVALUE]
(for CSNO CSINFO (FONT ← (FONTSPEC.TO.FONTDESCRIPTOR FONTSPEC 255)) from 0 to 255
do (SETQ CSINFO (\READCHARSET FONT CSNO '((AC ACFONT.FILEP ACFONT.GETCHARSET)
(STRIKE STRIKEFONT.FILEP
STRIKEFONT.GETCHARSET))
FROMDIR))

;; NIL means empty

(CL:IF CSINFO
(INSTALLCHARSETINFO FONT CSINFO CSNO)
(\SETCHARSETINFO FONT CSNO (SLUGCSINFO FONT)))
finally (RETURN FONT))))

)

(FILESLOAD ACFONT)

;; For testing

(DEFINEQ

(IPF
[LAMBDA (PHASE FONTSPEC)

; Edited 4-May-2026 00:01 by rmk
; Edited 30-Mar-2026 12:44 by rmk
; Edited 21-Mar-2026 22:46 by rmk
; Edited 10-Mar-2026 00:54 by rmk

(MEDLEYFONT.READ.FONT (MEDLEYFONT.FILENAME FONTSPEC (IMPORTFONTS.DIRECTORY (OR (fetch (FONTSPEC
FSDEVICE
)
of FONTSPEC)
'DISPLAY)
NIL PHASE))
'ALL])

(IPFSIZES
[LAMBDA (FONTSPEC)

; Edited 4-May-2026 00:49 by rmk
; Edited 30-Mar-2026 12:45 by rmk
; Edited 23-Mar-2026 13:18 by rmk
; Edited 21-Mar-2026 00:59 by rmk
; Edited 16-Mar-2026 08:43 by rmk
; Edited 13-Mar-2026 10:33 by rmk

;; Returns the file sizes for all the phases of FONTSPEC

(if (type? FONTSPEC FONTSPEC)
then (SETQ FONTSPEC (\FONT.CHECKARGS FONTSPEC NIL NIL NIL NIL T))
else (SETQ FONTSPEC (FONTSPECFROMFILENAME FONTSPEC)))
(for PHASE DIR FONTFILE SIZE LASTSIZE in '(IMPORT MCCS COMPLETE FAKE DEPLOY)
eachtime (SETQ DIR (IMPORTFONTS.DIRECTORY FONTSPEC NIL PHASE)) when (SETQ FONTFILE (INFILEP (
MEDLEYFONT.FILENAME

FONTSPEC DIR)))

```
collect (SETQ FONTFILE (INFILEP (MEDLEYFONT.FILENAME FONTSPEC DIR)))
  [SETQ SIZE (CL:IF FONTFILE
    (GETFILEINFO FONTFILE 'LENGTH))]
  (PROG1 [LIST* PHASE SIZE (CL:IF LASTSIZE
    (CONS (IDIFFERENCE SIZE LASTSIZE)))]
    (SETQ LASTSIZE SIZE])
```

)

(DEFINEQ

(PEF

[LAMBDA (PHASES FONTSPEC CHARSET)

```
; Edited 5-May-2026 12:20 by rmk
; Edited 4-May-2026 08:27 by rmk
; Edited 12-Apr-2026 19:32 by rmk
; Edited 30-Mar-2026 09:14 by rmk
; Edited 25-Mar-2026 00:11 by rmk
; Edited 22-Mar-2026 00:19 by rmk
; Edited 16-Mar-2026 08:43 by rmk
; Edited 13-Mar-2026 10:33 by rmk
```

```
(if (type? FONTSPEC FONTSPEC)
  then (SETQ FONTSPEC (FONT.CHECKARGS FONTSPEC NIL NIL NIL NIL T))
  else (SETQ FONTSPEC (FONTSPECFROMFILENAME FONTSPEC)))
```

```
(for PHASE TITLETAG CHARSETNAME DIR FONTFILE inside PHASES
```

```
do ; Some phases may not have some charsets
```

```
(SETQ CHARSET (if (EQ CHARSET T)
  then (for C in (MEDLEYFONT.GETFILEPROP FONTFILE 'CHARSET) unless (OR (
    KANJICHARSETP
    C)
    (
    UNIHANCHARSETP
    C))
```

```
collect C)
elseif (CHARSET.DECODE CHARSET)
else 0))
```

```
(if (LISTP CHARSET)
  then (for C in CHARSET do (PEF PHASE FONTSPEC C))
  elseif (EQ PHASE 'DEPLOY)
  then (EDITFONT FONTSPEC CHARSET NIL NIL "Deployed")
  else (SETQ DIR (IMPORTFONTS.DIRECTORY FONTSPEC NIL PHASE))
    (SETQ FONTFILE (MEDLEYFONT.FILENAME FONTSPEC DIR))
    (if (NLSETQ (MEDLEYFONT.FILEP FONTFILE))
      then (SETQ TITLETAG (CL:IF (EQ PHASE 'MCCS)
        'MCCS
        (L-CASE PHASE T)))
        (SETQ CHARSETNAME (CHARSET.ENCODE CHARSET))
        (CL:UNLESS (OCTALNUM? CHARSETNAME)
          (SETQ TITLETAG (CONCAT CHARSETNAME " " TITLETAG)))
        (if [CAR (NLSETQ (RESETLST
          (IMPORTFONTS.CONTEXT PHASE DIR NIL (fetch (FONTSPEC FSDEVICE)
            of FONTSPEC))
          (EDITFONT (MEDLEYFONT.READ.FONT FONTFILE CHARSET)
            CHARSET NIL NIL TITLETAG)))]
          else (PRINTOUT T "Charset " (OCTALSTRING CHARSET)
            " of " FONTSPEC " not found in " DIR T)
            NIL)
        else (PRINTOUT T FONTSPEC " not found in " DIR T])
```

