Appendix E- Data Dictionary

This appendix contains a subset of the LPS detailed design data dictionary. The appendix contains an entry for each data dictionary item (parameters, variables, files, and database tables) cited in the module specifications in Appendix D. Each entry contains an informal description of the data item.

Formal definitions for each data item may be reviewed on-line by accessing the LPS detailed design model, LPS_DDS,onipdcase.gsfc.nasa.gov. Contact TBD for access to the on-line version of the detailed design model.

*
NO DESCRIPTION AVAILABLE
*

Band_File (store) =
*
A set of files of deinterleaved data with one file per band. Each file has been appended with the Status_Info and Drift_Time. Format 1 bands are 1-6; format 2 bands are 6, 7 and Pan.
*

Bands_Present (store) =
*
A data store containing the bands present in the associated processed contact.
*

Browse_File (store) =
*
A datastore containing one or two reduced data-volume image files of the band files of a sub-interval. If format 2, no browse file will exist.
*

CADU_Bit_Slip_Cnt (data flow) =
*
The count of the number of CADUs affected by bit slips.

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)
DATABASE RANGE/VALUE: X>=0 (Not Null)
*

CADU_CRC_Err_Cnt (data flow) =
*
Number of CRC errors encountered while CCSDS processing a Raw_WB_Data data set.

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)

DATABASE RANGE/VALUE: X>=0 (Not Null)
*

CADU_Flywheel_Cnt (data flow) =
*
Number of CADUs with flywheel errors encountered while
CCSDS processing a raw wideband data set.

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)
DATABASE RANGE/VALUE: X>=0 (Not Null)
*

CADU_Rcv_Cnt (data flow) =
*
Number of CADUs received in a raw wideband data set
(Raw_WB_Data).

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)
DATABASE RANGE/VALUE: X>=0 (Not Null)
*

CADU_Sync_Err_Cnt (data flow) =
*
Number of CADU sync errors encountered while CCSDS
processing a raw wideband data set (Raw_WB_Data).

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)
DATABASE RANGE/VALUE: X>=0 (Not Null)
*

CAL_File (store) =
*
NO DESCRIPTION AVAILABLE
*

Contact_Schedules (store) =
*
A schedule containing multiple start/stop times that Landsat 7 spacecraft
downlinks the wideband data to LGS. The schedule is coming from LGS in
a hardcopy form.
*

FS_RETURN_CODE (control flow) =
*
Frame Synching unit return status.
*

Failed_CADUs (store) =
*
NO DESCRIPTION AVAILABLE
*

Fill_CADU_Cnt (data flow) =
*
Count of the number of fill CADUs received during a contact.

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)
DATABASE RANGE/VALUE: X>=0 (Not Null)
*

Fill_Value (data flow) =
*
Predefined fill value. It is an operator input.

DATABASE DOMAIN NAME: Fill_Val_Char
DATABASE DATA TYPE: Char(1)
DATABASE RANGE/VALUE: Any Value
*

Frame_Sync (store) =
*
NO DESCRIPTION AVAILABLE
*

I (data flow) =
*
Depth of data interleaving (Set to 1 for Landsat)
*

IDP_Acct (store) =
*
Scene metadata associated with the subinterval ID on a scene basis.
*

Inverted_CADU_Cnt (data flow) =
*
Count of the CADUs originally received with inverted polarity

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)
DATABASE RANGE/VALUE: X>=0 (Not Null)
*

LDT_DAN_Transfer_State (store) =
*
A database table containing state information whether DANS are currently being sent from the LPS to the LP DAAC.
*

LDT_File_Group_Info (store) =
*
File group information for each subinterval.
*

LDT_File_Set_Info (store) =
*
File set information database table.
*

LPSJournal (store) =
*
LPS system journal name
*

LPS_Configuration (store) =
*
A database table containing the setup information used to startup LPS.
*

LPS_File_Info (store) =
*
A database table containing all L0R files generated by LPS L0R processing.
*

MFP_Acct (store) =
*
All quality and accounting information that is received on a major frame basis for a given subinterval. It consists of information on the major frame processing data, the Calibration filename and MSCD filename.
*

MSCD_File (store) =
*
A file containing the Scan Line data extracted from the two minor frames following the End of Line Code in each major frame for a given subinterval.
*

MaxBitFlips (data flow) =
*
NO DESCRIPTION AVAILABLE
*

Metadata_File (store) =
*
A file containing information on the Level OR processing for a specific subinterval.
*

Mjf (data flow) =
*
An instance of a pcd_mainMajorFramesStruct.
*

Natural (data flow, pel) =

*
The range 0-MAXINT.
*

OutputModulePtr (data flow) =
*
NO DESCRIPTION AVAILABLE
*

PCD_Acct (store) =
*
A database table that contains the statistics that are gathered from the processing of PCD bytes, the building of PCD Minor and Major Frames, and the extraction and interpretation of information from the PCD Major Frames.
*

PCD_File (store) =
*
A file containing the PCD major frames received during a subinterval on a full PCD cycle basis. A PCD cycle consists of four PCD major frames.
*

PCD_MJF_TIMEUPDATE (data flow) =
*
Major frame time update rate (4.096 sec).
*

PCD_Scene_Acct (store) =
*
A database table containing identified scenes and the associated scene information for the processed contact period.
*

Pointer (data flow, pel) =
*
NO DESCRIPTION AVAILABLE
*

Polarity_Change_Cnt (data flow) =
*
Count of polarity changes on the CADUs. If there are no polarity changes encountered, the value stored will be 0.

DATABASE DOMAIN NAME: Count_Int
DATABASE DATA TYPE: Number(8)
DATABASE RANGE/VALUE: X>=0 (Not Null)
*

Process_Id (store) =
*

A database table containing process leader IDs.
*

Processing_Version_Info (store) =
*
A database table containing processing identification for
each data set processed on this string.
*

RDC_Acct (store) =
*
Raw Data Capture accounting information.
*

RDP_Acct (data flow) =
*
Accounting information for return link report
*

RDP_Globals (store) =
*
Globals variables used by RDPS
*

RDP_Stats (store) =
*
The temporary store for the RDP statistics to be stored in
the database (RDP Acct)
*

RETURN_CODE (control flow) =
*
Unit return status.
*

Raw_WB_Sets (store) =
*
This store contains the Raw_WB_Data to be processed.
*

Real (data flow, pel) =
*
Real number.
*.

String (data flow) =
*
An array of characters
*

Sub_Intv (store) =
*
A database table containing the beginning and ending
major frame times corresponding to a predefined

subinterval time range for each contact.
*

Valid_Band_Parms (store) =
*
A database table containing validated band parameters entered by an operator that specify which bands to process for ACCA and Browse. Valid band numbers are 1-6. The Multi1, Multi2, and Multi3 are for multiband browse data. Subs and Wave are subsampling and wavelet reduction ratios, respectively. CCA_Method and CCA_Ratio are two user-defined ACCA parameters.
*

Valid_CCSDS_Parms (store) =
*
A database table containing validated CCSDS parameters.
*

Valid_MFP_Parms (store) =
*
A database table containing the validated MFPS setup parameters.
*

Valid_MFP_Thres (store) =
*
The validated MFPS threshold values.
*

Valid_PCD_Parms (store) =
*
The database table containing the validated PCD Parameters used in processing PCD data.
*

Valid_PCD_Thres (store) =
*
A database table containing the validated PCD threshold parameters used in processing PCD data.
*

Valid_RDP_Thres (store) =
*
A database table containing the validated RDP processing thresholds.
*

