


```

000000  TTTTTTTTTT  SSSSSSSS  LL      NN      NN  KK      KK
000000  TTTTTTTTTT  SSSSSSSS  LL      NN      NN  KK      KK
00      00      TT      SS      NN      NN  KK      KK
00      00      TT      SS      NN      NN  KK      KK
00      00      TT      SS      NNNN     NN  KK      KK
00      00      TT      SS      NNNN     NN  KK      KK
00      00      TT      SSSSSS  LL      NN  NN  KKKKKK
00      00      TT      SSSSSS  LL      NN  NN  KKKKKK
00      00      TT      SS      LL      NN  NNNN  KK      KK
00      00      TT      SS      LL      NN  NN  KK      KK
00      00      TT      SS      LL      NN  NN  KK      KK
00      00      TT      SS      LL      NN  NN  KK      KK
000000  TT      SSSSSSSS  LLLLLLLLLL  NN      NN  KK      KK
000000  TT      SSSSSSSS  LLLLLLLLLL  NN      NN  KK      KK

```

```

RRRRRRR  EEEEEEEEE  QQQQQQ
RRRRRRR  EEEEEEEEE  QQQQQQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RRRRRRR  EEEEEEEE  QQ      QQ
RRRRRRR  EEEEEEEE  QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EEEEEEEEE  QQQQ  QQ
RR      RR  EEEEEEEEE  QQQQ  QQ

```

01
%E
MA
MA

File: OTSLNK.REQ Edit: PLL1035

This file, OTSLNK.REQ, contains the definitions of all LINKAGE declarations for BLISS modules

```
*****
*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

```
Author: T. Hastings
1-11 - Add CALL RO. TNH 29-July-78
1-12 - Add JSB_CB_GET. TNH 2-Aug-78
1-13 - Change name to FORLNK.REQ. JBS 14-NOV-78
1-014 - Add copyright notice. JBS 16-NOV-78
1-015 - Change file name to OTSLNK.REQ. JBS 06-DEC-78
1-016 - Add linkage for BMF (Basic major frame ptr) - R11. DGP 17-Dec-78
1-017 - Fix some comments. JBS 18-DEC-78
1-018 - Change JSB_REC0 linkage to save registers needed for CH$FILL. DGP
27-Feb-79
1-019 - Similarly, change JSB_UDF0, since the UDF routines must
preserve the same registers to call the REC routines.
JBS 28-FEB-1979
1-020 - That change causes a similar change in JSB_REC1 and JSB_REC9.
JBS 28-FEB-1979
1-021 - Which in turn causes the same changes in JSB_UDF9.
JBS 28-FEB-1979
1-022 - Which in turn causes the same changes in JSB_DO_READ and
JSB_DO_WRITE. JBS 28-FEB-1979
1-023 - Add linkage JSB_CCB_A1 A0 for PUT relative with count. DGP 02-Mar-79
1-024 - Add linkage JSB_REC_IND for indexed file support. DGP 03-Apr-79
1-025 - Change linkage JSB_REC_IND to take 5 args. DGP 06-Apr-79
1-026 - Add linkage for conversion kernel routine. DGP 27-Jun-79
1-028 - Add linkage for Basic format routines. DGP 30-Jul-79
1-029 - Remove PRINT statement, for new BLISS compiler. JBS 02-OCT-1979
1-030 - Change JSB_FORMAT_A7 to A10. DGP 31-Oct-79
```

```

1-031 - Add new Linkage JSB_REC_WSL1. DGP 06-Nov-79
1-032 - Change JSB_UDFO, JSB_FMT0 so that they pass no arguments. SBL 5-Dec-1979
1-033 - Add linkage JSB_FMT1 for Fortran format interpreter. JAW
        08-Aug-1981
1-034 - Change JSB_A5_R11 to JSB_A6_R11, JSB_A10_R11 to JSB_A11_R11.
        PLL 16-Mar-1982
1-035 - Change JSB_DO_READ, JSB_REC_IND, JSB_REC0. PLL 1-Jun-1982

```

```

++
Linkage definitions for BLISS modules for CALL and JSB routines
The idea is to have all definitions here in one place
so that they can be changed easily and the entire
RTL recompiled.

```

```

+
Define symbols for register numbers used to pass parameters from one
module to another. Note: these symbols are used in the modules in
GLOBAL REGISTER declarations rather than below in this REQUIRE file
(where all registers appear as absolute numbers).

```

LITERAL

```

K_BMF_REG = 11,      ! Register used by Basic compiler to point
                    ! to last major frame
K_CCB_REG = 11;     ! Pointer to LUB/ISB/RAB

```

```

+
First define some macros for frequently used combinations.
Do not change the PRESERVE conventions for our sanity.
Also do not change the definitions of these combination, since
they are also used in defining local routines within a module that is CALLED!!!
NOTE: Local routines which are JSBed to from JSB procedures must
have LINKAGE definitions here even though only local procedures.
Otherwise, lose control of NOTUSED registers which must be same
or more inclusive for JSB routines called by JSB routines.
See DO_READ and DO_WRITE for examples.

```

MACRO

