
TITLE: dearepl Patch

DOCUMENT NUMBER: 36-58030.12 REVISION: 02

ORIGINATOR: James Francis

LETTER	SCO NO.	DESCRIPTION	APPROVED	DATE
01	36-986	Initial numeric release	jimf	11/12/98
02	36-1010	Fix typos	J.F	05/10/99

```
# ------------------------------------------------
#
# $$Source: /nfs/acis/h3/acisfs/configcntl/patches/dearepl/dearepl.pkg,v $$
#
# DEA Engineering Replacement Patch Specification File
#
# Version:
#   The part number and version of this release are
#   described below under the "partnumber" and
#   "version" keywords.
#
# Description:
#   This is a Patch Specification File. The detailed
#   documentation for this file is provided after the
#   NOTES: keyword below.
#
# Format:
#   This is a line-oriented file.
#
#   Comments are indicated by a leading '#'.
#   Blank lines are ignored.
#
#   Keyword pairs are assigned as "keyword = value",
#   where:
#       ident           - The CVS/RCS identification string
#       partnumber      - The partnumber of the patch
#       version         - The release version of the patch
#       environment     - Either "flight", or "engineering"
#       sco             - The software change order of the released patch
#       reason          - Short reason for this version
#
#   Lists of information consist of the list name
#   followed by the next item to be placed into the
#   list. The lists are:
#
#       source <name> <partext> - This specifies a source file
#                       which should be reviewed when
#                       the package is released. At this time,
#                       these entries are only used for documentation
#                       purposes and aren't used to build run-time
#                       products. The run-time products are produced
#                       by the .mak file. <partext> refers to the part
#                       number extension of the file relative to the
#                       base part number of the patch.
#
#       object <name>   - This specifies an object file
#                       which must be built and linked for
#                       the patch, where <name> is the name
#                       of the file to be built and linked with.
#
#       func <oldname>  <newname> -
#                       This specifies a function
#                       which must be overridden for the
#                       patch to work. <oldname> is the
#                       old subroutine name, and <newname>
#                       is the new subroutine which replaces
#                       the old.
#
#       bcmd <name>     - This specifies a literal bcmd input
#                       file which must be built and included
#                       in the load for the patch. These typically
#                       hold indepenent specially built patches
#                       which do not have to be linked with the
#                       reset of the system in order to work, such
#                       as inline patches.
#
#       spr <number>    - This identifies a Software Problem Report
#                       which is addressed by this patch.
#
#       ser <number>    - This identifies a Software Enhancement Request
#                       which is addressed by this patch.
#
#       tool <number>   - This identifies a Software Diagnostic Tool
#                       which is addressed by this patch.
#
#       docref <number> - This identifies an exisiting design or
#                       requirements reference which is pertinent
#                       to the patch.
#
#       approval <rev> <sco> <signer> <date> <text>
#                       - Sign-off on a previous release
#
#   At the end of the file, the 'NOTES:' keyword
#   delimits the notes section of the file. All lines
#   following this keyword line are treated as the
#   release notes for this patch. These notes should be
#   included in all patch releases and option suite documentation.
#
#   The notes sections are delimited by section keywords. Any text
#   from the start of the NOTES section until the first keyword is
#   treated as a general description of the patch.
#
#      COMMAND IMPACT:  - This section describes the impact of the patch
#                       on commanding of the instrument.
#
#      TELEMETRY IMPACT: - This section describes the impact of the patch
#                        on the telemetry produced by the instrument.
#
#      SCIENCE IMPACT:   - This sections describes the impact of the patch
#                        on the science data produced by the instrument.
#
#      :END              - Delimits the end of the notes section
#
# Version Log:
# $$Log: dearepl.pkg,v $
# $Revision 1.4  1999/05/10 17:26:03  jimf
# $Fix typos
# $
# Revision 1.3  1998/11/12  17:29:15  jimf
# Specify ECO #.
#
# Revision 1.2  1998/11/10  20:37:15  jimf
# Assign partnumbers and small documentation cleanup.
#
# Revision 1.1  1998/11/02  19:52:54  jimf
# Initial spec/makefiles
#$
#
# ------------------------------------------------

# Identification Information
ident = $$Id: dearepl.pkg,v 1.4 1999/05/10 17:26:03 jimf Exp $$

partnumber  = 36-58030.12
version     = 02
environment = engineering
sco         = 36-1010
reason      = Fix typos
```

```
# Release history information
approval 01 36-986 jimf 11/12/98 Initial numeric release

# Product and source file information
object dearepl.o

func DeaDevice::sendCmd Test_DeaDevice::sendCmd
func DeaDevice::readReply Test_DeaDevice::readReply
func DeaDevice::isCmdPortReady Test_DeaDevice::isCmdPortReady
func DeaDevice::isReplyReady Test_DeaDevice::isReplyReady
func DeaCcdController::updateRegister Test_DeaCcdController::updateRegister
func DeaManager::writeData Test_DeaManager::writeData
func DeaManager::checkLoads Test_DeaManager::checkLoads

source dearepl.pkg  01
source dearepl.mak  02
source dearepl.C    03

# Initiating action information
tool PENDING

# Documentation references

#-------------------------------------------------
NOTES:
This patch provides the basic capability to fake
the existence of a DEA. This patch is used when
no DEA box is available, or one wants to test
without actually talking to the DEA.

COMMAND IMPACT:
This "fakes" the existence of the DEAs. Commands
which read and write PRAM, SRAM or DEA hardware
will not crash, but won't work either.

TELEMETRY IMPACT:
This will produce true fiction from the DEAs.

SCIENCE IMPACT:
Can't do any, since the patch replaces the
interface to the real DEAs.

:END
```