Valid_Scene_Parms (store) =
*
A database table containing the validated general mission information and parameters that are provided by the IAS and used to calculate the longitude and latitude, the WRS Scene Id, and sun elevation and azimuth.

```
*

Valid_WRS_Parms (store) =
*
A database table containing the validated Worldwide
Reference System information for each WRS scene.
*

buff (data flow) =
*
NO DESCRIPTION AVAILABLE
*

flipped_bits (data flow) =
*
NO DESCRIPTION AVAILABLE
*

frames (data flow) =
*
NO DESCRIPTION AVAILABLE
*

fsBytes (data flow) =
*
NO DESCRIPTION AVAILABLE
*

fsControl (data flow) =
*
NO DESCRIPTION AVAILABLE
*

fsData (data flow) =
*
NO DESCRIPTION AVAILABLE
*

fsDataSize (data flow) =
*
NO DESCRIPTION AVAILABLE
*

fsl (data flow) =
*
Offset of frame to first byte of encoded data for RS
(Set to 4 for Landsat)
*

gftPath (data flow) =
*
The pointer to the Galoia field tables for RS
*
```

idp_ACCAState (store) =
*
NO DESCRIPTION AVAILABLE
*

idp_BandState (store) =
*
A global table used by band child process
*

idp_BrowseState (store) =
*
A global store containing global information used in
browse child process
*

idp_MainState (store) =
*
A global store containing global information used in
main IDPS process
*

idp_accaDBOpenFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaExitFlag (data flow) =
*
Control flag for processing. It tells the shutdown procedure
the result of processing: LPS_SUCCESS, LPS_FAILURE, or
IDP_ABORTED_TERMINATION
*

idp_accaFileDescrs (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaFillValue (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaFillValueArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaMethod (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaRatio (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaSceneArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaSceneDataArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaSceneIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaSceneScores (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_accaSubIntvId (data flow) =
*
Identifies the subinterval for the contact period
*

idp_bandContactSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_bandMajorFrameIndexArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_bandMajorFrameTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_bandSceneCenterArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_bandSceneInfoArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

```
idp_bandSceneStartArg (data flow) =
* The time associated with the start of a scene in
  milliseconds from the beginning of the current
  day. *

idp_bandSceneStopArg (data flow) =
* The time associated with the end of a scene in
  milliseconds from the beginning of the current
  day. *

idp_browseDBOpenFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseExitFlag (data flow) =
*
Status flag for idp_Browse process
*

idp_browseFileDescrs (data flow) =
*
NO DESCRIPTION AVAILABLE
*
```

```
idp_browseFileList (data flow) =
*
Linked list of multiband browse filenames
*

idp_browseFilenameArg (data flow) =
*
Filename for current multiband browse file
*

idp_browseFillValue (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseMulti1 (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseMulti2 (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseMulti3 (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseMultiDescr (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseSceneIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseSubIntvId (store) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseSubSampled (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseSubs (data flow) =
*
NO DESCRIPTION AVAILABLE
*
```

```
idp_browseWave (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_browseWavelet (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_cfrFileDescrsArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_cfrTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_childDBOpenFlagArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_childExitFlagArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_childFifoIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_childFifoKeyArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_childFileDescrsArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_childFillValueArg (data flow) =
*
Fill value for band files
*

idp_childListArg (data flow) =
*
NO DESCRIPTION AVAILABLE
```

*

idp_childShutdownArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

```
idp_childSubIntvIdArg (data flow) =
*
Subinterval identifier
*

idp_db_buFilenameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_db_buSubIntvIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_db_gfvFillValueArg (data flow) =
*
Fill value for the band data
*

idp_db_gipParmsArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_firstTimeIndexArg (data flow) =
* The value that points to the major frame whose time
  is closest to the scene start time. *


idp_lastTimeIndexArg (data flow) =
*  A value that points to the major frame whose
   time comes closest to the end of scene time. *


idp_mainChild (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_mainExitFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*

idp_mainLastSubIntvFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*

invertedFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*
```

ldt_DTATimeArg (data flow) =
*
The time a DTA was received from the LP DAAC.
*

ldt_contactSeqIdArg (data flow) =
*
A contact sequence ID, together with a file version ID, uniquely identifies an LDTS file set.
*

ldt_dlfOnlyDeleteIngestedFilesArg (control flow) =
*
When TRUE, indicates that only files which have been ack'd as successful in DDN's should be deleted.

When FALSE, indicates all files in a file set should be deleted.
*

ldt_dlfOverrideRetentionArg (control flow, pel) =
*
When TRUE, indicates that files should be deleted regardless of whether or not they have been marked for retention.

When FALSE, indicates that only files should be deleted only if they have not been marked for retention.

The LDTS DDN server should NEVER delete files which have been marked for retention. However the operator does have that option.
*

ldt_fileNameArg (data flow) =
*
The name of an LPS output file.
*

ldt_fileVersionIdArg (data flow) =
*
An identifier which distinguishes between LPS output file sets processed from the same raw data. A file version ID, together with a contact sequence ID, uniquely identifies an LDTS file set.
*

ldt_getretContactSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

ldt_getretFileVersionIdArg (data flow) =

*
NO DESCRIPTION AVAILABLE
*

ldt_getretRetentionStateArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

ldt_gettodTimeOutListArg (data flow) =
*
This object is a list of structures which identify a file
set whose DAN has timed out (waiting for a DDN) and which
give the time at which this occurred.
*

ldt_gftsStartTimeArg (data flow) =
*
The optional user specified report start time for a file
transfer summary report.
*

ldt_gftsStopTimeArg (data flow) =
*
The optional user specified report stop time for a file
transfer summary report.
*

ldt_ingestedArg (data flow) =
*
An indication of whether or not an LPS output file set has
been ingested, i.e., whether or not each of the files in
the file set has been successfully transferred to the LP
DAAC by the LP DAAC.
*

ldt_nas (data flow) =
*
Structure containing NASA standard time.
*.

ldt_rdSleepDuration (data flow) =
*
The amount of time between ldt_RcvDTA's checks for incoming
traffic on the socket with the LP DAAC.
*

ldt_sdAppendItemArg (data flow) =
*
This object represents a byte string which is to be appended to a DAN byte stream.
*

ldt_sdContactFileNamesArg (data flow) =
*

This object represents a list of all file names in a
particular file set.
*

ldt_sdContactSeqIdArg (data flow) =
*
The contact sequence ID which, together with a file version number, will uniquely
identify a set of LPS output files for which a DAN is to be sent or resent by
LDTS.
*

ldt_sdDANArg (data flow) =
*
This object contains all information necessary in a data availablility notification.
*

ldt_sdFileSetListArg (data flow) =
*
Array containing multiple pairs of contact sequence ID and
file version ID and a count of the pairs.
*.

ldt_sdFileVersionIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

ldt_sdNonexistentFilesArg (data flow) =
*
This object represents a list of all file names in a
particular file set which as given do not represent files
which exist on the LPS host machine.
*

ldt_sdResendFlagArg (control flow, pel) =
*
This object will indicate to the LDTS Send DAN function
whether the MACS is requesting that a DAN be sent for the
first time (SEND) or whether it is requesting that a
previously encountered DAN be sent again (RESEND).
*

ldt_sdUnreadableFilesArg (data flow) =
*
This object represents a list of all file names in a
particular file set which are not world readable.
*

ldt_settoContactSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

ldt_settoFileVersionIdArg (data flow) =

*
NO DESCRIPTION AVAILABLE
*

ldt_settoNextTimeoutArg (data flow, pel) =
*
The time at which a DAN will next timeout if a DDN is not
received by that time.
*

ldt_suspendArg (data flow) =
*
This object represents the value which will be used to update the Suspended attribute in the LDT_Output_File_Set_Info database table.
*

ldt_updretContactSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

ldt_updretFileVersionIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

ldt_updretRetainArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

ldt_xferStateArg (data flow) =
*
This object represents the state of DAN Transfer for LDTS. Only when DAN transfer is enabled are DAN's sent or resent to the LP DAAC.
*

