

```

%* macro to check result code of previous macro and abort the program if it is
greater then INRC;
%macro MU_CHECKRC(CONDITION =
                    ,RC          =
                    ,rcprefix =
                    ,MSG1       =
                    ,MSG2       =
                    ,MSG3       =
                    ,ALERTID    =
                    ,SETABORT = N
                    ) / store source des='V1.0.0.2';
%*****
*****
* FILENAME:      MU_CHECKRC.SAS
* DEVELOPER:     (b) (6)
* PLATFORM:      SAS 9.1.3, 9.2 on PC
*
* MACROS USED:   None
* ASSUMPTIONS:   Called from inside another macro and passes a minimum of two
parameters: CONDITION and MSG1
*
*
* DESCRIPTION:
* Macro reviews value of CONDITION parameter and if it is true, produces an ALERT
message box
* with message text that has been provided in MSG1 and if provided MSG2 parms.
* Macro also creates GLOBAL macro variable based upon value passed into RCPREFIX
parm with value passed via RC parm.
*
* If no values have been passed via CONDITION or MSG1, the macro will print an
ALERT message box
* indicating these fields are required and stop (ABORT) the program.
*
* USAGE NOTES:
* Example 1: If the macro variable _default_bign_in_data exists, IN_DATA will be
set to the value of &_default_bign_in_data.
* If it does not exist, IN_DATA will be set to a null value.
* %MU_CHECKRC(CONDITION = &SKIPVARS_NOT_IN_ORDVARS=1
*             ,RC          = 1
*             ,rcprefix = MYMAC
*             ,MSG1       = Some variables listed in the SKIPVARS parameter were not
listed in the
*             ,MSG2       = SORT_ORDER_VARS - skipped
*             ,ALERTID    = R
*             )
* Produces the following ALERT message box and a global macro variable MYMAC_RC=1
* A_LERT_R:
-----
* A_LERT_R: Some variables listed in the SKIPVARS parameter were not listed in the
* A_LERT_R: SORT_ORDER_VARS - skipped

```

```

* A_LERT_R:
-----
*
*
* Example 2: Use macro to print out an information message, no matter
what...notice condition value.
*
* %MU_CHECKRC(CONDITION = 1=1
*             ,RC      = 0
*             ,rcprefix = mymac
*             ,MSG1     = The Macro Parameter [&mParm_name] was set to an internal
DEFAULT value ==> %str(&md_default_value)
*             )
*
* Produces the following ALERT message box and a global macro variable
MD_MACRO_DUMMY_RC=0
* A_LERT_I:
-----
* A_LERT_I: The Macro Parameter [OUT_DATA] was set to an internal DEFAULT value
==> SUMM_DATA
* A_LERT_I:
-----
*
*
* PARAMETERS:
* CONDITION      = an algorithm (REQUIRED). Example: &calc_value = 20
* MSG1           = Message to print in ALERT box (REQUIRED).
* MSG2           = Additional Message line if message is to appear on two lines or
too long to fit on one line
* MSG3           = Additional Message line
* ALERTID        = One of the following letters: [I]nformation (Default),
[R]equired, ???.
```

If no value is entered, program will print I.

```

* SETABORT       = If program is to stop (ABORT) if algorithm entered in CONDITION
parm is true, set this parm to Y
*               Any other value entered will not cause program to stop.
* RC             = Return Code numeric value to be passed back if condition is
true
*               Macro initializes this value to 0 (zero) at beginning of
processing.
* RCPREFIX       = Prefix to be given to Global macro variable created by this
macro containing the return code
*               If no value is entered, then macro variable created containing
return code will be _RC.
*               To create variable based upon calling macro, enter
&SYSMACRONAME or hard code calling macro name
*
*
*****
* DATE          INITIALS      MODIFICATION DESCRIPTION
```

```

*****
*
*****
*   © 2011 Pharmaceutical Product Development, Inc.
*   All Rights Reserved.
*****
*****;

%local  prtmsg abort efront e1 e2 e3 i;
%global &sysmacroname._RC;

%let &sysmacroname._RC = 0;
%let efront      = &sysmacroname%str(:);

%*
-----;
%* -- Review RCPREFIX entry for issues, initialize as necessary  ----;
%*
-----;
%if %bquote(&rcprefix) ne %str( ) %then %do;

    %let rcprefix = %upcase(&rcprefix);

    %* -- Invalid value for macro variable name
    -----;
    %if %sysfunc(nvalid(&rcprefix._rc, v7)) = 0 %then %do;
        %let e1 = %str(&efront RCPREFIX [&rcprefix] is not a valid macro variable
name);
        %let &sysmacroname._RC = 1;
    %end;
%end;

%if %length(&e1) = 0 %then %do;

    %* -- Determine if the Macro Variable RC name has already been defined as a
LOCAL macro variable --;
    %if %symlocal(&rcprefix._rc) = 1 %then %do;
        %let e1 = %str(&efront &RCPREFIX._RC has already been defined as a LOCAL
macro variable );
        %let &sysmacroname._RC = 2;
    %end;

    %else %if %symexist(&rcprefix._rc) = 0 %then %do;
        %* -- Global Macro Variable RC name does not exist, create and
initialize here --;
        %global &rcprefix._rc;
        %let &rcprefix._rc = 0;
    %end;
%end;

