





```
.title pli_rt_macros - Defines general pli runtime macros
.ident /1-0037
; Edit: CGN003
; Edit: WHM002
```

```
*****
*
* 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: Bill Matthews

Functional Description:  
Define PL/I Runtime macros.

Modifications:

- 1-002 Bill Matthews 11-Apr-1983  
Removed \$defdat macro. DATDEF.SDL now generates \$defdat.
- 1-003 Chip Nylander 08-Aug-1983  
Added str\_l\_parent to the stream control block structure.

case macro - generates a case instruction

```
.MACRO CASE,SRC,DISPLIST,TYPE=W,LIMIT=#0,NMODE=S^# ,?BASE,?MAX
CASE'TYPE SRC,LIMIT,NMODE'<<MAX-BASE>/2>-1
BASE:
.IRP EP,<DISPLIST>
.WORD EP-BASE
.ENDR
MAX:
.ENDM CASE
```

```

++
macro to generate a offset list for a data structure
    it is useful for input argument lists positively indexed from ap, and
    work areas allocated in call stack and negatively indexed from fp.

```

```

call: $offset initial,direction,<<lab1,[size]>>,...,<labn,[size]>>

```

```

where:    initial is a required value for the intial index when
          originating a data structure definition.  it is normally
          (+) 4 for argument lists and 0 for work areas.

```

```

          direction is a keyword that must be:
              positive - for structures growing up in memeory
              negative - for structures growing down in memeory
              or blank, in which case POSITIVE is assumed.

```

```

          the label, size list is the symbolic name for the location
          and the optional size of the element.  if blank, size is
          assumed to be 4 ( one longword ).

```

```

to permit the definition of an indefinitely large number of labels,
the macro may be continued.  in this case the INITIAL and
DIRECTION arguments must be blank.

```

```

--
.MACRO $OFFSET INITVALUE,DIRECTION,SYMLST
.save
.psect $abs$,abs
.if b,initvalue
.if nb,direction
.error ; direction must be blank when contiuing;
.mexit
.endc
.iff
dir...=1
.=initvalue
.if nb,direction
.if idn <direction>,<positive>
.iff
.if idn <direction>,<negative>
dir...=-1
.iff
.error ; DIRECTION must be POSITIVE,NEGATIVE, of blank;
.endc
.endc
.endc
.endc
.irp sym,<symlst>
$offst1 sym
.endr
.restore
.ENDM $OFFSET

.MACRO $OFFST1 SYM,SIZ=4
.if lt,siz

```

```
.error                ;***** siz parameter negative *****;
.endc
.if      lt,dir...
.blkb   -siz
.endc
.if nb,sym
sym:
.endc
.if      gt,dir...
.blkb   siz
.endc
.ENDM   $OFFST1

:
: psect definition MACRO
:
: .MACRO  RTSHARE          ; define shared psect
: .psct   pli$code,pic,rd,shr,nowrt,long
: .ENDM   RTSHARE

:
: Define the PL/I Runtime read-only data
:
: .macro  pli_v1_read_only_data
:
: define the ascii character set for collate
:
: pli$ab_colat::
: x=0
: .rept 128
: .byte  x
: x=x+1
: .endr
:
: x=0
: .rept 128
: .byte  x
: x=x+1
: .endr

:
: define packed decimal constants
:
: pli$b_pacn1::
: .packed -1
: pli$b_pac1::
: .packed 1
: pli$b_pac0::
: .packed 0
: pli$b_pac5::
: .packed 5
:
: .endm  pli_v1_read_only_data

:
: .macro  pli_v2_read_only_data
: pli$b_pac_2_power_00::
: .packed 0000000001
```