lpsAddressType (data/control flow) =
*
Address pointer type definition.
*

lpsBooleanType (control flow) =
*
Boolean type definition. This must be an 8 bit unsigned integer type due to its use in the CADU annotation structures (Ann_VCDU, ANN_CADU, lpsAnnotatedCADUStruct).
*

lpsInt16Type (data/control flow) =
*
16 bit integer type definition.
*

lpsInt32Type (data/control flow) =
*
32 bit integer type definition.
*

lpsShmKeyType (data flow) =
*
NO DESCRIPTION AVAILABLE
*

```
lpsTimeStruct (data flow) =
*
LPS time structure formats
*

lpsUInt16Type (data/control flow) =
*
16 bit unsigned integer type definition.
*

lpsUInt32Type (data/control flow) =
*
32 bit unsigned integer type definition.
*

lps_AlignedBands (store) =
*
The aligned band structure.
*

lps_Globals (store) =
*
A table containing lps global information
*

lps_Journal (store) =
*
A global store containing lps error or event/status messages
*

lps_annCadu (data flow) =
*
NO DESCRIPTION AVAILABLE
*

lps_ccsdsParmArg (data flow) =
*
A pointer to the CCSDS Parameter structure
*

lps_commitArg (control flow) =
*
A flag to indicate database commit or rollback
*

lps_configurationArg (data flow) =
*
A buffer pointer pointed to LPS_Configuration record
*

lps_dbErrorArg (data flow) =
*
Database error number
*
```

lps_dbErrorMsgArg (data flow) =
*
Truncated LPS database error message
*

lps_fifoInfoArg (data flow) =
*
fifo buffer address
*

lps_fifoInfoBytesArg (data flow) =
*
Total number of LPS FIFO message expected in bytes
*

lps_fifoInfoBytesRcvArg (data flow) =
*
The number of bytes actually received from the message queue
*

lps_fifoKey (data flow) =
*
An assigned key for LPS fifo channel
*

lps_fifoKeyArg (data flow) =
*
An assigned key to create a LPS FIFO or open an existing LPS
FIFO.
*

lps_fifoKeyIdMap (store) =
*
A global lookup table containing the fifoChannelId, fifoKey, fifoId, and
the number of parties connected to the fifo channel
*

lps_fifoWaitArg (control flow) =
*
A flag to indicate a wait i/o or no_wait i/o used
*

lps_fileNameArg (data flow) =
*
A LPS L0R filename with path.
*

lps_filePathArg (data flow) =
*
A LPS L0R file pathname
*

lps_fileSubIntvIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

```
lps_fileTypeArg (data flow) =
*
The L0R file type identifiers. They are defined as follows:

        ["LPS_PCD"    PCD file
         |"LPS_BAND1"        Band 1 file
         |"LPS_BAND2"        Band 2 file
         |"LPS_BAND3"        Band 3 file
         |"LPS_BAND4"        Band 4 file
         |"LPS_BAND5"        Band 5 file
         |"LPS_BAND6"        Band 6 file
         |"LPS_BAND7"        Band 7 file
         |"LPS_BANDPAN"  Band PAN file
         |"LPS_BROWSE"       Browse file
         |"LPS_CAL"     Cal file
         |"LPS_MSCD"          MSCD file
         |"LPS_META"]         Metadata file
*

lps_logLineNumbArg (data flow) =
*
The source code line number provided by preprocessor
*

lps_logMessageArg (data flow) =
*
LPS status or error message string
*

lps_logPriorNumbArg (data flow) =
*
LPS message priority level
*

lps_logUnitNameArg (data flow) =
*
The source file name provided by preprocessor
*

lps_mainContactIdArg (data flow) =
*
*.

lps_procChildNameArg (data flow) =
*
Child process load module or script file
*

lps_procChildOptionArg (data flow) =
*
A passing argument list vctor to the child process
*

lps_procChildPIDArg (data flow) =
```

```
*
A child process identifier returned by kernel
*

lps_procChildStatusArg (data flow) =
*
A integer to represent the child exit status
*

lps_procDBArg (control flow) =
*
A flag to indicate whether the database connection is needed
*

lps_procHandleArg (data flow) =
*
A function call pointer pointed to a process handling function
*

lps_procWaitStatusArg (data flow) =
*
Child exit status
*

lps_shmBlkArg (data flow) =
*
Shared memory block address
*

lps_shmBlkMfpPcd (data flow) =
*
NO DESCRIPTION AVAILABLE
*

lps_shmBlkRdpMfp (data flow) =
*
Array of LPS_SHMBLK_RDPMFP_NUMCADUS Annotated  Cadus in a Shared
Memory  Block
*

lps_shmBlkTypeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

lps_shmCntrlInfo (store) =
*
NO DESCRIPTION AVAILABLE
*

lps_shmKeyArg (data flow) =
*
The shared memory key
*
```

lps_shmNumbFBlkArg (data flow) =
*
The total number of free blocks created in the free block list.
*

```
lps_shmNumbSegArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

lps_shmPartSegSizeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

lps_shmRdBlkArg (data flow) =
*
An address pointed to a read block
*

lps_shmRsrcIdArg (data flow) =
*
A LPS shared resource identifier
*

lps_shmRsrcInfoArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

lps_shmRsrcKey (data flow, pel) =
*
The shared resource key used by all interested parties
*

lps_shmSegPtr (data flow, pel) =
*
The address of a shared memory segment
*

lps_shmSemIdArg (data flow) =
*
Semaphore Id
*

lps_shmWaitArg (control flow) =
*
A integer indicating whether a blocking i/o or  non-blocking i/o used
*

lps_shmWrBlkArg (data flow) =
*
An address pointed to a write block in shared memory resource
*

lps_shm_MFPPCD (store) =
*
NO DESCRIPTION AVAILABLE
*
```

lps_shm_RDPMFP (store) =
*
The shared memory segment between RDPS and MFPS
*

lps_subIntvArg (data flow) =
*
A pointer to the subinterval structure
*

lps_subIntvId (data flow) =
*
Subinterval identifier
*

lps_subIntvIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

lps_subIntvSequenceIdArg (data flow) =
*
Subinterval sequence identifier.
*

lps_time1Arg (data flow) =
*
LPS time format
*

lps_time2Arg (data flow) =
*
LPS time structure format
*

lps_timeArg (data flow) =
*
LPS time structure format
*

lps_timeCurrentTimeArg (data flow) =
*
A time format
*

lps_timeNumbUnitArg (data flow) =
*
A number of time duration unit
*

lps_timeOffsetArg (data flow) =
*
LPS time structure format

```
*

lps_timeStringArg (data flow) =
*
LPS time string type
*
```

```
lps_timeStructFormatArg (data flow) =
*
Time structure format
*

m (data flow) =
*
Number of bits per signal for Reed_Solomon (set to 4 for Landsat)
*

mac_MetaDataGlobal (store) =
*
Meta file descriptor for a subinterval
*

mac_acNextCaptureInfoArg (data flow) =
*

mac_acSignalNumberReceived (control flow) =
*
  Global variable indicating either that auto capture
  process has not received a signal or the number of
  the signal received.  Possible values are MAC_AC_NOSIGNAL,
  MAC_AC_SCHED_UPDATE_SIG, and MAC_AC_TERM_SIG.
*

mac_acSleepTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_filenameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_formatArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_groupIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_ingestIASFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_mainContSeqIdArg (data flow) =
*
A unique number assigned to a scheduled contact period or
```

acquisition entry in Contact_Schedules store.
*.