```

!+
! CALL interface with CCB passed in R11 (in and/or out)
!-

```

```

CALL_CCB_R11 =
  CALL: GLOBAL(CCB=11) %,

```

```

!+
! JSB interface with CCB passed in R11, 1 arg in R0 and 1 arg in R1 and
! only uses R0 and R1
!-

```

```

JSB_CCB_A1_A0 =
  JSB(REGISTER = 1, REGISTER = 0): GLOBAL(CCB=11) NOTUSED(2,3,4,5,6,7,8,9,10) %,

```

!+
! CALL interface with BMF passed in R11 (in and/or out)
!-
!

CALL_BMF_R11 =
CALL: GLOBAL(BMF=11) %,

!+
! CALL interface with 1st arg in R0
!-
!

CALL_A0 =
CALL (REGISTER = 0): %,

!+
! JSB interface with CCB passed in R11, no args and uses R0-R5
!-
!

JSB_CCB_R5 =
JSB: GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %,

!+
! JSB interface with CCB passed in R11, no args and only uses R0, R1
!-
!

JSB_CCB_NO_ARGS =
JSB: GLOBAL(CCB=11) NOTUSED (2,3,4,5,6,7,8,9,10) %,

!+
! JSB interface with CCB passed in R11, 1 arg in R0, and only uses R0, R1
!-
!

JSB_CCB_A0 =
JSB (REGISTER = 0): GLOBAL(CCB=11) NOTUSED (2,3,4,5,6,7,8,9,10) %,

!+
! JSB interface with CCB passed in R11, 1 arg in R0, and preserves
! through R5. Needed for MOVCS or calling routines that use MOVCS.
!-
!

JSB_CCB_A0_R5 =
JSB (REGISTER = 0): GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %,

!+
! Same as above, but with 2 arguments.
!-
!

JSB_CCB_A1_R5 =
JSB (REGISTER = 0, REGISTER = 1): GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %,

!+
! JSB interface with CCB passed in R11, 1 arg in R2, and only uses R0, R1, R2
! Needed when input arg is referenced after a CALL or JSB,
! so do not need to copy to R2.
!-
!

!-

```
JSB_CCB_A2 =
  JSB (REGISTER = 2): GLOBAL(CCB=11) NOTUSED (3,4,5,6,7,8,9,10) %.
```

!+

```
JSB interface with CCB passed in R11, 1 arg in R2, and preserves
through R5. Needed in place of JBS_CCB_A2 to do MOVCS, or call
routines which do.
```

!-

```
JSB_CCB_A2 R5 =
  JSB (REGISTER = 2): GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %.
```

!+

```
JSB interface for Fortran format interpreter with CCB passed in
R11, two arguments passed in R10 and R9, and routine value
returned in R8.
```

!-

```
JSB_CCB_FMT1 =
  JSB : GLOBAL(CCB = 11, EL SIZE = 10, DT SEEN = 9, FMT_CODE = 8)
    NOPRESERVE (2,3) NOTUSED (4,5,6,7) %.
```

!+

```
Support for Indexed files.
Pass arguments (6) in R0:R5 and CCB is passed in R11.
```

!-

```
JSB_CCB_A6 R5 =
  JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3,
    REGISTER = 4, REGISTER = 5):GLOBAL (CCB = 11) NOTUSED (6, 7, 8, 9, 10) %.
```

```
JSB_CCB_A5 R5 =
  JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3,
    REGISTER = 4):GLOBAL (CCB = 11) NOTUSED (6, 7, 8, 9, 10) %.
```

!+

```
JSB interface with CCB passed in R11, arg1 in R2, arg2 in R0,
and only uses R0, R1, R2.
Needed when input arg1 is referenced after a CALL or JSB, so save
copying to R2.
```

!-

```
JSB_CCB_A2 A0 =
  JSB (REGISTER = 2, REGISTER = 0): GLOBAL(CCB=11) NOTUSED (3,4,5,6,7,8,9,10) %.
```

!+

```
JSB interface (no CCB), args in R0 and R9
```

!-

```
JSB_A0 A1 R8 =
  JSB (REGISTER = 0, REGISTER = 1) : NOPRESERVE (2,3,4,5,6,7,8) %.
```

!+

B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
[
\
]
^
_
`
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
{|}~

! JSB for Basic format routines - Plain F and E format. Pass
6 args and preserve all other registers. (1 optional arg)
! -

JSB_A6_R11 =
JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3, REGISTER = 4, REGISTER = 5) :
PRESERVE (6, 7, 8, 9, 10, 11) %;

!+ JSB for Basic format routines - Fancy F and E formats. Pass 11 args and
preserve all other registers. (4 optional args)
! -

JSB_A11_R11 =
JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3, REGISTER = 4,
REGISTER = 5, REGISTER = 6, REGISTER = 7, REGISTER = 8, REGISTER = 9, REGISTER = 10) :
PRESERVE (11) %;

!+ JSB interface (without CCB), no args in registers
! -

JSB_NO_ARGS =
JSB: NOTUSED (2,3,4,5,6,7,8,9,10) %;