```
plisb_pac_2_power 01::  
  .packed 0000000002  
plisb_pac_2_power 02::  
  .packed 0000000004  
plisb_pac_2_power 03::  
  .packed 0000000008  
plisb_pac_2_power 04::  
  .packed 0000000016  
plisb_pac_2_power 05::  
  .packed 0000000032  
plisb_pac_2_power 06::  
  .packed 0000000064  
plisb_pac_2_power 07::  
  .packed 0000000128  
plisb_pac_2_power 08::  
  .packed 0000000256  
plisb_pac_2_power 09::  
  .packed 0000000512  
plisb_pac_2_power 10::  
  .packed 0000001024  
plisb_pac_2_power 11::  
  .packed 0000002048  
plisb_pac_2_power 12::  
  .packed 0000004096  
plisb_pac_2_power 13::  
  .packed 0000008192  
plisb_pac_2_power 14::  
  .packed 0000016384  
plisb_pac_2_power 15::  
  .packed 0000032768  
plisb_pac_2_power 16::  
  .packed 0000065536  
plisb_pac_2_power 17::  
  .packed 0000131072  
plisb_pac_2_power 18::  
  .packed 0000262144  
plisb_pac_2_power 19::  
  .packed 0000524288  
plisb_pac_2_power 20::  
  .packed 0001048576  
plisb_pac_2_power 21::  
  .packed 0002097152  
plisb_pac_2_power 22::  
  .packed 0004194304  
plisb_pac_2_power 23::  
  .packed 0008388608  
plisb_pac_2_power 24::  
  .packed 0016777216  
plisb_pac_2_power 25::  
  .packed 0033554432  
plisb_pac_2_power 26::  
  .packed 0067108864  
plisb_pac_2_power 27::  
  .packed 0134217728  
plisb_pac_2_power 28::  
  .packed 0268435456
```

```

pli$b_pac_2_power_29::
.packed 0536870912
pli$b_pac_2_power_30::
.packed 1073741824
pli$b_pac_2_power_31::
.packed 2147483648
.endm

```

```

.end

```

```

;+
; define condition handler control blocks
;-

```

```

.MACRO SDEFCND

```

```

$defini cnd

```

```

$def cnd_l_link .blkl 1 ;link entry
$def cnd_l_enabl .blkl 1 ;enable value
$def cnd_l_addr .blkl 1 ;address of handler routine
$def cnd_l_arg .blkl 1 ;condition handler argument
;usually fcb of associated file
$def cnd_l_flags .blkl 1 ;flags longword
_vieid cnd,0,<-
<nocall,1,m>,-
;
;does not require call interface
>
$def cnd_k_length ;length of block

```

```

$defend cnd

```

```

.mdelete $defcnd
.ENDM SDEFCND

```

```
:+
: define convert case table indices
:-
```

```
.MACRO $DEFCVTIND
```

```
: NOTE WELL: CHANGES MADE HERE MUST BE REFLECTED IN THE CODE GENERATOR,
: THE RUNTIME FORMAT CONVERSION ROUTINES, THE RUNTIME
: CONVERT ROUTINE, AND ANYWHERE ELSE THAT BREAKS. YOU'LL
: ALSO WANT TO CHECK PUT LIST ITEM, PUT EDIT ITEM, GET LIST
: ITEM AND GET EDIT ITEM, IF YOU ADD ENTRIES. GOOD LUCK
```

```
$defini cvtind
```

```
Sequlst cvt_k_dst_.,0,1,< -      :define destination indices
<pic>, -                          :picture
<fixb>, -                         :fixed binary
<fltb>, -                         :float binary
<fixd>, -                         :fixed decimal
<fltd>, -                         :float decimal
<char>, -                         :character
<vcha>, -                        :character varying
<bit>, -                          :bit
<abit>, -                        :aligned bit
>
```

```
Sequlst cvt_k_src_.,0,9,< -      :define source indices
<pic>, -                          :picture
<fixb>, -                         :fixed binary
<fltb>, -                         :float binary
<fixd>, -                         :fixed decimal
<fltd>, -                         :float decimal
<char>, -                         :character
<vcha>, -                        :character varying
<bit>, -                          :bit
<abit>, -                        :aligned bit
>
```