(AEF

[LAMBDA (FONTSPEC CHARSET CLOSE)

```
; Edited 4-May-2026 08:22 by rmk
; Edited 5-Apr-2026 11:50 by rmk
; Edited 21-Mar-2026 15:09 by rmk
; Edited 14-Mar-2026 12:59 by rmk
; Edited 12-Mar-2026 22:56 by rmk
```

:: Show CHARSET for all phases of FONTSPEC

(CL:WHEN CLOSE (EFCLOSE))
(PEF 'IMPORT MCCS COMPLETE FAKE DEPLOY)
 FONTSPEC CHARSET])

(IEF
 [LAMBDA (FONTSPEC CHARSET)

; Edited 4-May-2026 08:34 by rmk
; Edited 12-Mar-2026 11:38 by rmk
; Edited 10-Mar-2026 01:02 by rmk
; Edited 4-Mar-2026 00:31 by rmk
; Edited 1-Mar-2026 23:56 by rmk
; Edited 27-Feb-2026 14:24 by rmk

(PEF 'IMPORT FONTSPEC CHARSET])

(MEF
 [LAMBDA (FONTSPEC CHARSET)

; Edited 4-May-2026 08:35 by rmk
; Edited 12-Mar-2026 11:38 by rmk
; Edited 10-Mar-2026 01:01 by rmk
; Edited 4-Mar-2026 00:31 by rmk
; Edited 1-Mar-2026 23:57 by rmk
; Edited 9-Oct-2025 20:38 by rmk

(PEF 'MCCS FONTSPEC CHARSET])

(CEF
 [LAMBDA (FONTSPEC CHARSET)

; Edited 4-May-2026 08:35 by rmk
; Edited 12-Mar-2026 11:38 by rmk
; Edited 10-Mar-2026 01:03 by rmk
; Edited 4-Mar-2026 00:31 by rmk
; Edited 2-Mar-2026 00:00 by rmk
; Edited 9-Oct-2025 22:58 by rmk

(PEF 'COMPLETE FONTSPEC CHARSET])

(FEF
 [LAMBDA (FONTSPEC CHARSET)

; Edited 4-May-2026 08:35 by rmk
; Edited 23-Mar-2026 13:18 by rmk
; Edited 19-Mar-2026 00:42 by rmk
; Edited 12-Mar-2026 11:38 by rmk
; Edited 10-Mar-2026 01:03 by rmk
; Edited 4-Mar-2026 00:31 by rmk
; Edited 2-Mar-2026 00:00 by rmk
; Edited 9-Oct-2025 22:58 by rmk

(PEF 'FAKE FONTSPEC CHARSET])

(DEF
 [LAMBDA (FONTSPEC CHARSET)

; Edited 4-May-2026 08:34 by rmk
; Edited 12-Mar-2026 11:38 by rmk
; Edited 10-Mar-2026 01:02 by rmk
; Edited 4-Mar-2026 00:31 by rmk
; Edited 1-Mar-2026 23:56 by rmk
; Edited 27-Feb-2026 14:24 by rmk

(PEF 'DEPLOY FONTSPEC CHARSET])

(EFCLOSE
 [LAMBDA NIL
 (for W in (OPENWINDOWS) when (AND (WINDOWPROP W 'FONT)
 (WINDOWPROP W 'CHARITEMS))
 do (CLOSEW W])

; Edited 9-Oct-2025 19:48 by rmk

)