mac_mainFileVerNumArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_mainGroupIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_maindelRawDataFileArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_maindelRawDataFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaContactSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaFileCrtTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaFilePathArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaFileSrcIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaFileVerNumArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaLpsConfArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

```
mac_metaSubIntvArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaSubIntvSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_metaSwVerNumArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_opsSeverityListArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_ppIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_processArrayArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_rptContactSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_rptFileVerNumArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_statusArrayArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mac_subIntvStartTimeArg (data flow) =
*
Subinterval start time
*.

mac_subIntvStopTime (data flow) =
*
NO DESCRIPTION AVAILABLE
```

*

mf_offset (control flow) =
*
NO DESCRIPTION AVAILABLE
*

```
mfpAvInfoStruct (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_EolTol (data flow) =
* maximum number of frames away from the nominal EOL
  minor frame count to continue checking for the
  existence of an EOL minor frame.  (Checking ceases
  when an EOL frame pair is found.)  *

mfp_FailedMjfs (store) =
* The tossed CADUs that have not been able to be define.*.

mfp_Global (store) =
*
NO DESCRIPTION AVAILABLE
*

mfp_MajSyncSize (data flow) =
* for sync (or EOL) minor frames that are partially
  fill, the minimum number of data bytes (non-fill)
  that must be present for the frame to be accepted
  as a sync (or EOL) minor frame. *

mfp_MajSyncTol (data flow) =
* limit for number of bit errors allowed in the sync
  minor frame.  Exceeding this limit constitutes
  prima-facie evidence that the frame is NOT a sync.
  (re-used for each EOL minor frame. *

mfp_MjfQa (store) =
*
NO DESCRIPTION AVAILABLE
*

mfp_Parms (store) =
*
NO DESCRIPTION AVAILABLE
*

mfp_StatusInfo (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_SubQa (store) =
*
NO DESCRIPTION AVAILABLE
*

mfp_Thres (store) =
*
NO DESCRIPTION AVAILABLE
```

*

```
mfp_VfyEolIndexArg (control flow) =
* index to .mfp_avInfo[] where the 1st EOL frame starts *.

mfp_VfyEolOffsetArg (control flow, del) =
* byte offset in VCDU data area for start of 1st EOL
  minor frame *.

mfp_VfyEolmfcArg (control flow, del) =
* mf_count for 1st EOL minor frame  *.

mfp_VfyMfcArg (control flow) =
* minor frame count to be checked against the standard EOL
  pattern  *.

mfp_VfyNummfArg (control flow, del) =
* number of minor frames to extract (1-22) *.

mfp_VfyStartMfcArg (control flow) =
* mfc for the 1st minor frame to extract *.

mfp_VfySyncBegEndArg (control flow) =
* flag to tell mfp_FindMjfSync() to look at the beginning
  of .mfp_avInfo[] or at the end for a sync frame.
  0 = beginning,   1 = end           *.

mfp_VfySyncIndexArg (control flow) =
* index to .mfp_avInfo[] for CADU that contains the start
  of the sync minor frame *.

mfp_VfySyncOffsetArg (control flow, del) =
* byte offset in VCDU data area for sync start *.

mfp_VfymfBufArg (data flow) =
* buffer for minor frames (85 bytes each).  Either
  data or 0-fill will be transferred.  *.

mfp_VfymfBytesArg (control flow, del) =
* # bytes of data output to  mfp_VfymfBufArg, NOT counting
  the fill bytes copied  *.

mfp_arrayMaxArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_arrayStarterArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_bdLoadLocArg (data flow) =
* An array of pointers to the location in lps_AlignedBands that data will be placed
for each band/detector combination.
*.
```

mfp_bitValueArg (data flow) =
* The bit value result from the resolution of the data
group word.
*.

mfp_caduLocArg (data flow) =
*
The location of data to be extracted.
*

mfp_caduSeqErrThr (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_caduSyncErrCnt (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_calDeintAlignDataArg (data flow) =
* A structure containing the deinterleaved and aligned
calibration data by band and detector. The structure also
includes quality information for the major frame*

mfp_calDetSubInfoArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_calFileName (data flow) =
* The calibration file name for the current subinterval. *.

mfp_calFpArg (data flow) =
*
The file pointer to the calibration file.
*.

mfp_calMjfTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_calMscdArg (data flow) =
* Contains the quality information and starting and end location to
the MSCD and calibration data *.

mfp_calNewSubIntvArg (control flow) =
*
An integer value which determines a new subinterval, the end of contact,
no subinterval change, end of one subinterval and the beginning of another
subinterval, and none of above at the beginning of contact period.
*.

```
mfp_calPcdStatSetFlgArg (control flow) =
* A flag indicating if status information is available *.
```

mfp_calScanDirArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_calStartArg (data flow) =
*
Defines the starting location of the calibration data
*

mfp_chckAdjFactorArg (data flow) =
* A adjustment value for correctly positioning the pointer to
the next minor frame *.

mfp_chckCaduLocArg (data flow) =
*
The location of data to be extracted.
*

mfp_chckLengthArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_chckSplitArg (control flow) =
* A flag indicating if a minor frame spans 2 cadus*.

mfp_dataGroupArg (data flow) =
* The five byte data group as that condenses to one bit.
*.

mfp_dataPeriod (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_eolErrFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_fillValue (data flow, cel) =
*
NO DESCRIPTION AVAILABLE
*

mfp_filledData (data flow, cel) =
* Prefixed set of bytes full of mfp_fillValue for inserting into aligned
band data when the complete major frame is missing.*.

mfp_formatList (data flow) =
*
This structure is an double array of size 80 by 2 that contains the band and detector
number

exactly as defined in the format layout of the DFCB.
*

mfp_fullFillCnt (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_holdQa (data flow) =
* For holding QA totals for tossed VCDUs*

mfp_l0rCalMscdArg (data flow) =
* The starting location to where the mscd/calibration
data begins*.

mfp_l0rDetSubInfoArg (data flow) =
* Identifies the current subinterval condition*.

mfp_l0rMjfTimeArg (data flow) =
* The major frame time*.

mfp_l0rNewSubIntvArg (data flow) =
*
An integer value which determines a new subinterval, the end of contact,
no subinterval change, end of one subinterval and the beginning of another
subinterval, and none of above at the beginning of contact period.
*

mfp_l0rPcdStatSetFlgArg (control flow) =
* A flag indicating if status information is available*.

mfp_l0rScanDirArg (data flow) =
* The scan direction *.

mfp_mainAvInfoArg (data flow) =
* The structure with pointers to shared memory *

mfp_mainBadTimeInfoArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainBandsArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainCalMscdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainCheckTypeArg (data flow) =
*
NO DESCRIPTION AVAILABLE

*

```
mfp_mainDataWordArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainDetSubInfoArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainEstimatedArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainFirstFullDataWordArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainFirstGoodMnfArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainLastFullDataWordArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainLastGoodMnfArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainMjfPrevTimeArg (data flow) =
*
The previous major frame time.
*.

mfp_mainMjfReuseNumArg (data flow) =
* (int) re-use count for CADUs at end of major frame VCDUset
  to be re-used for next major frame.
  0-N = count of data/fill CADUs to re-use
  -1  = re-use last (data) CADU only, re-generate any fill *.

mfp_mainMjfSetEndArg (control flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainMjfSetStartedArg (control flow) =
* True: Major Frame Synch VCDUs detected.
  False: Otherwise *
```

mfp_mainMjfSetStateArg (control flow) =
* An integer value representing the reason for terminating
the previous invocation *

mfp_mainMjfTcFramesArg (data flow) =
*
Structure to help identify the time code minor frames and how complete they are.
*.

mfp_mainMjfTimeArg (data flow) =
*
The time associated with one major frame. This time
is extracted from minor frame 2 through 5. It is
calculated according to the minor frame format as stated
in the DFCB.
*.