```
$defend cvtind
```

```
.MDELETE $DEFCVTIND
.ENDM $DEFCVTIND
```

```

:~+
:~ define file display block
:~   this block is used to fill in the users file display block by the
:~   DISPLAY built in subroutine.
:~
:~ .MACRO $DEFDSP
:~
:~ $defini dsp
:~
$def  block_size          .blk1  1
$def  bucket_size       .blk1  1
$def  creation_date     .blk1  2
$def  expiration_date   .blk1  2
$def  extension_size    .blk1  1
$def  file_id           .blk1  6
$def  file_size         .blk1  1
$def  fixed_control_size .blk1  1
$def  index_number      .blk1  1
$def  maximum_record_number .blk1  1
$def  maximum_record_size .blk1  1
$def  multiblock_count  .blk1  1
$def  multibuffer_count .blk1  1
$def  owner_group       .blk1  1
$def  owner_member      .blk1  1
$def  retrieval_pointers .blk1  1
$def  linesize         .blk1  1
$def  pagesize         .blk1  1
$def  page_number      .blk1  1
$def  line_number      .blk1  1
$def  column_number    .blk1  1
$def  number_of_keys   .blk1  1
$def  dtr              .blk1  2 ;displayed attributes (see below)
$def  device           .blk1  1
$def  spool_device     .blk1  1
$def  file_organization .blkb  3
$def  group_protection .blkb  6
$def  owner_protection .blkb  6
$def  system_protection .blkb  6
$def  world_protection .blkb  6
$def  expanded_title   .blkb 130
$def  dspend
:~
:~ _vield dtr,0,<-
:~   <fortran_format,1,m>-
:~   <block_boundary_format,1,m>-
:~   <supersede,1,m>-
:~   <temporary,1,m>-
:~   <block_io,1,m>-
:~   <deferred_write,1,m>-
:~   <carriage_return_format,1,m>-
:~   <rewind_on_open,1,m>-
:~   <current_position,1,m>-
:~   <write_check,1,m>-
:~   <fixed_length_records,1,m>-
:~   <rewind_on_close,1,m>-
:~   <ignore_line_marks,1,m>-

```



```

: +
: define environment block
:   this control block is used to pass environment information from
:   pli$$envir to pli$open, for inclusion in the dynamically allocated
:   xab's.
: -

.MACRO $DEFENV
$defini env

$def  env_l_status      .blk1 1      ;status from env processing
$def  env_q_cdate      .blk1 2      ;creation date
$def  env_q_exdate     .blk1 2      ;expiration date
$def  env_l_fileidto   .blk1 1      ;addr of file_id_to
$def  env_l_fxctlto    .blk1 1      ;addr of fixed_control_to
$def  env_w_prot       .blkw 1      ;protection bits
$def  env_w_owngroup   .blkw 1      ;owner_group
$def  env_w_ownmem     .blkw 1      ;owner_member
$def  env_k_len
$def  _viold env,0,<-
      <create_dat,1,m>, -      ;creation_date was specified
      <expire_dat,1,m>, -     ;expiration_date was specified
      <fileid_to,1,m>, -      ;file_id_to was specified
      <fixedctl_to,1,m>, -    ;fixed_control_to was specified
      <protect,1,m>, -       ;*_protection was specified
      <uic,1,m>, -           ;owner_* was specified
      <close,1,m>, -        ;process environment for close
      >