(DEFINEQ

(SHOWCHARS

[LAMBDA (FONT FROMCHAR TOCHAR ONELINE)

; Edited 5-Oct-2025 17:41 by rmk
; Edited 7-Sep-2025 20:29 by rmk
; Edited 2-Sep-2025 10:26 by rmk
; Edited 24-Jul-2025 11:30 by rmk
; Edited 8-Jun-2025 20:05 by rmk
; Edited 26-Jan-2024 14:18 by mth
; Edited 1-Aug-2020 09:27 by rmk:

```
[SETQ FONT (FONTCREATE (OR FONT '(CLASSIC 12]
(RESETLST
  [LET ((OLDFONT (DSPFONT NIL T))
        CHARS)
        (CL:UNLESS (CHARCODEP FROMCHAR)
          (SETQ FROMCHAR (OR (CHARCODE.DECODE FROMCHAR T)
                             FROMCHAR)))
        (SETQ CHARS (if (LISTP FROMCHAR)
                        elseif (CHARCODEP FROMCHAR)
                            then (CL:UNLESS (CHARCODEP TOCHAR)
                                             (SETQ TOCHAR (OR (CHARCODE.DECODE TOCHAR)
                                                             FROMCHAR)))
                            (for C from FROMCHAR to TOCHAR collect C)
                        else (CHCON FROMCHAR)))
        /RESETSAVE OLDFONT '(PROGN (DSPFONT OLDVALUE)
        (TERPRI)
        (for C in CHARS do (PRINTOUT T .FONT OLDFONT (CONCAT (OCTALSTRING (\CHARSET C))
                                                                ""
                                                                (OCTALSTRING (\CHAR8CODE C)))
                          10 .FONT FONT (CHARACTER C))
                          (CL:UNLESS ONELINE (PRINTOUT T T))
        (TERPRI))
```

(CSSOURCE
[LAMBDA (PHASE FONT INDIRECT)

; Edited 22-Mar-2026 00:26 by rmk

:: Shows the source properties of all the charsets

```
(CL:UNLESS (FONTP FONT)
  (SETQ FONT (IPF PHASE FONT)))
(LIST* PHASE (FONTPROP FONT 'SPEC)
  (for CS CSINFO SOURCE (FONTSPEC ← (FONTPROP FONT 'SPEC)) from 0 to (MAXCHARSET FONT)
    everytime (SETQ CSINFO (\GETCHARSETINFO FONT CS)) when CSINFO unless (fetch (CHARSETINFO
                                                                                      CSSLUGP)
                                                                                      of CSINFO)
              unless [AND INDIRECT (EQUAL FONTSPEC (CHARSETPROP CSINFO 'SOURCE)]
              collect (LIST CS (CHARSETPROP CSINFO 'SOURCE])
```

(FONTDEFFONTS
[LAMBDA (FACES)

; Edited 22-Mar-2026 12:48 by rmk

```
(SETQ FACES (for F inside FACES collect (\FONTFACE F)))
(for FD FONTLIST in FONTDEFS do (for FP in (CDR (ASSOC 'FONTPROFILE FD))
  do (for FONT in (CDDR FP) when FONT unless (MEMB (CAR FONT)
                                                    '(PDF POSTSCRIPT))
      when (OR (NOT FACES)
              (MEMBER (FONTFACE (fetch (FONTSPEC FSPACE) of FONT)
                                FACES))
            )
      do (push FONTLIST FONT)))
  finally (RETURN (SORTFONTSPECS (CL:REMOVE-DUPLICATES FONTLIST :TEST (FUNCTION EQUAL])
```

)

(FILESLOAD EDITFONT)

FUNCTION INDEX

AEF	11	FONTDEFFONTS	13	IMPORTFONTS.FONT SPECS	4
CEF	12	IEF	12	IMPORTFONTS.NOCACHE	6
CSSOURCE	13	IMPORT.DISPLAY	9	IMPORTFONTS.SUBDIR	7
DEF	12	IMPORTFONTS	1	IPF	10
EFCLOSE	12	IMPORTFONTS.AVAILABLE	8	IPFSIZES	10
FAKEFACE	8	IMPORTFONTS.CLEAR	7	LEGACYDISPLAYFONT	10
FAKEFACE.FROMFILE	9	IMPORTFONTS.CONTEXT	5	MEF	12
FAKEFACE.FROMFONT	9	IMPORTFONTS.DIRECTORY	6	PEF	11
FEF	12	IMPORTFONTS.DIRSIZE	7	SHOWCHARS	12
FONT.TO.MCCS	3	IMPORTFONTS.EXISTS?	8		