mfp_mainMjfValidArg (control flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainNewSubIntvArg (control flow) =
*
An integer value which determines a new subinterval, the end of contact,
no subinterval change, end of one subinterval and the beginning of another
subinterval, and none of above at the beginning of contact period.
*.

mfp_mainNumBadDgsArg (data flow) =
* The number of data groups in the major frame time code that were found to
condense outside the tolerance value.not-defined.*.

mfp_mainNumDisArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainPcdInfoPtrArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainPcdStatSetFlgArg (control flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainPreAvInfoArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainQaSetArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainScanBitArg (control flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainScanDirArg (data flow) =
* An integer for the scan direction. 1 is forward, 0 is reverse*.

mfp_mainSeqEstCountArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainSeqFlagArg (control flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainSpacecraftIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainTCMnfArg (data flow) =
*
The start of the time code minor frame.
*.

mfp_mainTcTypeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainThresFlagsArg (control flow) =
* Threshold flags All Boolean*.

mfp_mainValidFlagArg (control flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mainVcduSetArg (data flow) =
* The argument is assigned Either mfp_mjfVcduSet or mfp_tossedVcduSet
for passing to mfp_MainChkMissing *.

mfp_majVoteTol (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_majorFrameTimeArg (data flow) =
*
A structure to temporarily hold the different pieces of time information.
The days,hours, minutes, seconds, milliseconds, and 16th of a millisecond.

*.

mfp_maxAlignValue (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mjfQa (data flow) =
* For holding QA totals for tossed VCDUs*

mfp_mjfReuseNum (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mjfVcduSet (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mnfCtrThr (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_mscdCalMscdArg (data flow) =
* The starting location to where the mscd data begins*.

mfp_mscdDataArg (data flow) =
*
The mirror scan correction data collected on a major frame basis for a
given subinterval. The MSCD data is located immediately following the end of
line code pattern and is extracted from 2 contiguous minor frames.
The structure includes quality information for the major frame.*.

mfp_mscdDetSubInfoArg (data flow) =
* Conditions for a subinterval *.

mfp_mscdFileName (data flow) =
* The MSCD file name for the current subinterval.*.

mfp_mscdFpArg (data flow) =
*
The file pointer to the mirror scan correction data file.
*.

mfp_mscdMjfTimeArg (data flow) =
* The major frame time *.

mfp_mscdNewSubIntvArg (data flow) =
*
An integer value which determines a new subinterval, the end of contact,

no subinterval change, end of one subinterval and the beginning of another subinterval, and none of above at the beginning of contact period.
*

mfp_nextAvInfo (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_nextAvInfoFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_parms (data flow) =
* MFPS parameters from the database. *.

mfp_partFillCnt (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_partMnfTol (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_scanDirArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_sensorAlignInfo (data flow) =
*
Information provided by IAS and used by the MFPS to perform integer-pixel alignment.
*

mfp_shmAddList (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_spacecraftId (data flow) =
*
There are 4 bits for the spacecraft id in time code minor frames 2 - 4, group 8a-e.
*.

mfp_splitArg (control flow) =
* A flag to indicate that the data will span two CADUs.*.

mfp_subIntv (data flow) =
*
Info for one sub interval.
*

mfp_subIntvClose (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_subIntvDelta (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_subQa (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_thres (data flow) =
*
The threshold values for MFPS.
*

mfp_timeCodeErrCnt (data flow) =
* Total number of CADUs with time code errors*

mfp_timeRangeTol (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_tossedVcduSet (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mfp_typeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

mo (data flow) =
*
RS code parameter used in RS generator polynomial(Set to 6 for Landsat)
*

mode (data flow) =
*
NO DESCRIPTION AVAILABLE
*

numFrames (data flow) =
*
NO DESCRIPTION AVAILABLE
*

num_bytes (data flow) =

*
NO DESCRIPTION AVAILABLE
*

offset (data flow) =
*
NO DESCRIPTION AVAILABLE
*

params (data flow) =
*
NO DESCRIPTION AVAILABLE
*

pcd_Global (store) =
*
Global data variables that will be used by the PCDS.
*

pcd_accSubIntvFileRecArg (data flow) =
*
An argument containing a subinterval file structure node.
*

pcd_contactSeqId (data flow) =
*
NO DESCRIPTION AVAILABLE
*

pcd_cycleList (data flow) =
*
The global linked list containing the assembled PCD cycles.
*

pcd_eocFlag (data flow) =
*
NO DESCRIPTION AVAILABLE
*

pcd_fileVersionNum (data flow) =
*
NO DESCRIPTION AVAILABLE
*

pcd_foutArg (data flow) =
*
Interpolated dependent variable.
*

pcd_fxArg (data flow) =
*
Array of independent variables.
*

pcd_mainActSceneArg (data flow) =
*
A pointer to pcdActSceneStruct which is passed between
modules as an argument.
*

pcd_mainAttDataArg (data flow) =
*
Attitude data array.
*

pcd_mainAttTimeArg (data flow) =
*
Attitude time tag array.
*

pcd_mainContactSeqIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

pcd_mainDataBreakAcctArg (data flow) =
*
A structure containing the count and position of each data
break occurance.
*

pcd_mainDataWordArg (data flow) =
*
A structure containing a pointer to a buffer of
maajority voted PCD data words, and its associated
accounting information.
*

pcd_mainDeclnArg (data flow) =
*
Declination angle.
*

pcd_mainEndMjfNoArg (data flow) =
*
The ending fill major frame number.
*

pcd_mainEndMnfNoArg (data flow) =
*
The ending fill minor frame number.
*

pcd_mainEphDataArg (data flow) =
*
Ephemeris data array.
*

pcd_mainEphTimeArg (data flow) =
*
Ephemeris time tag array.
*

pcd_mainEtmToGciMatrixArg (data flow) =

*
A 3x3 transformation matrix from ETM+ to GCI.
*

pcd_mainFileVersionArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

pcd_mainFirstSceneArg (data flow) =
*
A pointer to pcdNomSceneStruct which is passed between
modules as an argument. The first WRS scene center.
*

pcd_mainHdsArg (data flow) =
*
The horizontal display shift in kilometers.
*

pcd_mainIntAttArg (data flow) =
*
Interpolated attitude data.
*

pcd_mainIntEphArg (data flow) =
*
Interpolated ephemeris data.
*

pcd_mainJulTimeArg (data flow) =
*
The modified Julian date.
*

pcd_mainLatArg (data flow) =
*
The geodetic latitude of the look point.
*

pcd_mainLongArg (data flow) =
*
The longitude of the view point.
*

pcd_mainLosGciArg (data flow) =
*
A 3 x 1 line of sight unit vector in GCI coordinates.
*

pcd_mainMafTimeArg (data flow) =
*
A pointer to a list of type pcdScTimeStruct which is passed
between modules as an argument.

```
*

pcd_mainMajVoteDataWordArg (data flow) =
*
An argument containing the majority voted PCD data word result
*
```

```
pcd_mainMajorFramesArg (data flow) =
*
A pointer to a list of PCD major Frames.
*

pcd_mainMinorFramesArg (data flow) =
*
A pointer to the pcdMinorFrameStruct.
*

pcd_mainMissingCyclesArg (data flow) =
*
An argument containing the number of missing PCD cycles
prior to the current PCD cycle.
*

pcd_mainMissingDataWordCntArg (data flow) =
*
An argument containing the number of PCD words missing.
*

pcd_mainNextWrsSceneArg (data flow) =
*
A pointer to pcdNomSceneStruct which is passed between
modules as an argument.  The next WRS scene center.
*

pcd_mainNomSceneArg (data flow) =
*
A pointer to pcdNomSceneStruct which is passed
between modules as an argument.
*

pcd_mainNumPtrArg (data flow) =
*
The number of input points.
*

pcd_mainOffsetArg (data flow) =
*
An argument containing the value identifying the cut-off
position for valid PCD words.
*

pcd_mainOrbitNumArg (data flow) =
*
An argument containing the calculated orbit number
*

pcd_mainPcdCycleArg (data flow) =
*
An instance of a PCD cycle structure.
*
```