```

```

%end;

%else %if %symexist(&rcprefix._rc) = 1 %then %do;
    %* -- Global Macro Variable RC name exist, reset here --;
    %let &rcprefix._rc = 0;
%end;
%end;
%*
-----
-----;
%* -- Review CONDITION and MSG1 entries for issues, initialize as necessary
-----;
%*
-----
-----;

%* -- Check that the macro required macro parameters CONDITION and MSG1 have been
passed --;
%if %bquote(&condition) = %str( ) or %bquote(&msg1) = %str( ) %then %do;
    %let e2 = %str(&efront Must pass values for both CONDITION and MSG1 );
    %if &&&sysmacroname._RC > 0 %then %let &sysmacroname._RC =
&&&sysmacroname._RC%str( 3);
    %else
        %let &sysmacroname._RC = 3;
%end;

%* -- Check that the condition contains an expression --;
%else %if %sysfunc(indexc(%bquote(&condition) , %str(=<>))) = 0 %then %do; %* No
logical symbols found;
%* -- Look for GE LE GT LT EQ -----;
%mu_wordscan(string = %bquote(&condition)
              ,root = _cword
              ,numw = nwords);

%if &nwords > 2 %then %do;
    %let _cword2 = %upcase(&_cword2);

    %if %bquote(&_cword2) = %str(GE) or %bquote(&_cword2) = %str(LE) or
        %bquote(&_cword2) = %str(EQ) or %bquote(&_cword2) = %str(NE) or
        %bquote(&_cword2) = %str(GT) or %bquote(&_cword2) = %str(LT) %then
%let e3 =;
    %else %let e3 = 1;

%end;
%else %let e3 = 1;

%if %str(&e3) = 1 %then %do;
    %let e2 = %str(&efront CONDITION must be a logical expression comparing
two or more items);
    %let e3 = %str(&efront CONDITION value passed to macro ==>
%bquote(&condition) );

```

```

        %if &&&sysmacroname._RC > 0 %then %let &sysmacroname._RC =
&&&sysmacroname._RC%str( 4);
        %else                                     %let &sysmacroname._RC = 4;
        %end;
    %end;

    %*
-----
-----;
    %* -- One of the above issues has occurred, stop processing and write message to
log --;
    %*
-----
-----;
    %if %length(&e1.&e2.&e3) > 0 %then %do;
        %let abort = 1;
        %let prtmsg = 1;
        %let alertid= P;

        %do i = 1 %to 3;
            %let msg&i = %str(&e&i);    %* Overwrite message text provided via parms
with issue messages;
        %end;
    %end;

    %*
-----
-----;
    %*-- Review Condition passed to macro and process message if True ---;
    %*
-----
-----;
    %else %if %sysfunc(ifn( %bquote(&CONDITION ) , 1, 0)) %then %do;

        %let prtmsg = 1;

        %if %upcase(%bquote(&setabort)) = Y %then %do;
            %let abort = 1;
            %let alertid = P;
        %end;

        %else %if %bquote(&alertid) = %str( ) %then %let alertid = I;

        %else %do;    %* If an invalid AlertID was provided, set to to R;
            %if %length(&alertid) >1 %then %let alertid = R;
            %else %do;
                %let alertid = %upcase(%bquote(&alertid));
                %if %verify(%bquote(&alertid), ICRP) %then %let alertid = R;
            %end;
        %end;
    %end;

```

```

        %end;

        /* -- Set RC value. If provided, Use it. If not and alertID is not I, set
to 9;
        %if          %bquote(&rc) ne %str( ) %then %let &rcprefix._rc = &RC;
        %else %if %bquote(&alertid) ne I %then %let &rcprefix._rc = 9;
        %end;

        /*
-----
-----;
        /* -- Write Alert to program log if condition was met or an issue was found with
parameter entries --;
        /*
-----
-----;
        %if &prtmsg = 1 %then %do;
            %if &abort = 1 %then %let msg1 = %str(&msg1) %str( - ABORT);
            %put %str(ALERT_&ALERTID:
-----
-);
            %put %str(ALERT_&ALERTID: &MSG1);
            %if %bquote(&MSG2) ne %str( ) %then %put %str(ALERT_&ALERTID: &MSG2);
            %if %bquote(&MSG3) ne %str( ) %then %put %str(ALERT_&ALERTID: &MSG3);
            %put %str(ALERT_&ALERTID:
-----
-);
        %end;
        %else %let &sysmacroname._RC = -1;

        /*
-----
-----;
        /* -- If requested, program will Abort processing -----;
        /*
-----
-----;
        %if &abort = 1 %then %abort;

%mend MU_CHECKRC;

```