```
!++
Now define the LINKAGE declarations.
Use names associated with the entry point rather than
the type of linkage, so that we can easily change
the linkage for an entry point without changing that
for other entry points.
Note: entry points that are dispatched to using a table
must have the same linkage name. In this case the LINKAGE
name is associated with the name of the dispatch table and
the call is made using the general LINKAGE form.

NOTUSED restriction!!! Because each JSB declaration must be
aware of all JSB routines which are in turn called. The NOTUSED
registers can only be the same as the caller (if also A JSB routine)
or include additional registers as well as being the same.
Thus, this file documents the calling tree for JSB linkages
so that the NOTUSED declarations can be kept in agreement.
--
```

LINKAGE

```
!+
Default CALL using CCB as a GLOBAL register,
all args in arg list.
-

CALL_CCB =          CALL_CCB_R11,

!+
CALL from BASIC compiled code, which uses R11 to point to the
major frame.
-

CALL_BMF =          CALL_BMF_R11,

!+
This is a linkage for BASS$REC_WSL1 to allow one arg to be passed.
-

JSB_REC_WSL1 =      JSB_CCB_A0_R5,

!+
CALL passing first arg in R0.
Used by FORENTRY module to make multiple entry points
all branch to FIOBEG.
-

CALL_FIOBEG =      CALL_A0,

!+
UDF initialization (user data formatting level of abstraction)
Arg is adr. of format statement.
JSBs to record level initialization (JSB_REC0).
-

JSB_UDF0 =          JSB_CCB_R5,
```



```
!+
! JSB to plain formatting routines for Basic.
!-
```

```
JSB_FORMAT_A6 = JSB_A6_R11,
```

```
!+
! JSB to fancy formatting routines for Basic.
!-
```

```
JSB_FORMAT_A11 = JSB_A11_R11,
```

```
!+
! UDF termination (user data formatting level of abstraction)
! JSBs to DO_READ (JSB_DO_READ) or DO_WRITE (JSB_DO_WRITE).
!-
```

```
JSB_UDF9 = JSB_CCB_R5,
```

```
!+
! UDF read routine
! JSBs to record level (JSB_REC1).
!-
```

```
JSB_DO_READ = JSB_CCB_A1_R5,
```

```
!+
! UDF write routine
! JSBs to record level (JSB_REC1).
!-
```

```
JSB_DO_WRITE = JSB_CCB_A0_R5,
```

```
!+
! Format interpreter initialization: FORMAT_ADR = arg is adr. of format statement
! JSBs to nothing.
!-
```

```
JSB_FMT0 = JSB_CCB_NO_ARGS,
```

```
!+
! Format interpreter main processing
!-
```

```
JSB_FMT1 = JSB_CCB_FMT1,
```

```
!+
! JSB to REC level of index file support
!-
```

```
JSB_REC_IND = JSB_CCB_A5_R5,
JSB_REC_IND1 = JSB_CCB_A8_R5,
```

```
!+
! Record level RMS interface level initialization.
! JSBs to nothing.
```

!-

JSB_RECO = JSB_CCB_R5,
JSB_REC2 = JSB_CCB_A0_R5,

+ Record level RMS interface level finished one buffer
JSBs to nothing.
-

JSB_REC1 = JSB_CCB_R5,

+ Record level RMS interface termination of statement.
JSBs to nothing.
-

JSB_REC9 = JSB_CCB_R5,

+ Push current LUB/ISB/RAB: LOGICAL_UNIT is unit no., LUN_Min is min. no.
JSBs to nothing.
-

JSB_CB_PUSH = JSB_CCB_A2_A0,

+ PUT relative with count
-

JSB_PUT = JSB_CCB_A1_A0,

+ Pop current LUB/ISB/RAB
JSBs to nothing.
-

JSB_CB_POP = JSB_CCB_NO_ARGS,

+ Return current LUB/ISB/RAB to free storage (open error or close)
JSBs to nothing.
-

JSB_CB_RET = JSB_CCB_NO_ARGS,

+ Get adr. of current LIB/ISB/RAB (called only from non-shared routines)
since harder to have a data entry vector which is the same
if module were to become shared or vice versa.
-

JSB_CB_GET = JSB_CCB_NO_ARGS,

+ JSB to kernel conversion routine

!-

JSB_CVT_KERNEL = JSB_A0_A1_R8,

!+
!-
!-

JSB to CALL_VFE routine, args on stack

JSB_CALL_VFE = JSB_NO_ARGS;

! End of file OTSLNK.REQ