pcd_mainPcdStartTimeArg (data flow) =
*
An argument containing the subinterval PCD start time.
*

pcd_mainPosArg (data flow) =
*
The spacecraft GCI position.
*

pcd_mainPositionArg (data flow) =
*
A pointer to a list of type pcdPositionStruct which is passed
between modules as an argument.
*

pcd_mainPrevWrsSceneArg (data flow) =
*
A pointer to pcdNomSceneStruct which is passed between
modules as an argument. The previous WRS scene center.
*

pcd_mainProcStatusArg (data flow) =
*
An argument containing the PCDS processing status.
*

pcd_mainQutnArg (data flow) =
*
Attitude quaternion.
*

pcd_mainRightAsnArg (data flow) =
*
Right ascension angle.
*

pcd_mainScTimeArg (data flow) =
*
A pointer to a list of type pcdScTimeStruct which is passed
between modules as an argument.
*

pcd_mainSceneArg (data flow) =
*
A pointer to a listed list of scene center description.
*

pcd_mainSceneDescrArg (data flow) =
*
A pointer to lpsSceneDescrStruct which is passed between
modules as an argument.
*

pcd_mainSceneParmArg (data flow) =
*
A pointer to a list of type pcdSceneParmStruct which is passed
between modules as an argument.
*

```
pcd_mainShmMemAddrArg (data flow) =
*
A pointer to the shared memory structure from MFPS
containing the PCD bytes.
*

pcd_mainStartMjfNoArg (data flow) =
*
The starting fill major frame number.
*

pcd_mainStartMnfNoArg (data flow) =
*
The starting fill minor frame number.
*

pcd_mainStartPosArg (data flow) =
*
An argument containing the value identifying the starting
position for valid PCD words
*

pcd_mainSubIntvFileListArg (data flow) =
*
A linked list of subinterval file structure pointers
*

pcd_mainSubIntvIdArg (data flow) =
*
The current suubinterval ID associated with the current
PCD cycle.
*

pcd_mainSunAzArg (data flow) =
*
Sun azimuth angle.
*

pcd_mainSunElevArg (data flow) =
*
Sun elevation angle.
*

pcd_mainSunPosArg (data flow) =
*
A 3 x 1 sun position vector.
*

pcd_mainSyncPosArg (data flow) =
*
A structure containing the position of each PCD word cycle sync
pattern located in the PCD word stream
*
```

pcd_mainTimeArg (data flow) =
*
The time at which the data interpolation is needed.
or
The time at which GHA is computed.
*

pcd_mainTripDataWordArg (data flow) =
*
An argument containing a pointer to triple data word structure.
*

pcd_mainVelArg (data flow) =
*
Velocity data.
*

pcd_mainWordArg (data flow) =
*
A structure containing the PCD data words.
*

pcd_minorFrames (data flow) =
*
A pointer to a minor frame structure linked list.
*

pcd_numArg (data flow) =
*
Number of input points.
*

pcd_procReturnCode (data flow) =
*
The PCDS processing status returned to calling environment.
*

pcd_storeBandInfoArg (data flow) =
*
An argument containing a pointer to a bands present
structure array.
*

pcd_storeBandInfoCountArg (data flow) =
*
The number of bands present records in pcd_storeBandInfoArg.
*

pcd_subIntvChangedFlag (data flow) =
*
A flag identifying that there has been a subinterval change
during processing:
 LPS_TRUE - Subinterval change
 LPS_FALSE - No subinterval change.

*

pcd_subIntvIdList (data flow) =
*
A structure containing the PCDS subinterval processing
identification for the contact period.
*

pcd_validParms (data flow) =
*
A structure containing the valid PCD parameters.
*

pcd_validSceneParms (data flow) =
*
A structure containing the validated scene parameters.
*

pcd_validThres (data flow) =
*
An instance of type pcdValidThresStruct.
*

pcd_xArg (data flow) =
*
Array of independent variables.
*

pcd_xinArg (data flow) =
*
Value of independent variable to which a corresponding
dependent variable is being sought.
*

poa (data flow) =
*
RS code parameter used in generator polynomial(Set to 1 for Landsat)
*

rdc_AcctFile (store) =
*
Raw data capture accounting information.
*

rdc_PIDRestageFile (store) =
*
Contains the save process Id used to manually stop this process.
*

rdc_PIDSaveFile (store) =
*
Contains the save process Id used to manually stop this process.
*

rdc_PIDTempFile (store) =
*

Contains the capture process Id and is used to manually
stop data capture.
*

rdc_calcActualStartTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_calcActualStopTimeArg (data flow) =
*
Actual (current) time capture ends.
*

rdc_calcExpectedVolArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_calcNumberOfBuffersArg (data flow) =
*
Number of buffers filled used to calculate rdc_mainCptrDataVolArg
and rdc_mainCptrTransRateArg.
*

rdc_calcReceivedVolArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_calcSizeOfBufferArg (data flow) =
*
Number in megabytes used to calculate rdc_mainCptrDataVolArg.
*

rdc_calcTransRateArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_calcTransTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_dbOnLineArg (data flow) =
*
Flag if set to LPS_TRUE signifies that the data file is
on line.

Note: This flag looses its integrity should the user
manually delete files.
*

rdc_dbOnLineFNArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_dbWriteAcctFDArg (data flow) =
*
File descriptor for the capture accounting file.
*

rdc_dbWriteCptrModeArg (data flow) =
*
Option in rdc_Main set to "-u" if it is an unscheduled contact.
If the rdc_mainSchedIdArg is "null" and rdc_mainCptrModeArg is
also "null", the contact is considered manually driven. If
rdc_mainSchedIdArg has a valid value and rdc_mainCptrModeArg is
equal to "-u", it is considered invalid.

SCHEDULE ID	UNSCHEDULED	RESULT
--	--	Manual
--	-u	Unscheduled
-s	--	Scheduled
-s	-u	Invalid

*

rdc_dbWriteSchedIdArg (data flow) =
*
TBD
*

rdc_deleteDataFileOnlyArg (data flow) =
*
This is a user option. If rdc_deleteDataFileOnlyArg is
LPS_TRUE then the capture file is deleted whether it has
been updated in the database or not. If the file has
not been updated in the database, the accounting file
is not deleted.
*

rdc_deleteFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_initAcctFNArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_initFileDescArg (data flow) =
*
File descriptor for raw data capture file.
*

rdc_initFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_initStopTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_initStringIdArg (data flow) =
*
This string Id is an argument passed to RDCS needed to construct
the raw data capture filename.
*

rdc_labelDeviceArg (data flow) =
*
This is the device name where the removable medium is
located containing the capture accounting information used
to generate the label for short term storage.
*

rdc_mainCptrAcctFNArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainCptrActualStartTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainCptrActualStopTimeArg (data flow) =
*
Actual (current) time capture ends.
*

rdc_mainCptrDbFlagArg (data flow) =
*
If connection to the database is successful this flag
is set to LPS_TRUE.
*

rdc_mainCptrExpectedVolArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainCptrFileDescArg (data flow) =
*
File descriptor for raw data capture file.
*

rdc_mainCptrFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainCptrModeArg (data flow) =
*
Option in rdc_Main set to "-u" if it is an unscheduled contact.
If the rdc_mainSchedIdArg is "null" and rdc_mainCptrModeArg is
also "null", the contact is considered manually driven. If
rdc_mainSchedIdArg has a valid value and rdc_mainCptrModeArg is
equal to "-u", it is considered invalid.