$defend env

.MDELETE $DEFENV
.ENDM $DEFENV

```

```

:~+
:~ define runtime file control block
:~-

```

```

.MACRO $DEFFCB
$defini fcb

```

```

$def fcb_l_next .blkl 1 ;addr of next open file
$def fcb_l_previous .blkl 1 ;addr of last open file
$def fcb_l_error .blkl 1 ;most recent error code
$def fcb_l_attr .blkl 1 ;current file attributes
$def fcb_l_dtrr .blkl 1 ;declared file attributes
$def fcb_l_buf .blkl 1 ;address of file buffer
$def fcb_l_buf_end .blkl 1 ;address of end of buffer (stream)
; length of allocated buffer (record)
$def fcb_l_buf_pt .blkl 1 ;address of next byte to be processed
; in buffer (stream)
$def fcb_q_rfa .blkl 2 ;rfa of last record processed (record)
$def fcb_w_revision .blkw 1 ;version of the file control block (set
; by valloc, checked by open)
$def fcb_w_linesize .blkw 1 ;linesize (stream)
$def fcb_w_pagesize .blkw 1 ;pagesize (stream)
$def fcb_w_column .blkw 1 ;current column (stream)
$def fcb_w_line .blkw 1 ;current line number (stream)
$def fcb_w_page .blkw 1 ;current page number (stream)
$def fcb_c_strlen .blkw 1 ;length of the vestigial file control
; block used for get and put string
$def fcb_l_prn .blkl 1 ;printer control buffer (stream)
$def fcb_l_kcb .blkl 1 ;addr of key control block (record)
$def fcb_b_numkcb .blkb 1 ;number of keys (record)
$def fcb_b_extra .blkb 3 ;extra
$def fcb_b_ident .blkb 1 ;files declared name
$def fcb_w_ident_len .blkw 1 ;length of files declared name
$def fcb_b_ident_nam .blkb 32 ;text of files declared name
$def fcb_b_rab .blkb 68 ;record access block
$def fcb_b_fab .blkb 80 ;file access block
$def fcb_b_nam .blkb 56 ;name block
$def fcb_b_esa .blkb 128 ;expanded string area
$def fcb_l_condit .blkl 1 ;entry for handling conditions which
$def fcb_l_cndaddr .blkl 4 ;jsb's to cndaddr (filled in by open)
$def fcb_b_envir .blkb ;start of the environment block
$def fcb_c_len .blkw ;length of the control block

_vield atr,0,<- ;attributes (used for attr and dtrr)
<eof,1,m>,- ;end of file
<opened,1,m>,- ;file is opened
<comma_exp,1,m>,- ;comma is expected in list input
<recur,1,m>,- ;ast re-entrant lock
<update,1,m>,- ;update attribute
<output,1,m>,- ;output attribute
<input,1,m>,- ;input attribute
<print,1,m>,- ;print attribute
<keyed,1,m>,- ;keyed attribute
<direct,1,m>,- ;direct attribute
<seql,1,m>,- ;sequential attribute

```

```
<stream,1,m>,-      ;stream attribute
<record,1,m>,-      ;record attribute
<scalvar,1,m>,-     ;scalar_varying
<app,1,m>,-         ;append
<recidacc,1,m>,-    ;record_id access allowed
<indexed,1,m>,-     ;file is indexed
<bfall,1,m>,-       ;buffer is allocated for rewrite
<currec,1,m>,-      ;rms current record context is in-
                    ; correct by pl/i rules
<delete,1,m>,-      ;last completed operation was delete
<write,1,m>,-       ;last completed operation was write
<app_comma,1,m>,-   ;ignore linemarks was not specified
<blockio,1,m>,-     ;block i/o access allowed
<string,1,m>,-      ;string i/o in progress
<vcha,1,m>,-        ;varying character string target
<virgin,1,m>,-      ;file was just opened
<fltrg,1,m>,-       ;floating target in get
```

&gt;

\$defend fcb

```
.MDELETE $DEFFCB
.ENDM $DEFFCB
```

```
:+
: define options for get statement
:-
```

```
.MACRO $DEFGETOPT
```

```
$defini getopt
```

```
$def getopt_l_fxdctl .blkl 1 ;address of fixed control area
$def getopt_l_prompt .blkl 1 ;address of prompt (char var)
$def getopt_b_tmo .blkb 1 ;timeout in seconds
$def getopt_b_bits .blkb 1 ;additional bit options
$def getopt_c_len ;
```

```
_yield get,0,<- ;
<no_echo,1,m>,- ;use read no echo (1)
<use_tmo,1,m>,- ;use timeout specified above (2)
<upper_case,1,m>,- ;convert to upper case (4)
<no_filter,1,m>,- ;use read no filter (8)
<extra,1,m>,- ;extra, not used by rms (10)
<purge_tah,1,m>,- ;purge type ahead (20)
<prompt,1,m>,- ;use prompt above (40)
> ;
```