dearepl/dearepl.mak


```
# ============================================================
#
# $$Source: /nfs/acis/h3/acisfs/configcntl/patches/dearepl/dearepl.mak,v $$
#
# Patch Name: DEA Replacement Patch Makefile
#
# Description:
#   This file builds the components of the DEA Replacement
#   patch, to be linked with either a required patch set or
#   optional patch set.
#
# Products:
#   dearepl.o - Implementation of DEA function stubs.
#
# $$Log: dearepl.mak,v $
# $Revision 1.1  1998/11/02 19:52:53  jimf
# $Initial spec/makefiles
# $$
#
# ============================================================

.SUFFIXES:  .bcmd .map .ab .o .lst .c .C .h .H

FS        = /nfs/acis/h3/acisfs/flightbld/flight1.5
BEPMAP    = $(FS)/cur_bep/acisBepRom.loc.map

CC       = acis-g++ -fvtable-thunks -c
CFLAGS   = -g -fconserve-space -fno-implement-inlines -O -DNDEBUG $(WARN) $(INCLUDES) -m
no-gpopt -G 0
INCLUDES = -I. -I.. -I$(FS)/acis_h -I$(FS)
WARN     = -Wall -Woverloaded-virtual -Wconversion -Wshadow -Wcast-align \
         -Wcast-qual -Waggregate-return

.C.o :
        $(CC) $(CFLAGS) -Wa,"-alh" $< -o $@ > $(*).lst

.S.o :
        $(CC) $(CFLAGS) -Wa,"-alh" $< -o $@ > $(*).lst

dearepl.o: dearepl.C

clean:
        rm -f *.o *.bcmd *.lst
```

```
/* ============================================================
 *
 * $$Source: /nfs/acis/h3/acisfs/configcntl/patches/dearepl/dearepl.C,v $$
 *
 * Patch Name: DEA Replacement Patch Source
 *
 * Description:
 *  This implements the classes which replace the interface
 *  to the DEA. This patch is used when one does not have
 *  a DEA attached, or wants to avoid commanding the DEA.
 *
 *  A detailed implementation description is pending.
 *
 * References:
 *
 * $$Log: dearepl.C,v $
 * $Revision 1.2  1998/11/10 20:37:14  jimf
 * $Assign partnumbers and small documentation cleanup.
 * $$
 *
 * ============================================================ */

#include "filesdevices/deadevice.H"
#include "filesdea/deaccdcontroller.H"
#include "filesprotocols/deamanager.H"

// ----------------------------------------------------------------------
class Test_DeaDevice : public DeaDevice
{
public:

    void     sendCmd(unsigned cmdWord);
    unsigned readReply();
    Boolean  isCmdPortReady() const;
    Boolean  isReplyReady() const;

protected:
private:
};

void     Test_DeaDevice::sendCmd(unsigned cmdWord) {};
unsigned Test_DeaDevice::readReply()               { return 0; };
Boolean  Test_DeaDevice::isCmdPortReady() const    { return BoolTrue; };
Boolean  Test_DeaDevice::isReplyReady() const      { return BoolTrue; };


// ----------------------------------------------------------------------
class Test_DeaCcdController : public DeaCcdController
{
public:
  Test_DeaCcdController(DeaBoardId deaBoard, DeaCcdController*&activeref, unsigned&broad
castMaskRef);

  Boolean updateRegister(RegId reg);
};

Test_DeaCcdController::Test_DeaCcdController(DeaBoardId deaBoard,
                                             DeaCcdController*&activeref,
                                             unsigned&broadcastMaskRef)
: DeaCcdController(deaBoard, activeref, broadcastMaskRef)
{
};

Boolean Test_DeaCcdController::updateRegister(RegId reg)
{
  return BoolTrue;
};

// ----------------------------------------------------------------------
class Test_DeaManager : public DeaManager
{
public:
  Test_DeaManager(unsigned semid);
  Boolean writeData(CcdId ccdid, unsigned addr, unsigned value);
  Boolean checkLoads();
};

Test_DeaManager::Test_DeaManager(unsigned semid)
: DeaManager(semid)
{
};

Boolean Test_DeaManager::writeData(CcdId ccdid, unsigned addr, unsigned value)
{
  return BoolTrue;
};

Boolean Test_DeaManager::checkLoads()
{
  proxy->setLoadValid();
  return BoolTrue;
};
```