SCHEDULE ID	UNSCHEDULED	RESULT
--	--	Manual
--	-u	Unscheduled
-s	--	Scheduled
-s	-u	Invalid

*

rdc_mainCptrReceivedVolArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainCptrStopTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainCptrTransRateArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainCptrTransTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainIsolateArg (data flow) =
*
Option for isolating capture process - Set capture process
to nonpreemptive, nondegrading priority, and set process
to restricted processor. Integer identifies processor number
to restrict process to.
*

rdc_mainSchedIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainStopTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainSuspendArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_mainSuspendGIDArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_pLabelChanIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_pLabelFNArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_pLabelSchedIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_pLabelStartTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_pLabelStopTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_pLabelStringIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_processorNumArg (data flow) =
*
This number represents the restricted processor used
to capture data.
*

rdc_restageDeviceArg (data flow) =
*
This is the device name where the removable medium is located

containing the capture accounting information to be restaged.
*

rdc_shutDwnFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_shutDwnWaitOptArg (data flow) =
*
Amount of time rdc_ShutDownRDC is to wait before starting
termination of RDCS. It is set to the stop time minus the
current time.
*

rdc_stpSvRsArg (control flow) =
*
This argument identifies whether the save process is to be
stopped, or the restage process is to be stopped.
*

rdc_suspendGIDArg (data flow) =
*
Level 0R processes group id number. This id number is necessary
to suspend Level 0R processes.
*

rdc_svAcctFNArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_svDeviceArg (data flow) =
*
This is the device name where the capture data is to be
saved to removable media for short-term storage.
*

rdc_testCptrModeArg (data flow) =
*
Option in rdc_Main set to "-u" if it is an unscheduled contact.
If the Schedule Id is "null" and the capture mode is
also "null", the contact is considered manually driven. If
Schedule Id has a valid value and the capture mode is
equal to "-u", it is considered invalid.

SCHEDULE ID	UNSCHEDULED	RESULT
--	--	Manual
--	-u	Unscheduled
-s	--	Scheduled
-s	-u	Invalid

*

rdc_testDTActualStopTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testDTExpectedVolArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testDTFileDescArg (data flow) =
*
File descriptor for raw data capture file.
*

rdc_testDTReceivedVolArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testDTTransRateArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testDTTransTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testIsolateArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testSchedIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testStopTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testSuspendArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_testSuspendGIDArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_udbAcctFNArg (data flow) =
*
Filename for RDCS accounting file.
*

rdc_udbCptrModeArg (data flow) =
*
Option in rdc_Main set to "-u" if it is an unscheduled contact.
If the rdc_mainSchedIdArg is "null" and rdc_mainCptrModeArg is
also "null", the contact is considered manually driven. If
rdc_mainSchedIdArg has a valid value and rdc_mainCptrModeArg is
equal to "-u", it is considered invalid.

SCHEDULE ID	UNSCHEDULED	RESULT
--	--	Manual
--	-u	Unscheduled
-s	--	Scheduled
-s	-u	Invalid

*

rdc_udbDbFlagArg (data flow) =
*
If connection to the database is successful this flag
is set to LPS_TRUE.
*

rdc_udbSchedIdArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeAcctFNArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeActualStartTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeActualStopTimeArg (data flow) =
*
Actual (current) time capture ends.
*

rdc_writeCptrModeArg (data flow) =
*
Option in rdc_Main set to "-u" if it is an unscheduled contact.
If the rdc_mainSchedIdArg is "null" and rdc_mainCptrModeArg is
also "null", the contact is considered manually driven. If
rdc_mainSchedIdArg has a valid value and rdc_mainCptrModeArg is
equal to "-u", it is considered invalid.

SCHEDULE ID	UNSCHEDULED	RESULT
--	--	Manual
--	-u	Unscheduled

```
  -s        --        Scheduled
  -s        -u         Invalid
*

rdc_writeFileNameArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeIsolateArg (data flow) =
*
Option for isolating capture process - Set capture process
to nonpreemptive, nondegrading priority, and set process
to restricted processor.  Integer identifies processor number
to restrict process to.
*

rdc_writeReceivedVolArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeStopTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeStringIdArg (data flow) =
*
This string Id is an argument passed to RDCS needed to construct
the raw data capture filename.
*

rdc_writeSuspendArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeTransRateArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdc_writeTransTimeArg (data flow) =
*
NO DESCRIPTION AVAILABLE
*

rdp_BCHQuadFile (store) =
*
Identical to rdp_bchMsnQuadTable, except that it is
stored as a file.
*
```

```
rdp_BCHTables (store) =
*
Global memory area used to hold the tables needed to
perform the various calculations for BCH error
detection & correction
*
```

rdp_MainFSyncGlobals (store) =
*
Gives a number of useful identifiers needed by the
RDPS/MFPS in situations where passing parameters is
inconvenient or impossible. Note that the variable,
rdp_mfsBytesReceived must be listed first as it is a
64-bit, type 'long long' variable. The maximum number
of CADUs in a shared memory block is given by
LPS_SHMBLK_RDPMFP_NUMCADUS
*

rdp_bchCADUArg (data flow) =
*
Annotated CADU
*

rdp_bchDenominatorArg (data flow) =
*
Value of denominator of a fraction expressed in the Galois
field.
*

rdp_bchELArg (data flow) =
*
Type of locator polynomial in the range [1, 8]
*

rdp_bchMsnCodeBlockArg (data flow) =
*
A de-interleaved mission Codeblock
*

rdp_bchMsnErrorArg (data flow) =
* An array of up to three positions for the errors in the mission codeblock *.

rdp_bchMsnItemCountArg (data flow) =
*
Number of items in rdp_bchMsnErrorArg array
*

rdp_bchMsnQuadTable (data flow) =
*
rdp_bchMsnQuadTable
[rdp_BCHMSNWRDLEN - 1]
[rdp_BCHMSNWRDLEN - 1]
[2] (three dimensions)

Table used to solve quadradic eqns for the Msn
locator poly
*

rdp_bchMsnRootCntArg (data flow) =

*
Dynamic size of rdp_BCHPtrErrorArg array
*

rdp_bchMsnRootCountArg (data flow) =
*
Count of the number of distinct roots of the locator
polynomial
*

rdp_bchMsnStartArg (data flow) =
*
Root to locator polynomial found by a straight search.
Will be used to reduce the polynomial from a cubic
to a quadratic.
*

rdp_bchMsnSyndromesArg (data flow) =
*
Holds values for the syndromes for the mission codeblock.
*

rdp_bchMsn_gf (data flow) =
*
rdp_bchMsn_gf[RDP_BCHMSNWRDLEN]
Powers of alpha.the generator polynomial for the Galois
Field GF(2^10 - 1)
*

rdp_bchMsn_syn_table (data flow) =
*
lpsUInt32Type [128][256]
Provides means for calculating the syndromes for a
mission codeblock. First dimension ([128]) is used for
the position of the byte being processed (0 through 127).
The second dimension is used for the binary value of the
byte (0 through 255).
*

rdp_bchMustBCHArg (control flow) =
*
BCH control flag
True = Perform BCH Decode independent of CRC errors
False = Do not perform BCH Decode if there are NO
CRC errors
*

rdp_bchNumeratorArg (data flow) =
*
Value of numerator of a fraction expressed in the Galois
field.
*