```
$defend getopt
```

```
.MDELETE $DEFGETOPT
.ENDM $DEFGETOPT
```





```
:+
: run time constant definitions
:-
```

```
.MACRO $DEFPL1RTCONS
```

```
$defini plx
```

```
Sequ plisc_deflms      132      :default line size
Sequ plisc_defmrs     512      :default maximum record size
Sequ plisc_version     3       :version of fcb
Sequ plisc_cr         141      :carriage return for prn files
```

```
$defend plx
```

```
.MDELETE $DEFPL1RTCONS
.ENDM $DEFPL1RTCONS
```

```
:+
: define options for put statement
:-
  .MACRO $DEFPUTOPT
    $defini putopt

$def  putopt_l_fxdctl .blkl 1      ;address of fixed control area
$def  putopt_b_bits  .blkb 1      ;additional bit options
$def  putopt_c_len
    _vield put,0,<-                ;
    <can_con_o,1,m>,-             ;cancel control o
    >                               ;

$defend putopt

.MDELETE $DEFPUTOPT
.ENDM $DEFPUTOPT
```

```
:+  
: define runtime scanc/spanc table  
:-
```

```
.MACRO $DEFRTSCAN
```

```
$defini rtscan
```

```
$equ plisc_space 1 ;scan mask for space  
$equ plisc_comma 2 ;scan mask for comma  
$equ plisc_point 4 ;scan mask for decimal point  
$equ plisc_e 8 ;scan mask for e or E  
$equ plisc_tab 16 ;scan mask for tab  
$equ plisc_blank 17 ;scan mask for space and tab
```

```
$defend rtscan
```

```
.MDELETE $DEFRTSCAN  
.ENDM $DEFRTSCAN
```

```
:+
: define runtime stack
:-
```

```
.MACRO $DEFSTK
```

```
$defini stk
```

```
.=.-12
```

```
$def stk_l_cnd_lst .blkl 1 ; condition handler list
$def stk_l_arg_list .blkl 1 ; argument pointer
$def stk_l_display .blkl 1 ; parent's display pointer
$def stk_l_cnd_hnd .blkl 1 ; address of condition handler
$def stk_l_psl .blkl 1 ; psw and register save mask
$def stk_l_ap .blkl 1 ; saved argument pointer
$def stk_l_fp .blkl 1 ; saved frame pointer
$def stk_l_pc .blkl 1 ; saved program counter
$def stk_l_regs .blkl 1 ; saved registers
```

```
$defend stk
```

```
.mdelete $DEFSTK
.ENDM $DEFSTK
```

```

: +
: define stream i/o control block
: -

```

```

.MACRO SDEFSTR

```

```

$defini str

```

```

$def str_l_sp      .blkl 1      ;format interpreter stack pointer
$def str_l_fp      .blkl 1      ;format pointer
$def str_l_parent  .blkl 1      ;parent pointer
$def str_l_fs      .blkl 1      ;format status
$def str_l fld_pt  .blkl 1      ;address of next free char in field
$def str_l fld_end .blkl 1      ;address of end of field + 1
$def str_b_field   .blkb 1008   ;field storage
$def str_l_stack_end .blkl 511  ;stack storage
$def str_l_stack   .blkl 1      ;top of stack
$def str_c_len     .blkl 1      ;length of stream block

_vield str 0, <-
  <missing,1,m>, - ;status bits for stream block
  <edit,1,m>, -    ;last format parameter was missing = 1
  <string,1,m>, - ;edit directed = 1, list directed = 0
  <blankend,1,m>, - ;get or put string in progress = 1
  <gfloat,1,m>, - ;append blanks to end of field
  <null_line,1,m>, - ;processing gfloat value
  > ;last record read was null

```

```

$defend str

```

```

.MDELETE SDEFSTR

```

```

.ENDM SDEFSTR

```

```

.end

```