rdp_bchPtrBuffArg (data flow) =

```
*
Pointer to Pointer codeblock
*

rdp_bchPtrErrorArg (data flow) =
*
An array of up to three positions for errors in the
pointer codeblock.
*

rdp_bchPtrItemCountArg (data flow) =
*
Number of items in rdp_bchPtrErrorArg array
*

rdp_bchPtrQuadTable (data flow) =
*
rdp_bchPtrQuadTable[rdp_BCHPTRWRDLEN - 1]
[rdp_BCHPTRWRDLEN - 1]
[2] (three dimensions)
Table used to solve quadradic eqns for the Ptr
locator poly
*

rdp_bchPtrRootCntArg (data flow) =
*
Dynamic size of rdp_bchPtrErrorArg array
*

rdp_bchPtrRootCountArg (data flow) =
*
Count of the number of distinct roots of the locator
polynomial.
*

rdp_bchPtrStartArg (data flow) =
*
Root to locator polynomial found by a straight search.
Will be used to reduce the polynomial from a cubic
to a quadratic.
*

rdp_bchPtrSyndromesArg (data flow) =
*
Holds values of the syndromes for the Pointer area.
*

rdp_bchPtr_gf (data flow) =
*
rdp_bchPtr_gf[RDP_BCHPTRWRDLEN]
Powers of alpha. the generator polynomial for the Galois
Field GF(2^5 - 1)
*
```

rdp_bchPtr_syn_table (data flow) =
*
lpsUInt16Type [4][256]
Provides means for calculating the
syndromes for a mission codeblock.
First dimension ([4]) is used for the
position of the byte being processed
(0 through 3). The second dimension is
used for the binary value of the byte
(0 through 255).
*

rdp_bchShmPtrArg (data flow) =
*
Pointer to a shared memory block used to store annotated
CADUs
*

rdp_bchSigma1Arg (data flow) =
*
Cofficient of quadratic term in locator polynomial
*

rdp_bchSigma2Arg (data flow) =
*
Cofficient of linear term in locator polynomial.
*

rdp_bchSigma3Arg (data flow) =
*
Constant term in locator polynomial
*

rdp_bchThresholdArg (data flow) =
*
Error threshold for BCH EDAC processing
*

rdp_bchTrans_Table (data flow) =
*
rdp_bchTrans_Table[256].
For each of the 256 possible values for a byte. This
table "spreads out" the 8 bits into each of the eight
bytes in the 'long long' variable. Each bit in the
byte is placed sequentially right justified, with
binary zero fill to the left into the corresponding
byte of the table entry. This table is essential
for de-interleaving of the Mission and Mission BCH zone
into the 8 Mission Codeblocks.
*

rdp_crcFirstEntryArg (control flow) =
*
Flag to indicate first block to CRC for contact period - LPS_TRUE =

*

rdp_crcOctetsArg (data flow) =
*
Number of octets in input data block
*.

rdp_crcPolyArg (data flow) =
*
17 bit CRC polynomial
*

rdp_crcRegFlagArg (data flow) =
*
Indicates if CRC registers are filled with ones or zeros. Always set to ones filled for Landsat.
*

rdp_crcShmPtrArg (data flow) =
*
Pointer to a shared memory block that stores annotated VCDUs for CRC processing.
*

rdp_crcTableArg (data flow) =
*
Table for CRC calculations
*

rdp_crcThresholdsArg (data flow) =
*
Error thresholds for CRC EDAC processing
*

rdp_crcVCDUArg (data flow) =
*
Current Annotated VCDU being processed by CRC
*

rdp_dataFilePtr (data flow) =
*
Pointer to the Raw Data File
*

rdp_failedFilePtr (data flow) =
*
Pointer to the RDPS Trouble File
*

rdp_mainAlignedCADUArg (data flow) =
*
Pointer to byte aligned CADU.
*

```
rdp_mainCCSDSIndexArg (data flow) =
*
Database index for CCSDS parameters
*

rdp_mainCCSDSParmsArg (data flow) =
*
CCSDS Parameters for RDPS Frame Sync Processing
*

rdp_mainContactIdArg (data flow) =
*
Identifies the data set for a contact period
*

rdp_mainDataFilePtrArg (data flow) =
*
The file pointer for the raw data file
*

rdp_mainDataFilenameArg (data flow) =
*
The name of the raw data file for the specified contact period
*

rdp_mainEOFArg (control flow, pel) =
* Flag indicting whether eond-of-file condition has been encountered *

Natural
.

rdp_mainFailedFilePtrArg (data flow) =
*
The file pointer for the Failed CADU file.
*

rdp_mainFirstEntryArg (control flow) =
*
Flag to indicate first block to RDPS for contact period. LPS_TRUE =
*



rdp_mainInvertedCADUArg (data flow, pel) =
*
Indicates whether a received CADU was inverted when it
was received.  0 = NO, 1 = YES.
*

rdp_mainMustBCHArg (control flow) =
*
BCH control flag
True = Perform BCH Decode independent of CRC errors
False = Do not perform BCH Decode if there are NO
```

CRC errors
*

rdp_mainNumFlippedBitsArg (data flow) =
*
Number of flipped bits in a CADUs Synch pattern.
*

rdp_mainPositionInBuffArg (data flow) =
*
A signed value giving the starting position in bytes,
counting from zero for the offset of the current CADU
relative to the start of the data buffer. If a CADU
starts in the previous buffer, the value is negative.
For the current buffer it is zero or positive..
*

rdp_mainRawBlockAddrArg (data flow) =
* Address of block containing the CADUs from the raw input file. *

rdp_mainRawDataSizeArg (data flow, del) =
*
Number of bytes of data obtained from last read from raw data file that
holds the input CADUs
*

rdp_mainShmPtrArg (data flow) =
*
Pointer to a shared memory block used to store annotated
CADUs.
*

rdp_mainSlippageArg (data flow) =
*
A signed integer providing the magnitude and direction of a bit slip. Left
slips are negative and right slips are positive. For a example, a slip of -3
indicates a left slip of 3 bits.
*

rdp_mainStateArg (data flow) =
*
Gives software status associted with the given CADU.
Values: SEARCH, CHECK,.LOCK and FLYWHEEL.
*

rdp_mainSyncThresholdArg (data flow) =
*
Error threshold for frame sync processing
*

rdp_mainThresholdsArg (data flow) =
*
RDP processing thresholds stored in database by the MACS.
*

```
rdp_mainVersionArg (data flow) =
*
RDPS file version
*

rdp_mfsBytesReceived (data flow) =
*
Number of bytes received PRIOR to the current physical
block of data.
*

rdp_mfsSharedMem (data flow) =
*
Pointer to shared memory block.
*

rdp_mfsSyncThreshold (data flow) =
*
The number of CADUs with synch errors that are allowed
before the operator is notified.
*

rdp_rsFirstEntryArg (control flow) =
*
Flag to indicate first block for RS processing for
contact period  -  LPS_TRUE = first block
*

rdp_rsShmPtrArg (data flow) =
*
Pointer to a shared memory block used that stores annotated VCDUs for RS
processing.
*

rdp_rsThresholdsArg (data flow) =
*
Error thresholds for RS processing
*

rdp_status (data flow) =
*
Status of RDPS processing (LPS_SUCCESS or LPS_FAILURE)
*

rdp_thresholdsArg (data flow) =
*
RDP error thresholds retrieved from the database
*

rs_data (data flow) =
*
Pointer to current CADU for RS EDAC processing
*
```

rsmode (data flow) =
*
Flag to indicate use of convention(0) or symbol(1) representation for RS
(Set to 1 for Landsat)
*

slipped_bits (data flow) =
*
NO DESCRIPTION AVAILABLE
*

startThisBuff (data flow) =
*
NO DESCRIPTION AVAILABLE
*

state (data flow) =
*
NO DESCRIPTION AVAILABLE
*

stats (data flow) =
*
NO DESCRIPTION AVAILABLE
*

t (data flow) =
*
Number of RS correctable errors per codeblock (Set to 2 for Landsat)
*

vf (data flow) =
*
Number of virtual fill symbols in data for RS(Set to 5 for Landsat)
*